

Simulation von Massenspektren und Entwicklung eines Systems zur Reaktionsvorhersage

Den Naturwissenschaftlichen Fakultäten
der Friedrich-Alexander-Universität Erlangen-Nürnberg

zur

Erlangung des Doktorgrades

vorgelegt von

Robert Höllering

aus München

Als Dissertation genehmigt von den
Naturwissenschaftlichen Fakultäten der Universität Erlangen-Nürnberg

Tag der mündlichen Prüfung:

26. Februar 1998

Vorsitzender der Promotionskommission:

Prof. Dr. D. Kölzow

Erstberichterstatter:

Prof. Dr. J. Gasteiger

Zweitberichterstatter:

Priv. Doz. Dr. T. Clark

Meinem Doktorvater

Herrn Prof. Dr. Johann Gasteiger

danke ich für die anregenden Diskussionen und die vielfältige Unterstützung, die wesentlich zum Gelingen der vorliegenden Arbeit beigetragen haben.

Mein besonderer Dank gilt ferner:

- Herrn Dr. Klaus-Peter Schulz und Herrn Dr. Wolfgang Hanebeck für die Bereitstellung des Programms FRANZ, das die Daten geliefert hat, die den Ausgangspunkt dieser Arbeit darstellen. Außerdem danke ich ihnen für die vielen fruchtbaren Diskussionen sowie für die verwendeten Statistikprogramme bzw. das Programm VEGAS, mit dem Massenspektren und Fragmentierungsschemata dargestellt werden können.
- Herrn Prof. Dr. Jure Zupan für die Einführung in das Gebiet der neuronalen Netze während mehrerer Aufenthalte bei uns im Arbeitskreis.
- Frau Dr. Vera Simon für die Unterstützung bei der Arbeit mit neuronalen Netzen und die gute Zusammenarbeit im gemeinsamen Projekt.
- Herrn Dr. Walter Vetter von der F. Hoffmann-LaRoche AG Basel für die fruchtbaren Diskussionen und die Tests der Programme FRANZ und MASSIMO.
- Frau Dr. Larissa Steinhauer für die Methoden zur Lösung der Differentialgleichungen, die Anbindung von Tcl und die Implementierung der neuronalen Netze für EROS7.
- Frau Dr. Susanne Bauerschmidt für die Erstellung der Molekülorbital-orientierten Datenstruktur RICOS und die angenehme Zusammenarbeit während der Massenspektroskopiearbeiten und der Erstellung von EROS7.
- Herrn Thomas Kostka dafür, daß ich die von ihm erstellten Regeln zur Simulation des Abbaus von s-Triazinherbiziden in meiner Arbeit vorstellen durfte.
- Herrn Dr. Ulrich Hondelmann, Herrn Dr. Wolf-Dietrich Ihlenfeldt, Herrn Dr. Markus Wagener und Herrn Dr. Wolfgang Witzenbichler für die Entwicklung von Programmen zur Darstellung der Ergebnisse in dieser Arbeit und als Hilfsmittel für die Ein- und Ausgabe von Daten.
- Herrn Dr. Markus Wagener, Herrn Dr. Wolfgang Witzenbichler, Herrn Paul Selzer und Herrn Andreas Teckentrup für die stabile Hard- und Softwareumgebung.

Den nicht namentlich erwähnten Mitarbeitern des Arbeitskreises danke ich für die herzliche Arbeitsatmosphäre und die Hilfen bei administrativen Angelegenheiten.

Nicht zuletzt bedanke ich mich für die finanzielle Unterstützung:

- beim Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie,
- bei der Deutschen Bundesstiftung Umwelt sowie
- dem Institut für organische Chemie, Lehrstuhl I, Prof. Dr. Paul v. R. Schleyer.

Meinen Eltern

1. Einleitung	1
2. Einsatzgebiete der Reaktionssimulation in der Chemie	3
2.1. Reaktionsvorhersage	3
2.2. Syntheseplanung	3
2.3. Kombinatorische Chemie	3
2.4. Reaktionsführung	4
2.5. Massenspektroskopie	4
2.5.1. Strukturaufklärung	4
2.5.2. Sicherung der Spektrenqualität, Überprüfung von Datenbanken	7
3. Komponenten eines Systems zur Simulation von Reaktionen	9
3.1. Grundsätzlicher Aufbau eines Systems zur Simulation von Reaktionen	9
3.2. Repräsentation chemischer Strukturen (Bindungsliste)	9
3.3. Beschreibung von Reaktionstypen	9
3.4. Bewertung der Reaktionen	10
3.5. Massenspektrenvorhersage	10
3.6. Verwendung zur Syntheseplanung	10
4. Simulation von Massenspektren auf der Basis von Bindungslisten. 13	
4.1. Methodik	13
4.2. Erstellung der Wissensbasis	15
4.2.1. Fragmentierungsregeln	15
4.2.2. Datengewinnung mit FRANZ	16
4.2.2.1. Reaktionswahrscheinlichkeiten	17
4.2.2.2. Sicherheit der Bewertungen	17
4.2.2.3. Umlagerungen als Reaktionstyp	19
4.2.2.4. Heuristiken	21
4.2.3. Modifikation der Regeln	24
4.2.3.1. Weiterentwicklung	24
4.2.3.2. Erhöhung der Genauigkeit der Reaktionswahrscheinlichkeiten	27
4.2.4. Berechnung physikochemischer Effekte	28
4.2.5. Ableitung der Wahrscheinlichkeitsfunktionen	29
4.2.5.1. Regeln mit Umlagerungsreaktionstypen	30
4.2.5.2. Regeln ohne Umlagerungsreaktionstypen	37
4.3. Simulation von Massenspektren mit MASSIMO	44
4.3.1. Verwendete Regeln	44
4.3.2. Beispiele für die Leistungsfähigkeit des Ansatzes	48
4.3.2.1. Vergleich von simulierten und experimentellen Spektren	48
4.3.2.2. Selektion eines Strukturvorschlages durch Spektrenvergleich	50
4.3.2.3. Auffinden falscher Spektren in Datenbanken	52
4.3.2.4. Unterstützung von Massenspektroskopieexperten durch die Simulation von Spektren	54
4.3.3. Erweiterung der Regeln um die Wasserstoffumlagerung	55
4.3.3.1. Erstellung der Regel für die Wasserstoffumlagerung	56
4.3.3.2. Andere Quellen für die Wahrscheinlichkeiten der Umlagerungen	59

4.4.	Zusammenfassung der Ergebnisse	61
4.5.	Grenzen der Strukturrepräsentation und des verwendeten Systems zur Generierung der Reaktionen	62
4.5.1.	Kodierung von Radikalkationen	62
4.5.2.	Mesomerie	63
4.5.3.	Reaktionsbeschreibung	65
4.6.	Schlußfolgerungen	67
5.	RICOS, eine MO-orientierte Datenstruktur für Moleküle	69
5.1.	Vorteile gegenüber einer Bindungsliste	69
5.1.1.	Kodierung von π -Systemen	69
5.1.2.	σ -Ionisation	70
5.1.3.	Mehrzentrenbindungen	71
5.1.4.	Komplexbindungen	71
5.1.5.	Trennung von Bindungen und Nachbarschaften	72
5.1.6.	Gruppen und Aggregate	72
5.1.7.	Genealogische Variablen	72
5.2.	Vorteile von RICOS bei der Reaktionsgenerierung	73
5.2.1.	Reaktionen von Komplexen und Boranen	73
5.2.2.	Keine Erzeugung von Grenzstrukturen	73
5.2.3.	Steuerung des Reaktionsnetzwerks	74
6.	Das Programmsystem EROS7.....	77
6.1.	Konzeption	77
6.2.	Aufbau.....	77
6.3.	Funktionsweise	78
6.3.1.	Lesen der Regeln	78
6.3.2.	Reaktionsvorhersage	78
6.3.3.	Abschluß der Simulation	79
6.4.	Vorbehandlungen für Ensembles	79
6.4.1.	Ausgangsmaterialien	80
6.4.2.	Eduktensembles	80
6.5.	Definitionsmöglichkeiten der Reaktionssubstruktur.....	80
6.5.1.	Lineare, verzweigte und ringförmige Reaktionssubstrukturen	80
6.5.2.	Mehrere Teilsstrukturen	81
6.5.3.	Suche der Reaktionssubstruktur (RSS)	82
6.5.4.	Symmetriezahl der Reaktion	83
6.5.5.	Bedingungen	84
6.5.6.	Optionale Atome	85
6.6.	Durchführung der Reaktionen.....	86
6.6.1.	Reaktionsgenerierung	86
6.6.1.1.	Erzeugung von Zwischenprodukten und Übergangszuständen	86
6.6.1.2.	Reaktionen auf der Datenstruktur RICOS	87
6.6.1.3.	Ausschluß der Bildung ungültiger π -Systeme	90
6.6.1.4.	Valence-Bond-Emulation (Bindungslisten)	91
6.6.1.5.	Automatische Verteilung der Elektronen	92
6.6.1.6.	Weitere Möglichkeiten der Reaktionsgenerierung	95
6.6.2.	Berechnung der Reaktivität	98

6.6.3.	Anpassung der Symmetriezahl einer Reaktion	99
6.6.4.	Die Verteilungsfunktion	99
6.7.	Arten der Reaktionsführung: Reaktoren und Phasen	100
6.7.1.	Grundtypen der Reaktionsführung	101
6.7.1.1.	Unterscheidung kontinuierlich / absatzweise	101
6.7.1.2.	Absatzweise betriebener, idealer Rührkessel (AIK)	102
6.7.1.3.	Ideales Strömungsrohr (IR)	102
6.7.1.4.	Kontinuierlich betriebener, idealer Rührkessel (KIK)	102
6.7.1.5.	Rührkesselkaskade	102
6.7.1.6.	Vergleich der Reaktionsführungen	102
6.7.1.7.	Monomolekularer Zerfall	103
6.7.2.	Die Phasenmodi und ihre Auswahl der Edukte	103
6.7.2.1.	Der Phasenmodus Rührkessel	104
6.7.2.2.	Der Phasenmodus Monomolekular	105
6.7.2.3.	Der Phasenmodus Grenzflächenreaktionen	105
6.7.2.4.	Der Phasenmodus Inert	106
6.7.2.5.	Der Phasenmodus laminares Strömungsrohr	106
6.7.3.	Phasenübergänge und Phasentransferreaktionen	107
6.7.4.	Aufbau der Reaktoren mit Phasen in verschiedenen Modi	107
6.7.4.1.	Wann werden welche Phasenmodi eingesetzt?	107
6.7.4.2.	Absatzweise betriebener, idealer Rührkessel (AIK)	108
6.7.4.3.	Ideales Strömungsrohr (IR)	108
6.7.4.4.	Kontinuierlicher, idealer Rührkessel (KIK)	108
6.7.4.5.	Rührkesselkaskade	109
6.7.4.6.	Rührkessel mit organischer und wäßriger Phase	110
6.7.4.7.	Rührkesselkaskade im Gegenstrom	110
6.7.4.8.	(Pseudo-)Monomolekularer Zerfall	111
6.7.4.9.	Laminares Strömungsrohr	112
6.7.4.10.	Kombinatorische Chemie	112
6.7.4.11.	Weitere Aufbauschemata für Reaktoren	115
6.8.	Steuerung der Reaktionsfolge	115
6.9.	Kinetik.	117
6.9.1.	Kinetik mit Geschwindigkeitskonstanten	117
6.9.1.1.	Ausschluß unbedeutender Reaktionen	119
6.9.1.2.	Bestimmung der Integrationszeit	119
6.9.1.3.	Vergleich der Integrationsmethoden	119
6.9.1.4.	Ergebnisse am Beispiel vom Abbau des Herbizids Atrazin	120
6.9.1.5.	Michaelis-Menten-Kinetik und Reaktionen nullter Ordnung	122
6.9.2.	Wahrscheinlichkeitskinetik	122
6.10.	Einsatz von EROS7 zur Simulation massenspektroskopischer Prozesse	123
6.10.1.	Einstellungen in den Regeln, Unterschiede zu MASSIMO	123
6.10.2.	Simulation von Massenspektren	125
6.11.	Implementationsstrategien und Anbindung der Regeln	128
6.12.	Implementation des Programmsystems EROS7	129
6.12.1.	Aufbau der Reaktionsgenerierung	129
6.12.2.	Anbindung der Wissensbasis (Regeln) an EROS7	131
6.12.2.1.	Teile der Regelfunktion	131
6.12.2.2.	Datenaustausch zwischen den Regeln und dem Kernsystem	132

6.12.2.3.	Anbindung der Regelfunktion	134
6.12.2.4.	Anbindung anderer Sprachen, wie zum Beispiel Tcl	135
6.12.3.	Zugriffe auf die chemische Datenstruktur RICOS	136
6.12.3.1.	Indizes der Atome, Elektronensysteme usw.	136
6.12.3.2.	Eigenschaften	137
6.12.3.3.	Umordnung der Elektronensysteme (Reaktionen)	137
6.13.	Verwendung der Reaktionsgenerierung auch in anderen Programmen	138
7.	Anwendungen von EROS7	141
7.1.	Auf organisch-chemische Reaktionen	141
7.1.1.	Laborchemie und Chargenprozesse	141
7.1.1.1.	Bromierung von Phenol	141
7.1.2.	Kontinuierliche Prozesse, wie sie großtechnisch durchgeführt werden	145
7.1.2.1.	Estersynthese mit gleichzeitiger Extraktion	145
7.1.3.	Abbau von Umweltchemikalien	147
7.1.3.1.	Abbau von Prometon	148
7.1.3.2.	Jährliche Ausbringung von s-Triazinen	151
7.1.4.	Pharmakokinetik	152
7.1.4.1.	Abbaureaktionen	152
7.1.4.2.	Mehrfachdosierung	153
7.1.4.3.	Ausscheidung	154
7.1.4.4.	Absorption des Arzneimittels	155
7.1.4.5.	Das Gewebe als Arzneimittelspeicher	156
7.1.4.6.	Wirkkompartiment	158
7.1.4.7.	Unregelmäßige Dosierung	164
7.1.4.8.	Kinetik nach Michaelis-Menten und nullter Ordnung	164
7.1.5.	Syntheseplanung	166
7.1.6.	Erzeugung eines vollständigen Satzes von Verbindungen	169
7.1.6.1.	Mögliche Produkte einer Diels-Alder-Reaktion	169
7.1.6.2.	Chlorierte Dioxine	171
7.1.7.	Kombinatorische Chemie	177
7.1.7.1.	1,4-Benzodiazepine	177
7.1.7.2.	Substituierte Pyrazole	180
7.1.7.3.	Tripeptide aus drei möglichen Aminosäuren	182
7.2.	Simulation von Massenspektren	186
7.3.	Anwendungen mit mehreren Reaktoren	188
7.3.1.	Reaktionsvorhersage und Simulation der Massenspektren der Produkte	188
7.3.2.	Simulation des Spektrums einer Reaktionsmischung	194
8.	Zusammenfassung und Ausblick	197
8.1.	Massenspektrensimulation	197
8.2.	Neue Datenstruktur und Reaktionsvorhersage	197
8.3.	Arten der Reaktionsführung	198
9.	Für die Darstellungen verwendete Programme.	201
10.	Literatur	203

Anhänge

Anhang A. User-Manual zum EROS7-System	211
A.1. Aufbau von EROS7	211
A.2. Benutzerschnittstelle	211
A.2.1. Die Regeln	211
A.2.2. Die Aufrufparameter	212
A.2.2.1. Eingabedatei	212
A.2.2.2. Regeln	212
A.2.2.3. Steuerung der Reaktionsgenerierung	213
A.2.2.4. Traceoutput	213
A.2.2.5. Ausgabedateien	213
A.3. Ein Beispielaufruf von EROS7	214
Anhang B. Manual zum Abfassen neuer Regeln für EROS7	215
B.1. Gliederung der Regeln	215
B.2. Globale Regelinformationen	216
B.2.1. Variablen	217
B.2.2. Name und Datum	219
B.2.3. Reaktoren und Phasen	219
B.2.4. Phaseneigenschaften	221
B.2.4.1. INERT	222
B.2.4.2. MONOMOLEC	222
B.2.4.3. MIX (Modus von EROS6) und MIX_NO_A_A	222
B.2.4.4. SURFACE	222
B.2.4.5. TUBE (Modus von EROS5)	223
B.2.5. Kinetik	223
B.2.6. Weitere globale Einstellungen	226
B.2.7. Vorbehandlungsfunktionen für die Ausgangsmaterialien	226
B.2.8. Vorbehandlungsfunktionen für die Edukte	227
B.2.9. Die Verteilungsfunktion	227
B.2.10. Die Modellfunktionen	230
B.2.11. Die Cleanup-Funktion	231
B.3. Die einzelnen Reaktionsregeln	231
B.3.1. Regelname, Attribute	232
B.3.2. Die Reaktionssubstruktur	232
B.3.3. Bedingungen	233
B.3.3.1. Funktionalität	234
B.3.3.2. Der Rückgabewert	236
B.3.4. Die Reaktionsfunktion	237
B.3.4.1. Eduktfunktion	237
B.3.4.2. Reaktionsgenerierung	237
B.3.4.3. Produktfunktion	246
B.3.4.4. Ein sehr einfaches Beispiel	247
B.3.4.5. Neuronale Netze	249
Anhang C. 103 Moleküle für die MS-Bewertung	251

Anhang D. Von FRANZ bewertete Reaktionen.	253
D.1. Bei Verwendung von Umlagerungsreaktionstypen	254
D.1.1. Carbonyleliminierung Klasse 1-Daten	254
D.2. Ohne Umlagerungsreaktionstypen	260
D.2.1. α -Spaltung	260
D.2.2. Oniumreaktionen	264
D.2.3. Carbonyleliminierungen	265
D.2.4. Mc Lafferty-Reaktionen	266
Anhang E. Chlorierte Dioxine	267
E.1. Monochlordioxine	267
E.2. Dichlordioxine	267
E.3. Trichlordioxine	267
E.4. Tetrachlordioxine	268
E.5. Pentachlordioxine	269
E.6. Hexachlordioxine	269
E.7. Heptachlordioxine	269
E.8. Octachlordioxin	270
Anhang F. Verwendete Regeldateien	271
F.1. Regeln für den Abbau von s-Triazinen mit EROS7	271
F.2. Regeln für die Simulation von Massenspektren mit EROS7	284
Anhang G. Publikationen.	297
Anhang H. Lebenslauf	299

1. Einleitung

In den letzten Jahrzehnten hat die Computertechnologie eine starke Entwicklung erlebt und damit fast alle wissenschaftlichen und gesellschaftlichen Bereiche verändert. In nahezu jedem Büro ist heute ein Computer zu finden, der einem bei der Bewältigung der wachsenden Aufgaben zur Seite steht. Dies trifft auch in zunehmendem Maße auf die Chemie zu. Um so wichtiger wird das Gebiet der Simulation von Reaktionen. Da die Reaktionen ein Kernpunkt chemischer Experimente sind, sind die Bereiche zahlreich, in denen die Reaktionsvorhersage eingesetzt werden kann.

In unserer Zeit, in der alle Vorgänge immer stärker rationalisiert werden, wird der Raum für eine zeit- und kostenintensive Forschung selten. Hinzu kommen speziell in der Chemie zunehmende Auflagen des Umweltschutzes. Immer mehr Abwägungen und Versuche bei gleichzeitig geringerem Zeit- und Kostenaufwand sind nur unter Zuhilfenahme des Computers möglich. Hierzu werden entsprechende Simulationsverfahren benötigt.

Im ersten Teil der vorliegenden Arbeit wird versucht die im Massenspektrometer ablaufenden Prozesse [1] und damit die Spektren organischer Verbindungen vorherzusagen. Für einen routinemäßigen Einsatz sind dazu Verfahren notwendig, die es gewährleisten, daß die Ergebnisse spätestens nach einigen Minuten bereit stehen. Das schränkt die Methoden meist auf empirische und semiempirische Verfahren ein. Im Kapitel 4 werden die Reaktivitäten der Fragmentierungen und Umlagerungen nach einer Elektronenstoßionisation unter Verwendung von Techniken des automatischen Lernens aus Datenbanken abgeleitet. Die gewonnenen Zusammenhänge zwischen der Reaktivität und der Struktur können anschließend zur Simulation von Massenspektren eingesetzt werden.

In einem zweiten Teil dieser Arbeit wird die Entwicklung eines neuen Reaktionsvorhersageprogramms EROS7 (*Erzeugung von Reaktionen für die Organische Synthese*) vorgestellt, das auf einer Molekülorbital-orientierten chemischen Datenstruktur basiert. Damit kann eine Reihe von Reaktionen simuliert werden, die die meisten bisherigen Systeme nicht vornehmen können, da sie auf der Darstellung der Strukturen als Bindungslisten beruhen. Bindungslisten sind Darstellungen der Strukturen nach Lewis, bei denen Benzol alternierende Einfach- und Doppelbindungen besitzt und für jede Bindung die Bindungsordnung angegeben wird. EROS7 wurde so konzipiert, daß es in vielen Bereichen der Chemie eingesetzt werden kann. Es ist ein Expertensystem, das eine abgetrennte Wissensbasis besitzt, in der die Reaktionstypen und deren Funktionen zur Berechnung der Reaktivität kodiert sind. Dieses System wurde in Zusammenarbeit mit Frau Dr. Susanne Bauerschmidt und Frau Dr. Larissa Steinhauer erstellt.

2. Einsatzgebiete der Reaktionssimulation in der Chemie

2.1. Reaktionsvorhersage

Unter Reaktionsvorhersage versteht man die Simulation von einzelnen Reaktionen und deren Bewertung. Faßt man diese Reaktionen zu einem gesamten Reaktionsnetzwerk zusammen und liegen kinetische Daten als Bewertung der Reaktionen vor, lassen sich die Mengen in einer Reaktionsmischung nach Ablauf einer bestimmten Zeit vorhersagen. Gelten die Bewertungen für einen weiten Bereich der betrachteten Chemie, kann man so bestimmen, welche Produkte und wie viel bei einer geplanten Reaktion zu erwarten sind, ohne diese tatsächlich durchzuführen. Bei einer hohen Zuverlässigkeit des Systems kann es den synthetisch tätigen Chemiker bei der Planung seiner Experimente unterstützen. Er kann, bevor er eine unter Umständen aufwendige Reaktion versucht, testen, ob die geplante Reaktion eine annehmbare Ausbeute des gewünschten Produkts erwarten läßt.

Da in jüngerer Zeit der Umweltschutz mehr und mehr in den Vordergrund tritt, können mit einem solchen System auch die entstehenden Nebenprodukte mit ihren Mengen bestimmt werden. Sind unter den Nebenprodukten bedenkliche oder gar umweltgefährdende Stoffe dabei, können die Reaktionen und Bedingungen am Computer optimiert werden, bis man zu einem befriedigenden Ergebnis gekommen ist. Da dieser Vorgang im virtuellen Labor auf dem Computer erfolgt ist, ist dabei auch kein Abfall entstanden, um dessen Entsorgung man sich möglicherweise kümmern müßte.

2.2. Syntheseplanung

Die Reaktionsvorhersage ist eine indirekte Hilfe für den synthetischen Chemiker, da sie nur seine Vorschläge bewertet. Der direkte Ansatz wäre, ihm ausgehend von der Zielverbindung einen oder mehrere mögliche Wege der Synthese anzubieten und ihre Güte aufgrund von abgeschätzten Ausbeuten und der Verfügbarkeit bzw. den Kosten der Ausgangsmaterialien anzugeben. Gibt man einem System, das ganze Reaktionsnetzwerke vorhersagen kann, statt der verschiedenen Reaktionstypen retrosynthetische Schritte und kann es überdies während dieser Schritte kleine Moleküle hinzufügen, die bei der Umkehrung der retrosynthetischen Schritte - den Synthesereaktionen - abgespalten werden, ist es möglich, mit einem System zur Reaktionsvorhersage auch Syntheseplanungen vorzunehmen.

2.3. Kombinatorische Chemie

Ein neuerer Zweig der Chemie, die kombinatorische Chemie, versucht durch die gleichzeitige Synthese und anschließende Selektion aus der Vielzahl erzeugter Substanzen schneller zu einer pharmakologisch wirksamen Verbindung zu kommen. Um das Auffinden von der Zufälligkeit weg zu einer zielgerichteten Strategie hin zu entwickeln und zu einem höchst wirksamen Medikament bei guter Verträglichkeit zu kommen, müssen Tausende von Verbindungen auf ihre Ähnlichkeit hin untersucht werden. Dies kann wegen der großen Zahl der zu untersuchenden Substanzen nur durch entsprechende Methoden vom Computer erledigt werden. Dazu benötigt man allerdings zunächst die Strukturen aller im Versuch erzeugten Moleküle. Da die Moleküle in der Regel zu zwei Verbindungsklassen gehören, die über eine einfache und gut kontrollierbare Reaktion zu den erzeugten Produkten reagieren, kann bei einer entsprechenden Steuerungsmöglichkeit auch hierfür ein Programm zur Vorhersage von Reaktionen eingesetzt

werden. Die Bewertung der einzelnen Reaktionen spielt für diesen Fall keine Rolle, da die Reaktionsbedingungen so gewählt werden, daß alle möglichen Produkte etwa in gleicher Menge entstehen.

2.4. Reaktionsführung

In der chemischen Reaktionstechnik kann ein System zur Simulation von Reaktionen ebenfalls eingesetzt werden, wenn für die einzelnen Reaktionen Geschwindigkeitskonstanten zur Verfügung stehen. Läßt man zudem jedes Molekül mit jedem reagieren, hat man einen der Prototypen der Reaktoren, den absatzweise betriebenen, idealen Rührkessel (AIK). Ideal deshalb, weil zu jedem Zeitpunkt eine vollständige Durchmischung angenommen wird. Absatzweise bedeutet, daß man zu Beginn der Reaktionen eine bestimmte Menge der Reaktanden vorgibt und über die Zeit die Abnahme ihrer Konzentrationen und die Zunahme der daraus gebildeten Produkte beobachtet [2]. (Siehe auch 2.1). Einsatzgebiete dieser Art der Reaktionsführung sind die Synthesen, wie sie üblicherweise im Forschungslabor durchgeführt werden, und Chargenprozesse, die zur Herstellung kleiner Mengen von Arzneimitteln verwendet werden.

Geht man von den absatzweise betriebenen Reaktoren auf die großtechnisch eingesetzten, kontinuierlich arbeitenden über, kommt man zunächst zum idealen Strömungsrohr. Beim idealen Strömungsrohr wird von einer turbulenten Pfropfströmung durch das Rohr ausgegangen. Daraus ergeben sich die gleichen Konzentrationsverläufe der Verbindungen wie bei einem AIK, wobei die Reaktionszeit des AIK einer Wegstrecke im Rohr entspricht, für die die Reaktanden für den Durchfluß die gleiche Zeit benötigen haben.

Das Modell des Reaktors für die meisten großtechnischen Prozesse ist der kontinuierlich betriebene, ideale Rührkessel [3]. Auch für ihn wird zu jeder Zeit vollständige Durchmischung angenommen. In der Praxis kann dies bei großen Reaktoren oft nicht ganz erreicht werden, da die Edukte der Umsetzung kontinuierlich in den Reaktor zugegeben und die Reaktionsmischung auch dauernd entnommen werden und große Mengen schlechter zu mischen sind. Die Ergebnisse mit und ohne Berücksichtigung der mangelnden Durchmischung unterscheiden sich aber meist nur wenig. Bei der Berechnung der Konzentrationsverläufe der Verbindungen im Reaktor müssen zur Kinetik eines AIK auch die Zu- und Abflüsse der Substanzen einbezogen werden. In der Praxis wird manchmal auch noch die Zielverbindung oder auch ein Stoff, der unerwünschte Nebenreaktionen auslöst, aus der Reaktionsmischung extrahiert [3]. Eine Simulation solcher Vorgänge erfordert auch die Zuordnung der Verbindungen zu Phasen mit unterschiedlichen Konzentrationen und einer Quantifizierung der Diffusion in die andere Phase sowie die Angabe der Volumenströme der Phasen.

2.5. Massenspektroskopie

Bei der Aufnahme von Elektronenstoßmassenspektren, der bezüglich des Aufbaus der Struktur informationsreichsten Art von Massenspektren, finden im Massenspektrometer abhängig von der Struktur der Verbindung verschiedene Ionisations-, Fragmentierungs- und Umlagerungsprozesse statt. Aus der Stoffmenge der entstandenen Kationen und deren Masse inklusive der Isotopensignale läßt sich das Massenspektrum berechnen. Sind die Reaktionsprozesse die Reaktionen in einem Reaktionsnetzwerk, wobei nur monomolekulare Reaktionen stattfinden, können die Konzentrationen der Ionen mit einem Reaktionsvorhersageprogramm bestimmt werden. Reaktionen von den abgespaltenen Neutralfragmenten können vernachlässigt werden, da sie nicht zum beobachteten Massenspektrum beitragen.

2.5.1. Strukturaufklärung

Die verschiedenen Spektroskopiearten werden meist zur Strukturaufklärung eingesetzt, so auch

die Massenspektroskopie. Aus dem Massenspektrum einer unbekanntem Verbindung möchte man dabei deren Struktur ableiten. Abbildung 1 zeigt den häufigsten Einsatz der Massenspektroskopie durch den synthetisch arbeitenden Chemiker. Der Chemiker hat eine Idee, die gewünschte Verbindung - das Synthesziel - zu synthetisieren. Nach der Reaktion erhält er ein Gemisch verschiedener Produkte, das er mit einer GC-MS-Kopplung analysiert. Dabei wird das Substanzgemisch über den Gaschromatographen aufgespalten und mit einem Massenspektrometer das Spektrum und über den Gesamtionenstrom die Menge des Stoffes bestimmt.

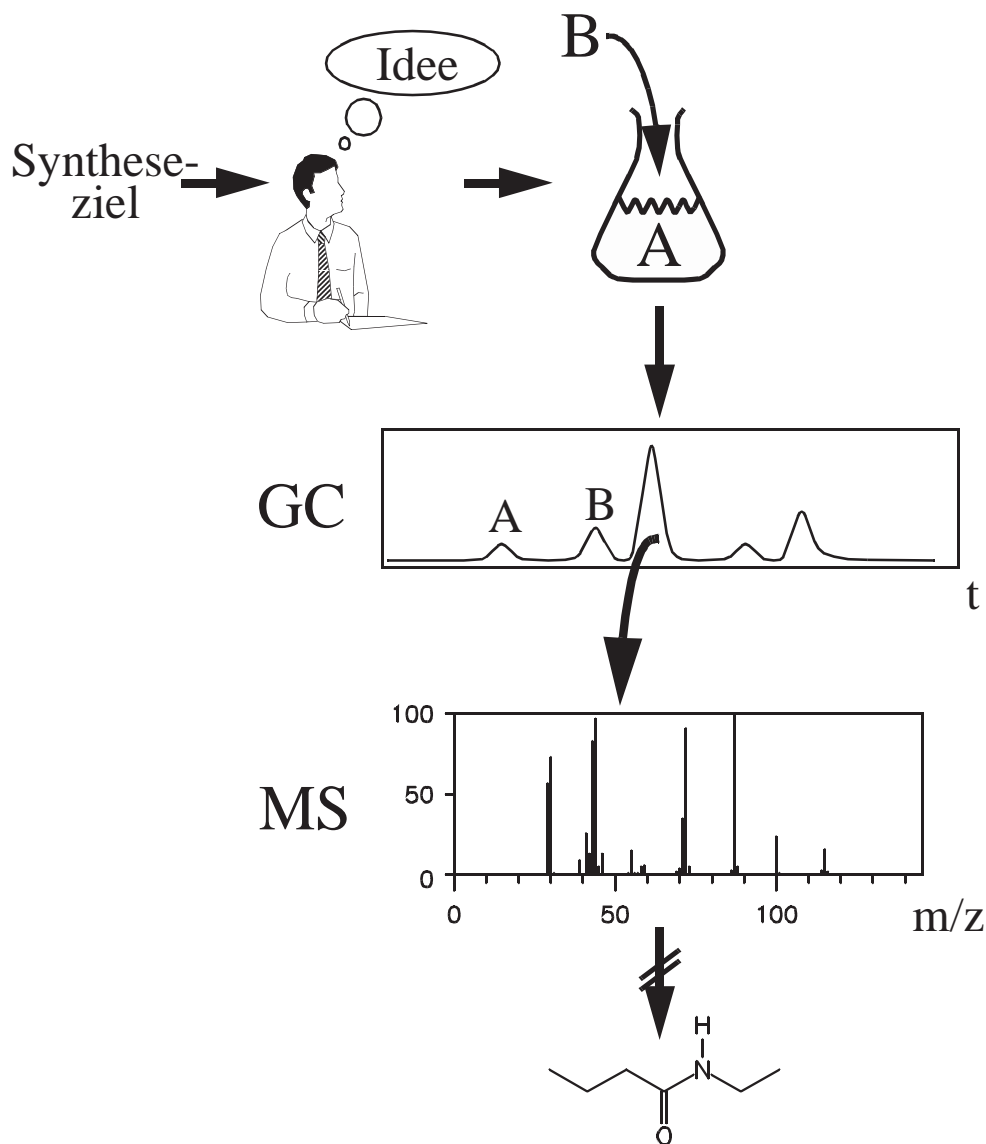


Abbildung 1: Üblicher Einsatz einer spektroskopischen Methode. Bei der Massenspektroskopie ist jedoch die Ableitung der Struktur aus dem Spektrum nur bedingt möglich.

Das Elektronenstoßmassenspektrum enthält zwar viele Informationen über die Struktur der Verbindung, sie lässt sich aber leider nur teilweise aus dem Spektrum ableiten. Hierzu gibt es Ansätze, in der Substanz enthaltene oder nicht enthaltene Strukturelemente, wie funktionelle Gruppen, vorherzusagen [4]. Die gesamte Struktur lässt sich jedoch nicht aus dem Massenspektrum ableiten. Deshalb wird versucht den umgekehrten Weg der Simulation des Spektrums (siehe 2.5) zu beschreiten, der in Abbildung 2 gezeigt ist. Der Chemiker kann in der Regel neben der Zielstruktur mehrere weitere möglicherweise entstandene Produkte angeben. Simuliert man für all diese Strukturen das Massenspektrum und vergleicht sie mit dem experimentellen Spek-

trum aus der GC-MS, kann man über deren Ähnlichkeit auf die richtige Struktur schließen.

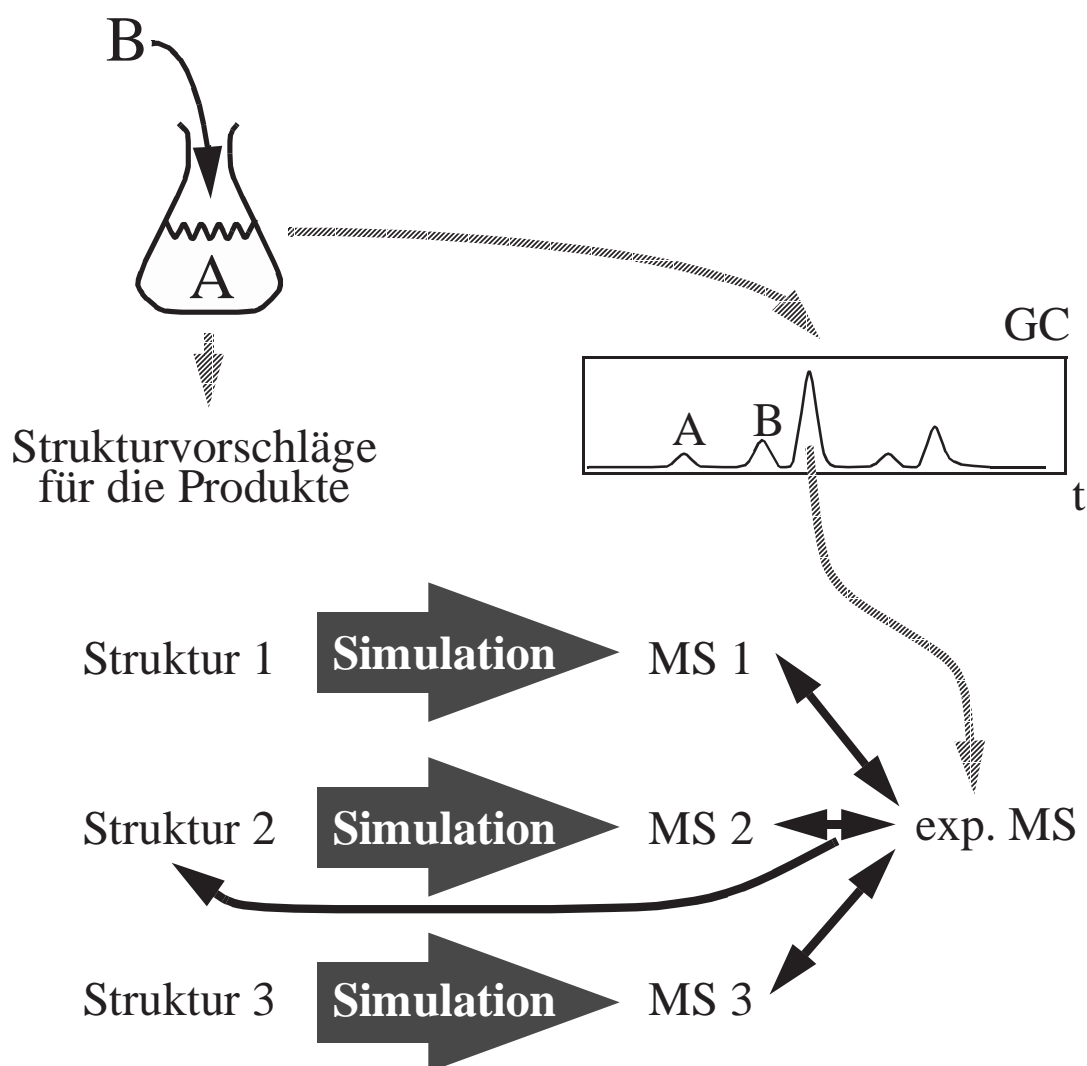


Abbildung 2: Indirekte Identifizierung der Struktur einer Verbindung.

Statt der Erzeugung der Strukturvorschläge durch den Chemiker, ist es auch möglich, diese durch den Computer erzeugen zu lassen. Hierzu benötigt man die oben erwähnte Ableitung von Strukturmerkmalen aus dem Spektrum. Kennt man nun aus den verschiedenen Spektroskopiearten, wie magnetisches Kernresonanzspektrum (NMR), Infrarotspektrum (IR), Massenspektrum (MS) und möglicherweise der Elementaranalyse vorhandene und abwesende Strukturelemente und die Summenformel, kann man mit einem Strukturgenerator unter diesen Randbedingungen alle möglichen Strukturenarten bestimmen. Kann man schließlich neben dem Massenspektrum auch alle anderen Spektren vorhersagen, verwendet man die erhaltenen möglichen Strukturen als Vorschläge und simuliert für sie die Spektren (siehe Abbildung 3).

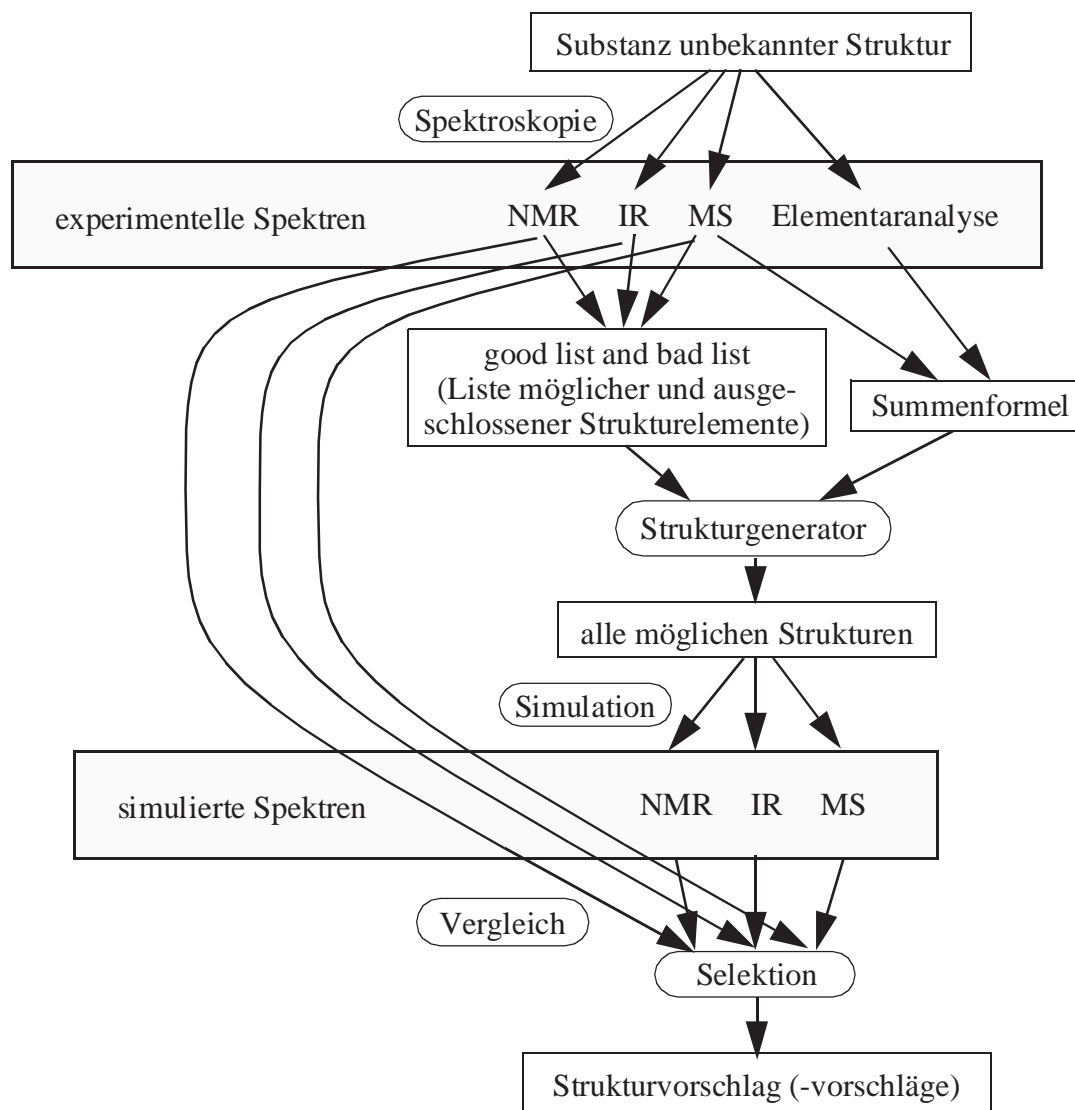


Abbildung 3: Automatische Ermittlung von Strukturvorschlägen aus den Spektren.

Alle Strukturen, deren vorhergesagte Spektren nicht mit dem experimentell bestimmten übereinstimmen, werden verworfen. Zurück bleibt eine kleine Zahl von Strukturvorschlägen. Systeme, die diese Techniken in sich vereinen, sind beispielsweise CHEMICS [5] und SpecInfo [6][7].

2.5.2. Sicherung der Spektrenqualität, Überprüfung von Datenbanken

Bei der Fülle von Spektren, die in einer Abteilung für Strukturaufklärung anfallen, speziell, wenn man an GC-MS-Spektren denkt, können nicht alle Spektren vom Experten ausgewertet werden. Dies könnte zur Folge haben, daß auch Spektren mit einer falsch zugeordneten Struktur in die Datenbanken aufgenommen werden. Hat man dagegen ein Programm, das Massenspektren zuverlässig vorhersagen kann, ist es möglich den Experten zu entlasten, daß er sich nur diejenigen Spektren mit ihren Strukturen genauer ansehen muß, bei denen das experimentelle und das simulierte Spektrum der zugeordneten Struktur größere Differenzen aufweisen. Auf diese Weise kann die Qualität einer Datenbank gesichert werden, während sie aufgebaut wird.

Da dies in der Vergangenheit nicht immer gegeben war und vor einigen Jahren die Analytik

noch nicht so ausgereift war, muß man damit rechnen, daß älteren Spektren, die in manchen heute eingesetzten Datenbanken enthalten sind, nicht immer die richtige Struktur zugeordnet ist [8]. Mit einem System zur Vorhersage von Massenspektren könnte man den Inhalt älterer Datenbanken automatisch überprüfen. Struktur-Spektren-Paare, die möglicherweise nicht zusammengehören, könnte dann ein Experte auf die richtige Zuordnung überprüfen. Dies ist im Gegensatz zur manuellen Kontrolle aller Einträge einer Datenbanken im Bereich des Möglichen, wenn eine gute Spektrenvorhersage zur Verfügung steht.

3. Komponenten eines Systems zur Simulation von Reaktionen

3.1. Grundsätzlicher Aufbau eines Systems zur Simulation von Reaktionen

Ein System zur Vorhersage von Reaktionen besteht grundsätzlich aus mehreren Teilen. Offensichtlich gehören die chemische Datenstruktur und der Reaktionsgenerator sowie der Aufbau des Reaktionsnetzwerks dazu. Hierin unterscheiden sich die Programme zur Reaktionssimulation nur wenig. Alle Programme zur Vorhersage von Reaktionen sind wissensbasierte Systeme, die auch Module für die Beschreibung der Reaktionstypen und Bewertung der Reaktionen enthalten. Die Art, in der die Reaktionstypen und Bewertungen festgelegt werden, teilt die existierenden Programme jedoch in mehrere Gruppen, was in den nächsten Abschnitten beschrieben wird. Eine Übersicht über alle Teile eines Programms zur Reaktionsvorhersage gibt Abbildung 4.

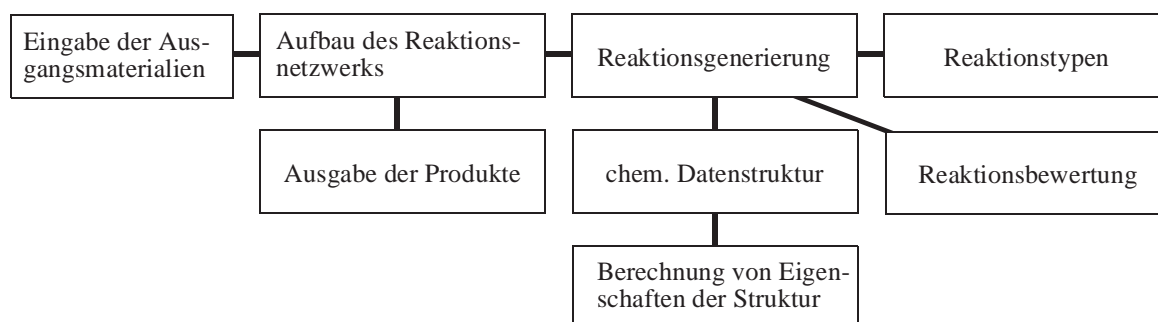


Abbildung 4: Generelles Aufbauschema eines Systems zur Reaktionsvorhersage.

3.2. Repräsentation chemischer Strukturen (Bindungsliste)

Die überwiegende Zahl der Programme zur Reaktionsvorhersage verwendet die Darstellung der chemischen Struktur als Bindungsliste, also in der Weise, wie man sie nach Lewis schreibt (z.B. Benzol mit alternierenden Einfach- und Doppelbindungen). Dies liegt einerseits daran, daß sich die Reaktionen für diese Darstellung der chemischen Struktur auf einfache Weise als Änderung der Bindungsordnungen und Zahl der freien Elektronen beschreiben lassen und andererseits an der Tatsache, daß eigentlich bei alle Reaktionen in computerlesbarer Form, die in Datenbanken gespeichert sind, die Edukte und Produkte auf diese Weise kodiert sind. Aus ihnen lassen sich Bewertungen für einzelne Reaktionstypen ableiten. Die Beschränkungen der Darstellung als Bindungsliste, daß beispielsweise keine Verbindungen mit Mehrzentren-Bindungen wie Borane dargestellt, sowie die Struktur von Substanzen mit komplexen π -Systemen nicht eindeutig kodiert werden können, hat man dabei für die einfachere Kodierung der Strukturen und Reaktionen in Kauf genommen.

3.3. Beschreibung von Reaktionstypen

Bei der Beschreibung der Reaktionstypen, die ausgeführt werden, teilen sich die Systeme dagegen in zwei Gruppen. Auch die Vorläuferprogramme des im Rahmen dieser Arbeit entwickelten

Programms EROS7 (*Erzeugung von Reaktionen für die Organische Synthese*) gehören zu diesen Gruppen. EROS5 [9] hat fest eingebaute Definitionen von Reaktionstypen, die allerdings gesteuert werden können. EROS6 [10][11][12] hat dagegen eine externe Wissensbasis mit einer eigenen Sprache, in der die Reaktionstypen kodiert werden.

3.4. Bewertung der Reaktionen

Auch bei dem Programmteil, der die erzeugten Reaktionen bewertet, gibt es starke Unterschiede. Grundsätzlich gibt es die Möglichkeit einer Reaktion zuzuordnen, ob sie stattfindet oder nicht. Zur Gruppe der Programme, die selbst entscheiden, welche Reaktionen ablaufen, gehört auch EROS5, das aufgrund von eingestellten, topologischen Eigenschaften und der Reaktionsenthalpie selbst entscheidet. Neben diesen Systemen gibt es eine weitere Gruppe von Programmen, die die Entscheidung, ob eine Reaktion stattfindet oder nicht, dem Chemiker überlassen, wie das in umgekehrter Richtung arbeitende Syntheseprogramm IGOR [13]. Darüber hinaus kann man durch die Angabe von Geschwindigkeitskonstanten oder Reaktionswahrscheinlichkeiten genauer angeben, wie gut eine Reaktion abläuft. EROS6 kann aus physikochemischen Eigenschaften der Edukte und Produkte am Reaktionszentrum auch Geschwindigkeitskonstanten oder Reaktionswahrscheinlichkeiten bestimmen und sie für die Berechnung der Endkonzentrationen verwenden.

3.5. Massenspektrenvorhersage

Erweitert man ein System zur Vorhersage von Reaktionen um ein Modul, das abhängig von der Ladung und den Konzentrationen der Produkte sowie ihren atomaren Zusammensetzungen die Intensitäten der Signale im Massenspektrum berechnet, kann das Reaktionsvorhersageprogramm auch zur Simulation von Massenspektren eingesetzt werden. Dabei wird ferner davon ausgegangen, daß die in der Ionisationskammer ablaufenden Prozesse mit dem System als Reaktionen formuliert werden können.

Die in CHEMICS (siehe 2.5.1) eingebaute Vorhersage von Massenspektren enthält etwa 150 Fragmentierungsregeln [5]. Die einzelnen Fragmentierungen werden allerdings nur mit den beiden Werten null und eins (Reaktion läuft ab bzw. läuft nicht ab) bewertet. Es kann deshalb keine Intensitäten der Signale simulieren [14].

Im Gegensatz dazu kann MASSIMO [15], das aus EROS6 hervorgegangen ist und um die notwendigen Programmteile ergänzt wurde, mit entsprechenden Bewertungen in den Regeln auch Intensitäten der Signale vorhersagen und kommt mit etwa einem Dutzend verschiedener Fragmentierungs- und Umlagerungstypen aus. Die Regeln sind die Wissensbasis von MASSIMO, in denen die Reaktionstypen und Bewertungsfunktionen zur Berechnung der Reaktionswahrscheinlichkeiten kodiert sind.

3.6. Verwendung zur Syntheseplanung

Soll ein Reaktionsvorhersageprogramm zur Syntheseplanung eingesetzt werden, müssen an das System zusätzliche Forderungen gestellt werden. Bei vielen Reaktionen entstehen kleine Moleküle als Coprodukte, wie zum Beispiel Wasser bei einer Veresterung. Für eine Syntheseplanung müssen diese während des retrosynthetischen Schrittes hinzugefügt werden. Da diese kleinen Moleküle nicht zusammen mit dem Synthesziel angegeben werden, muß sie das Programm während der retrosynthetischen Schritte hinzufügen können. Weil EROS5 auch noch die Bewertung für die Syntheseplanung gegenüber der Reaktionsvorhersage umstellen konnte, war es möglich, mit EROS5 sowohl Reaktionsvorhersagen wie Syntheseplanungen vorzunehmen.

Bei der Weiterentwicklung der Syntheseprogramme ging man oft von den reinen retrosynthetischen Schritten zu wesentlich komplexeren Transforms über, die häufig funktionelle

Gruppen kanonisieren und damit auch Elemente austauschen. Unter Kanonisierung versteht man hier die Umwandlung verschiedener, gleich reagierender und ineinander überführbarer funktioneller Gruppen in eine stellvertretende. Daneben können sich auch mehrere Teile der Verbindung gleichzeitig verändern, ohne das als Reaktion zu formulieren, oder eine umfangreiche Suche und Bewertung von strategischen Bindungen durchgeführt werden. Dies stellt an die Syntheseplanung Anforderungen, die meist von den Programmen zur Reaktionsvorhersage nicht erfüllt werden konnten. Durch den Wechsel von retrosynthetischen Schritten zu Transforms, ist es gelungen, durch das Überspringen mehrerer Schritte gezielter zu den Vorstufen zu gelangen. [16] Die Einführung der komplexeren Transforms war auch der Grund, weshalb sich die Syntheseplanungsprogramme aus ihren Vorläufern zur Reaktionsvorhersage als eigenständige Werkzeuge entwickelt haben. Dies ist so geschehen bei der Fortentwicklung von EROS5, wobei EROS6 für die Reaktionsvorhersage und WODCA (*Workbench for the Organisation of Data for Chemical Applications*) [16] für die Syntheseplanung hervorgingen. Da EROS6 und WODCA aus der gleichen Entwicklung stammen, haben sie intern auch die gleiche Darstellung der chemischen Struktur als Bindungsliste und eine externe Wissensbasis mit den Reaktionstypen bzw. Transforms. Den gleichen Aufbau hat unter anderem auch das Syntheseplanungsprogramm LHASA (*Logic and Heuristic Applied to Synthesis Analysis*) [17].

4. Simulation von Massenspektren auf der Basis von Bindungslisten

4.1. Methodik

Diejenigen Massenspektren, die am meisten Information über die Struktur enthalten und schon seit längerem aufgenommen werden, sind die 70eV Elektronenstoßmassenspektren. Sie ergeben sich dadurch, daß die Moleküle durch einen Elektronenstoß ionisiert und gleichzeitig angeregt werden. Die energiereichen Radikalkationen lagern sich um und fragmentieren unter Abspaltung von Neutralfragmenten. Abhängig von der Häufigkeit der verschiedenen Kationen und ihren Massen ergibt sich die Intensitätsverteilung bei den unterschiedlichen Massen im Massenspektrum.

Ist es möglich die Umlagerungen und Fragmentierungen inklusive deren Umsätze zu berechnen, kann man auch die sich daraus ergebenden Massenspektren simulieren. Die wichtigsten Umlagerungs- und Fragmentierungstypen sind schon eine Weile bekannt und können beispielsweise in [1] und [18] nachgelesen werden. Über die Reaktivitäten der einzelnen Umlagerungen und Fragmentierungen findet man in der Literatur dagegen oft nur grobe Bewertungen oder gar keine Angaben. Es stehen jedoch Datenbanken mit 70eV Elektronenstoßmassenspektren und ihren Strukturen mit einigen tausend Einträgen [19][20] zur Verfügung. Solche Spektrensammlungen sind oft auch in Programmsysteme integriert, mit denen Ähnlichkeits- und Substruktursuchen auf der darunterliegenden Datenbank durchgeführt werden können, wie MassLib [21][22] oder SpecInfo [6][7]. Mit diesen Programmen hat man starke Werkzeuge zur Strukturauflösung, sofern viele Massenspektren ähnlicher Substanzen in der Datenbank enthalten sind. Hat man dagegen die Reaktivitäten der Umlagerungs- und Fragmentierungsregeln, ist es möglich, auch für solche Substanzen Massenspektren zu simulieren, für die keine ähnlichen in der Datenbank enthalten sind.

Da Massenspektren die ganze Information über die Struktur der Verbindung enthalten, sollte es möglich sein, die fehlenden Reaktivitäten für die im Massenspektrometer ablaufenden Reaktionen abzuleiten. Dabei ist es nicht nötig, möglichst viele ähnliche Strukturen in der verwendeten Datenbank zu haben. Wichtig ist, daß für alle Reaktionstypen der gesamte Raum der elektronischen und sterischen Verhältnisse (alle Kombinationsmöglichkeiten der Verhältnisse) des Reaktionszentrums bei den Fragmentierungen abgedeckt ist. Dies kann aber auch in ganz verschiedenen Molekülen erreicht werden. Als Ergebnis der Analyse der Massenspektren aus der Datenbank erhält man für jeden Reaktionstyp eine Datenbank von Reaktionen mit ihren Reaktivitäten (siehe Abbildung 5). Stellt man nun einen funktionalen Zusammenhang zwischen den Eigenschaften, z.B. physikochemischen Parametern, des Reaktionszentrums und der Reaktivität her, kann man für beliebige Reaktionszentren die Reaktivität bestimmen. Der funktionale Zusammenhang kann dabei explizit in Form einer mathematischen Funktion oder implizit in einem neuronalen Netz beschrieben sein. Das Programm FRANZ (*F*ragmentation and *R*earrangement *A*nalyzer) [23][24] ist das Programm zur Analyse der Massenspektren und erstellt aus einer Datenbank mit Massenspektren und den dazugehörigen Strukturen für die verschiedenen, vorgegebenen Fragmentierungstypen Datenbanken mit Reaktionen, denen eine Reaktionswahrscheinlichkeit zugeordnet wurde. Über eine Statistik oder neuronale Netze lassen sich aus diesen die bewerteten Fragmentierungsregeln ableiten.

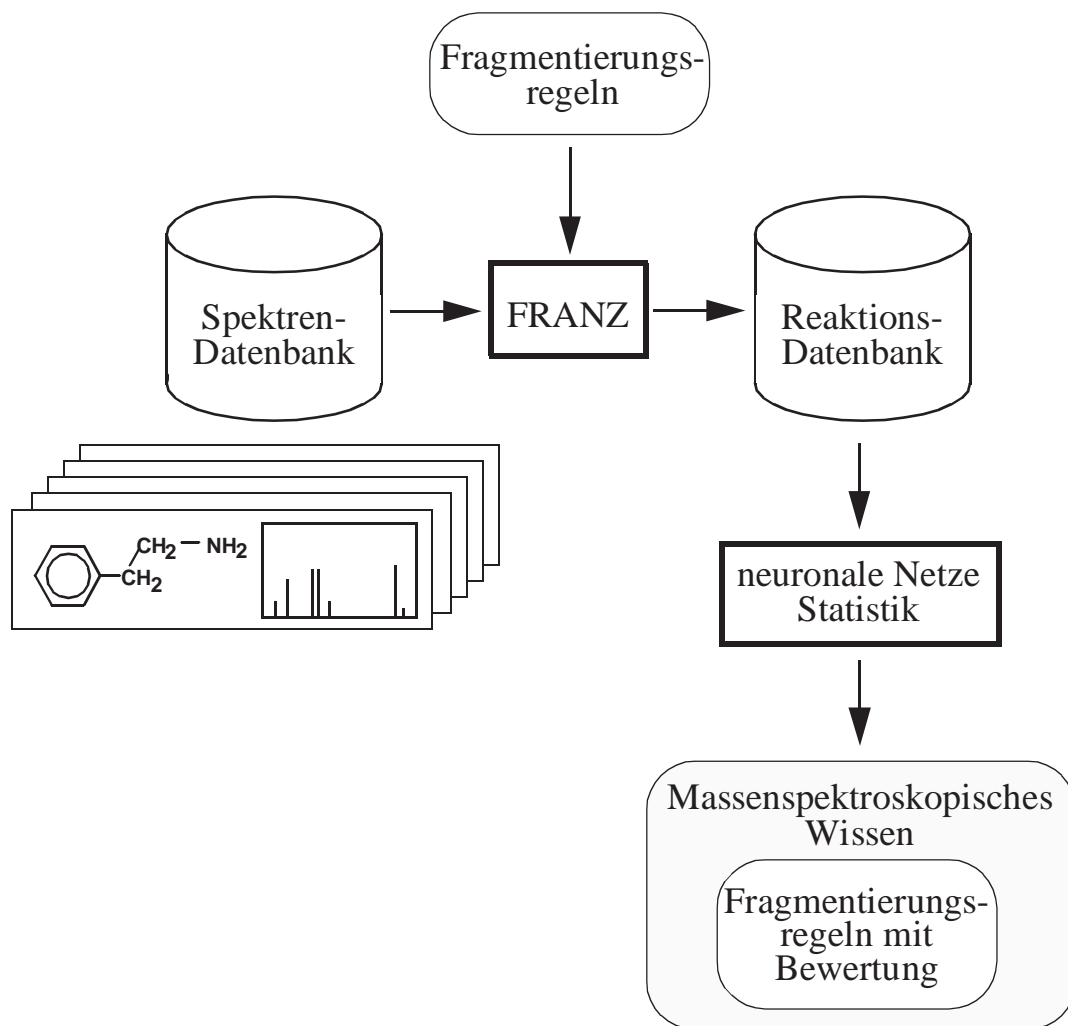


Abbildung 5: Ableitung der bewerteten Fragmentierungsregeln aus Datenbankspektren.

Ausgehend von den Fragmentierungsregeln sowie den funktionalen Zusammenhängen zwischen den Reaktivitäten und den Eigenschaften des Reaktionszentrums für die einzelnen Regeln lassen sich dann Massenspektren simulieren. Mit MASSIMO (MAss Spectra SIMulatOr) [23][15] kann man dann Massenspektren ausgehend von der Struktur einer Verbindung und unter Verwendung der mit FRANZ gewonnenen bewerteten Fragmentierungsregeln simulieren.

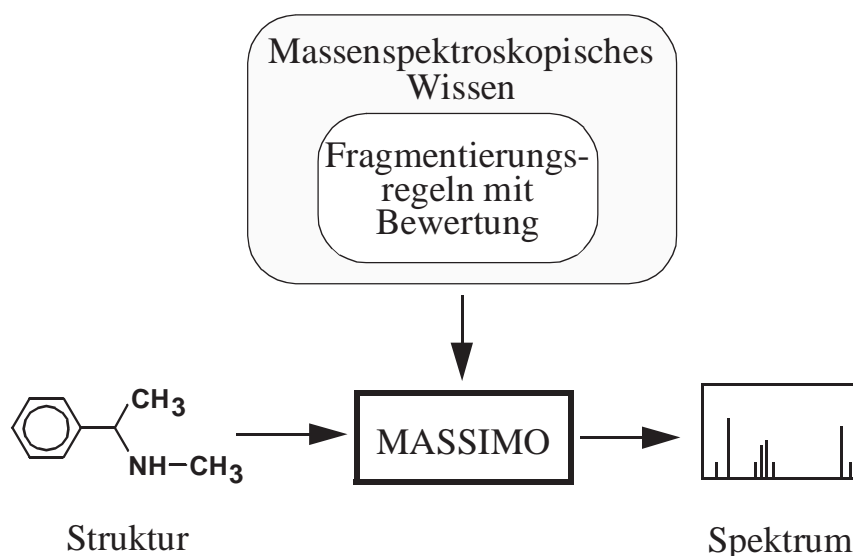


Abbildung 6: Simulation von Massenspektren aufgrund des abgeleiteten massenspektroskopischen Wissens.

Gelingt es nicht, den gesamten Raum der elektronischen und sterischen Verhältnisse des Reaktionszentrums eines Reaktionstyps abzudecken, werden Reaktionen, bei denen die Verhältnisse des Reaktionszentrums in einem unbesetzten Teil des Raums zu liegen kommen, möglicherweise falsch bewertet. In solchen Fällen gibt es dann Einschränkungen für die Anwendbarkeit der bewerteten Fragmentierungsregeln.

4.2. Erstellung der Wissensbasis

Die Wissensbasis besteht aus den Regeln, in denen die verschiedenen Fragmentierungs- und Umlagerungstypen kodiert sind. Für die Analyse von Massenspektren enthalten die Regeln keine Reaktivitätsbewertung.

4.2.1. Fragmentierungsregeln

Begonnen wurde mit den Fragmentierungs- und Umlagerungsreaktionen, wie sie in Abbildung 7 gezeigt sind, sowie der n - und π -Ionisation. Angaben wie 3–7 besagen, daß an dieser Stelle eine Kette von Atomen einer Länge zwischen drei und sieben ist. Dies ist der Satz aller möglichen Reaktionstypen. Welche Reaktionen auch tatsächlich im Massenspektrometer stattfinden und in welchem Ausmaß, wird von FRANZ (Fragmentation and Rearrangement ANalyzer) aus den Reaktionstypen, dem experimentellen Massenspektrum und der Struktur der Verbindung ermittelt [23][24]. Ergebnis der Berechnungen von FRANZ sind dann Datenbanken für die einzelnen Reaktionstypen mit Reaktionen diesen Typs, denen eine Reaktionswahrscheinlichkeit zugeordnet ist.

Fällen im Bereich niedriger Massen abgeschnitten sind. Um auch schnelle Folgereaktionen erkennen zu können, bei denen die Masse des Zwischenprodukts nicht im Spektrum zu sehen ist, wird auch von Produkten, deren Signale nicht zu sehen sind, noch eine Reaktionsebene erzeugt. Sind auch deren Produkte nicht im Massenspektrum zu sehen, werden sie verworfen.

Da einige Massenspektren bei fast allen Massen im Spektrum mindestens ein Signal von bis zu zwei Prozent haben, kann bei FRANZ ein Schwellwert eingestellt werden, ab dem ein Signal als gesichert im Massenspektrum enthalten angesehen wird.

Ist das ganze Reaktionsnetzwerk erzeugt, werden die Umlagerungssysteme zusammengefaßt, da ihre Ionen im Massenspektrum nicht unterschieden werden können, und für alle Fragmentierungen aus den Signalintensitäten im Massenspektrum die Reaktionswahrscheinlichkeiten berechnet. Da auch verschiedene Reaktionswege zu gleichen oder gleichschweren Kationen führen können, kann nicht für jede Reaktion eine eindeutige Reaktionswahrscheinlichkeit bestimmt werden. FRANZ ermittelt deshalb für jede Reaktion auch eine Unter- und Obergrenze für die Wahrscheinlichkeit. Ob die ermittelte Reaktionswahrscheinlichkeit eher an der Ober- oder Untergrenze liegt, wird durch Heuristiken festgelegt (siehe 4.2.2.4).

4.2.2.1. Reaktionswahrscheinlichkeiten

Die Reaktionswahrscheinlichkeiten werden aus den Intensitäten der Signale des experimentellen Massenspektrums berechnet. Zur Ableitung der Reaktionswahrscheinlichkeiten werden Massenspektren aus Datenbanken verwendet, die oft nicht die zu fordernden Qualitätskriterien aufweisen. So werden die Massenspektren meist unterhalb einer Masse von m/z 20 bis 40 abgeschnitten und manchmal enthalten die Massenspektren bei fast jeder Masse zwar ein sehr kleines, aber wahrnehmbares Signal. Dies wird von FRANZ dadurch berücksichtigt, daß Reaktionen, deren Produktkationen eine kleinere als die minimale beobachtete Masse im Spektrum besitzen, nicht berücksichtigt werden und Signale mit einer Intensität unterhalb eines Schwellwerts als nicht sicher im Massenspektrum enthalten angesehen werden. Da meist auch die Intensitäten der Isotopensignale, die zur Aufteilung der Intensitäten der Signale auf nicht isomere Kationen gleicher Masse verwendet werden, geringer als die theoretischen Werte sind, wird vor der Auswertung zu Signalen mit einer geringeren Intensität ein kleiner Betrag hinzuaddiert, z.B. 0.5 ‰ bei Signalen bis zu einer Intensität von 9 ‰.

Nachdem alle Ionen, die durch Umlagerungen miteinander verknüpft sind, zu einer virtuellen Struktur - dem Umlagerungssysteme - zusammengefaßt wurden, werden für die Reaktionen über die folgende Gleichung die Reaktionswahrscheinlichkeiten P_i berechnet. Die Gleichung entspricht einer Kinetik erster Ordnung, U_i sind die Umsätze und z_i die Symmetriezahlen für die von A ausgehenden Reaktionen, $[A]_0$ die Anfangskonzentration und $[A]_t$ die Restkonzentration der Ausgangsverbindung nach dem betrachteten Reaktionszeitraum [25].

$$P_i = 1 - \left(\frac{[A]_t}{[A]_0} \right)^{\frac{U_i}{\sum_j U_j z_j}} \quad (1)$$

In gleicher Weise werden auch die Unter- und Obergrenzen berechnet, wobei der Aufbau des Reaktionsnetzwerks, das Auftreten mehrerer Fragmente mit gleicher Masse und der Abgleich der Isotopensignale mit den experimentellen Signalen eine Rolle spielen (siehe 4.2.2.2).

4.2.2.2. Sicherheit der Bewertungen

Es gibt verschiedene Gründe, durch die FRANZ die Möglichkeit genommen werden kann, den Reaktionen eindeutige Wahrscheinlichkeiten zuzuordnen. Wird beispielsweise ein Fragment auf zwei verschiedenen Reaktionswegen gebildet, kann dies vollständig über den einen, wie auch über den anderen Weg oder einer Mischung aus beiden geschehen.

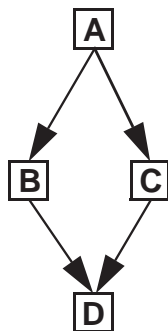


Abbildung 8: Zwei Wege zum gleichen Fragment.

Für jede dieser Reaktionen ist die Untergrenze der Reaktionswahrscheinlichkeit null, wenn die Fragmente B und C nicht im Spektrum zu sehen sind, und die Obergrenze die Wahrscheinlichkeit, wenn das Fragment D ausschließlich über diesen Weg gebildet wird. Sind die Fragmente B und/oder C im Massenspektrum zu sehen, erhöhen sich die Ober- und Untergrenzen der Reaktionswahrscheinlichkeiten für die Reaktionen $A \rightarrow B$ und $A \rightarrow C$ um den Teil, der der im Spektrum sichtbaren Fragmente B und C entspricht. Für reaktive Reaktionen mit einer Wahrscheinlichkeit größer ca. 0.7 hat dies zur Folge, daß über diese Reaktion eigentlich keine sichere Aussage für die Reaktionswahrscheinlichkeit mehr getroffen werden kann, da sie zwischen 0.0 und 0.7 bzw. noch höher liegen, also fast alle Wahrscheinlichkeitswerte besitzen kann.

Als Güte der Bewertung verwendet FRANZ die Differenz zwischen Unter- und Obergrenze der ermittelten Reaktionswahrscheinlichkeiten. Überschreitet diese Differenz den Wert 0.2 nicht, wird sie der Reaktionsdatenbank für Klasse-1-Daten hinzugefügt, für einen Wert bis 0.4 der für Klasse-2-Daten. Die Grenzwerte von 0.2 und 0.4 sind dabei einstellbar. Darüber hinaus enthält FRANZ auch noch einen Mechanismus, um den Datenraum der Reaktionen möglichst gleichmäßig abzudecken. Dazu werden die einzelnen Reaktionen in verschiedene Äquivalenzklassen eingeteilt und von Reaktionen einer sehr häufig vorkommenden Klasse wird nur ein Teil der Reaktionen in die Datenbank aufgenommen.

Da solche konkurrierenden Reaktionswege auch systematisch für Verbindungen mit einer bestimmten Substruktur vorkommen können und dann für diese in den von FRANZ erstellten Reaktionsdatenbanken keine Reaktionen enthalten sind, ist bei der Erstellung der Reaktionstypen darauf zu achten, daß dieser Fall nicht eintreten kann.

Isomere Fragmente können durch ihre gleiche Masse und Isotopensignale im Massenspektrum nicht unterschieden werden. Die Kationen, die ineinander umlagern, werden deshalb - wie schon erwähnt - zu Umlagerungssystemen zusammengefaßt. Haben mehrere Umlagerungssysteme oder Fragmente außerhalb von Umlagerungssystemen das gleiche Isotopenmuster oder Muster, die sich bezüglich der Intensitätsauflösung nicht unterscheiden, können deren Intensitäten ebenfalls nicht festgestellt werden. Auch in solchen Fällen ist die Untergrenze der Reaktionswahrscheinlichkeiten der Reaktionen von und zu solchen Fragmenten oder Umlagerungssystemen gleich null. Kann ein Folgefragment eindeutig einer Signalgruppe im Massenspektrum zugeordnet werden, erhöht sich die Untergrenze auf den Betrag, der nötig ist, um dieses Folgeprodukt zu bilden.

Da bei Umlagerungsreaktionstypen grundsätzlich eine zum Edukt isomere Verbindung entsteht, stellen diese Reaktionstypen eine besondere Schwierigkeit dar. Für diese Reaktionstypen, die immer in Umlagerungssystemen liegen, kann FRANZ nur sehr ungenaue Reaktionswahrscheinlichkeiten ermitteln, da sich die Signale des Edukts und Produkts im Massenspektrum nicht unterscheiden und somit deren Konzentrationen nicht bestimmt werden können. Stellt FRANZ eine Untergrenze der Wahrscheinlichkeit größer null fest, wird die Umlagerungsreak-

tion als reaktiv eingestuft. Analoges gilt für Reaktionen mit einer Obergrenze der Wahrscheinlichkeit von null. Sie werden als nicht reaktiv klassifiziert. FRANZ kann für Reaktionstypen, die eine Umlagerung kodieren, nur in wenigen Fällen den Reaktionen eine Reaktivität zuordnen, die dann nur angibt, ob diese Umlagerung stattfindet oder nicht. Umlagerungsreaktionstypen haben daneben noch weitere Effekte, die im nächsten Kapitel beschrieben sind.

4.2.2.3. Umlagerungen als Reaktionstyp

Bei der Verwendung der Regeln, die die Wasserstoffumlagerung einschließen, aus Abbildung 7 auf Seite 16 hat sich gezeigt, daß FRANZ fast jedem Signal im Massenspektrum ein Fragment zuordnen kann. Dies erreicht FRANZ häufig dadurch, daß Wasserstoffatome solange radikalisch umgelagert werden, bis ein Fragment der Masse abgespalten werden kann, das im experimentellen Massenspektrum enthalten ist. Häufig hat dies nicht nur unübersichtliche Reaktionswege zur Folge, die nur teilweise korrekt sind, sondern bietet oft auch Reaktionswege, bei denen ein Fragment auf mehr als einer Reaktionssequenz gebildet werden kann. Das wiederum hat eine Verringerung der Sicherheit bei der Ermittlung der Reaktionswahrscheinlichkeit zur Folge.

Ein weiteres Problem erkennt man, wenn man zwei α -Spaltungen betrachtet, die von zwei verschiedenen Fragmenten eines Umlagerungssystems ausgehen. Diese Reaktionen finden beispielsweise im Massenspektrometer bei der Aufnahme der Spektren von Fettsäuren und deren Estern statt. Abbildung 9 zeigt hierfür ein Beispiel, wobei der unterlegte Teil das nur teilweise gezeigte Umlagerungssystem darstellt. Die Reaktionen unten sind die konkurrierenden α -Spaltungen.

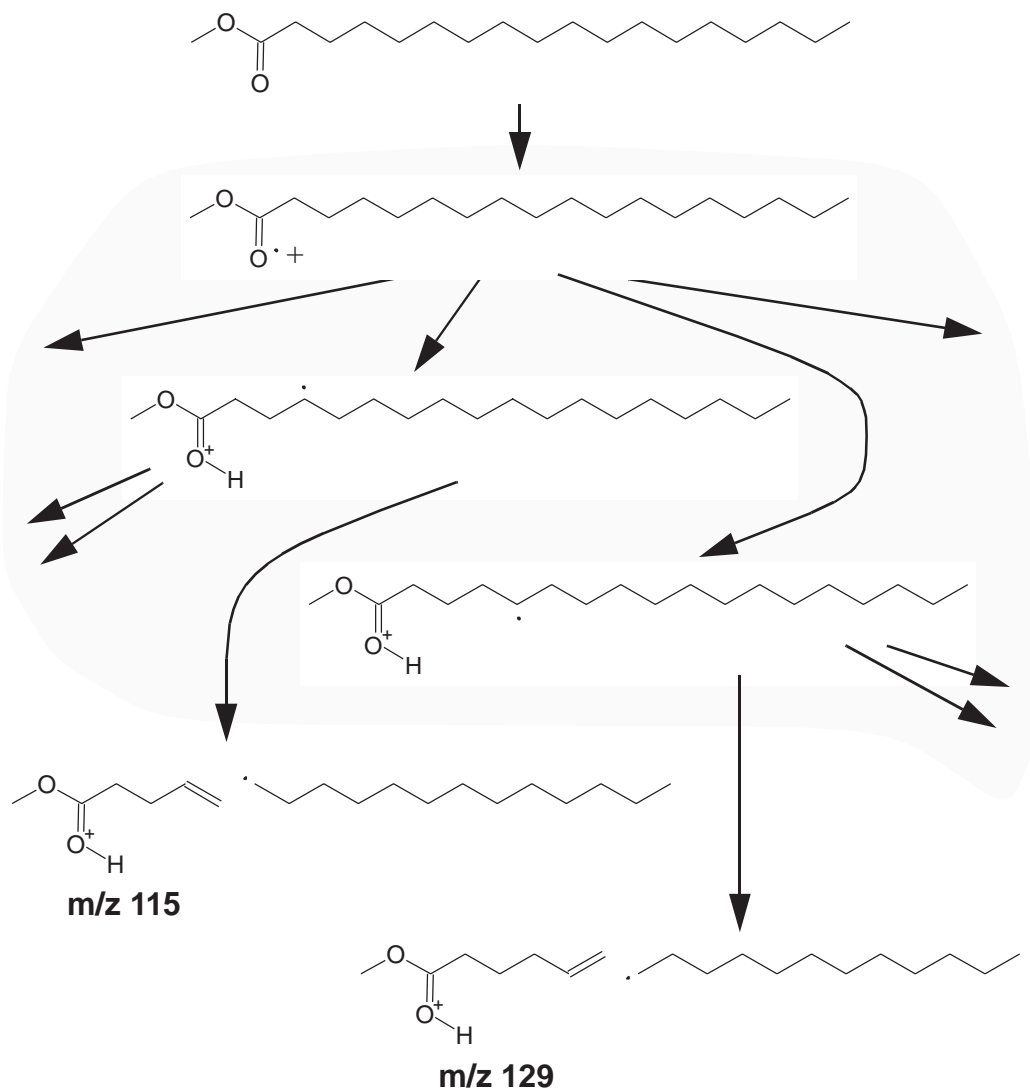


Abbildung 9: Konkurrierende α -Spaltungen aus einem Umlagerungssystem.

Da FRANZ, wie schon in 4.2.2.2 gesagt, keine Konzentrationen für die Fragmente eines Umlagerungssystem berechnen kann, werden die Ionen des Umlagerungssystem zusammengefaßt und FRANZ bestimmt nur die Gesamtkonzentration aller Umlagerungsprodukte. Daß das rechte Produkt der α -Spaltung (m/z 129) etwa in doppelt bis dreimal so großem Ausmaß gebildet wird wie das linke (m/z 115) (siehe Abbildung 10), kann sicherlich nicht an den unterschiedlich stark ablaufenden α -Spaltungen liegen, da die elektronischen und energetischen Verhältnisse vergleichbar sind. Da FRANZ das Umlagerungssystem zusammenfaßt und nur eine Konzentration dafür besitzt, verwendet FRANZ diese Konzentration auch als Eduktkonzentration für die Berechnung der Reaktionswahrscheinlichkeiten. Die unterschiedlichen Produkt- und gleichen Eduktkonzentrationen haben verschiedene Reaktionswahrscheinlichkeiten zur Folge. Dies kann man allerdings nicht, wie oben gesagt wurde, durch die Verhältnisse am Reaktionszentrum erklären. Die unterschiedliche Bildung der Produkte der α -Spaltungen liegt vielmehr daran, daß die Wasserstoffumlagerungen mit unterschiedlichen Reaktivitäten ablaufen und daher zu unterschiedlichen Konzentrationen für die Edukte der nachfolgenden α -Spaltungen führen. Dies kann auch dadurch belegt werden, daß das Signalmuster eines Fettsäureesters gut reproduziert werden kann (siehe Abbildung 10), wenn ausgehend von den Molekülionen zu 10% eine α -Spaltung stattfindet und die restlichen 90% eine Wasserstoffumlagerung eingehen, wobei

diese zu 15% über einen 5-, zu 80% über einen 6- und zu 5% über einen 7-gliedrigen Übergangszustand erfolgt und bei allen α -Spaltungen ein Alkylradikal abgespalten wird. [47]

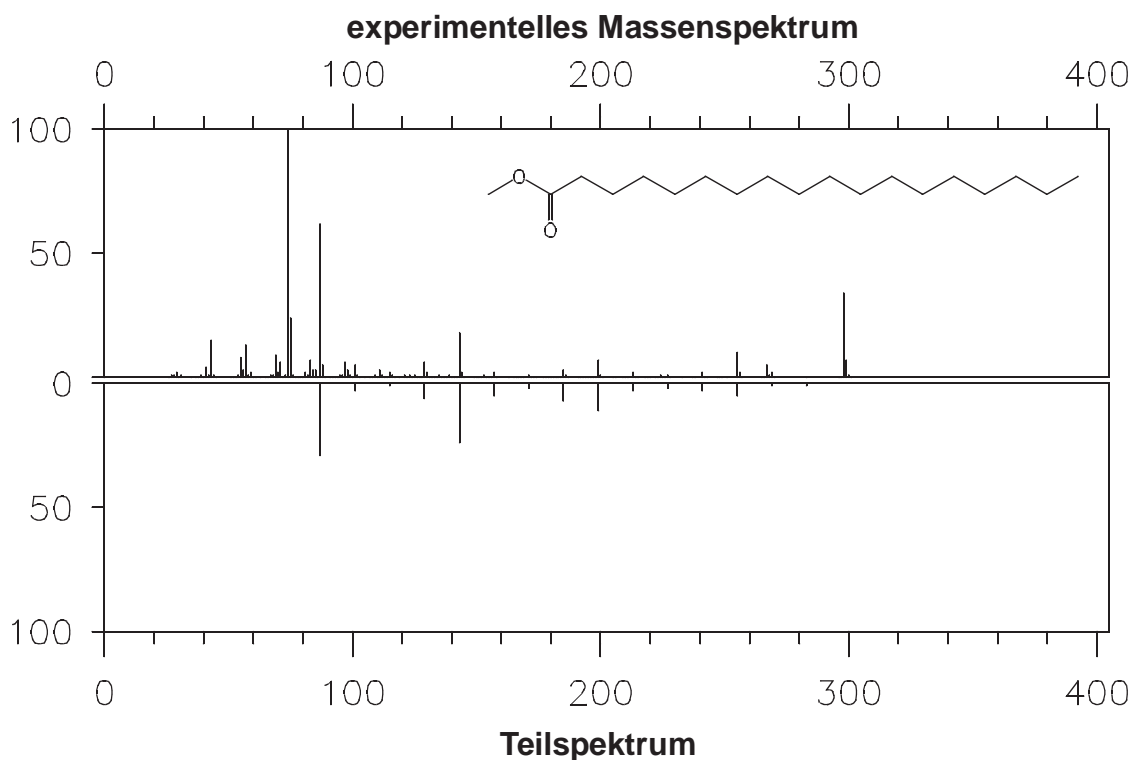


Abbildung 10: Vergleich des Teilspektrums [47], in dem nur die Signale enthalten sind, deren Ionen nur durch Wasserstoffumlagerungen und α -Spaltungen gebildet werden, mit dem experimentellen Massenspektrum [22] von Methylstearat.

4.2.2.4. Heuristiken

Da nicht alle gebildeten Fragmente im Massenspektrum unterschieden werden können, benötigt FRANZ einige zusätzliche Annahmen über die stattfindenden Reaktionen. Diese gliedern sich in zwei Gruppen, diejenigen, die Reaktionen, die aufgrund der Reaktionstypen möglich sind, ausschließen und solche, die nur die angenommene Reaktionswahrscheinlichkeit zwischen der Unter- und Obergrenze beeinflussen, während die Grenzen und damit die Sicherheit der Bewertung unverändert bleiben.

Heuristiken zum Ausschluß von Reaktionen

Eine dieser Heuristiken ist die Beurteilung der cyclischen Übergangszustände hinsichtlich der Ringspannung. Die Bildung eines solchen Übergangszustands wird von FRANZ aus den Definitionen der Reaktionssubstruktur, wenn sie zwei Teilsubstrukturen enthält, die über eine Kette von Atomen miteinander verbunden sind, und der Reaktion erkannt. Ist die notwendige Flexibilität der Kette durch Mehrfachbindungen oder starre Ringsysteme nicht ausreichend gegeben, wird die Reaktion verworfen.

Daneben ist eine zweimalige radikalische Wasserstoffumlagerung auf eine funktionelle Gruppe bei der Elektronenstoßmassenspektroskopie nur in wenigen Fällen, wie der Abspaltung von Wasser aus Aldehyden (Signal von etwa 1–2%) [47] oder der doppelten McLafferty-Umlagerung, von Bedeutung. Erstellt man für diese Fälle spezielle Reaktionstypen (Umlagerung mit anschließender Spaltung) und beschränkt die radikalische Wasserstoffumlagerung auf die einmalige Umlagerung auf eine funktionelle Gruppe (siehe Abbildung 11), reduziert man das Umlagerungsnetzwerk der Wasserstoffumlagerungen auf einen Bruchteil des unbeschränkten

Netzwerks ohne nennenswerten Informationsverlust (siehe Abbildung 12). Um von einer Ausgangsstruktur alle möglichen Reaktionen zu finden, ist es notwendig, die Reaktionen aller mesomeren Grenzstrukturen zu generieren.

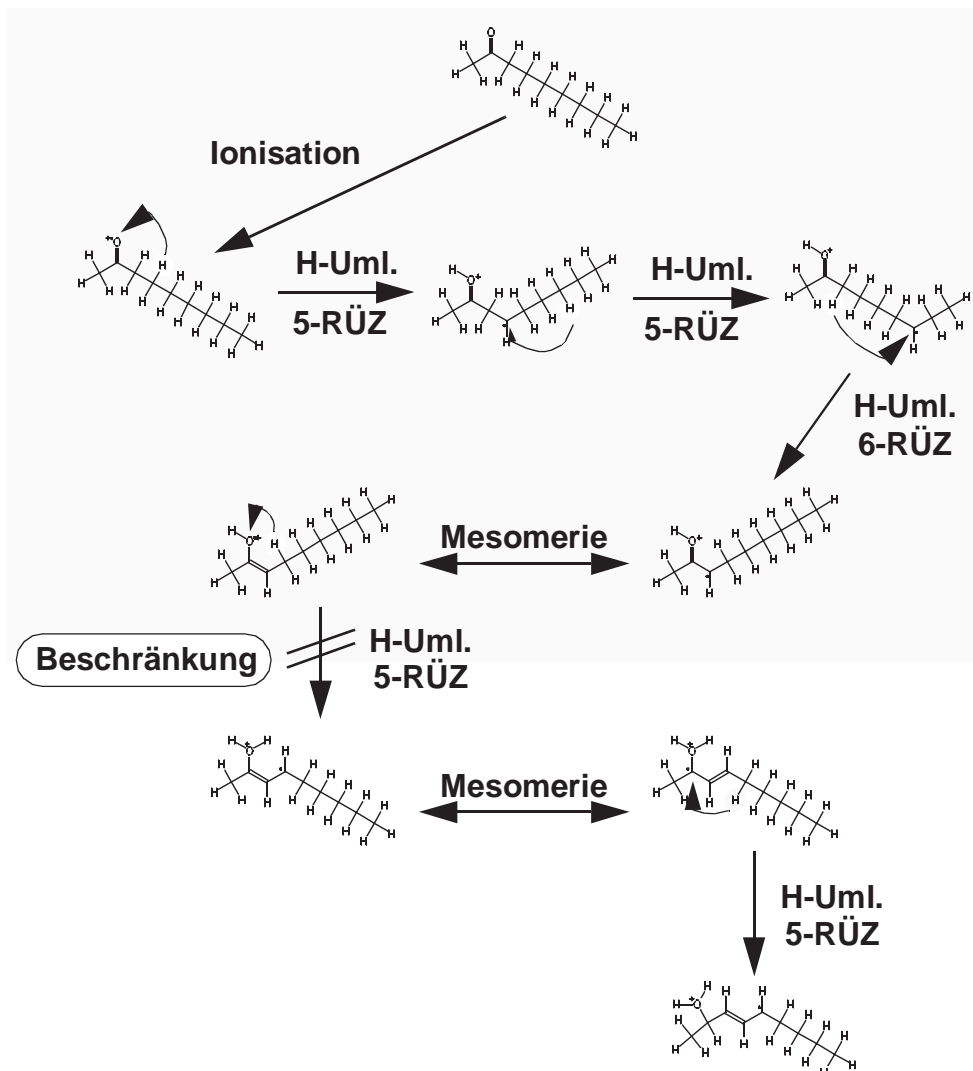


Abbildung 11: Beschränkung der Wasserstoffumlagerung bei 2-Nonanon. Nur der grau unterlegte Teil wird erzeugt. H-Uml. steht für Wasserstoffumlagerung und 5-RÜZ für einen 5-Ring-Übergangszustand.

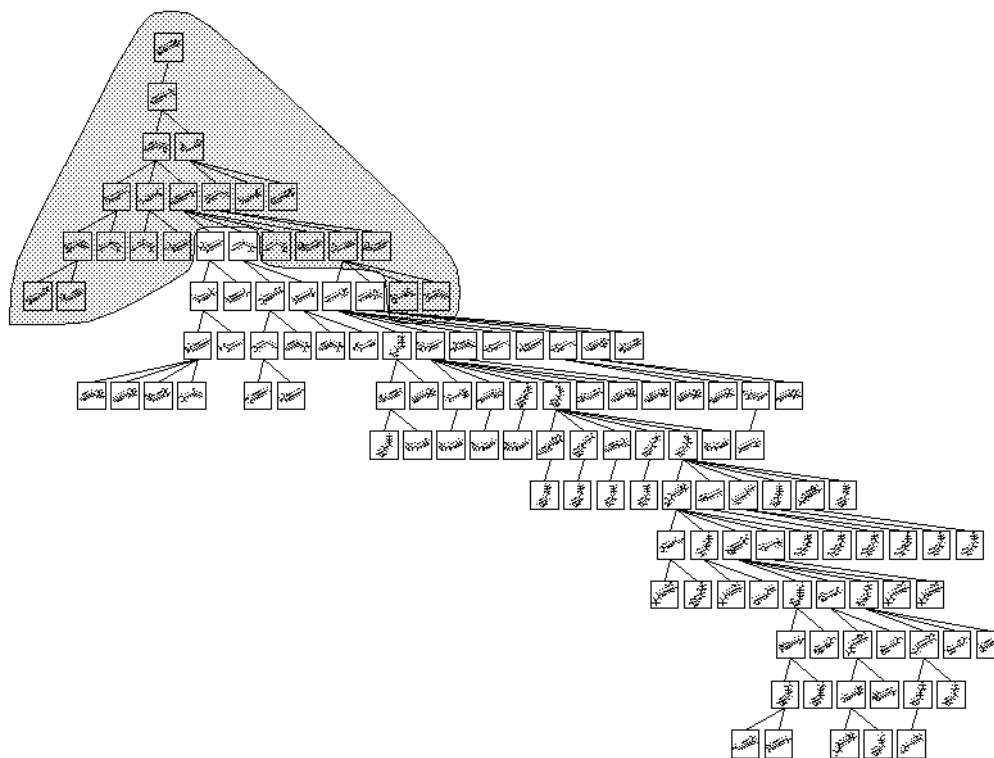


Abbildung 12: n-Ionisation und Umlagerungsnetzwerk des 2-Nonanons. Der unterlegte Teil ist der Teil des beschränkten Netzwerks, wobei das Netzwerk so dargestellt ist, daß jede Reaktion nur einmal gezeichnet ist, aber ein bestimmtes Kation an mehreren Stellen gebildet werden kann. Ist ein Kation mehrfach enthalten, sind seine Reaktionen nur an einer Darstellung angegeben.

Durch den Ausschluß der Reaktionen einer doppelten radikalischen Wasserstoffumlagerung erhält man bei der Analyse von Massenspektren kleinere Unsicherheitsbereiche bei den Reaktionswahrscheinlichkeiten (siehe 4.2.2.2) und hat durch die geringere Anzahl von Reaktionen eine kürzere Berechnungszeit sowohl bei der Analyse als auch bei einer möglichen Simulation, die die gewonnenen Fragmentierungsregeln mit Bewertung verwendet.

Heuristiken zur Verteilung der Umsätze

FRANZ führt ausgehend vom Molekülion alle möglichen Fragmentierungs- und Umlagerungsreaktionen durch und bestimmt aus den Signalen im Massenspektrum die Konzentrationen der Kationen. Die Reaktionswahrscheinlichkeiten, die FRANZ den einzelnen Reaktionen zuordnet, werden aus den Umsätzen bestimmt, die sich aus den Mengen des Kations und aller über dieses gebildeten Ionen ergibt. Die Verteilung der Umsätze ist eine Heuristik, die eingesetzt wird, wenn ein Fragment ausgehend von zwei oder mehr Umlagerungssystemen bzw. Fragmenten gebildet wird. Hier werden die Umsätze proportional zur Anzahl der Ionen im Umlagerungssystem verteilt. Fragmente, die keinem Umlagerungssystem angehören, haben die Größe eins. Ist die Anzahl der Reaktionsschritte vom unionisierten Molekül bis zum Fragment, das auf mehreren Wegen gebildet wird, unterschiedlich groß, werden die Umsätze so auf die Reaktionsfolgen verteilt, daß das Ion zum überwiegenden Maße über den kürzesten Reaktionsweg entsteht. Die Umsätze über die verschiedenen Wege werden um so stärker differenziert, je größer der Unterschied in der Zahl der Reaktionsschritte ist, und sinkt bei gleichem Unterschied mit der Zahl der Folgereaktionen.

Daneben kann FRANZ die Reaktionswahrscheinlichkeiten nach Möglichkeit auch an vorgege-

bene Werte angleichen. Dies setzt voraus, daß man für alle Reaktionen auch brauchbare Werte für die Wahrscheinlichkeiten hat. So können die Bewertungen der Reaktionen schrittweise verbessert werden. Für die Entwicklung der ersten Wahrscheinlichkeitsfunktionen bzw. neuronalen Netze, die in dieser Arbeit vorgestellt werden, kann man dieses Feature jedoch nicht heranziehen.

4.2.3. Modifikation der Regeln

Bei näherer Betrachtung der Reaktions- und Umlagerungsnetzwerke, die FRANZ für verschiedene Verbindungen ausgehend von ihrem Massenspektrum erzeugt hat, fiel auf, daß FRANZ fast alle Signale im Spektrum erklären konnte, dies aber häufig nur über Wasserstoffumlagerungen erreichte. Da diese für die Ableitung von Bewertungsfunktionen zur Vorhersage von Massenspektren Schwierigkeiten machen (siehe 4.2.2.3), wurden die Wege, auf denen die Fragmente gebildet werden, genauer untersucht und die Reaktionsregeln daraufhin modifiziert.

4.2.3.1. Weiterentwicklung

Es wurde festgestellt, daß einige Fragmente ausschließlich über Wasserstoffumlagerungen mit großen Ringübergangszuständen erzeugt werden können und daher einen geringen Umsatz aufweisen sollten. Die Wasserstoffumlagerung war für sechs- bis zehngliedrige Ringübergangszustände zugelassen. Sie wurden auf fünf- bis siebengliedrige Ringe beschränkt, bei denen die Reaktionen sowohl enthalpisch, wie auch entropisch im günstigen Bereich liegen. Diese Ringgrößen haben sich auch für die Simulation als geeignet erwiesen (siehe Abbildung 10 auf Seite 21). Um für die Umlagerungen auf Heteroatome, die exotherm und somit begünstigt sind, auch 1,2-Wasserstoffverschiebungen zuzulassen, wurden Umlagerungen für diese über drei- bis siebengliedrige Ringe erlaubt. Umlagerungen über viergliedrige Ringe, die energetisch am ungünstigsten sind, konnten beim Reaktionsgenerator von FRANZ und MASSIMO nicht ausgeschlossen werden. Zusammengefaßt kann gesagt werden, daß die radikalische Wasserstoffumlagerung für große Ringübergangszustände beschränkt und gleichzeitig um Umlagerungen mit kurzen Distanzen ergänzt wurde.

Daneben wurden induktive Spaltungen, deren positive Ladung an einem Kohlenstoffatom lokalisiert ist, ausgeschlossen, da sie bei der Fragmentierung im Massenspektrometer eine untergeordnete Rolle spielen. Werden FRANZ auch solche Reaktionsmöglichkeiten, bei denen Carbene entstehen, angeboten, entstehen oft alternative Reaktionssequenzen zu gleichschweren Ionen, was FRANZ die Möglichkeit der eindeutigen Zuordnung von Reaktionswahrscheinlichkeiten nimmt.

Gleichschwere Kationen, wie die der Carbene, werden durch eine 1,2-Eliminierung von Wasserstoffmolekülen gebildet [27]. Deshalb wurde der Reaktionstyp von Abbildung 13 den Regeln hinzugefügt.

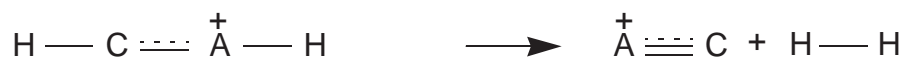


Abbildung 13: Abspaltung molekularen Wasserstoffs.

Damit von FRANZ bei primären Alkoholen nicht systematisch alternative Reaktionswege und damit unsichere Reaktionswahrscheinlichkeiten zur Bildung des $\text{R}-\text{C}\equiv\text{O}^+$ -Ions und aller seiner Folgeprodukte, der Serie der Alkylfragmente, gefunden werden, wurde diese Reaktion ausgeschlossen, wenn das Radikalzentrum und die positive Ladung an einem Atom lokalisiert sind (siehe Abbildung 14). Ohne diesen Ausschluß könnten die α -Spaltung (Abspaltung eines Wasserstoffradikals) und die Eliminierung von Wasserstoffmolekülen in beiden Reihenfolgen auf das Molekülion angewendet werden, wobei das $\text{R}-\text{C}\equiv\text{O}^+$ -Ion entsteht, und FRANZ könnte für die Reaktionen keine sicheren Reaktionswahrscheinlichkeiten berechnen (siehe 4.2.2.2).

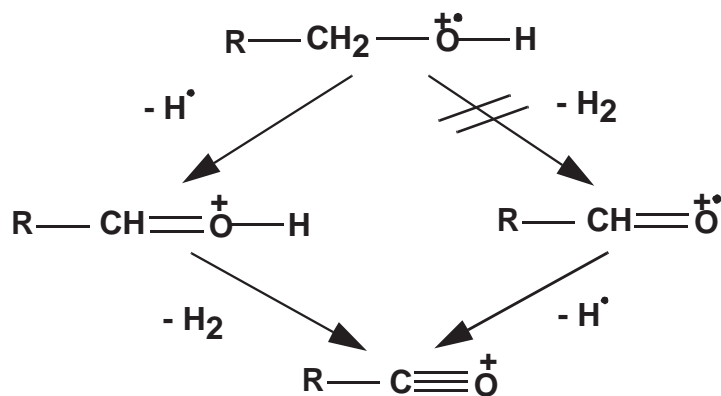


Abbildung 14: Alternative Reaktionswege bei primären Alkoholen.

Durch die Einführung der Eliminierung von Wasserstoffmolekülen können auch die Signale M-1 und M-3 im Spektrum von Methanol (Abbildung 15) erklärt werden. M steht dabei für die Molekülmasse. Das kleine Signal bei M-2 muß der sicheren Ermittlung der Reaktionswahrscheinlichkeiten durch FRANZ geopfert werden, da keine Möglichkeit bestand, selektiv die Weiterreaktion des Fragmentes mit der Masse M-2 zu verhindern. Dies ist allerdings nur von geringer Bedeutung, da die Signale bei M-1, M-2 und M-3 bei längeren Alkylketten ohnehin nicht mehr in den Spektren enthalten sind.

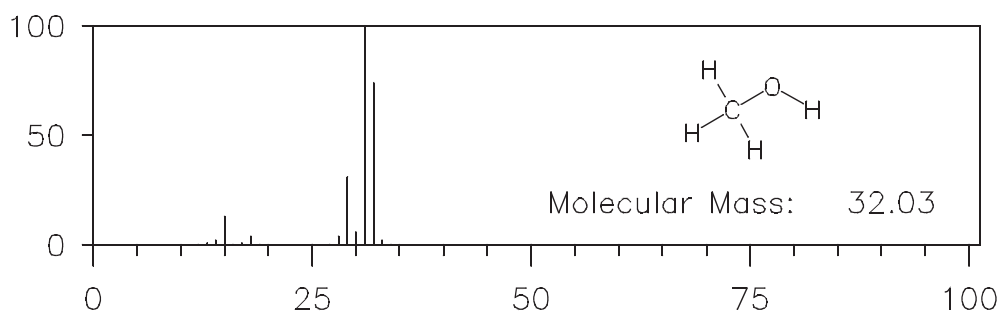


Abbildung 15: Massenspektrum von Methanol.

Analoges gilt für die Spektren der Verbindungen Methylmercaptan (Abbildung 16) und Methylamin (Abbildung 17).

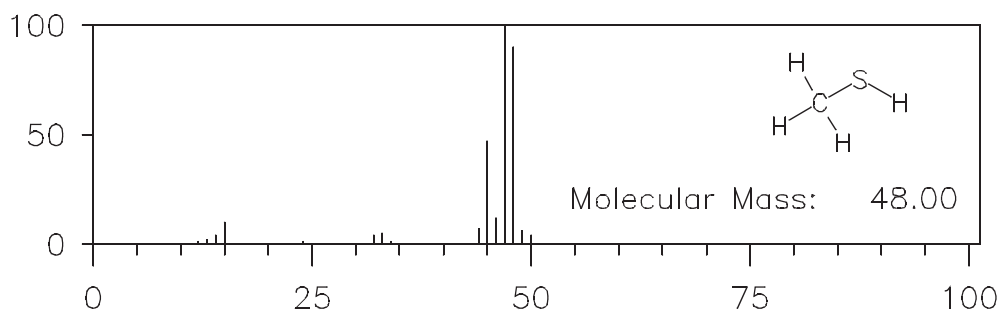


Abbildung 16: Massenspektrum von Methylmercaptan.

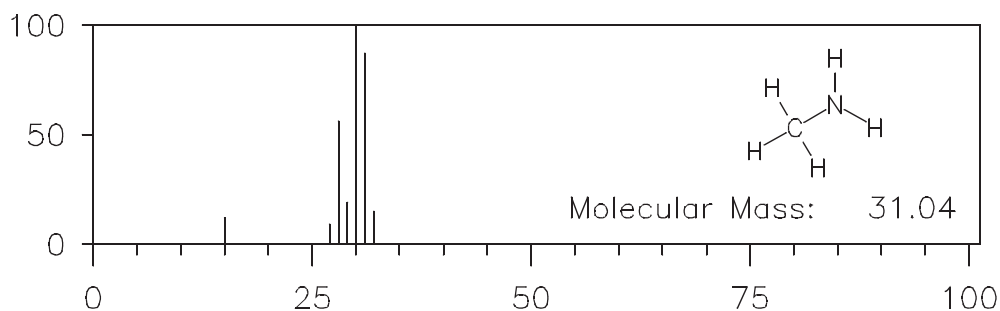


Abbildung 17: Massenspektrum von Methylamin.

Das für primäre Alkohole gebildete $R-C\equiv O^+$ -Ion spaltet Kohlenmonoxid ab und fragmentiert danach wie Alkylkationen. Dazu wird der Reaktionstyp der induktiven α -Spaltung benötigt [27], der ebenfalls neu eingeführt wurde.

(A, B = nicht H):



Abbildung 18: Induktive α -Spaltung.

Die Carbonyleliminierung, die nur einen Spezialfall der induktiven Spaltung aus einer anderen mesomeren Grenzstruktur darstellt, wurde so erweitert, daß an Stelle des Sauerstoffatoms auch andere Heteroatome stehen können. Diese Reaktionen finden bei der Fragmentierung von Nitrilen und Aminen Anwendung [18].

Desweiteren wurde die Eliminierung aus dem Satz möglicher Reaktionstypen genommen, da sie nur in Spezialfällen entscheidende Signale im Massenspektrum erklärt. Die Verdrängungsreaktion wurde für ionisierte Carbonylverbindungen ausgeschlossen, da dieser Reaktionstyp für derartige Verbindungen nicht stattfindet, sowie zur Bildung von dreigliedrigen Ringen zugelassen, die in [18] beschrieben sind und in den Massenspektren von Alkylhalogeniden mit Signalen von 10% und mehr zu sehen sind. Außerdem wurde die Halogenspaltung so erweitert, daß auch Halogenwasserstoffkationen abgespalten werden können.

Die Reaktionstypen der n- und π -Ionisation, sowie die α -Spaltung und die Oniumspaltung blieben unverändert. Der sich daraus ergebende Regelsatz ist in Abbildung 19 gezeigt.

α -Spaltung:



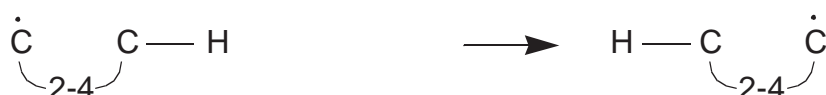
induktive Spaltung (X = Heteroatom, A = nicht H):



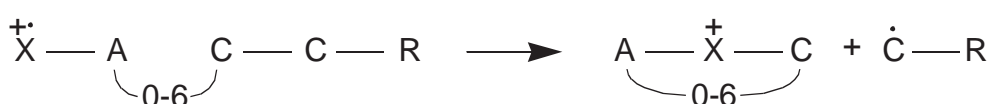
induktive α -Spaltung (A, B = nicht H):



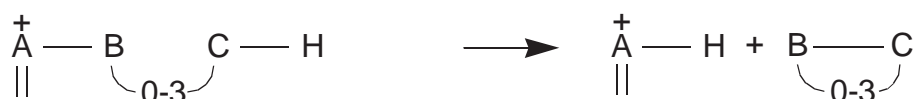
H-C-Wasserstoffumlagerung:



Verdrängungsreaktion (X = Heteroatom, C = kein π -Zentrum):



Onium-Reaktion (A kein Kohlenstoff, nicht radikalisches Ion):



Halogenabspaltung (X = Cl, Br, I):



Carbonyleliminierung (X = Heteroatom, nicht radikalisches Ion):



H₂-Eliminierung:



H-Het-Wasserstoffumlagerung (X = Heteroatom, C = kein π -Zentrum):



Abbildung 19: Weiterentwickelter Regelsatz.

4.2.3.2. Erhöhung der Genauigkeit der Reaktionswahrscheinlichkeiten

Um Regeln zu erhalten, an denen gezeigt werden konnte, daß mit diesem Ansatz Massenspektren simuliert werden können, wurden die verwendeten Reaktionstypen so modifiziert, daß die oben beschriebenen Schwierigkeiten bei der Zuweisung von Reaktionswahrscheinlichkeiten (siehe 4.2.2.2) durch FRANZ nicht auftreten sollten. Redundante Reaktionswege zum gleichen Fragment bzw. zu einem gleichschweren Ion und die in Kapitel 4.2.2.3 beschriebenen Probleme mit Umlagerungsreaktionstypen wurden vermieden, indem beide Reaktionstypen der Wasserstoffumlagerung (auf Kohlenstoff- bzw. Heteroatome) aus dem Satz möglicher Reaktionstypen

entfernt wurden. Um nicht einen größeren Prozentsatz der Signale im Massenspektrum von der Aufklärung durch FRANZ auszuschließen, wurden stattdessen Reaktionstypen eingeführt, die die Wasserstoffumlagerung und die anschließende Fragmentierung in sich vereinen. So wurden die Typen der McLafferty-Spaltung, die eine Wasserstoffumlagerung und eine sich anschließende α -Spaltung darstellt, und der HX-Eliminierung eingeführt, hinter der sich die Wasserstoffumlagerung mit nachgeschalteter induktiver Spaltung verbirgt. Diese beiden Reaktionstypen sind in Abbildung 20 dargestellt. X ist ein Heteroatom.

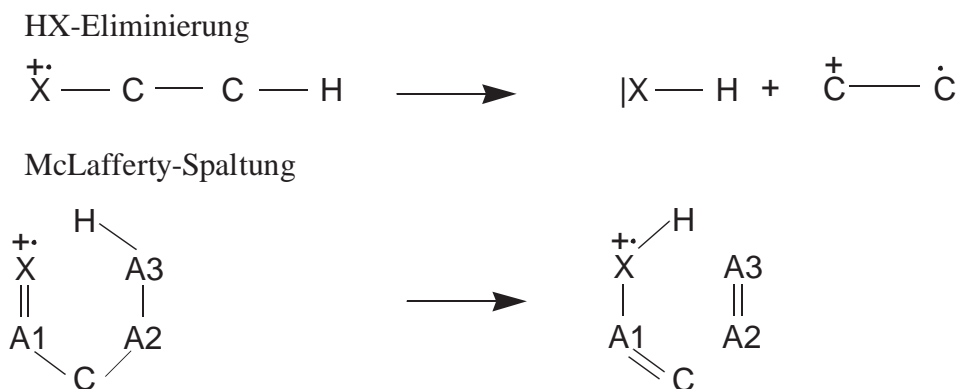


Abbildung 20: McLafferty-Spaltung und HX-Eliminierung.

Bei der McLafferty-Spaltung ist noch zu beachten, daß es sich beim dritten Atom um Kohlenstoff (C in Abbildung 20) handeln muß. Der Grund dafür liegt im untersuchten Datensatz von Molekülen (siehe Anhang C), der Ester enthält, bei denen eine McLafferty-Spaltung sowohl auf der Säure- wie auch auf der Alkoholseite stattfinden kann. Tritt beim betrachteten Ester eine doppelte McLafferty-Spaltung ein, kann FRANZ nicht entscheiden, in welcher Reihenfolge die McLafferty-Spaltungen stattfinden. Um die Situation der alternativen Reaktionswege zu vermeiden, wurde die McLafferty-Spaltung nur auf der Seite der Säure gestattet.

Da auch die Verdrängungsreaktion und die Halogenspaltung FRANZ in einigen Fällen alternative Reaktionswege eröffnen und somit in diesen Fällen keine eindeutigen Reaktionswahrscheinlichkeiten ermittelt werden könnten, wurden auch diese Reaktionstypen aus den Regeln entfernt, obgleich der Datensatz Halogenide enthält. Diese Halogenide wurden im Datensatz belassen, damit die Reaktivitätsfunktionen für die α -Spaltung und die induktive Spaltung auch für die Halogenide gültig sind. Dies hat zur Folge, daß einige Signale für die halogenhaltigen Substanzen von FRANZ nicht erklärt werden können, aber die durchgeführten Reaktionen werden richtig bewertet, wenn, wie im Fall der Verdrängungsreaktionen, die Folgereaktionen nur von geringer Bedeutung sind oder, wie im Fall der Halogenspaltung, die Reaktionen nur vernachlässigbare Umsätze besitzen, da im Datensatz nur Monohalogenide enthalten sind.

4.2.4. Berechnung physikochemischer Effekte

Grundlage für die Berechnung der Reaktivitäten sind die physikochemischen Eigenschaften am Reaktionszentrum, der Reaktionssubstruktur in den Edukten. Im Arbeitskreis von Professor Gasteiger wurden in den letzten 20 Jahren empirische Verfahren entwickelt, die energetische und elektronische Eigenschaften schnell berechnen können.

Zu den Eigenschaften zählen die Polarisierbarkeiten der Atome und Bindungen [28], die Partialladungen und die Elektronegativitäten. Diese werden über das PEOE-Verfahren (*partial equalization of orbital electronegativity*) berechnet. So erhält man die σ -Partialladungen und σ -Restelektronegativitäten der Atome [29][30]. Die Verteilung der Elektronen in π -Systemen basiert auf dem Konzept der Orbitalelektronegativitäten [31].

Zur Beschreibung der Resonanzeffekte zur Stabilisierung von Ladungen (+M und -M-Effekte)

sowie zur Quantifizierung von Hyperkonjugationseffekten existieren Methoden, die den Grad der Delokalisation bestimmen [32]. Bei der Verbesserung der Verfahren zur Bestimmung der Delokalisation von Ladungen wurde festgestellt, daß für eine optimale Beschreibung die für die Strukturrepräsentation verwendete, chemische Datenstruktur der Atom- und Bindungslisten nicht geeignet ist, da sie keine dreidimensionale Information über die Moleküle enthält, die für die Abschätzung der Überlappungsintegrale Voraussetzung wären [33]. Dieses Defizit wird teilweise durch die Einführung einer Datenstruktur wie RICOS (siehe Kapitel 5) beseitigt. Das Berechnungsverfahren müßte dazu allerdings grundlegend überarbeitet werden.

Die Berechnung von Bildungswärmen und Bindungsdissoziationsenergien, die zunächst auf acyclische, organische Substanzen beschränkt war [34], wurde auf ringförmige, neutrale, organische Verbindungen [35] sowie auf Radikale, Kationen und Radikalkationen erweitert [36][37], wie sie in der Elektronenstoßmassenspektroskopie auftreten. Jedoch sind auch hier die Abweichungen von den experimentell bestimmten Werten von Bildungsenthalpien für diejenigen ionisierten Moleküle größer als der tolerierbare Fehler, die orthogonale π -Systeme und freie Elektronenpaare enthalten. Hier gibt es Schwierigkeiten zu erkennen, welche Elektronensysteme miteinander konjugiert sind und welche nicht. Zusätzlich kann in der Bindungsliste in solchen Fällen oft nicht eindeutig ein bestimmtes Elektronensystem spezifiziert werden, aus dem bei der Ionisation das Elektron entfernt wurde. Für die übrigen Verbindungen liegen die Abweichungen von den experimentellen Werten etwa in der Größenordnung der Meßgenauigkeit. Zur Vermeidung dieser Abweichungen ist eine Verbesserung der Darstellung der chemischen Struktur sowie eine Neugestaltung des Algorithmus für die Berechnung erforderlich. Als Beispiel hierfür dienen ionisierte, ungesättigte Verbindungen, bei denen ein freies Elektronenpaar mit dem π -System (teilweise) konjugiert ist, wie bei den Jodtoluolen, Brombenzol oder 1,1-Difluorethen. Aber auch bei ionisierten Chinonen treten bei bestimmten mesomeren Grenzstrukturen starke Abweichungen auf. Für diese Substanzen sind Differenzen bei der Bildungswärme von ca. 200 bis 600 kJ/mol zu beobachten. Legt man der Berechnung der Bildungswärmen eine chemische Datenstruktur zugrunde, bei der die Konjugation von Doppel- und Dreifachbindungen sowie von freien Elektronenpaaren Bestandteil der chemischen Datenstruktur ist, müßten diese Probleme bewältigt werden können. Aus der Differenz der Bildungswärmen der Produkte und Edukte läßt sich auch die Reaktionsenthalpie berechnen.

Neben diesen Eigenschaften können auch rein formale Parameter der Atome, Bindungen und Moleküle für die Berechnung der Reaktivität herangezogen werden. Dazu gehören die Zugehörigkeit von Atomen zu Ringen in der Struktur, wozu neben dem kompletten Satz von Ringen auch der kleinste Satz kleinster Ringe (SSSR, *smallest set of smallest rings* [38][39]) bestimmt wird. Aber auch die Beteiligung an einem aromatischen System [36], die Zahl der Nachbaratome, der Wasserstoff- bzw. Nicht-Wasserstoff-Atome, Indikatoren, die die Möglichkeit der Bildung bzw. des Bruchs von Bindungen angeben, welche auf einer Valenztabelle beruhen, sowie einige weitere formale Parameter stehen zur Beschreibung des Reaktionszentrums zur Verfügung.

All diese Parameter werden vom eingebauten Subsystem PETRA (Parameter Estimation for the Treatment of Reactivity Applications) berechnet, das für die Reihenfolge der Ermittlung der verschiedenen Eigenschaften auch die Abhängigkeiten der einzelnen Berechnungsverfahren untereinander berücksichtigt. Dieses Subsystem wird in EROS6 und MASSIMO für die Ermittlung der physikochemischen Eigenschaften eingesetzt, die zur Evaluierung der Reaktivitäten verwendet werden.

4.2.5. Ableitung der Wahrscheinlichkeitsfunktionen

Nachdem die Reaktionstypen festgelegt wurden und mit FRANZ aus den Massenspektren zusammen mit deren Strukturen Reaktionsdatenbanken inklusive der Bewertungen der Reaktio-

nen erstellt wurden, kann man den Zusammenhang zwischen den Eigenschaften der Reaktionssubstruktur und den Reaktivitäten der Reaktionen herleiten. Dies können Wahrscheinlichkeitsfunktionen bzw. neuronale Netze sein. Dazu werden zunächst die Reaktionen, die der Klasse-1 zugeteilt wurden, für einen Reaktionstyp aus der Datenbank extrahiert und die physikochemischen Parameter für die Atome und Bindungen am Reaktionszentrum sowie die Eigenschaften der Moleküle berechnet. Das sind 11 Parameter pro Atom im Edukt und im Produkt, 12 Parameter je Bindung im Edukt oder Produkt, 1 Parameter je Edukt und Produkt sowie die Kettenlänge für jeden Reaktionstyp mit definierter Atomkette und die Zahl der Atome im Edukt. So erhält man einen Datensatz, bei dem jede Reaktion durch alle möglichen Parameter beschrieben wird, auch wenn diese auf die Beschreibung der Reaktivitäten der Reaktionen keinen Einfluß haben. Deshalb werden vor der Anwendung einer multilinenen Regressionsanalyse oder dem Training neuronaler Netze Variablen (Parameter) mit der Varianz null für den gesamten Datensatz (alle Reaktionen) sowie linear abhängige Variablen und Variablen mit einer Korrelation größer 0.98 (bzw. 0.8) entfernt. Da einige Parameter systematisch linear abhängig sind, gibt es eine Liste bevorzugt verwendeter Parameter zur Entscheidung, welcher der linear abhängigen oder hoch korrelierten Parameter verworfen wird. Zu den systematisch linear abhängigen Eigenschaften gehören beispielsweise die Differenz der σ -Elektronegativitäten zweier benachbarter Atome im Reaktionszentrum und die σ -Elektronegativitäten dieser Atome.

4.2.5.1. Regeln mit Umlagerungsreaktionstypen

Bei den Datensätzen, die sich ergeben, wenn der Satz von Reaktionstypen auch Umlagerungsreaktionstypen enthält, lassen sich zwischen den physikochemischen Parametern und den Reaktivitäten keine linearen Zusammenhänge finden. Für eine optimale Regression, bei der nur signifikante Variablen verwendet werden, erhält man ein Diagramm, bei dem die vorhergesagten gegen die von FRANZ bestimmten Reaktionswahrscheinlichkeiten aufgetragen sind, wie es in Abbildung 21 gezeigt ist. Der Korrelationskoeffizient von 0.5964 zeigt bei 9 signifikanten Variablen an, daß diese Regression unbrauchbar ist. Als Beispiel dient der Datensatz der Carbonyleliminierungsreaktionen (siehe Anhang D.1.1).

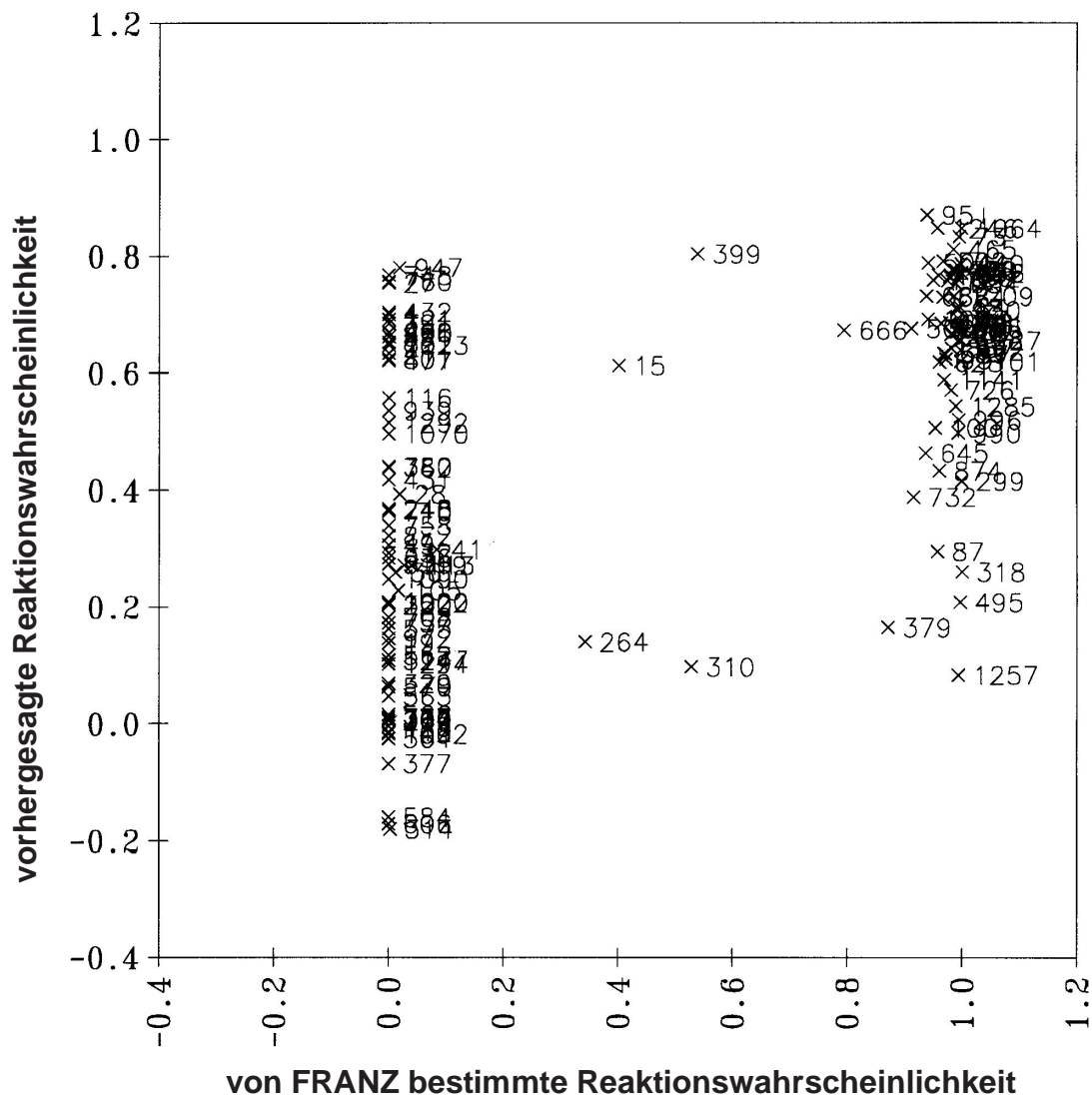


Abbildung 21: Diagramm, bei der die mit einer multilinearen Regression vorhergesagten Reaktionswahrscheinlichkeiten gegen die von FRANZ bestimmten (Sollwerte) aufgetragen sind. 139 Datenpunkte. Die Nummern der Datenpunkte sind die Nummern der Reaktionen im Anhang D.1.1.

Auch bei der Verwendung von 24 Variablen, von denen nur 9 als signifikant einzustufen sind, ergibt sich lediglich eine Steigerung des Korrelationskoeffizienten r von 0.5964 auf 0.6483. Neben den richtig klassifizierten Reaktionen enthält ein Datensatz auch Reaktionen mit durch Umlagerungsreaktionstypen von FRANZ falsch bestimmte Reaktionswahrscheinlichkeiten (siehe 4.2.2.3), die bis zu etwa zehn Prozent der Datenpunkte ausmachen. So besteht die Schwierigkeit, daß weder der Satz der Datenpunkte mit korrekter Reaktionswahrscheinlichkeit noch der Satz der Parameter bekannt ist, die die Abhängigkeit der Reaktionswahrscheinlichkeit beschreiben. Kennt man entweder die korrekten Datenpunkte oder die entscheidenden Variablen, gibt es etablierte Methoden zur Ermittlung der gewünschten Datenpunkte bzw. Parameter. Unternimmt man einen Versuch durch chemisches Wissen die höchst wahrscheinlich von FRANZ fehlklassifizierten Datenpunkte zu identifizieren und aus dem Datensatz zu entfernen, reduziert sich dadurch der Datensatz auf 115 Datenpunkte. So kann man die Qualität der Korrelation deutlich steigern (siehe Abbildung 22), obgleich sie noch weit von einem genügend gu-

ten Ergebnis entfernt ist, da zwischen den physikochemischen Parametern und den Reaktionswahrscheinlichkeiten keine linearen Zusammenhänge bestehen. Zudem müßten bei der Verwendung einer linearen Regressionsanalyse die vorhergesagten Wahrscheinlichkeiten anschließend auf den Wertebereich von 0.0 bis 1.0 beschränkt werden, da nur solche Werte für eine Simulation geeignet sind.

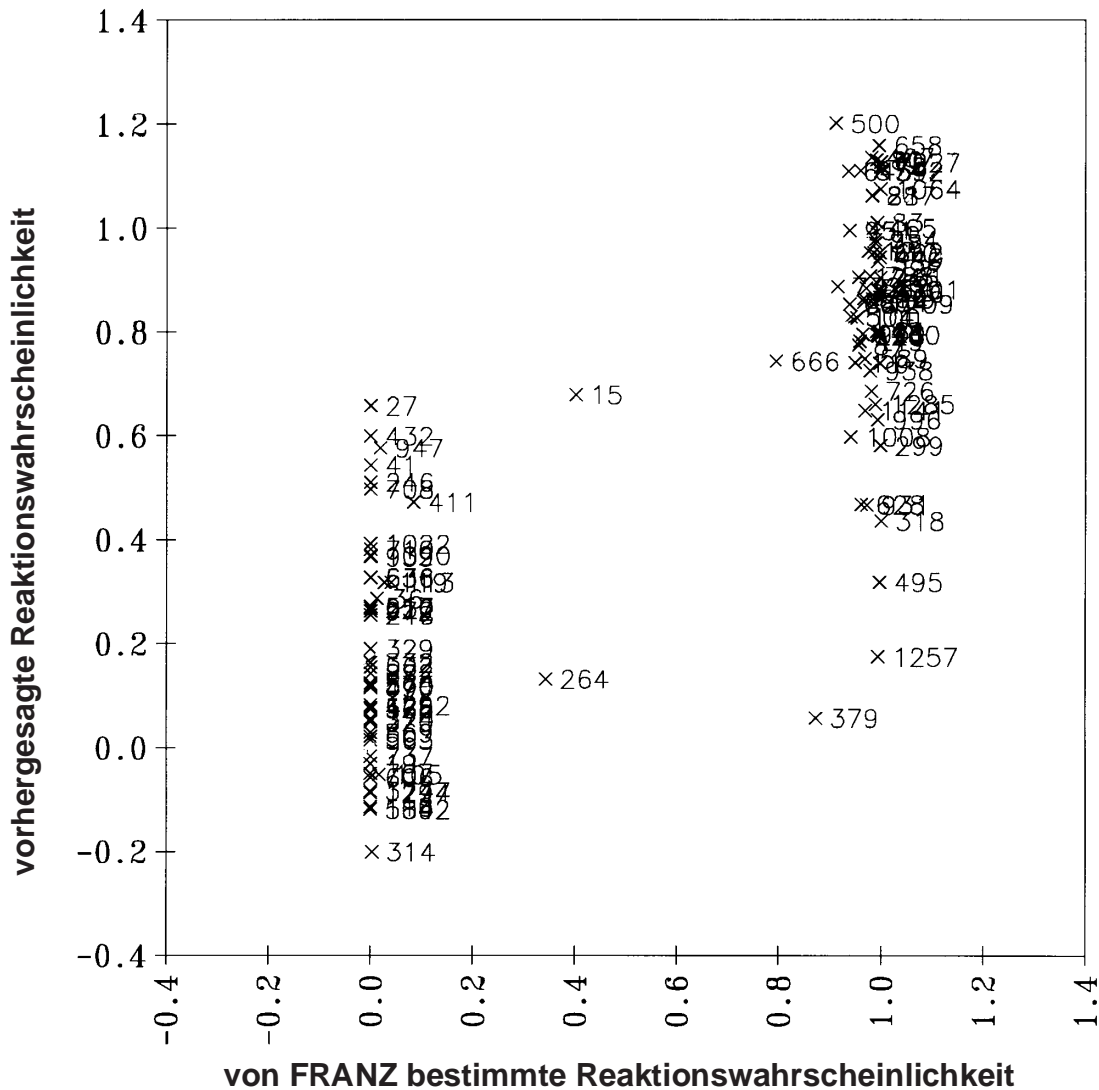


Abbildung 22: Diagramm, bei der die mit einer multilinearen Regression vorhergesagten Reaktionswahrscheinlichkeiten gegen die von FRANZ bestimmten (Sollwerte) aufgetragen sind. Nach manuellem Ausschluß falscher Datenpunkte: 115 Datenpunkte.
13 signifikante Variablen, Korrelationskoeffizient $r = 0.8376$

Da nur eine sehr geringe Anzahl von Datenpunkten eine mittlere Reaktivität besitzt, kann man auch an eine logistische Regressionsanalyse [40] denken, die zur Berechnung der Wahrscheinlichkeit nicht linear ist. Zuvor wird allerdings, wie in der multilinearen Regression, eine Linearkombination der verwendeten Variablen gebildet.

$$P = \frac{1}{1 + e^{-f}} \quad \text{mit} \quad f = c_0 + c_1 \cdot x_1 + c_2 \cdot x_2 + \dots \quad (2)$$

Betrachtet man das Diagramm in Abbildung 22 unter dem Aspekt, daß eine anschließende Funktion die vorhergesagten Reaktionswahrscheinlichkeiten oberhalb eines Schwellwerts in Richtung 1.0 und Werte unterhalb in Richtung 0.0 verschiebt, wie es bei der logistischen Regression durchgeführt wird, sind keine befriedigenden Ergebnisse zu erwarten, da die Linearkombination nicht in der Lage ist, die Datenpunkte richtig in zwei Gruppen zu trennen (siehe Abbildung 22).

Für Datensätze, bei denen weder die korrekten Datenpunkte noch die beschreibenden Variablen bekannt sind, wurde eine Technik entwickelt, die auch solche Daten behandeln kann [41][42]. Hierzu werden zwei verschiedene Typen neuronaler Netze verwendet. Kern dieser Methode ist, die Zahl der Variablen durch Reconstruction Learning [43], einem Backpropagationnetz [44] mit modifiziertem Trainingsalgorithmus, und eventuell manuell durch chemisches Wissen soweit zu reduzieren, daß man anschließend mit einem Kohonennetz [45] falsch klassifizierte Datenpunkte als ähnliche Datenpunkte mit deutlich abweichender Reaktionswahrscheinlichkeit erkennen und aus dem Datensatz entfernen kann. Kann nicht eindeutig entschieden werden, welcher Datenpunkt falsch bestimmt wurde, da etwa gleich viele ähnliche Datenpunkte mit hoher und niedriger Reaktivität enthalten sind, werden all diese sich widersprechenden Datenpunkte gelöscht. Danach können durch das Reconstruction Learning die für die Beschreibung der Reaktivität relevanten Variablen herausgefunden werden, wobei mit allen möglichen begonnen wird. Mit den so herausgefundenen Datenpunkten und Variablen wird nun das endgültige Backpropagationnetz trainiert, das auch nichtlineare Zusammenhänge beschreiben kann. Testet man nun die zuvor aus dem Datensatz herausgenommenen Datenpunkte mit dem so gewonnenen Backpropagationnetz auf ihre Reproduktion, stellt man fest, daß nur etwa die Hälfte dieser Datenpunkte stark von den vorhergesagten Werten abweicht. Der Ablauf ist in Abbildung 23 zusammengefaßt.

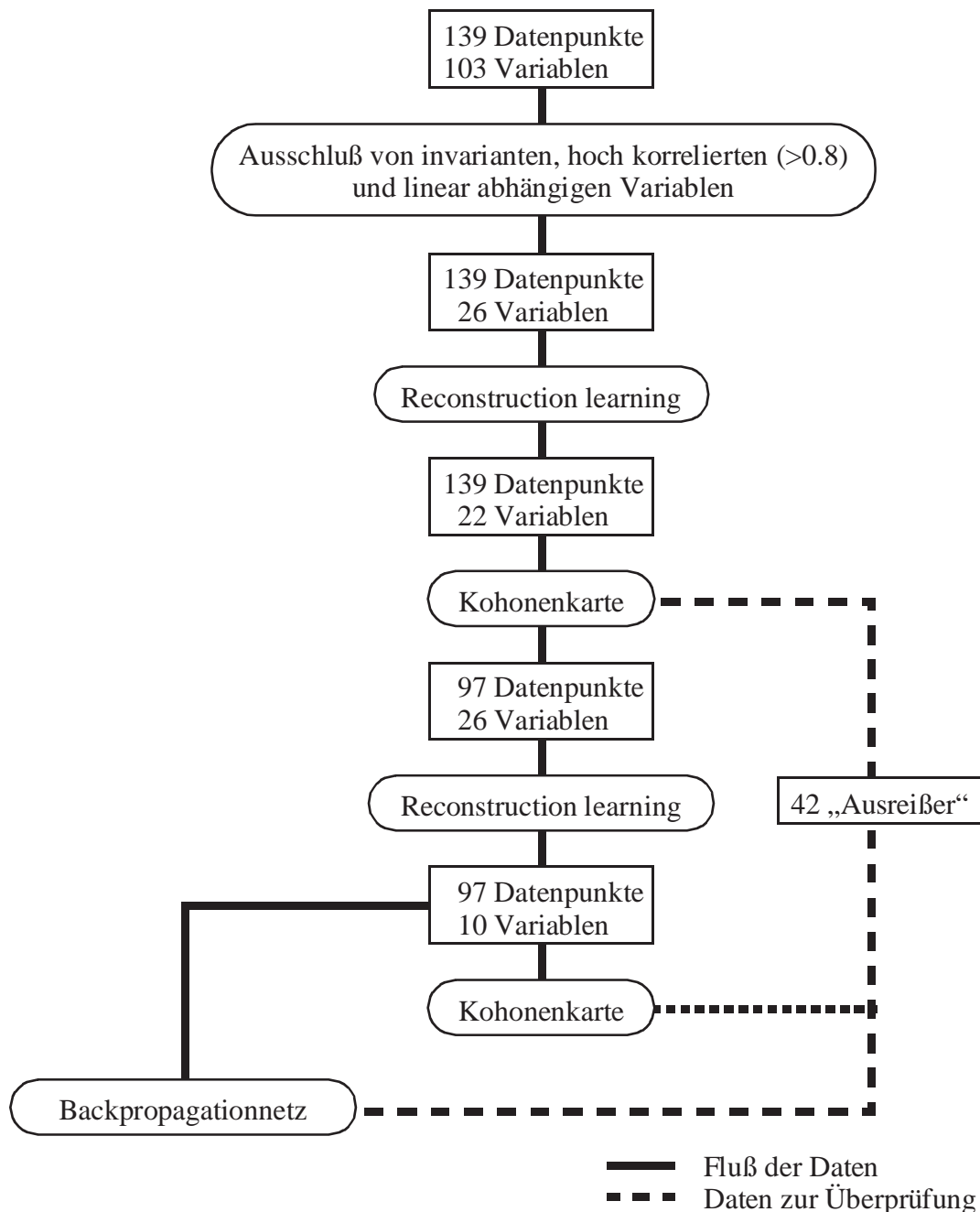


Abbildung 23: Beispiel des Ablaufschemas zum Training des Backpropagationnetzes.

Beim Training des Backpropagationnetzes muß noch darauf geachtet werden, daß dieses die Datenpunkte nicht auswendig lernt, sondern verallgemeinert. Man kann es dazu zwingen, indem man die Zahl der Gewichte, die von der Zahl der verwendeten Parameter, der Zahl der versteckten Schichten (hidden layer) und der Zahl der Neuronen in ihnen abhängt, soweit reduziert, daß man etwa einen Faktor 3:1 für die Zahl der trainierten Datenpunkte zur Zahl der Gewichte im Netz oder besser noch größer erhält. Ein höherer Faktor ist allerdings oft schwierig zu realisieren, da die Zahl der akkuraten Datenpunkte begrenzt ist.

Mit dieser Methode erhält man eine Gegenüberstellung der Sollwerte, die von FRANZ bestimmt wurden, und der vom Backpropagationnetz vorhergesagten wie in Abbildung 24. Das Backpropagationnetz wurde mit 97 Datenpunkten und einem RMS-Wert (Route Mean Square) von 0.024 trainiert. Die Netzarchitektur sind 10 Eingabeneuronen (Variablen), eine versteckte Schicht mit 2 Neuronen und ein Ausgabeneuron, also ein Netz mit 25 Gewichten (inkl. BIAS,

einem Neuron, das zur konstanten Verschiebung der Summe der gewichteten Eingabewerte eines Neurons immer den Wert eins liefert.). Damit ist das Verhältnis Datenpunkte (97) zu Gewichten (25), wie gefordert, größer 3. Sagt man mit diesem Netz für alle 139 Datenpunkte, inklusive der 42 Ausreißer, die Reaktionswahrscheinlichkeiten vorher, erhält man nur für 32 Reaktionen Fehler der Reaktionswahrscheinlichkeit größer 0.3.

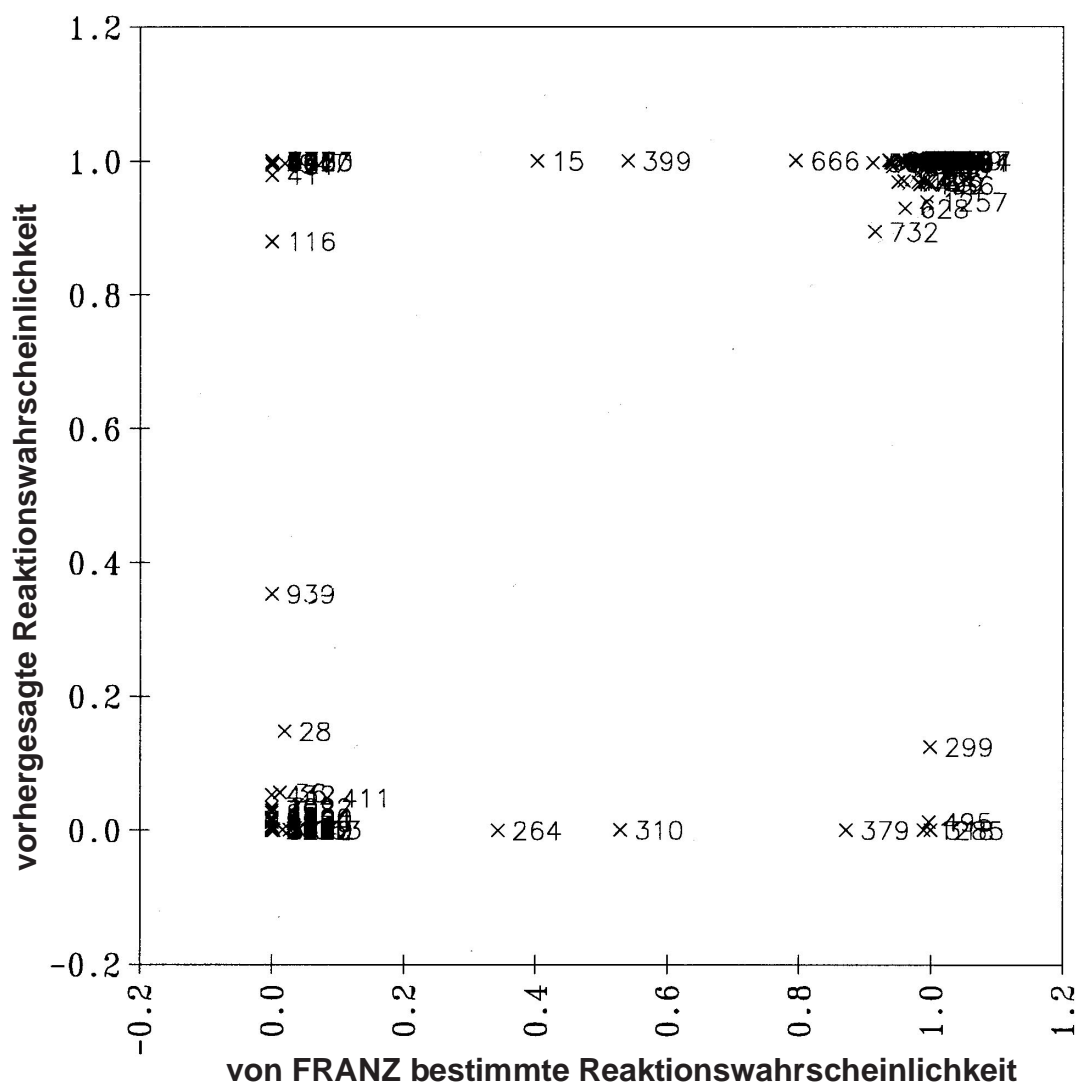


Abbildung 24: Diagramm, bei der die mit dem Backpropagationnetz vorhergesagten Reaktionswahrscheinlichkeiten für alle 139 Datenpunkte gegen die von FRANZ bestimmten (Sollwerte) aufgetragen sind.

Schließlich wird das so gewonnene Backpropagationnetz überprüft. Üblicherweise teilt man hierzu die Datenpunkte vor dem Training in einen Trainings- und Testdatensatz. Die Datenpunkte im Testdatensatz werden nach dem Training verwendet, um die Güte des Netzes festzustellen. In einigen Fällen, wie auch für die Vorhersage der Reaktionswahrscheinlichkeiten für die Carbonyleliminierung, reichen die Datenpunkte hierzu jedoch nicht aus. Ist die Gewinnung zusätzlicher Datenpunkte nur schwer möglich, muß eine andere Methode gefunden werden, das Netz zu testen. Im Fall der Carbonyleliminierung wurden die von FRANZ bewerteten Reaktionen in zwei Klassen eingeteilt (siehe 4.2.2.2): diejenigen, bei denen die Differenz der Unter- und Obergrenze für die Reaktionswahrscheinlichkeiten den Wert 0.2 (Klasse-1) bzw. 0.4 (Klasse-2) nicht überschreiten. Für die Ableitung des neuronalen Netzes wurden alle 139 Carbonyleliminierungen der Klasse-1 verwendet. Zur Überprüfung können die 224 etwas unsiche-

reren Datenpunkte der Klasse-2 eingesetzt werden. Besonderer Wert bei dieser Überprüfung muß dabei auf zusätzliche Datenpunkte in der Klasse-2 gelegt werden, da die Klasse-2 auch alle Datenpunkte der Klasse-1 enthält, die für die Ableitung des Netzes verwendet wurde. Die Datenpunkte, die sowohl in der Klasse-1, als auch in der Klasse-2 enthalten sind, können für die beiden Klassen von FRANZ unterschiedliche Werte zugeordnet bekommen. Ein Datenpunkt entspricht einer Reaktion, die FRANZ bei der Analyse der untersuchten Massenspektren und Strukturen öfters gefunden haben kann. Tritt eine bestimmte Reaktion mehrmals auf, trägt FRANZ den Mittelwert der aufgetretenen Reaktionswahrscheinlichkeiten in der Datenbank der Reaktionen unter der entsprechenden Klasse ein. Dies ist beispielsweise für die Reaktion 732 zu beobachten: Klasse-1 0.915; vorhergesagter Wert 0.893; Klasse-2 0.793.

Überprüft man nun das Backpropagationnetz mit den Datenpunkten der Klasse-2, haben 53 der 224 Datenpunkte eine Abweichung der Reaktionswahrscheinlichkeit größer 0.3 (siehe Abbildung 25). Vergleicht man diesen Anteil abweichender Reaktionswahrscheinlichkeiten für Datenpunkte der Klasse-2 mit denen der Klasse-1 (32 aus 139 mit einer Abweichung größer 0.3), stellt man fest, daß bei fast doppelter Anzahl der Datenpunkte der Prozentsatz nahezu identisch ist. Zusammenfassend kann man sagen, daß das Backpropagationnetz für die Schwierigkeiten ein brauchbares Ergebnis liefert.

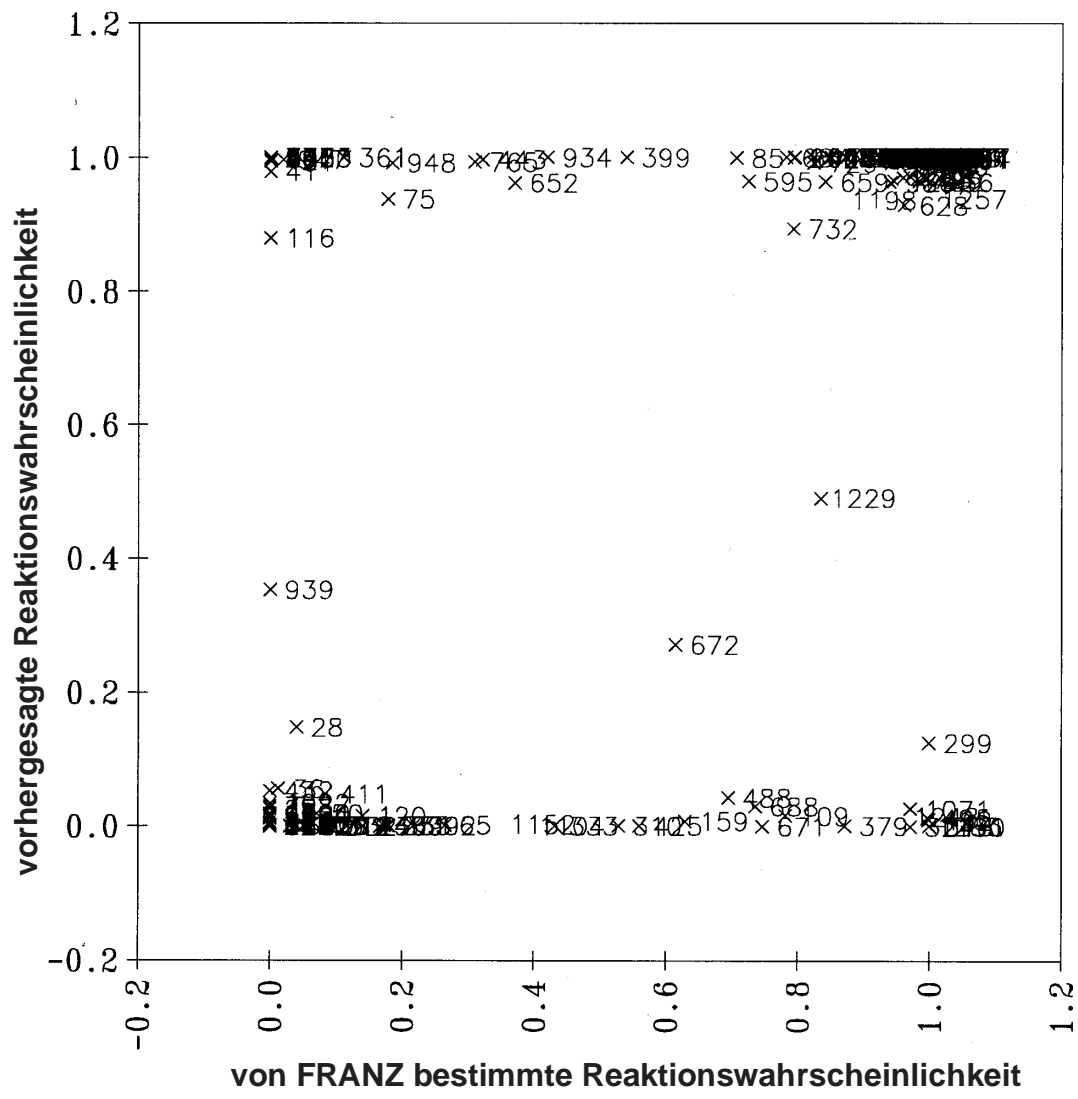


Abbildung 25: Diagramm der Klasse-2-Datenpunkte gegen die von FRANZ bestimmten Werte.

Schafft man es so für alle Reaktionstypen neuronale Netze zu trainieren und für die Umlagerungsreaktionen logistische Regressionsanalysen bzw. auch ein Backpropagationnetz zu trainieren, hat man bei einer Simulation, die die gewonnenen „Reaktivitätsfunktionen“ verwenden, immer noch die Probleme, daß für konkurrierende Reaktionen aus Umlagerungssystemen oft nicht, wie notwendig, unterschiedliche Reaktionswahrscheinlichkeiten vorhergesagt werden können (siehe 4.2.2.3).

4.2.5.2. Regeln ohne Umlagerungsreaktionstypen

Für die Ableitung der neuronalen Netze zur Bewertung der einzelnen Reaktionen unter Ausschluß von Umlagerungsreaktionstypen wurden die Fragmentierungstypen, wie in Kapitel 4.2.3.2 beschrieben, verwendet. Der gesamte Satz an Reaktionstypen ist in Abbildung 26 gezeigt.

α -Spaltung:



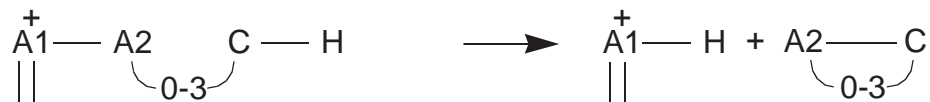
induktive Spaltung (X = Heteroatom, A = nicht H):



induktive α -Spaltung (A1, A2 = nicht H):



Onium-Reaktion (A kein Kohlenstoff, nicht radikalisches Ion):



Carbonyleliminierung (X = Heteroatom, nicht radikalisches Ion):



H₂-Eliminierung:



HX-Eliminierung



McLafferty-Spaltung

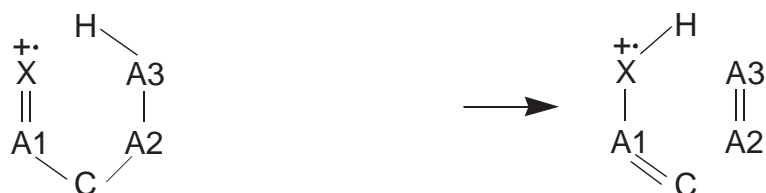


Abbildung 26: Regeln ohne Umlagerungsreaktionstypen.

Als Daten wurden die Strukturen und Massenspektren von 103 Moleküle genommen, die alle nur eine funktionelle Gruppe und keine Ringe enthalten. Ringe wurden ausgeschlossen, damit

auch Fragmentierungsreaktionstypen keine Umlagerungsreaktionen erzeugen, indem eine der Ringbindungen gebrochen wird. Die Moleküle dieses Datensatzes sind in Anhang C gezeigt und enthält:

- 6 Carbonsäuren
- 19 Ester (8 kurzkettige und 11 langkettige)
- 1 Amid
- 7 Aldehyde
- 5 Ketone
- 8 Acetale
- 2 Ketale
- 6 Ether
- 1 Thioether
- 1 Thiol
- 10 Alkohole (4 primär, 4 sekundär und 2 tertiär)
- 15 Amine (5 primär, 4 sekundär und 6 tertiär)
- 12 Halogenide (6 Chloride, 4 Bromide und 2 Iodide)
- 10 Alkene

Die Summe aller Signalintensitäten in den Massenspektren dieser Verbindungen konnte von FRANZ durchschnittlich zu 73% aufgeklärt werden. Die so erzeugten Reaktionsdatenbanken mit Reaktionswahrscheinlichkeiten wurden anschließend in gleicher Weise für die Ableitung der neuronalen Netze bzw. der expliziten Funktionen eingesetzt, wie es in 4.2.5.1 beschrieben ist (Regeln mit Umlagerungstypen). Um zu zeigen, daß so Massenspektren vorhergesagt werden können, wurden zunächst nur Bewertungen für folgende vier Fragmentierungstypen abgeleitet: α -Spaltung, Oniumspaltung, Carbonyleliminierung und die McLafferty-Spaltung (siehe Abbildung 27). Die Beschränkungen auf nur eine funktionelle Gruppe, keine Ringe und auf vier bewertete Fragmentierungstypen begrenzt die Einsatzmöglichkeit der so gewonnenen Regeln für die Simulation von Spektren durch das Programm MASSIMO auf Verbindungen, in deren Massenspektren keine anderen Fragmentierungstypen eine entscheidende Rolle spielen und bei denen die Fragmentierung einer funktionellen Gruppe nicht durch die Anwesenheit einer anderen beeinflußt wird.

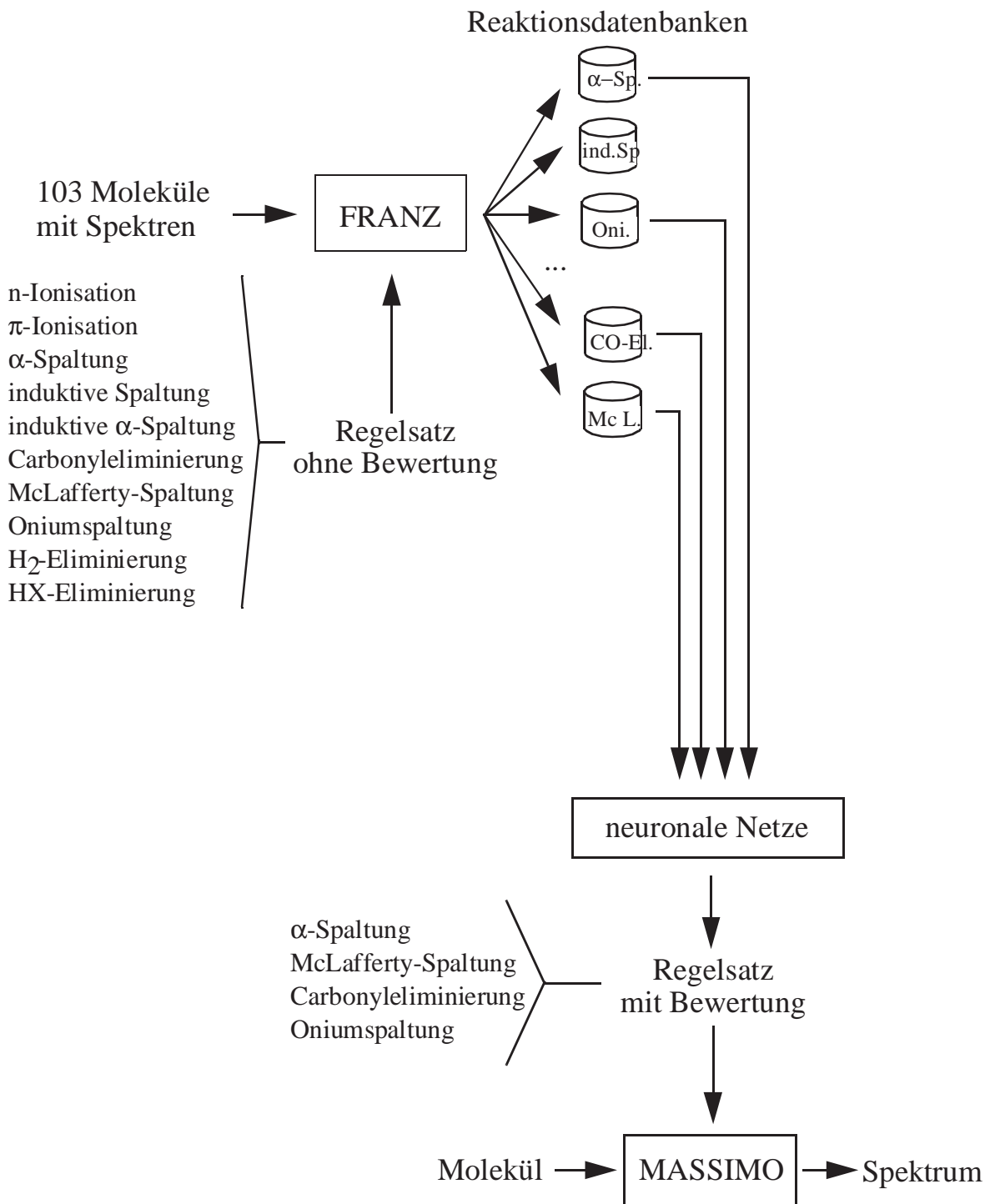


Abbildung 27: Ableitung der Bewertungen für die vier Fragmentierungstypen α -Spaltung, McLafferty-Spaltung, Carbonyleliminierung und Oniumspaltung.

Für die drei Fragmentierungstypen α -Spaltung, Oniumreaktion und Carbonyleliminierung wurden jeweils Backpropagationnetze ausgehend von den Datenpunkten in den Reaktionsdatenbanken der Klasse-1 trainiert. Die Reaktionen der Klasse-1 sind für die α -Spaltung in Anhang D.2.1, für die Oniumreaktion in Anhang D.2.2 und für die Carbonyleliminierung in Anhang D.2.3 gezeigt.

Ausgehend von 153 Datenpunkten der Klasse-1 konnte nach dem Ausschluß der „Ausreißer“ und damit der Reduktion der Datenpunkte auf 126 ein Backpropagationnetz abgeleitet werden [41], das die Reaktionswahrscheinlichkeiten der α -Spaltung aus 12 Variablen vorhersagt. Das Netz besitzt neben den 12 Eingabeneuronen eine versteckte Schicht aus drei Neuronen und ein Ausgabeneuron. Zählt man das BIAS-Neuron mit, das immer den Wert 1.0 liefert und somit den Übergangsbereich der sigmoiden Transferfunktion auf das Neuron in der nächsten Schicht verschiebt, besitzt dieses Netz 43 Gewichte. Es ist in Abbildung 28 dargestellt und hatte nach dem Training für die 126 Datenpunkte einen RMS-Wert von 0.095. 37 Werte hatten bei der Vorhersage eine Abweichung größer 0.1, vier größer 0.2 und ein Wert größer als 0.3. Kein Wert hatte eine Abweichung größer 0.4.

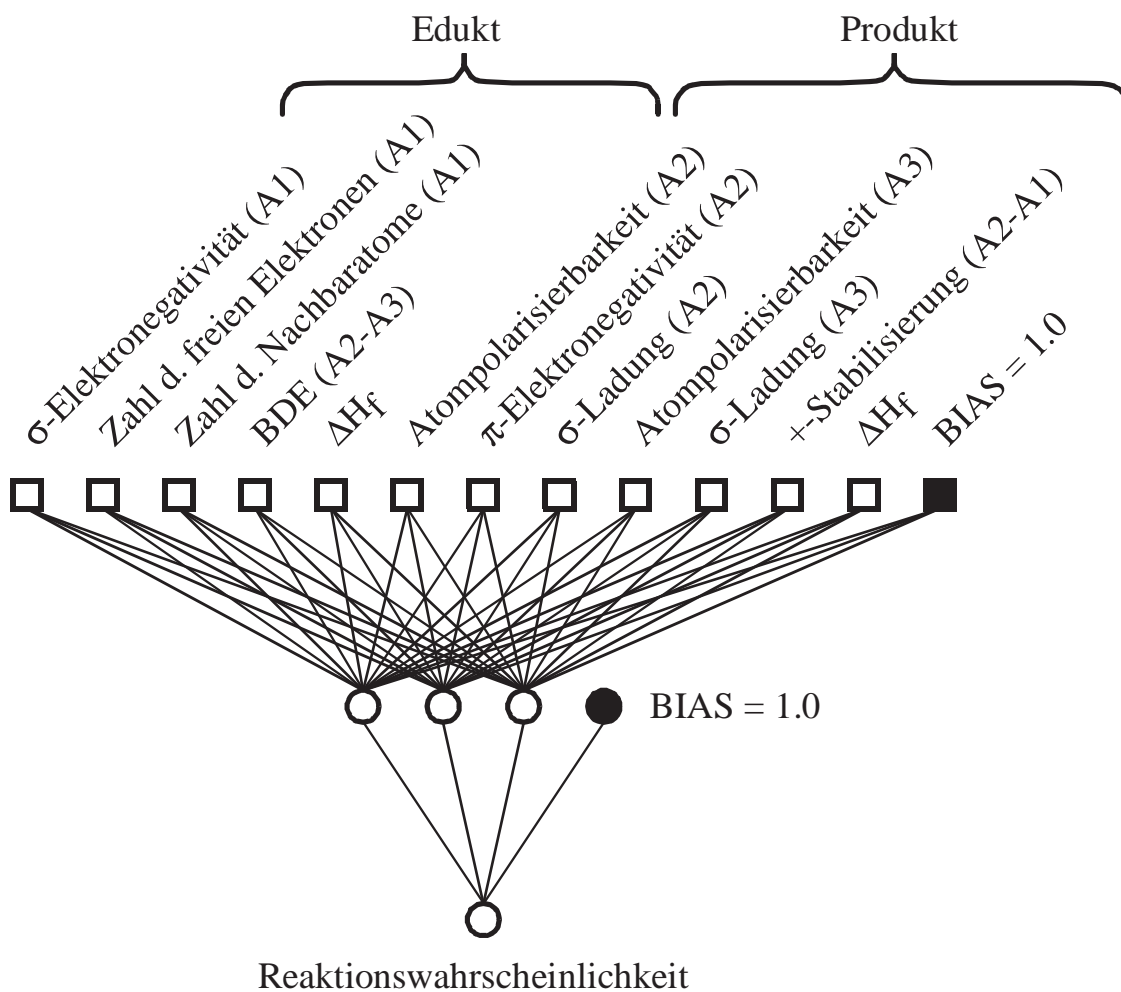


Abbildung 28: Aufbau des Backpropagationnetzes zur Vorhersage der Reaktionswahrscheinlichkeit der α -Spaltung.

Die Reproduktionsfähigkeit der Reaktionswahrscheinlichkeiten dieses Netzes ist in Abbildung 29 gezeigt. Hier sind die vorhergesagten gegen die von FRANZ vorgegebenen Wahrscheinlichkeiten aller 153 Klasse-1-Daten aufgetragen. Diesem Diagramm ist zu entnehmen, daß die Zusammenhänge vom Backpropagationnetz erkannt wurden und auch größtenteils reproduziert werden können. Die Zahl der Punkte der Reaktionen mit einer starken Abweichung der vorhergesagten Wahrscheinlichkeit deuten darauf hin, daß dem Netz nicht alle notwendigen Informationen zur Verfügung stehen. Dazu gehört sicherlich die Spezifikation der Anregungsenergie des Kations, die stark von seiner Entstehungsgeschichte abhängt. Die Energieverteilung der Ionen ändert sich einerseits mit der Reaktionsenthalpie, wenn das Ion genügend Energie be-

sitzt, um die Aktivierungsbarriere zu überwinden. Wird bei der Fragmentierung ein Neutralfragment abgespalten, verliert das Kation um so mehr Energie, je größer und schwerer das abgespaltene Molekül ist. Da im Massenspektrometer bevorzugt große Fragmente abgespalten werden, verliert das Kation bei solchen Reaktionen den größten Teil seiner Anregungsenergie. Die Reaktionswahrscheinlichkeit dieser kalten Ionen muß wesentlich kleiner sein als die von Kationen mit einer Anregungsenergie, wie sie nach einer Elektronenstoßionisation vorliegt.

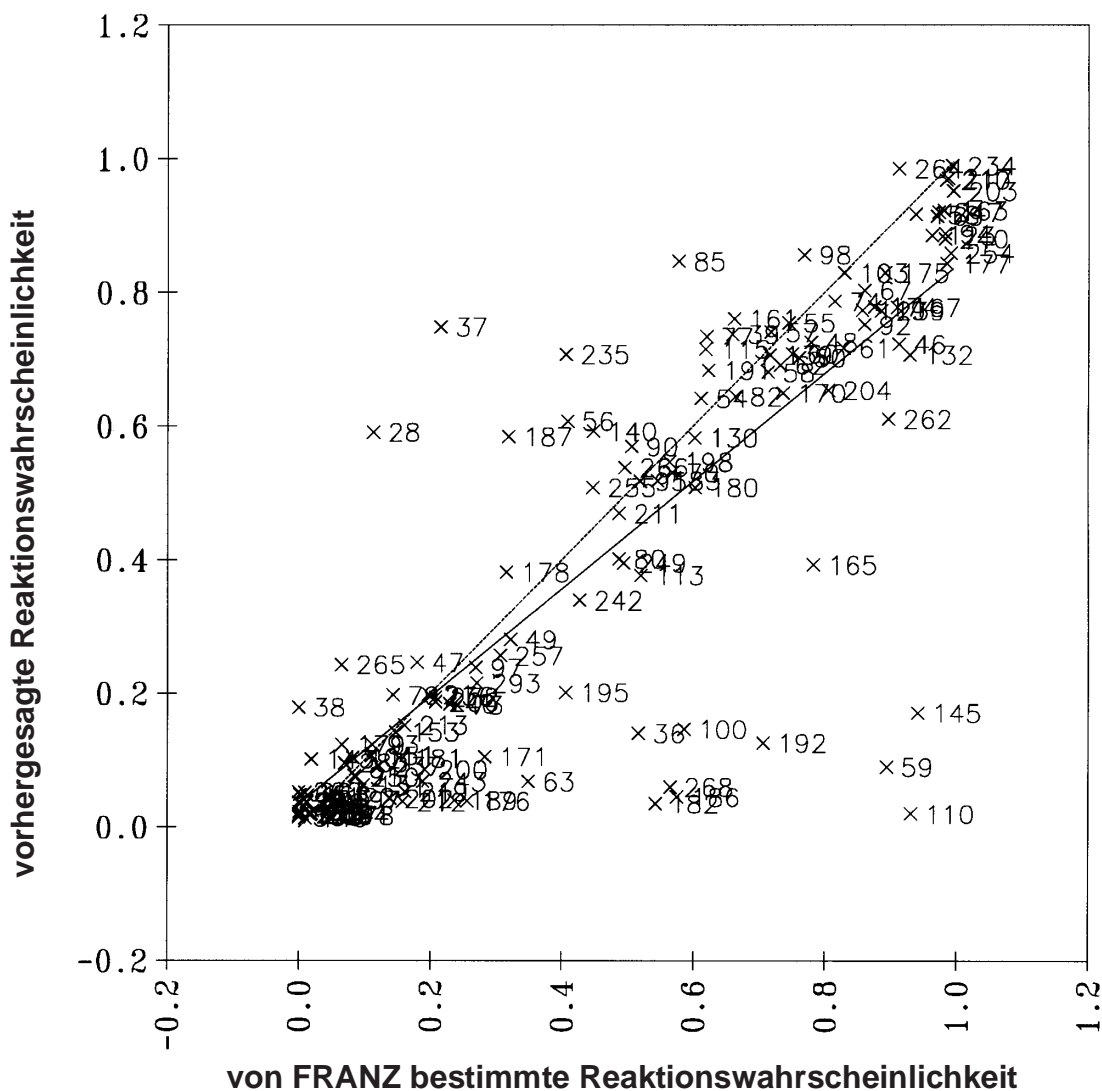


Abbildung 29: Korrelation der vom Backpropagationnetz vorhergesagten mit den von FRANZ ermittelten Reaktionswahrscheinlichkeiten für die α -Spaltung. Die durchgezogene Linie ist die Regressionsgerade und die gestrichelte die Winkelhalbierende, auf der alle Punkte idealerweise liegen sollten. Die Nummern der Datenpunkte sind die Nummern der Reaktionen im Anhang D.2.1.

Auch für die Oniumreaktion wurde mit 44 von 49 Klasse-1-Daten ein Backpropagationnetz trainiert, das 11 Variablen verwendet und zwei Neuronen in einer versteckten Schicht besitzt. Der RMS-Wert betrug 0.061, aber mit 27 Gewichten bei 44 Datenpunkten genügt es dem Kriterium, daß die Zahl der Datenpunkte mehr als dreimal so groß sein soll, wie die Zahl der Gewichte im Netz, nicht. Da eine weitere Reduktion der Variablen und damit der Gewichte ohne drastischen Verlust an Reproduktionsfähigkeit der trainierten Daten nicht möglich war und nicht so einfach

zusätzlich Datenpunkte beschafft werden konnten, wurde das Netz dennoch für die Simulation herangezogen. Daß dieses Netz auch für andere Reaktionen, die nicht trainiert wurden, die Reaktionswahrscheinlichkeiten korrekt vorhersagen kann, ist nicht unbedingt anzunehmen, da es bei diesem Verhältnis Datenpunkte zu Gewichten im Netz durchaus möglich ist, daß einige Datenpunkte vom Netz auswendig gelernt wurden und nicht überall der Zusammenhang verallgemeinert wurde. Die Korrelation der von diesem Netz vorhergesagten Datenpunkte der Klasse-1 der Oniumreaktionen ist in Abbildung 30 zu sehen.

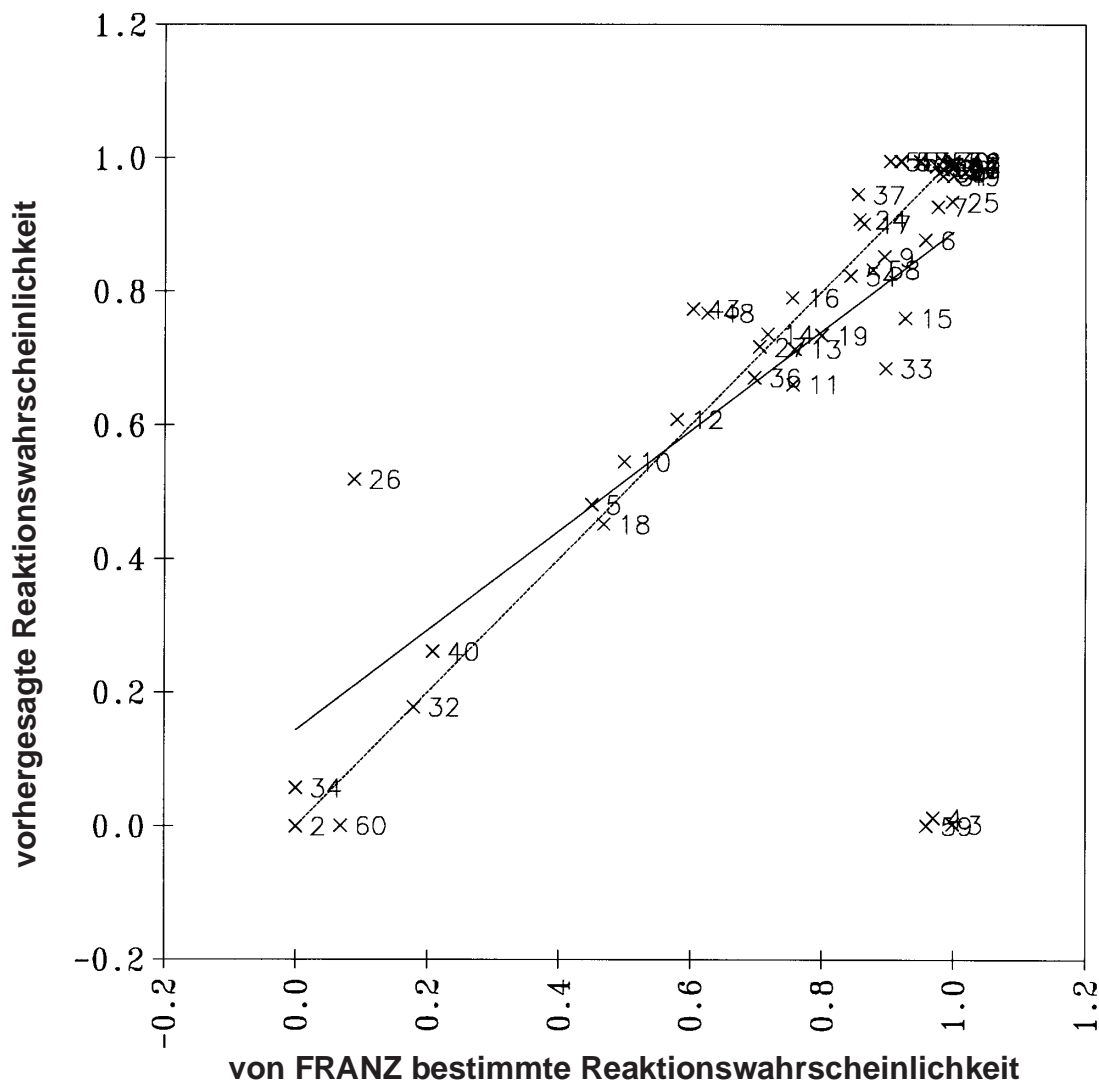


Abbildung 30: Korrelation der vom Backpropagationnetz vorhergesagten mit den von FRANZ ermittelten Reaktionswahrscheinlichkeiten für die Oniumreaktion. Die durchgezogene Linie ist die Regressionsgerade und die gestrichelte die Winkelhalbierende, auf der alle Punkte idealerweise liegen sollten. Die Nummern der Datenpunkte sind die Nummern der Reaktionen im Anhang D.2.2.

Noch schlechter war die Situation mit 13 Datenpunkten der Klasse-1 für die Carbonyleliminierung. Trainiert wurde ein Backpropagationnetz mit 11 der 13 Datenpunkte, 2 Variablen und einem Neuron in einer versteckten Schicht und einem RMS-Wert von 0.128. Abbildung 31 zeigt die Gegenüberstellung der vorhergesagten und der vorgegebenen Werte.

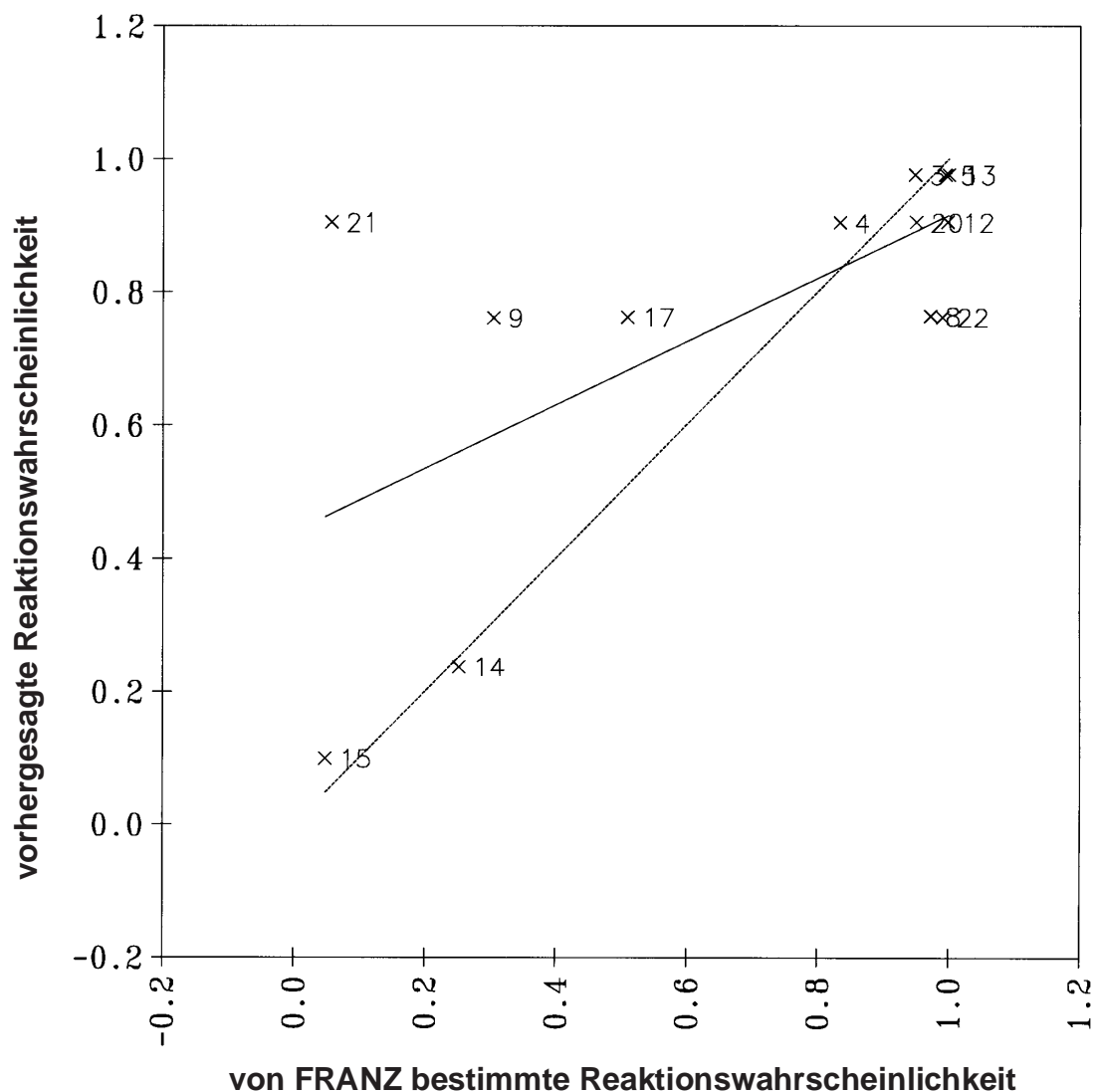


Abbildung 31: Korrelation der vom Backpropagationnetz vorhergesagten mit den von FRANZ ermittelten Reaktionswahrscheinlichkeiten für die Carbonyl-eliminierung. Die durchgezogene Linie ist die Regressionsgerade und die gestrichelte die Winkelhalbierende, auf der alle Punkte idealerweise liegen sollten. Die Nummern der Datenpunkte sind die Nummern der Reaktionen im Anhang D.2.3.

Für die McLafferty-Spaltung gab es schließlich nur noch vier Datenpunkte in der Klasse-1. Deshalb wurden alle 14 Datenpunkte der Klasse-2 (siehe Anhang D.2.4) herangezogen. Betrachtet man sich den Datensatz genauer, fällt auf, daß sich die McLafferty-Reaktionen in zwei Gruppen teilen. Zehn Reaktionen mit Reaktionswahrscheinlichkeiten von 0.787 bis 0.894 bilden eine Gruppe, bei denen das Wasserstoffatom von einer CH_2 -Gruppe auf das Carbonylsauerstoffatom umgelagert wird. Die andere Gruppe sind vier Reaktionen, bei denen das Wasserstoffatom von einer Methylgruppe abgezogen wird und Wahrscheinlichkeiten von 0.002 bis 0.544 haben. Demzufolge erhalten die McLafferty-Reaktionen die Mittelwerte der Reaktionswahrscheinlichkeiten 0.8361 (H von CH_2) bzw. 0.23 (H von CH_3).

4.3. Simulation von Massenspektren mit MASSIMO

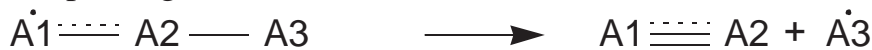
Hat man, wie in Kapitel 4.2 beschrieben, die Wissensbasis, also die Fragmentierungsregeln mit ihren Bewertungsfunktionen, erstellt, können mit dem Programm MASSIMO für beliebige Strukturen die Massenspektren vorhergesagt werden. Die Beliebigkeit der Strukturen gilt natürlich nur insoweit, als die Gültigkeit der Regeln nicht überschritten wird, die von den Fragmentierungstypen und Molekülen zusammen mit ihren Massenspektren abhängt, die zur Ableitung der Wissensbasis verwendet wurden. MASSIMO wendet, beginnend mit der Ausgangsverbindung, zunächst die zwei Ionisationstypen (n - und π -Ionisation) und dann die in der Wissensbasis kodierten Fragmentierungstypen auf alle entstehenden Produktkationen an. Für alle Reaktionen wird die Reaktionswahrscheinlichkeit und daraus für alle konkurrierenden Reaktionen die Stoffmenge der beteiligten Ionen berechnet. Dies wird solange fortgesetzt, bis entweder die Masse oder die Stoffmenge des Produkts unter einen bestimmten Schwellwert fällt.

4.3.1. Verwendete Regeln

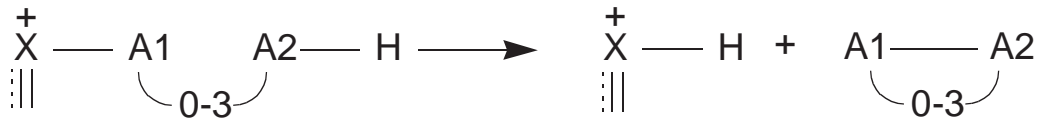
Als Regeln wurden neben den beiden Ionisationsarten (n - und π -) die vier Fragmentierungstypen verwendet, für die in Kapitel 4.2.5.2 eine Bewertungsfunktion oder eine neuronales Netz abgeleitet wurde. Sie sind in Abbildung 32 gezeigt.

n-Ionisation und π -Ionisation

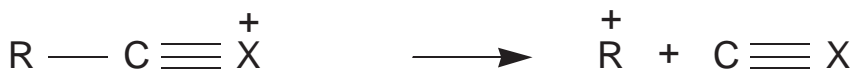
α -Spaltung:



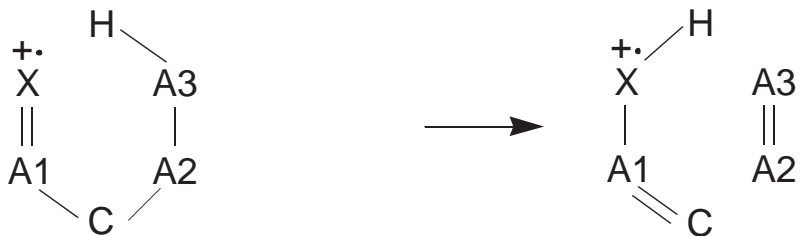
Onium-Reaktion (kein Atom Teil eines Rings, X kein Radikalzentrum):



Carbonyleliminierung (kein Radikalzentrum im Fragment)



Mc Lafferty-Spaltung:



- H ist ein Wasserstoffatom
- C ist ein Kohlenstoffatom
- X ist ein Heteroatom (nicht Kohlenstoff und nicht Wasserstoff)
- R ist ein beliebiges Atom außer Wasserstoff
- A, A1, A2, ... sind beliebige Atome

Abbildung 32: Fragmentierungstypen für die Vorhersage von Massenspektren.

So hat man Bewertungen für die vier Fragmentierungstypen α -Spaltung, Oniumreaktion, Carbonyleliminierung und McLafferty-Spaltung. Aber bevor die Fragmentierungen zum Zuge kommen, muß das Molekül ionisiert werden. Die Regeln für die beiden Ionisationsarten konnten aus dem Satz von Regeln für FRANZ entnommen werden, aber für die Ableitung der Wahrscheinlichkeitsfunktionen standen keine Daten zur Verfügung, da die verwendeten 103 Moleküle alle nur eine funktionelle Gruppe und somit meist nur einen Ionisationsort besitzen. Die Verteilung der Stoffmengen auf die unterschiedlichen Ionisationsprodukte wurde daher an die Boltzmann-Verteilung angelehnt. Die Umsätze für die verschiedenen Ionisationen U_i werden folgendermaßen bestimmt (Die Stoffmenge der nicht ionisierten Verbindung ist ohne Belang, da sie kein Signal zum Massenspektrum beisteuert.):

$$U_i = \frac{e^{-c \cdot IP_i}}{\sum_k e^{-c \cdot IP_k}} \quad (3)$$

IP ist dabei das erste Ionisationspotential in Elektronenvolt (eV), das aus den physikochemischen Parametern berechnet wird [26][46]. c ist eine Konstante, die an bifunktionellen Verbindungen wie 11-Aminododecansäuremethylester zu 2.0 bestimmt wurde. Diese Verbindungen haben zwei Ionisationsorte (ohne σ -Ionisation), bei denen die Ionisation an einem Zentrum deutlich dominiert. c wurde so gewählt, daß die Signalverhältnisse der beiden Signalgruppen für die Ionisationen (m/z 72 für das Produkt der McLafferty-Spaltung und m/z $M-1$, $M-15$ sowie m/z 44 für die Produkte der α -Spaltungen von der ionisierten Aminogruppe aus) im simulierten Massenspektrum gut wiedergegeben werden (siehe Abbildung 33).

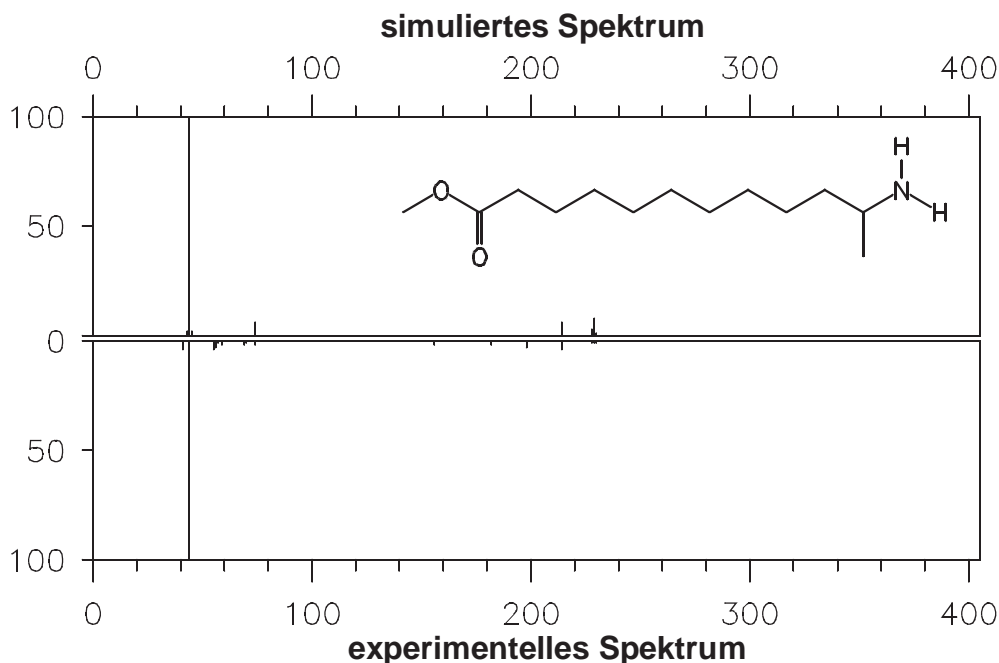


Abbildung 33: Spektrenvergleich für 11-Aminododecansäuremethylester (experimentelles Spektrum [7]).

Für 11-Aminododecansäuremethylester erhält man Ionisationspotentiale von 8.96 eV für die Aminogruppe und 10.23 eV für das freie Elektronenpaar an der Carboxylgruppe. Die verwendete Methode zur Berechnung der Ionisationspotentiale kann nur das niedrigste Potential einer funktionellen Gruppe berechnen. Die übrigen Ionisationen einer funktionellen Gruppe besitzen mit ihren höheren Ionisationspotentialen deutlich geringere Ionisationswahrscheinlichkeiten und können deshalb vernachlässigt werden. Aus diesem Grund wird auch für die Carbonylgruppe nur eine Ionisation erzeugt. Nach der Formel 3 ergibt sich dann eine Verteilung von 92.7% Ionisation an der Aminogruppe und nur zu 7.3% an der Carboxylgruppe, was recht gut mit dem Spektrum in Abbildung 33 vereinbar ist. Das Signal bei m/z 72 (siehe Abbildung 33) ist etwas kleiner als simuliert, da für Signale mit geringer Intensität im experimentellen Massenspektrum fast immer eine zu geringe Intensität bestimmt wird [8] und bei einer noch stärkeren Gewichtung der Ionisationsumsätze die Gefahr besteht, daß einige Signale überhaupt nicht mehr im simulierten Spektrum zu sehen sind, wenn das Ionisationspotential etwas zu hoch berechnet wird. Die Genauigkeit der berechneten Ionisationspotentiale ist etwa 0.3 eV, die der experimentell bestimmten ist 0.05 bis 0.2 eV [26].

Simuliert man beispielhaft das Massenspektrum von 3-Hexanon (siehe Abbildung 34) für Car-

bonylverbindungen, fällt auf, daß die beiden Carbonyleliminierungen (m/z 71 \rightarrow 43 und 57 \rightarrow 29) nach den α -Spaltungen vom Carbonylsauerstoffatom aus zu hoch bewertet sind. Das Signal bei m/z 72 ist das Produkt der McLafferty-Spaltung. Diese ist auch zu hoch bewertet, da für die wenig reaktiven McLafferty-Spaltungen der Mittelwert der Reaktionswahrscheinlichkeiten von 0.002 bis 0.544 genommen wurde. 3-Hexanon war im Trainingsdatensatz und hatte eine Wahrscheinlichkeit für die McLafferty-Reaktion von 0.079 (siehe Reaktion 9 im Anhang D.2.4). Die Signale bei den Massen m/z 41, 39 und 27 können nicht simuliert werden, da der Regelsatz die Wasserstoffmoleküleliminierung nicht enthält, durch die diese Signale aus den Fragmenten der Massen m/z 43 und 29 entstehen.

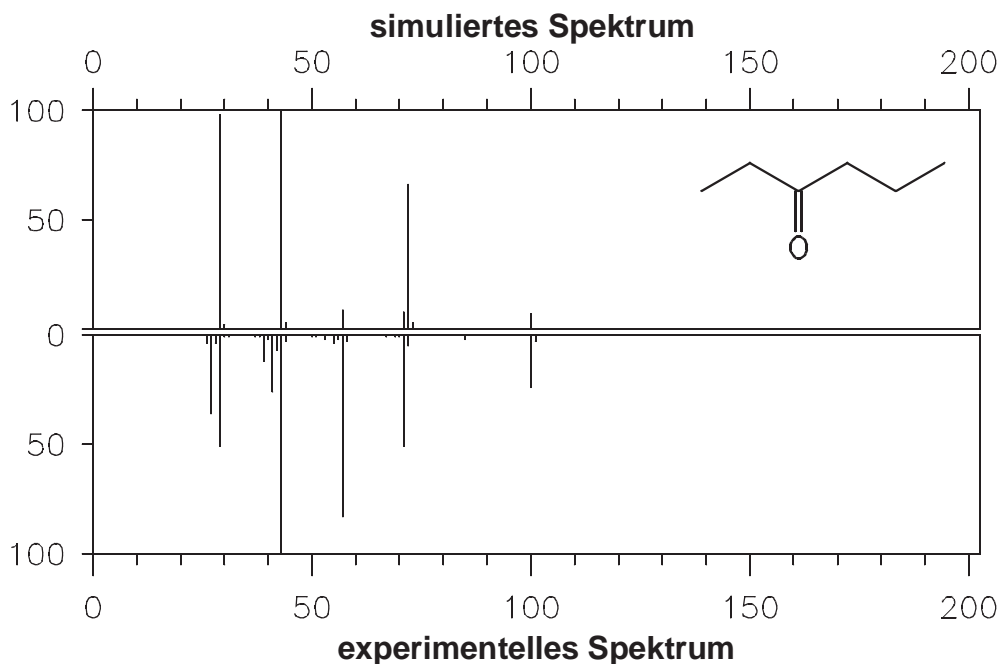


Abbildung 34: Spektrenvergleich für 3-Hexanon (experimentelles Spektrum [22]).

Die zu hohe Bewertung der Carbonyleliminierungen liegt daran, daß FRANZ bei der Bestimmung der Wahrscheinlichkeiten offensichtlich weitere zum Edukt der Reaktionen gleich schwere Ionen gefunden hat, die nicht zu einem Umlagerungssystem zusammengefaßt wurden, da Umlagerungen durch die Wahl der Regeln und Moleküle ausgeschlossen wurden. So wird die Stoffmenge auf die beiden Edukte verteilt und dem Edukt der Carbonyleliminierung möglicherweise eine zu geringe Stoffmenge zugewiesen, wodurch eine zu hohe Reaktionswahrscheinlichkeit ermittelt wird.

Nimmt man die Hälfte der Reaktionswahrscheinlichkeit, die das neuronale Netz für die Carbonyleliminierung liefert, können die Spektren von Carbonylverbindungen gut simuliert werden, wie auch das Spektrum von 3-Hexanon in Abbildung 35 zeigt.

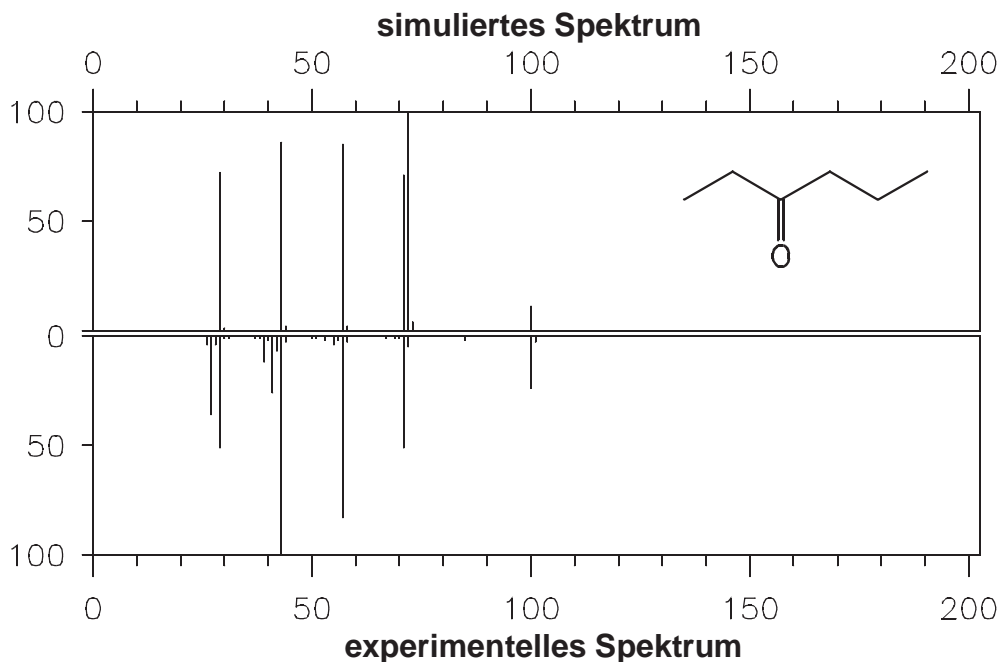


Abbildung 35: Spektrenvergleich für 3-Hexanon mit halber Reaktionswahrscheinlichkeit für die Carbonyleliminierung (experimentelles Spektrum [22]).

Gleiches gilt auch für die Oniumspaltung. Deshalb werden auch die Wahrscheinlichkeiten dieses Netzes für die Verwendung zur Simulation in MASSIMO mit 0.5 multipliziert.

4.3.2. Beispiele für die Leistungsfähigkeit des Ansatzes

Mit den Regeln, wie sie in Kapitel 4.3.1 beschrieben sind, also mit zwei Ionisations- (n und π) und vier Fragmentierungstypen (α -Spaltung, Oniumreaktion, Carbonyleliminierung und McLafferty-Spaltung), soll nun an einigen Beispielen die Simulation von Massenspektren gezeigt werden. Die im experimentellen Spektrum zusätzlich erscheinenden Signale stammen von Fragmentierungstypen, für die eine Vorhersagefunktion für die Reaktionswahrscheinlichkeit fehlt und deshalb nicht im Satz von Reaktionstypen für die Simulation enthalten sind. Beispiele für die Simulation von Massenspektren mit MASSIMO, bei denen der größte Teil der Summe der Signalintensitäten vorhergesagt werden kann, sind in den Abschnitten 4.3.2.2 bis 4.3.2.4 gezeigt. Die experimentellen Massenspektren sind den Datenbanken von MassLib [22] und Spec-Info [7] entnommen und waren mit Ausnahme von Butylmethylpropylamin mit seinem Spektrum (siehe Abbildung 43) nicht im Datensatz enthalten, mit dem die Wahrscheinlichkeitsfunktionen abgeleitet wurden.

4.3.2.1. Vergleich von simulierten und experimentellen Spektren

In den Abbildungen 36 und 37 sind die simulierten gegen die experimentellen Massenspektren von 3-Methoxy-3-methylhexan und N-Ethylbutylamid aufgetragen. Die Intensitäten der Signale von Fragmenten, die von den eingesetzten Regeln erzeugt werden, können gut vorhergesagt werden. Die nicht simulierten Signale werden von Fragmentierungstypen gebildet, für die keine Vorhersagefunktion für die Reaktionswahrscheinlichkeit vorhanden ist und deshalb nicht im Satz von Reaktionstypen enthalten ist, der zur Simulation verwendet wurde. In Abbildung 36 sieht man schön die Abstufung der verschiedenen α -Spaltungen ausgehend vom ionisierten Sauerstoffatom.

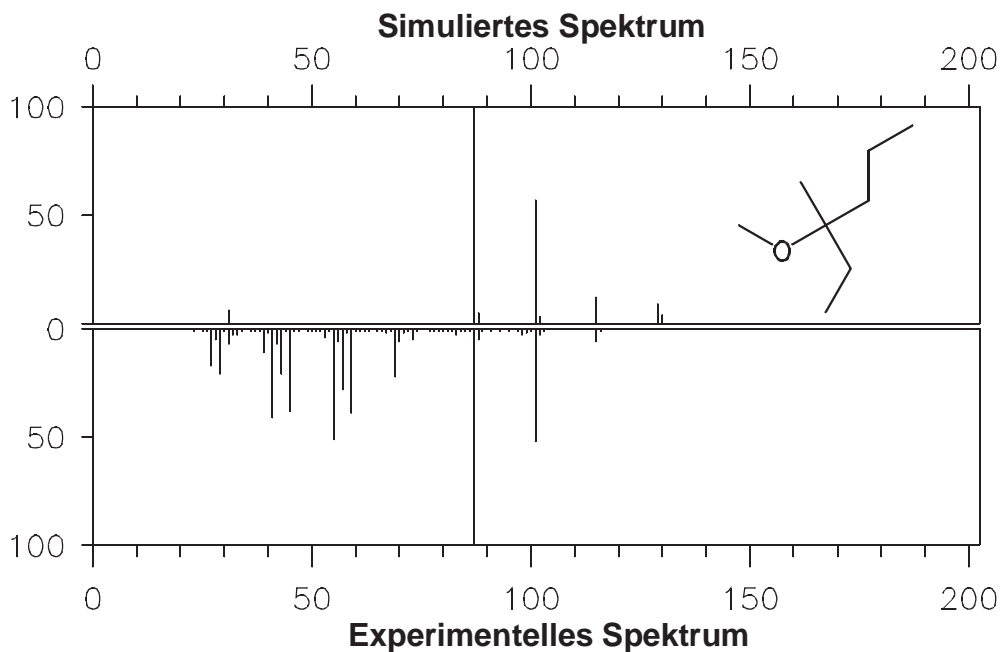


Abbildung 36: Simuliertes Massenspektrum von 3-Methoxy-3-methylhexan.

Ogleich im Trainingssatz neben Carbonsäuren und Estern nur ein nicht N-substituiertes Amid enthalten war, nämlich Hexylamid, werden die Signalintensitäten im Spektrum von N-Ethylbutylamid unter Berücksichtigung der fehlenden Reaktionstypen gut simuliert (nicht simulierte Signale siehe oben). Die neuronalen Netze für die Bestimmung der Reaktionswahrscheinlichkeiten konnten also verallgemeinern und damit auch Wahrscheinlichkeiten für Reaktionen mit ähnlichen physikochemischen Eigenschaften am Reaktionszentrum vorhersagen.

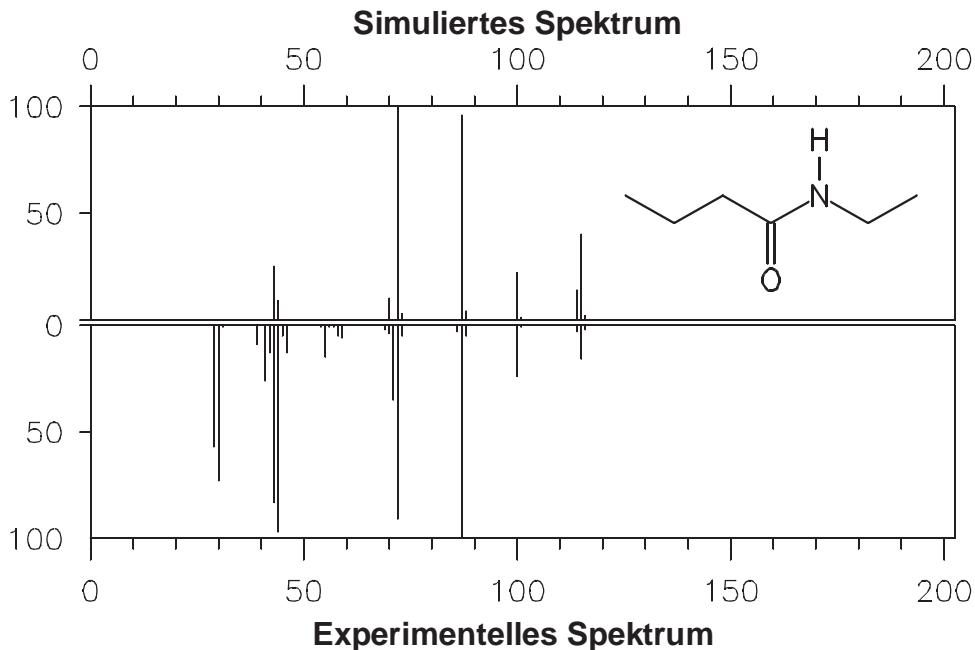


Abbildung 37: Simuliertes Massenspektrum von N-Ethylbutylamid.

Schließlich zeigt Abbildung 38 noch ein Beispiel für eine halogenhaltige, bifunktionelle Verbindung.

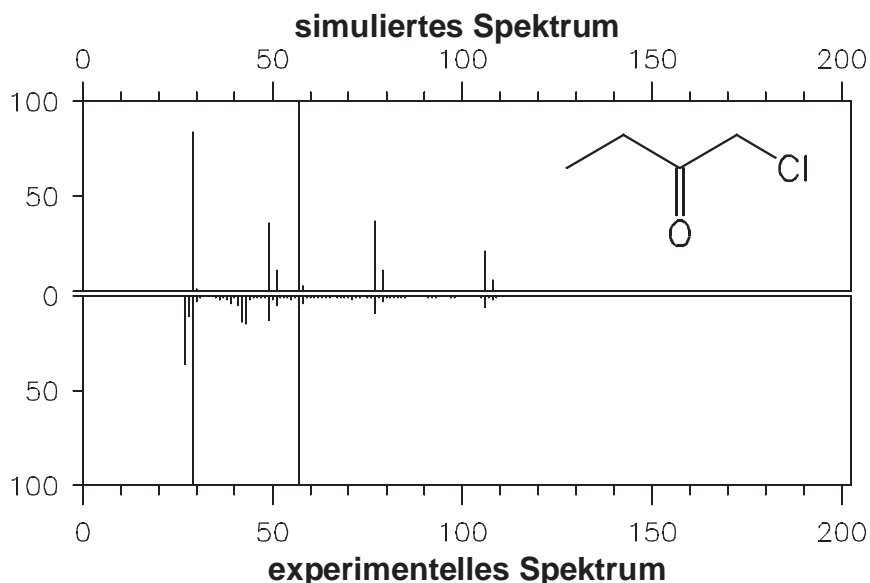


Abbildung 38: Simuliertes Massenspektrum von 1-Chlor-2-butanon.

4.3.2.2. Selektion eines Strukturvorschlages durch Spektrenvergleich

Ist es möglich für verschiedene, in Betracht kommende Verbindungen die Massenspektren zu simulieren, kann durch den Vergleich mit dem experimentellen Spektrum die richtige Struktur herausgefunden werden. Die Güte dieser Selektion ist um so besser, je besser die simulierten Spektren sind. Haben die Substanzen, aus denen die richtige ausgewählt werden soll, eine unterschiedliche Masse, kann schon über den Molekülpeak eine Entscheidung getroffen werden. Deshalb wurden für dieses Beispiel drei isomere Verbindungen der Summenformel $C_6H_{13}NO$ genommen, für die auf diese einfache Weise nicht zum Ziel gekommen werden kann. Dabei wurden Strukturen ausgewählt, deren experimentelles Massenspektrum zur Verfügung stand. Als einfaches Vergleichsmaß wird hier das Skalarprodukt verwendet, wobei die Massenspektren als Vektor aufgefaßt werden, dessen Elemente den Massen entsprechen und die Werte den Intensitäten. Ist das Signal bei m/z 57 der Basepeak, hat das 57. Element des Vektors den Wert $1.0 = 100\%$. Für Massen, die nicht im Spektrum beobachtet werden, ist der Wert null. Abbildung 39 zeigt nun die mit MASSIMO für die drei Strukturen simulierten Spektren und die Skalarprodukte dieser mit dem experimentellen Spektrum. Das Skalarprodukt hat einen Wertebereich von null bis eins, wobei identische Spektren den Wert eins besitzen. Das Spektrum der richtigen Struktur hat demnach das höchste Skalarprodukt.

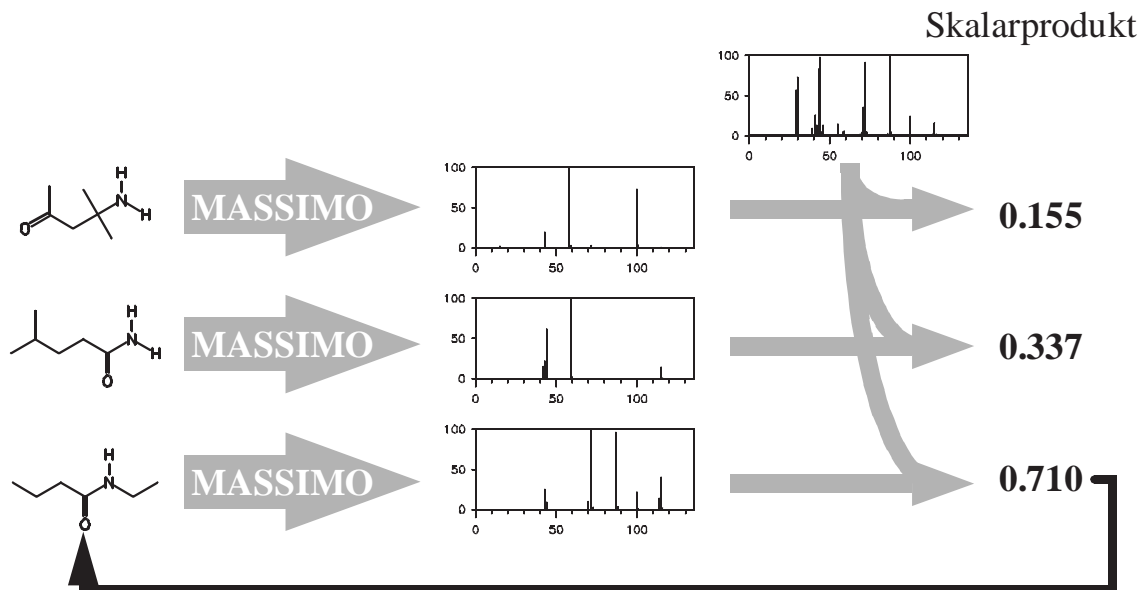


Abbildung 39: Selektion einer aus drei isomeren Verbindungen.

Da für alle drei Verbindungen das experimentelle Massenspektrum vorhanden war, wurde dieser Vergleich auch für die anderen zwei experimentellen Spektren durchgeführt. Alle drei Vergleiche sind in Abbildung 40 zusammengefasst. Eine Spalte entspricht dabei einer Selektion, deren letzte die Selektion in Abbildung 39 ist.

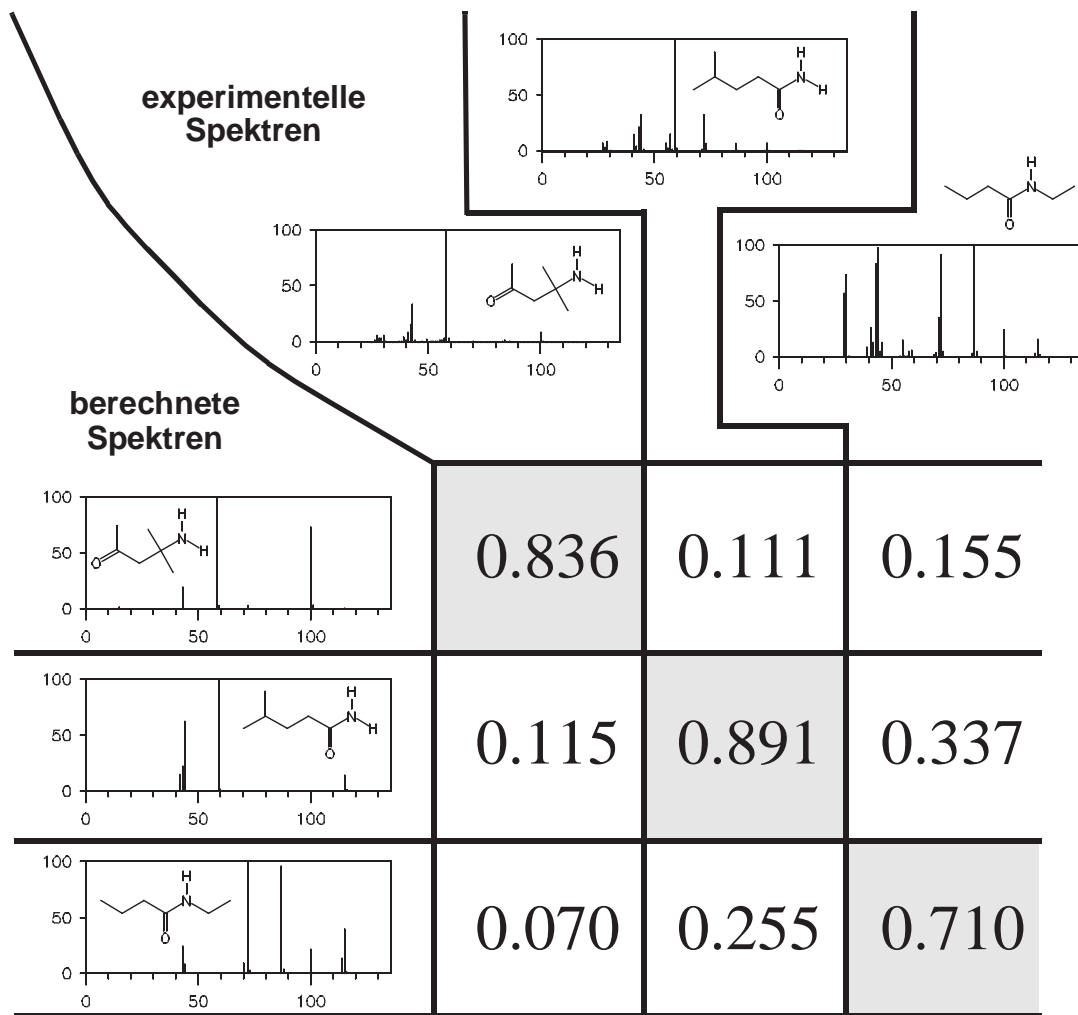


Abbildung 40: Vergleich aller drei Selektionen.

Vergleicht man diese drei Selektionen (Spalten) miteinander, stellt man fest, daß die anderen beiden Entscheidungen noch sicherer getroffen werden, da das Skalarprodukt für den Spektrenvergleich der richtigen Struktur noch größer und das mit dem nächstkleineren Wert noch kleiner ist. Die Felder für die richtigen Strukturen sind grau unterlegt.

4.3.2.3. Auffinden falscher Spektren in Datenbanken

Im Rahmen der Überprüfung, für welche Substanzen gute Massenspektren simuliert werden können, wurde auch das Spektrum von 11-Aminotridecansäuremethylester vorhergesagt und mit dem experimentellen aus der Datenbank (SpecInfo [7]) verglichen. Diese beiden Spektren sind in Abbildung 41 zu sehen.

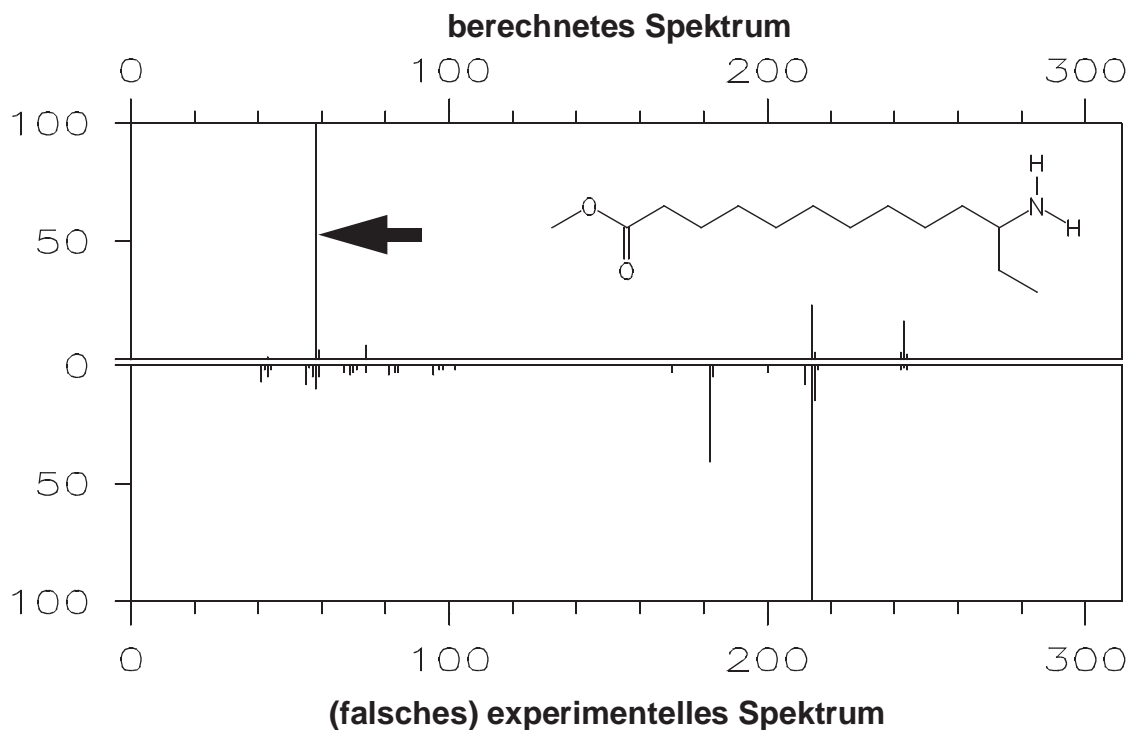


Abbildung 41: Spektrenvergleich von 11-Aminotridecansäuremethylester.

Beim Vergleich fällt auf, daß der Basepeak im vorhergesagten Spektrum bei m/z 58 (mit einem Pfeil markiert) im experimentellen Spektrum lediglich durch ein Signal von etwa 10% vertreten ist. Ähnlich verhält es sich in umgekehrter Weise mit dem Basepeak im experimentellen Spektrum. Betrachtet man auch den restlichen Teil der Spektren, kommt man zur Überzeugung, daß entweder das simulierte Spektrum falsch oder die beiden Spektren nicht zur selben Struktur gehören.

Bedenkt man, daß in Datenbanken oft auch ältere Spektren enthalten sind, bei denen die Zuordnung der Struktur durch die geringeren analytischen Möglichkeiten nicht mit der heute üblichen Sicherheit erfolgen konnte, muß man bei solchen Spektren durchaus mit der Angabe einer falschen Struktur rechnen. Darüber hinaus wurden viele dieser Spektren aus der Literatur abgetippt, was nicht in allen Fällen fehlerfrei gemacht wurde. Zudem wurden in der Literatur oft Signale mit geringer Intensität weggelassen. [8] Ausgehend von der angegebenen Struktur müßte allerdings der Peak bei m/z 58 (α -Spaltung von der Aminogruppe) die größte Intensität besitzen, was auch vom Massenspektroskopieexperten bestätigt wurde [8]. Auf diese Weise wurde also eine falsche Spektrum-Struktur-Zuordnung in einer Datenbank aufgefunden.

Mit einem Regelsatz, der für die Simulation nahezu aller organischer Verbindungen eingesetzt werden kann, könnte man den Inhalt von Datenbanken automatisch überprüfen. Die aussortierten Spektren sollten danach noch von einem Experten getestet werden, ob diese Entscheidung richtig war und es sich nicht doch um ein korrektes Spektrum handelt. Im Vergleich dazu müßten Experten ohne ein solches Hilfsmittel jedes in der Datenbank enthaltene Spektrum kontrollieren.

In gleicher Weise wurde die Fehlzuordnung für 1,1-Dimethoxyhexadecan (experimentelles Spektrum [7]) aufgedeckt. In diesem Fall konnte auch noch der Grund für die falsche Struktur angegeben werden. Bei der Aufnahme des Spektrums wurde die Verbindung 1,1-Dimethoxyhexadecan in das Inlett des Massenspektrometers geleitet, dort aber so stark erhitzt, daß die Verbindung Methanol eliminiert hat, und so das Spektrum des Pyrolyseprodukts 1-Methoxyhexadec-1-en aufgenommen wurde. [47] Dies wird auch von den von MASSIMO si-

mulierten Spektren bestätigt (siehe Abbildung 42).

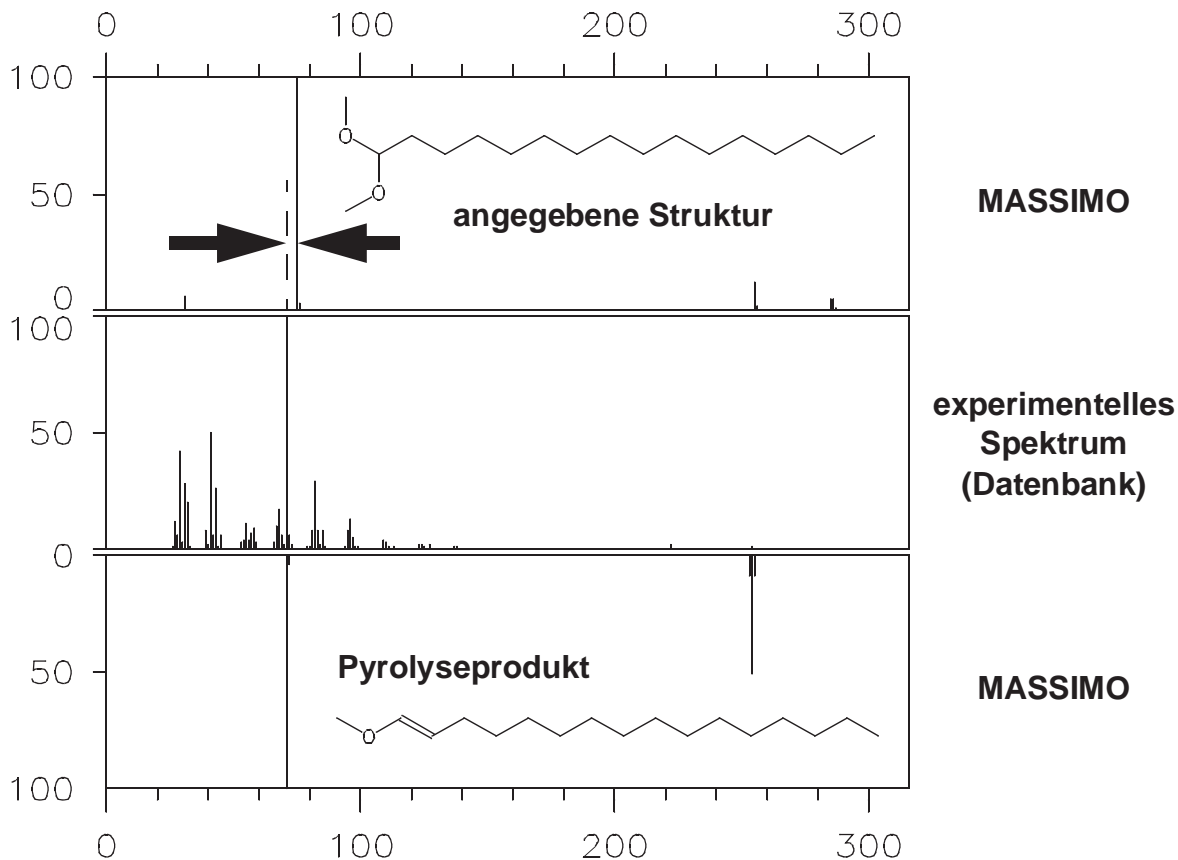


Abbildung 42: Statt von 1,1-Dimethoxyhexadecan wurde das Spektrum vom Pyrolyseprodukt 1-Methoxyhexadec-1-en aufgenommen.

4.3.2.4. Unterstützung von Massenspektroskopieexperten durch die Simulation von Spektren

Eine weitere Einsatzmöglichkeit der Spektrensimulation ist die Unterstützung des Massenspektroskopieexperten. Dies erfolgt zunächst, wie es in Kapitel 4.3.2.2 beschrieben wurde. Ist die Struktur dann aufgeklärt, muß vom Experten das Massenspektrum aufgrund dieser Struktur verifiziert werden. MASSIMO kann dabei dem Massenspektroskopieexperten zur Seite stehen und seinen Arbeitsaufwand reduzieren. Simuliert MASSIMO den größten Teil der Signale im Spektrum, wobei der Weg der Fragmentierungen zu den entsprechenden Ionen angegeben wird, gehören diese Signale zur bestimmten Struktur und der Experte muß sich nur noch um die wenigen Signale kümmern, die von MASSIMO nicht vorhergesagt werden. Abbildung 43 zeigt das simulierte und das experimentelle Spektrum von Butylmethylpropylamin.

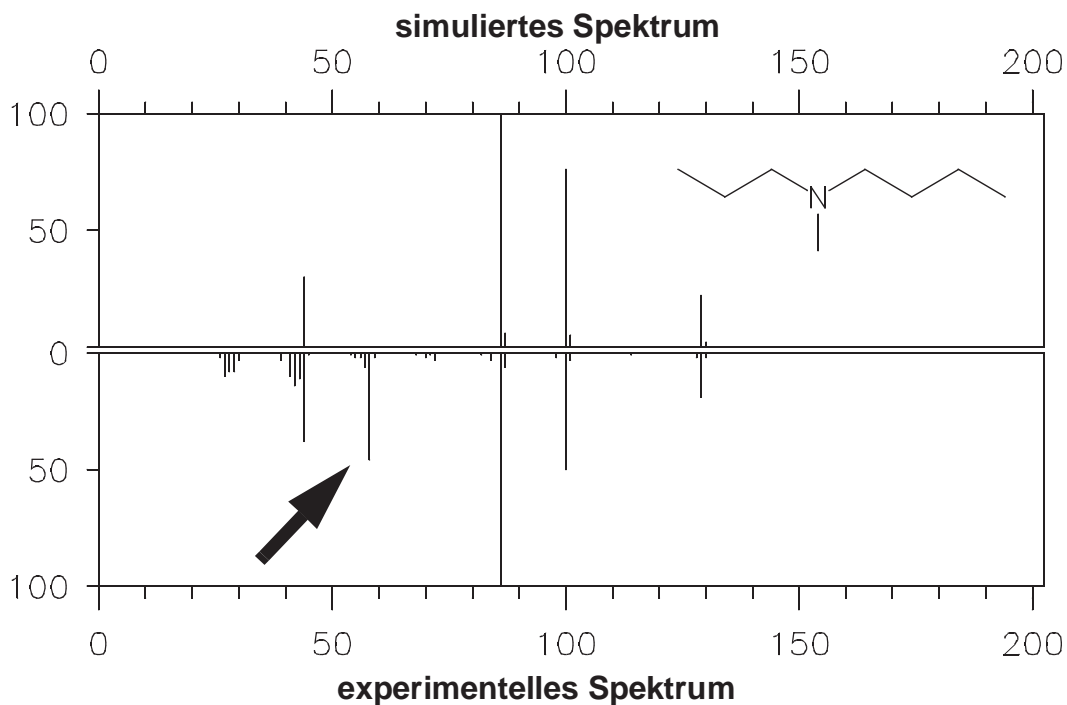


Abbildung 43: Spektrenvergleich von Butylmethylpropylamin.

Vier der fünf intensiven Signale werden von MASSIMO simuliert. Die Aufgabe des Experten hat sich auf die Erklärung des letzten Peaks bei m/z 58 reduziert. Das dazugehörige Fragment wird über die pericyclische Reaktion in Abbildung 44 aus den α -Spaltungsprodukten des Amins gebildet [18].

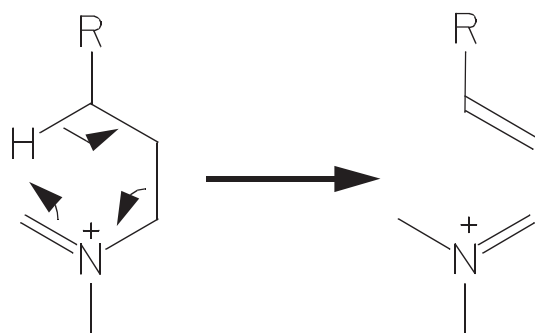


Abbildung 44: Pericyclische Reaktion zur Bildung des Fragments m/z 58.

Sagt MASSIMO ein Signal vorher, das nicht im experimentellen Spektrum zu sehen ist, ist zu überlegen, warum die Reaktion zur Bildung dieses Fragments für die angegebene Struktur nicht stattfindet bzw. ob es sich wirklich um die richtige Struktur handelt. Butylmethylpropylamin ist ein Beispiel, bei dem dies auch mit der beschränkten Einsatzfähigkeit der verwendeten Regeln erfolgreich gezeigt werden kann. Es ist im Trainingssatz zur Ableitung der Wahrscheinlichkeitsfunktionen enthalten gewesen. Für einen breiteren Einsatz müßten allerdings die Regeln für MASSIMO noch verbessert werden.

4.3.3. Erweiterung der Regeln um die Wasserstoffumlagerung

Bisher wurden Umlagerungsreaktionen und damit speziell alle Umlagerungsreaktionstypen wie die Wasserstoffumlagerung vermieden, da sie FRANZ bei der Bestimmung der Reaktionswahrscheinlichkeiten so große Probleme machen (siehe 4.2.2.3), daß mit den so gewonnenen Daten

keine erfolversprechenden Simulationen erwartet werden können. Es soll nun versucht werden, die Wasserstoffumlagerung nachträglich zum Satz der Fragmentierungstypen hinzuzufügen.

4.3.3.1. Erstellung der Regel für die Wasserstoffumlagerung

MASSIMO hat eine besondere Behandlung für Umlagerungsreaktionstypen, wie die Wasserstoffumlagerung. Für solche Reaktionen werden dann die Gleichgewichtskonstanten statt der Geschwindigkeitskonstante angegeben. Bevor MASSIMO von einem Kation die Fragmentierungen ausführt, werden alle Reaktionen von Umlagerungsreaktionstypen erzeugt, wobei dies auch für alle dabei entstehenden Produkte solange fortgesetzt wird, bis keine neuen Umlagerungsreaktionen mehr gefunden werden. Ist das Umlagerungsprodukt schon im Umlagerungssystem enthalten, wird die Reaktion verworfen, um das Umlagerungssystem mit seinen Gleichgewichtskonstanten eindeutig lösbar zu halten. Dahinter steht die Annahme, daß der zuerst gefundene, kürzeste Weg auch den größten Umsatz hat. Für alle weiteren Reaktionen, die ein Umlagerungsprodukt bilden, wird angenommen, daß sie nur einen geringen Beitrag zur Bildung beitragen, und werden vernachlässigt. Für die Bestimmung der Stoffmengen der Umlagerungsprodukte wird angenommen, daß alle Umlagerungen genügend Zeit haben, sich auch einzustellen. Dies ist zwar in der Realität nicht ganz gegeben. Es soll hier untersucht werden, wie gut es mit diesem einfachen Ansatz dennoch möglich ist, die Signale zu simulieren, deren Ionen über Wasserstoffumlagerungen gebildet werden. Bei langkettigen Carbonsäuren und Estern sieht man die Signale der Produkte der α -Spaltungen, die von den Produkten der Wasserstoffumlagerungen ausgehen. Diese wiederum werden aus der am Carbonylsauerstoffatom ionisierten Carbonsäure bzw. Ester gebildet. Abbildung 45 zeigt zunächst das simulierte Spektrum ohne die Wasserstoffumlagerung.

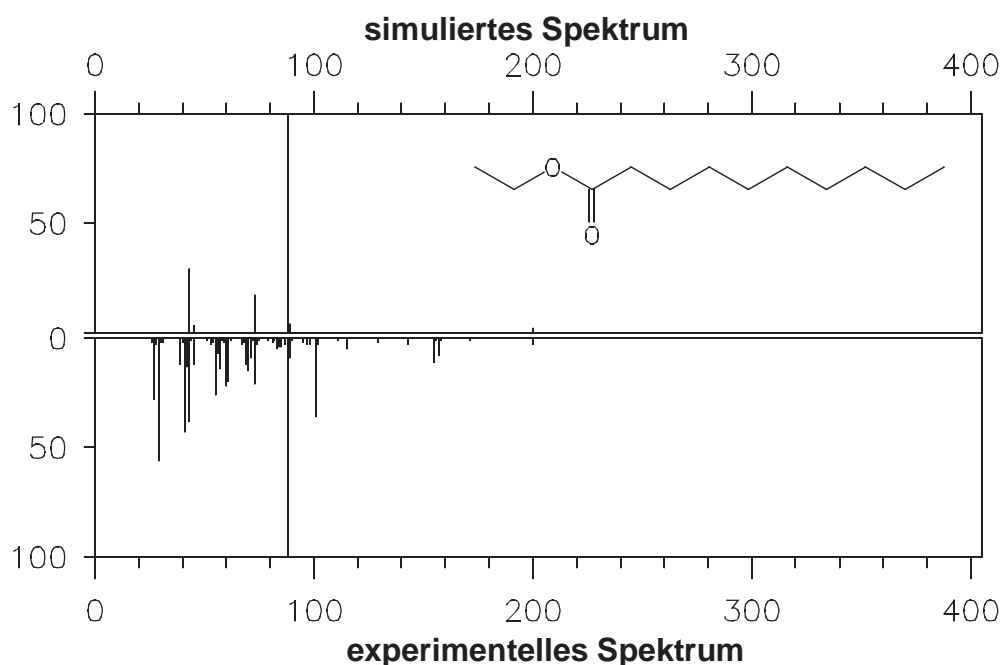


Abbildung 45: Simuliertes Massenspektrum für Decansäureethylester ohne Wasserstoffumlagerung (experimentelles Spektrum [22]).

Die Wasserstoffumlagerungen wurden dem erweiterten Regelsatz aus Kapitel 4.2.3.1 entnommen und auf die Kettenlängen von zwei bis vier beschränkt. Es wurden also Wasserstoffumlagerungen über 5-, 6- und 7-gliedrige, ringförmige Übergangszustände zugelassen. Die Reaktionstypen sind in Abbildung 46 gezeigt.

H-C-Wasserstoffumlagerung:



H-Het-Wasserstoffumlagerung (X = Heteroatom, C= kein π -Zentrum):



Abbildung 46: Wasserstoffumlagerungen.

Desweiteren wurden Wasserstoffumlagerungen auf ein Kohlenstoffradikal ausgeschlossen, wenn dieses mit einer positiven Ladung konjugiert ist, und mehrfache Wasserstoffumlagerungen auf eine funktionelle Gruppe wurden, wie in 4.2.2.4 beschrieben, unterdrückt. Die Gleichgewichtskonstante für Umlagerung auf ein Kohlenstoffatom wurde aufgrund von simulierten Testspektren auf 0.3 und die auf Heteroatome auf 0.03 gesetzt. Ansonsten wurden die Regeln inklusive der Bewertungen aus Kapitel 4.3.1 übernommen. Da das neuronale Netz zur Bewertung der α -Spaltungen nicht für Spaltungen nach Wasserstoffumlagerungen trainiert wurde und es in diesen Fällen für die Abspaltung von Wasserstoffradikalen viel zu hohe Wahrscheinlichkeiten liefert, wurde für diese Reaktionen die Wahrscheinlichkeit auf 6% der vom Netz gelieferten Wahrscheinlichkeit herabgesetzt. Mit den so erstellten Regeln erhält man das in Abbildung 47 gezeigte simulierte Spektrum.

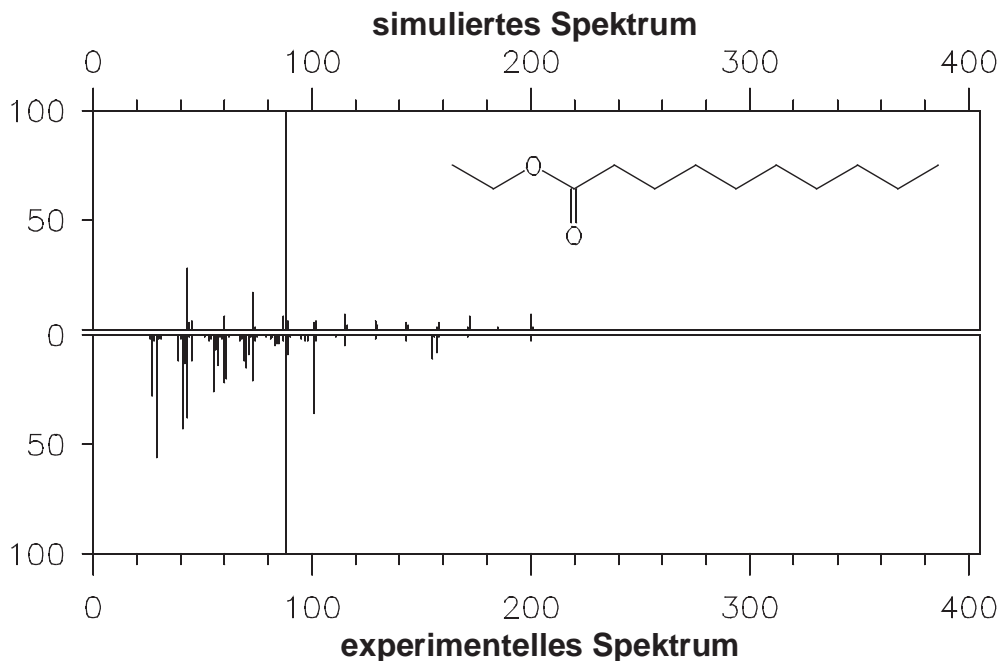


Abbildung 47: Simuliertes Spektrum von Decansäureethylester mit Wasserstoffumlagerung.

Bei näherer Betrachtung des simulierten Spektrums mit Wasserstoffumlagerung kann man vermuten, daß es so möglich sein könnte, die Signale der Produkte der α -Spaltungen nach einer Wasserstoffumlagerung vorherzusagen. An den Intensitäten sind aber noch Verbesserungen nötig und auch die Signale bei einer Masseneinheit höher, die von den α -Spaltungen in die andere Richtung stammen und im Massenspektrometer nicht vorkommen, müßten ausgeschlossen werden. Für eine Ähnlichkeitssuche, für die dieses Experiment durchgeführt wurde, könnte das

simuliertes Spektrum möglicherweise dennoch ausreichen.

Gewichtet man die verschiedenen Wasserstoffumlagerungen so, wie sie in Kapitel 4.2.2.3 für Fettsäuren und deren Ester als geeignet vorgestellt wurden, und schließt die α -Spaltungen in die falsche Richtung aus, kann das simulierte Spektrum verbessert werden.

Dazu wurden die α -Spaltungen ausgeschlossen, bei denen kein Radikal als Neutralfragment abgespalten wird, wodurch die α -Spaltungen in die falsche Richtung verhindert werden. Die Gleichgewichtskonstanten der Wasserstoffumlagerung auf ein Kohlenstoffatom wurden für 5-gliedrige Übergangszustände auf 0.16, für 6-gliedrige auf 0.35 und für 7-gliedrige auf 0.08 gesetzt. Für die Wasserstoffumlagerung auf ein Heteroatom wurde nur eine 1,3-Wasserstoffverschiebung erlaubt, die sicher als zwei 1,2-Verschiebungen entlang des π -Systems der Säure oder des Esters stattfindet. Die Gleichgewichtskonstante wurde abhängig von der Größe des Kations so bestimmt, daß sich für Decansäureethylester (38 Atome) ein Wert von 0.08 ergibt und für Methylstearat (59 Atome) 0.28. Das Spektrum von Decansäureethylester ist in Abbildung 48 gezeigt.

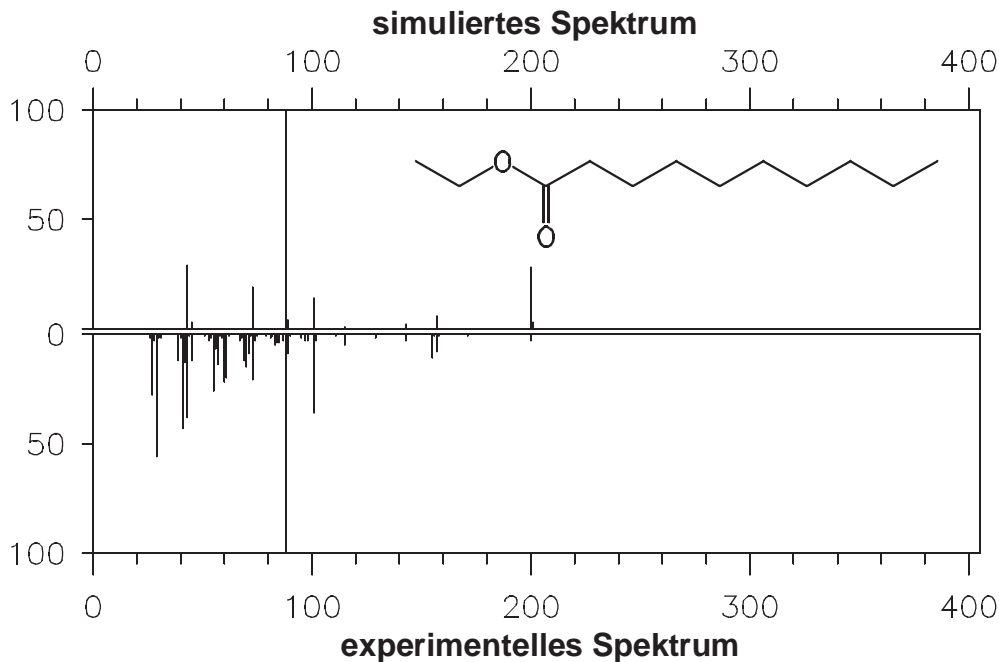


Abbildung 48: Simuliertes Spektrum von Decansäureethylester mit gewichteter Wasserstoffumlagerung.

Angeglichen wurden die Werte der Gleichgewichtskonstanten für die Wasserstoffumlagerung an Methylstearat, dessen Spektrum in Abbildung 49 gezeigt ist und gut mit dem experimentellen übereinstimmt. An Methylstearat wurde in Kapitel 4.2.2.3 die Güte der Gewichtung der verschiedenen, ringförmigen Übergangszustände gezeigt, an die die Gleichgewichtskonstanten angepaßt wurden (siehe Abbildung 10 auf Seite 21).

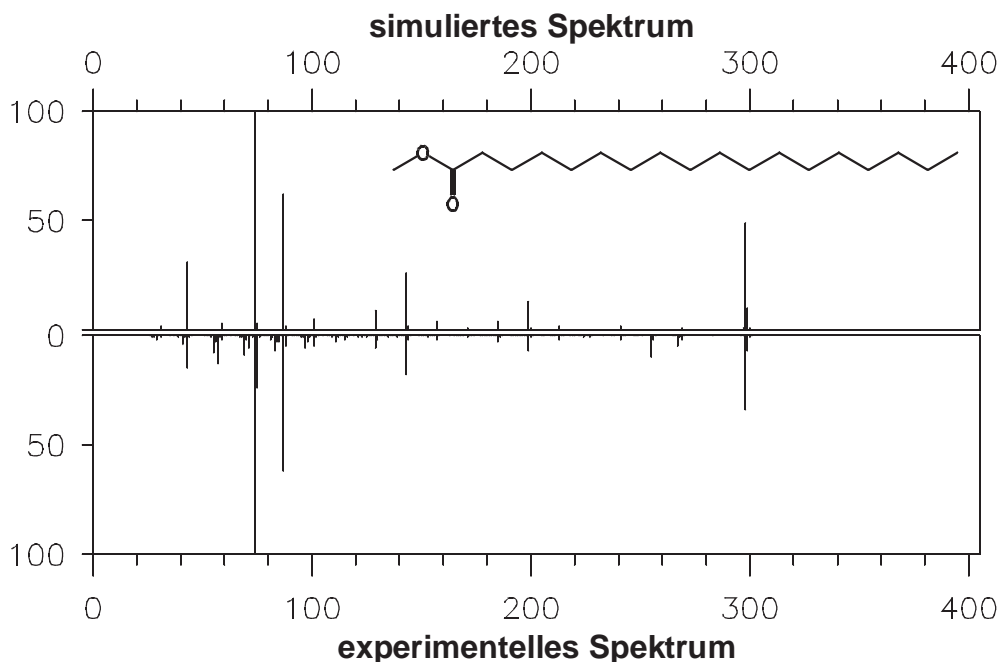


Abbildung 49: Simuliertes Spektrum von Methylstearat mit gewichteter Wasserstoffumlagerung.

Die so simulierten Spektren können sicherlich für die Suche in einer Spektrendatenbank verwendet werden.

4.3.3.2. Andere Quellen für die Wahrscheinlichkeiten der Umlagerungen

Eine Überlegung, an die Geschwindigkeitskonstanten der Wasserstoffumlagerungen zu kommen, war, diese aus semiempirisch berechneten Enthalpien und Entropien zu bestimmen. Für die Aktivierungsenergie der Wasserstoffumlagerung sind Werte in der Literatur [48] zu finden. Die Werte sind allerdings immer nur für das einfachste der möglichen Radikale angegeben. Um überprüfen zu können, ob die Werte auch für die im Massenspektrometer auftretenden Radikalkationen gelten, wurde zunächst getestet, ob die Ergebnisse reproduziert werden können. Für die Wasserstoffumlagerung über einen 6-gliedrigen Übergangszustand für das 2-Heptylradikal wird in [48] ein Wert von 21.99 kcal/mol angegeben. In Abbildung 50 sind die Ergebnisse für die Radikale 2-Heptyl-, 3-Nonyl- und 4-Undecylradikal für diese Umlagerung angegeben. Für das Heptylradikal ergibt sich eine Aktivierungsenergie von 21.13 kcal/mol gegenüber 21.99 kcal/mol aus [48], also keine allzu große Differenz. Für die anderen beiden Radikale ist die Aktivierungsenergie durch die bessere Stabilisierung der längeren Alkylketten um 1.17 kcal/mol niedriger. Alle Werte wurden mit der Methode AM1 [49] in VAMP5 [50] bestimmt.

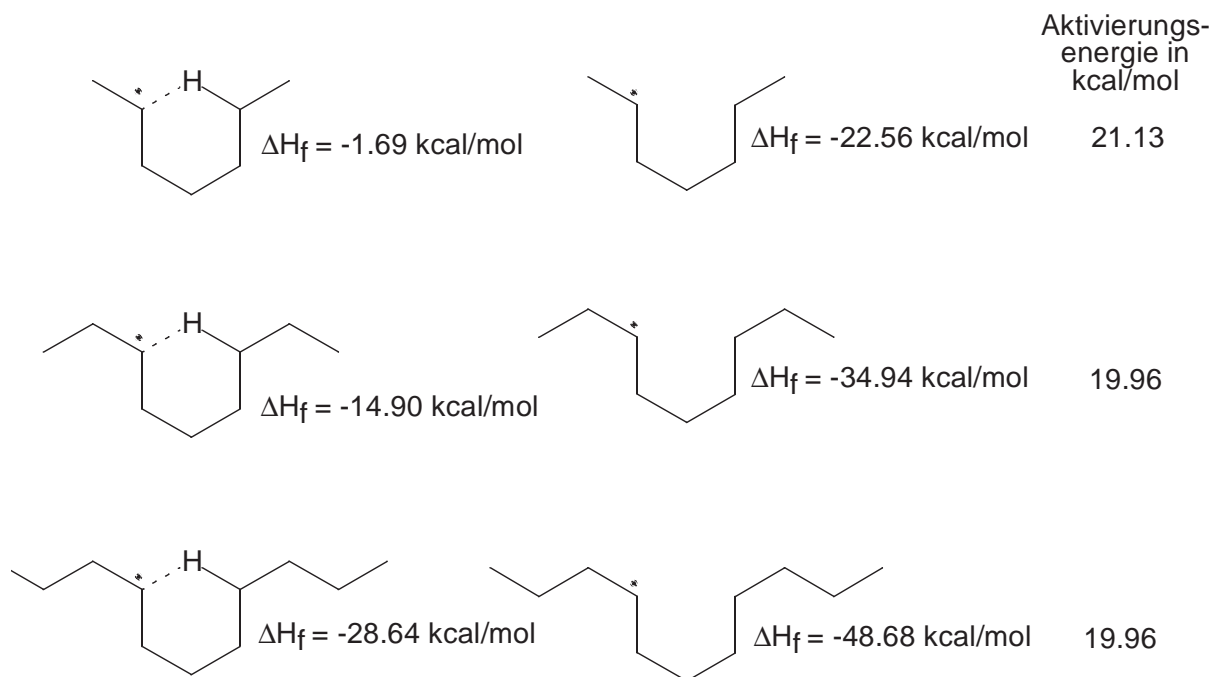


Abbildung 50: Mit AM1 bestimmte Aktivierungsenergien für die 1,5-Wasserstoffumlagerung für das 2-Heptyl-, 3-Nonyl- und 4-Undecylradikal.

Geht man nun zu einem Radikalkation über, das im Massenspektrometer von Tridecansäuremethylester auftritt, über, ist es nicht mehr möglich, den Übergangszustand sauber zu berechnen, da für die optimierte Struktur zwei statt einer negativen Eigenfrequenz berechnet werden. Die zweite Eigenfrequenz ist mit -5.82 cm^{-1} nahe null, läßt sich aber nicht ganz vermeiden. Die Strukturen und Werte sind in Abbildung 51 gezeigt.

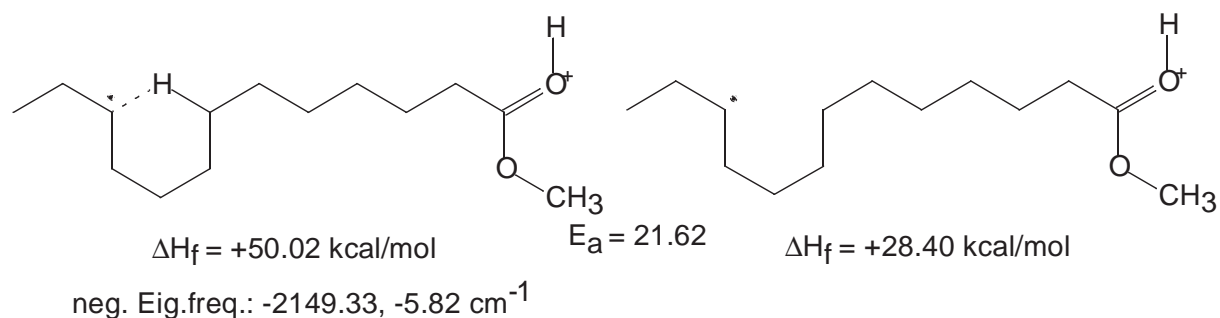
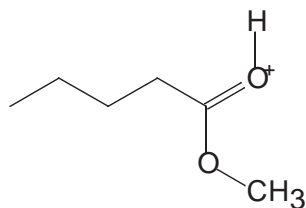
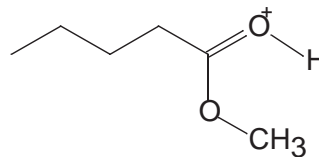


Abbildung 51: AM1-Berechnungen für das am Carbonylsauerstoffatom protonierte 11-Tridecansäuremethylesteradikal.

Die zweite negative Eigenfrequenz liegt an der protonierten Estergruppe, wie die Berechnungen am protonierten Pentansäuremethylester in Abbildung 52 für beide möglichen Orientierungen des Protons zeigen.



$\Delta H_f = +55.13 \text{ kcal/mol}$
neg. Eig.freq.: -5.98 cm^{-1}



$\Delta H_f = +58.61 \text{ kcal/mol}$
neg. Eig.freq.: -8.95 cm^{-1}

Abbildung 52: AM1-Rechnungen des protonierten Pentansäuremethylesters.

Da die Berechnungen für die tatsächlich im Massenspektrometer auftretenden Ionen nicht verifiziert werden konnten, obgleich die ermittelten Werte höchstwahrscheinlich doch zutreffend sind, sowie in der Literatur die Werte für 7- und 8-gliedrige Ringe als Übergangszustände nicht enthalten waren, wurden die Bestrebungen eingestellt, die Wahrscheinlichkeiten für die Wasserstoffumlagerungen semiempirisch zu bestimmen. Eine Berechnung der fehlenden Werte sowie der Entropien wäre wegen der großen Zahl der Freiheitsgrade für diese Radikale sehr zeitaufwendig gewesen und hätte nicht notwendigerweise brauchbare Ergebnisse geliefert.

4.4. Zusammenfassung der Ergebnisse

Es konnte gezeigt werden, daß für Verbindungsklassen, deren Massenspektren vorwiegend von den Fragmentierungstypen α -Spaltung, Oniumreaktion, Carbonyleliminierung, McLafferty-Spaltung und der Wasserstoffumlagerung bestimmt werden, mit dem eingeschlagenen Weg brauchbare Spektren simuliert werden können. Zu diesen Stoffgruppen gehören aliphatische, acyclische Amine, Alkohole, Ether, Acetale, Ketale und Fettsäureester. Effekte, die darauf zurückzuführen sind, daß in einer vorangegangenen Fragmentierung das Kation stark an Anregungsenergie verloren hat, können bei der Spektrensimulation mit MASSIMO nicht berücksichtigt werden.

Durch die unzureichende Kodierung der Radikalkationen als Bindungsliste, mit der einige Ionen nicht unterschieden werden können, und die fehlende Berücksichtigung der Anregungsenergie der Kationen war es für Reaktionstypen, wie die induktive Spaltung, nicht möglich, für diese Bewertungsfunktionen oder neuronale Netze abzuleiten. In Ermangelung der Unterscheidungsmöglichkeit einiger Kationen und alternative Reaktionswege zu gleich schweren Kationen wurden von FRANZ auch nicht stattfindenden Fragmentierungen Umsätze zugewiesen, weshalb einigen bewerteten Reaktionen eine falsche Reaktionswahrscheinlichkeit zugeordnet wurde. Diese Reaktionen stellen Ausreißer im Datensatz der Reaktionen während der Statistik bzw. dem Training der neuronalen Netze dar. Speziell die erwähnte, induktive Spaltung ist allerdings ein wichtiger Fragmentierungstyp für viele verschiedene Stoffklassen.

Bei der Zusammenstellung der Regeln zur automatischen Ableitung der Reaktionswahrscheinlichkeiten aus experimentellen Massenspektren ist nach Möglichkeit auf Umlagerungsreaktionstypen zu verzichten. Diese würden in vielen Fällen zur Folge haben, daß keine eindeutigen Wahrscheinlichkeiten bestimmt werden können. Eine Methode dies zu umgehen ist die Umlagerung und die sich anschließende Fragmentierung zu einem Reaktionstyp zusammenzufassen. Dadurch werden jedoch die Reaktionszentren erheblich komplexer, wobei oft an die Grenzen der Definitionsmöglichkeiten des Reaktionszentrums des Systems zur Reaktionsgenerierung von FRANZ und MASSIMO gestoßen wurde. Es konnten keine verzweigten Reaktionszentren festgelegt werden.

Für eine zuverlässige, breiter einsetzbare Simulation von Massenspektren müssen die Programme zur Analyse und Simulation zusammen mit den Reaktionstypen erheblich überarbeitet und

erweitert werden.

4.5. Grenzen der Strukturrepräsentation und des verwendeten Systems zur Generierung der Reaktionen

Die Strukturrepräsentation von FRANZ und MASSIMO beruht auf der Darstellung als Bindungsliste. Für Radikalkationen ist diese Kodierung oft nicht eindeutig, was auch auf die Generierung aller möglichen, mesomeren Grenzstrukturen und die Reaktionsgenerierung Auswirkungen hat.

4.5.1. Kodierung von Radikalkationen

Eine der einfachsten Strukturen, an denen gezeigt werden kann, daß die Kodierung von Radikalkationen durch die Darstellung als Bindungsliste nicht eindeutig ist, sind ionisierte Enolether (siehe Abbildung 53). Wird eines der beiden Elektronen aus einem der beiden freien Elektronenpaare am Sauerstoffatom des Enolethers entfernt, kann der Bindungsliste nicht entnommen werden, welches der beiden Orbitale, das mit einem oder das mit zwei Elektronen, mit der Doppelbindung konjugiert ist. Ist das Radikal mit der Doppelbindung konjugiert, kann die Bindung zur endständigen Methylgruppe, wie in Abbildung 53 angegeben, durch eine α -Spaltung gebrochen und als Radikal abgespalten werden. Ist das freie Elektronenpaar mit der Doppelbindung konjugiert, kann diese Reaktion nicht stattfinden.

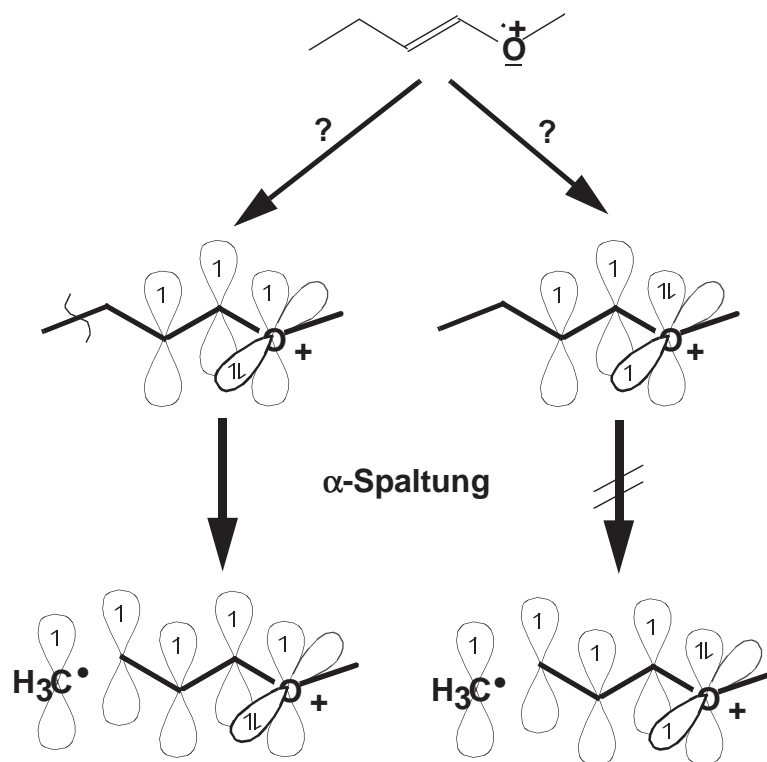


Abbildung 53: Uneindeutige Kodierung eines ionisierten Enolethers.

Der Unterschied zwischen den beiden ionisierten Formen des Enolethers liegt in der Rotation der beiden möglichen, mit der Doppelbindung konjugierten Orbitale um die C–O-Bindung. Desweiteren kann mit der verwendeten Bindungsliste keine Verbindung dargestellt werden, bei der eine σ -Bindung ionisiert wurde. Dies ist die einzige Möglichkeit Alkane, wie Propan (siehe Abbildung 54), zu ionisieren. Die ionisierten σ -Bindungen fragmentieren das Ion anschließend zu den anderen Kationen, deren Signale im Massenspektrum unterhalb der Masse des Molekül-

peaks beobachtet werden können.

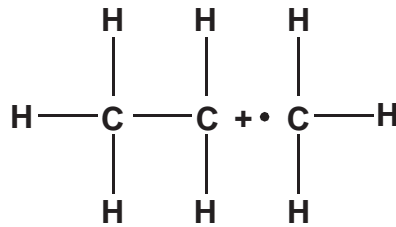


Abbildung 54: Ionisiertes Propan.

4.5.2. Mesomerie

Um alle möglichen Reaktionen zu finden, müssen die Reaktionszentren für die verschiedenen Reaktionstypen in allen unterschiedlichen mesomeren Grenzstrukturen gesucht werden, da für das Reaktionszentrum oft Bedingungen an die Bindungsordnungen gestellt sind. Um diese Grenzstrukturen zu erzeugen, werden sukzessive mehrere Regeln für die lokale Elektronenvertauschung angewendet.

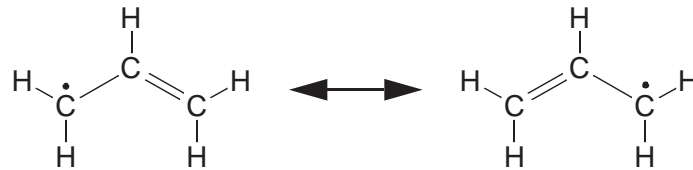


Abbildung 55: Mesomerie des Allylradikals.

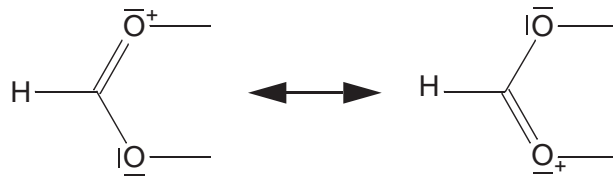


Abbildung 56: Mesomerie von ionisierten Acetalen nach der α -Spaltung.

Zur Überführung einer mesomeren Grenzstruktur in die andere, wie für das Allylradikal (siehe Abbildung 55) und das α -Spaltungsprodukt von Acetalen (siehe Abbildung 56), verwendet der in FRANZ und MASSIMO eingebaute Mesomeriegenerator neben anderen die in Abbildung 57 gezeigten lokalen Elektronenvertauschungen.

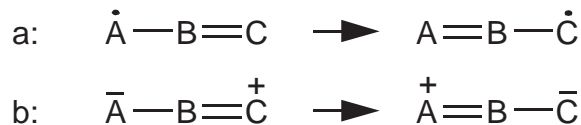


Abbildung 57: Zwei der lokalen Elektronenvertauschungen zur Erzeugung der mesomeren Grenzstrukturen.

Wendet man diese beiden Elektronenumordnungen nacheinander auf einen am freien Elektronenpaar des Carbonylsauerstoffatoms ionisierten Ester (Struktur A in Abbildung 58) an, kommt man zunächst mit der Elektronenvertauschung b zur Struktur B. Da die Kodierung der Struktur B uneindeutig ist (siehe Abbildung 59), kann auch eine Rotation der freien Elektronenpaare am Sauerstoffatom angenommen werden. Damit ist es auch möglich, daß in Struktur B das Radikal mit der Doppelbindung konjugiert ist und damit die Elektronenvertauschung a zur Struktur C

durchgeführt werden kann.

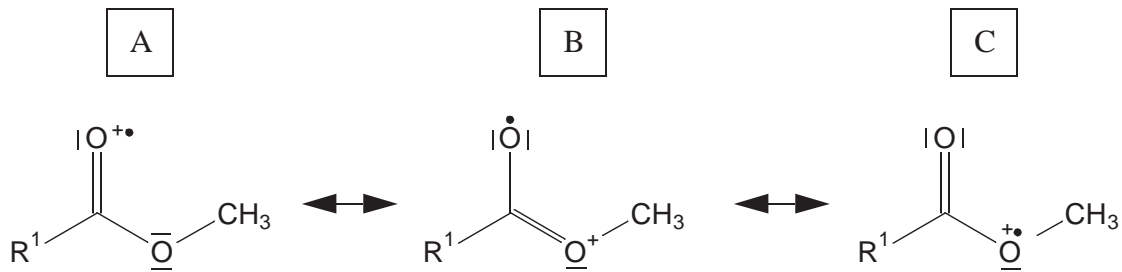


Abbildung 58: Erzeugte „mesomere“ Grenzstrukturen für einen Ester.

Vergleicht man nun die Strukturen A und C, so stellt man fest, daß diese „Grenzstrukturen“ eigentlich zu verschiedenen Strukturen gehören und nicht mesomer zueinander sind. Die Struktur B in Abbildung 58 hat zwei mögliche Interpretationen und das ist das Dilemma für die nicht mesomeren „Grenzstrukturen“. In Abbildung 59 sind die beiden Möglichkeiten als Strukturen B1 und B2 mit den einzelnen teilweise konjugierten Atomorbitalen gezeigt.

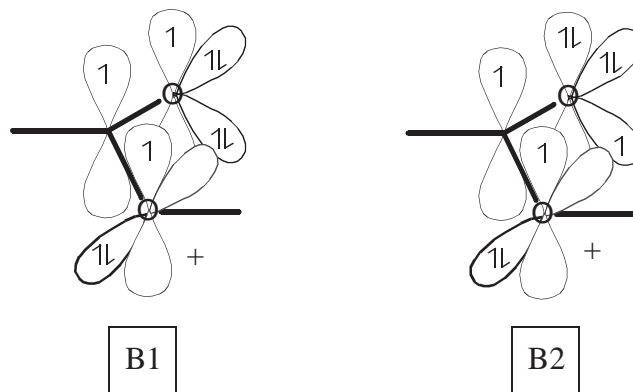


Abbildung 59: Zwei Interpretationsmöglichkeiten für Struktur B.

Durch die uneindeutige Kodierung (siehe 4.5.1) einiger Radikale und Radikalkationen kann nicht verhindert werden, daß vom Mesomeriegenerator in FRANZ und MASSIMO Grenzstrukturen erzeugt werden, die nicht zueinander mesomer sind. Tatsächlich handelt es sich bei den drei Strukturen A bis C in Abbildung 58 um drei verschiedene Formen des ionisierten Esters, von der nur die Struktur A eindeutig einer der ionisierten Formen zugeordnet werden kann (siehe Abbildung 60).

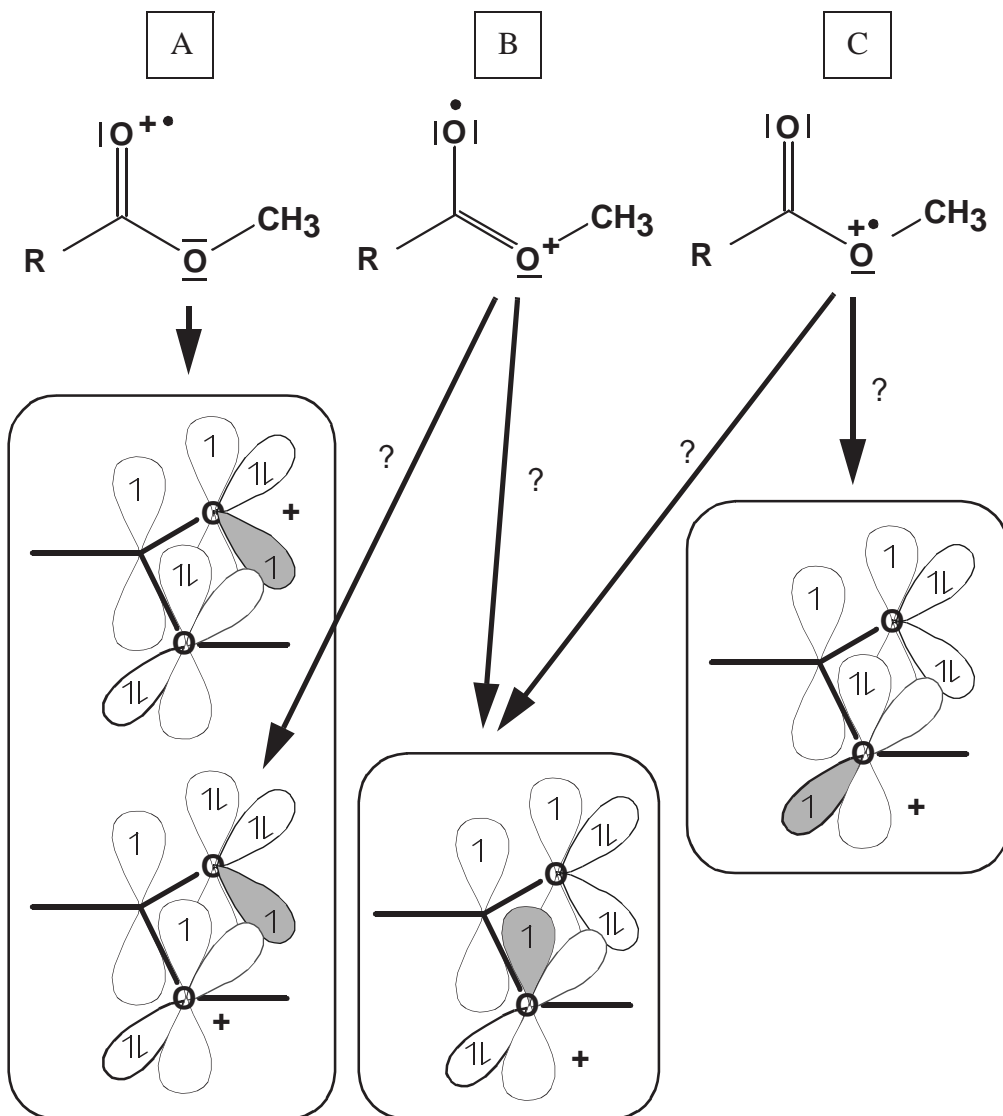


Abbildung 60: Zuordnungen der drei Grenzstrukturen zu den drei ionisierten Formen eines Esters.

Von all diesen „mesomeren“ Strukturen werden selbstverständlich auch Reaktionen durchgeführt, weshalb es bei größeren π -Systemen auch vorkommen kann, daß eine Reaktion von mehr als einer Grenzstruktur durchgeführt werden kann. Für FRANZ und MASSIMO ist es aber wichtig, daß jede der Reaktionen nur einmal gespeichert wird. Dazu werden alle Duplikate einer Reaktion verworfen. Da die Reaktion auch ausgehend von äquivalenten, anderen Atomen als Reaktionszentrum vorgenommen werden kann, wird zur Erkennung dieser Duplikate die Identität des Edukts und des Produkts herangezogen.

4.5.3. Reaktionsbeschreibung

Ein weiteres Defizit von FRANZ und MASSIMO liegt in der Reaktionsgenerierung selbst. Es kommt vor, daß eine Reaktion von einem bestimmten Edukt zu einem Produkt durch mehrere Reaktionstypen erfolgen kann. Nimmt man ein 1-Halogenpropyl-2-radikal und wendet darauf die beiden Reaktionstypen α -Spaltung (siehe Abbildung 61) und Halogenspaltung (siehe Abbildung 62) an, kommt man durch beide Reaktionstypen zu Propen und dem Halogenradikal (siehe Abbildung 63).



Abbildung 61: α -Spaltung.



Abbildung 62: Halogenspaltung.

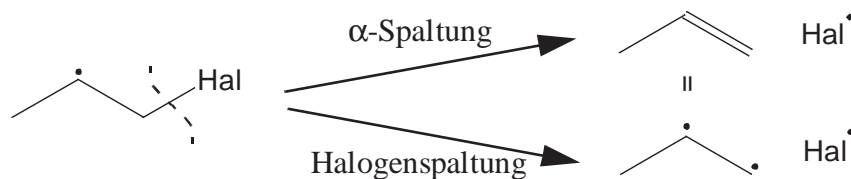


Abbildung 63: Konkurrenz zwischen der α -Spaltung und der Halogenspaltung. Die gestrichelte Linie deutet den Bindungsbruch an.

Unter der Erschwernis der mehrfachen Erzeugung ein und derselben Reaktion aufgrund der Generierung der mesomeren Grenzstrukturen, ist es schwer, ohne eine Priorität der Regeln die richtigen Reaktionen zu verwerfen. Da die Regeln den einzelnen Reaktionstypen keine Prioritäten zuordnen können, behalten FRANZ und MASSMIO die erste gefundene Reaktion für die Reaktion von einem Edukt zu einem Produkt.

Genau dieses Problem tritt auch bei den falschen Grenzstrukturen eines ionisierten Esters auf (siehe Abbildung 65). Hier ist es die Konkurrenz einer induktiven (siehe Abbildung 64) und einer α -Spaltung.



Abbildung 64: Induktive Spaltung.

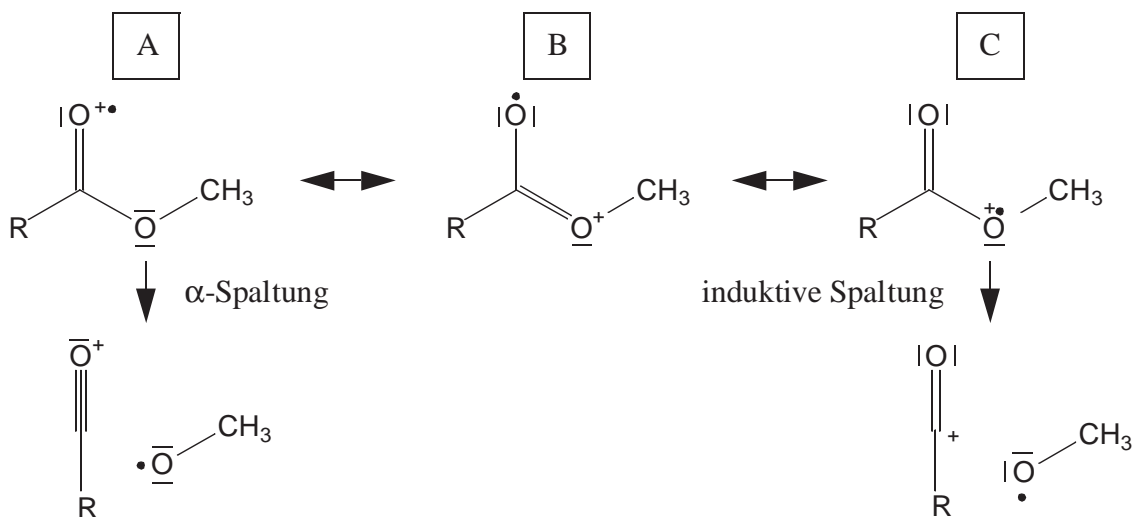


Abbildung 65: Konkurrenzreaktionen zu gleichen Produkten beim ionisierten Ester aus Grenzstrukturen, die nicht mesomer zueinander sind.

Darüber hinaus kann bei ionisierten Estern der Rest R aus zwei verschiedenen „Grenzstrukturen“ (Strukturen A und C), die nicht mesomer zueinander sind, durch eine α -Spaltung als Radikal abgespalten werden. Da diese Strukturen fälschlicherweise als mesomer zueinander

betrachtet werden, die Berechnung der physikochemischen Eigenschaften diese jedoch korrekterweise differenziert, unterscheiden sich diese Reaktionszentren und MASSIMO berechnet für die beiden Strukturen verschiedene Reaktionswahrscheinlichkeiten. Welche Struktur als Edukt der α -Spaltung zusammen mit ihrer Reaktionswahrscheinlichkeit gespeichert und für die Umsatzberechnung verwendet wird, ist von der Reihenfolge der Erzeugung der „Grenzstrukturen“ abhängig, die nicht mesomer zueinander sind. Diese Reihenfolge ist wiederum von der Nummerierung des Edukts abhängig und damit zufällig.

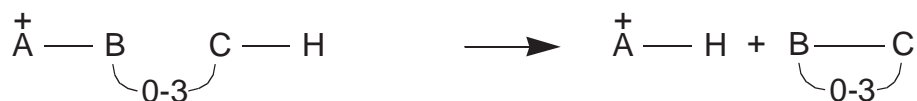


Abbildung 66: Oniumreaktion.

Auch bei der Oniumreaktion (siehe Abbildung 66) tritt sehr häufig der Fall ein, daß das gleiche Kation für verschiedene Kettenlängen gebildet wird. Da die sich in diesem Fall unterscheidenden Neutralfragmente in der Massenspektroskopie nur von geringem Interesse sind, werden sie oft, wie auch bei FRANZ, aus dem Produktensemble entfernt. So erhält man wieder mehrere gleiche Reaktionen, von denen man nur diejenige benötigt, bei der die Kette am kürzesten und damit die Wahrscheinlichkeit am höchsten ist.

Zusammenfassend kann man feststellen, daß für die einzelnen Reaktionstypen und auch die verschiedenen Reaktionen Prioritäten benötigt werden, daß die Reaktion aus den als gleich erkannten Reaktionen mit der höchsten Priorität behalten und weiter verwendet wird.

4.6. Schlußfolgerungen

Trotz der Beschränkungen (siehe 4.4 und 4.5) der auf der Strukturkodierung durch eine Bindungsliste aufbauenden Programme FRANZ und MASSIMO konnte gezeigt werden, daß eine Vorhersage der Massenspektren mit der eingesetzten Technik der automatischen Ableitung von Reaktivitäten aus experimentellen Massenspektren und der dazugehörigen Spektrensimulation möglich sein sollte. Dies konnte an einigen simulierten Spektren belegt werden.

Gründlichen Überlegungen, wie die notwendigen Verbesserungen an den Systemen erfolgen können, und eine Aufwandsabschätzung kamen zum Ergebnis, daß die Änderungen in fast allen Teilen der Programme vorzunehmen sind. Dies lag vorwiegend daran, daß der im Laufe der Jahre mehrfach erweiterte Quellcode in der Programmiersprache Fortran77 geschrieben wurde, die eine Datenabstraktion nicht unterstützt. So waren in den meisten Routinen direkte Zugriffe auf die chemische Datenstruktur enthalten, da dies für eine Bindungsliste einfach und schnell die erforderlichen Daten liefert. Die Erweiterung der chemischen Datenstruktur war aber gerade der Kernpunkt der Weiterentwicklung der Programmsysteme.

Da die Schätzung des Aufwands zur Verbesserung der Programme etwa dem entsprach, der für eine Neukonzeption und Implementation eines neuen, aufgabengerechten Systems veranschlagt wurde, und bei der Weiterentwicklung der alten Systeme Schwierigkeiten mit nicht klar ersichtlichen Abhängigkeiten nicht ausgeschlossen werden konnten, wurde begonnen, ein neues System - EROS7 - zu entwickeln, das die Einschränkungen mit starken Auswirkungen auf die Simulation von Massenspektren nicht mehr besitzt.

5. RICOS, eine MO-orientierte Datenstruktur für Moleküle

RICOS (*Representation of Inorganic, Coordinative and Organic Structures*) [51][52] ist eine Molekülorbital-orientierte chemische Datenstruktur für Moleküle und wurde von Frau S. Bauerschmidt in unserem Arbeitskreis entwickelt. Sie wurde für EROS7, das neue Reaktionsvorhersagesystem, verwendet, um die Beschränkungen der Darstellung als Bindungslisten (siehe 4.5) zu überwinden. Eine MO-orientierte Datenstruktur ist sie deshalb, weil sie konjugierte π -Systeme als solche kodiert und nicht, wie bei Bindungslisten, als alternierende Einfach- und Doppelbindungen.

5.1. Vorteile gegenüber einer Bindungsliste

5.1.1. Kodierung von π -Systemen

Wie schon erwähnt, ist RICOS eine MO-orientierte, chemische Datenstruktur, die die π -Systeme als solche kodiert. Dies soll nun am Beispiel der an einem freien Elektronenpaar des Carbonylsauerstoffatoms ionisierten Acrylsäure gezeigt werden. Struktur A in Abbildung 67 zeigt die Lewis-Struktur, also die Darstellung der Bindungsliste. Der besseren Übersicht halber wurden die an Kohlenstoffatome gebundenen Wasserstoffatome in den Strukturen weggelassen. Betrachtet man nun die Atomorbitale in der richtigen Hybridisierung und Ausrichtung, erhält man Struktur B. Die fünf Atomorbitale, die in der Bindungsliste als die beiden Doppelbindungen und ein freies Elektronenpaar am unteren Sauerstoffatom dargestellt sind, bilden zusammen ein π -System mit insgesamt sechs Elektronen. Dies ist neben den verbleibenden zwei freien Elektronenpaaren und dem Radikalzentrum in Struktur C, der Darstellung nach RICOS, zu sehen. Desweiteren hat die ionisierte Acrylsäure in der RICOS-Darstellung je ein σ -Elektronensystem für jede σ -Bindung zwischen den Kohlenstoffatomen sowie zu den Sauerstoff- und Wasserstoffatomen, die je zwei Elektronen besitzen.

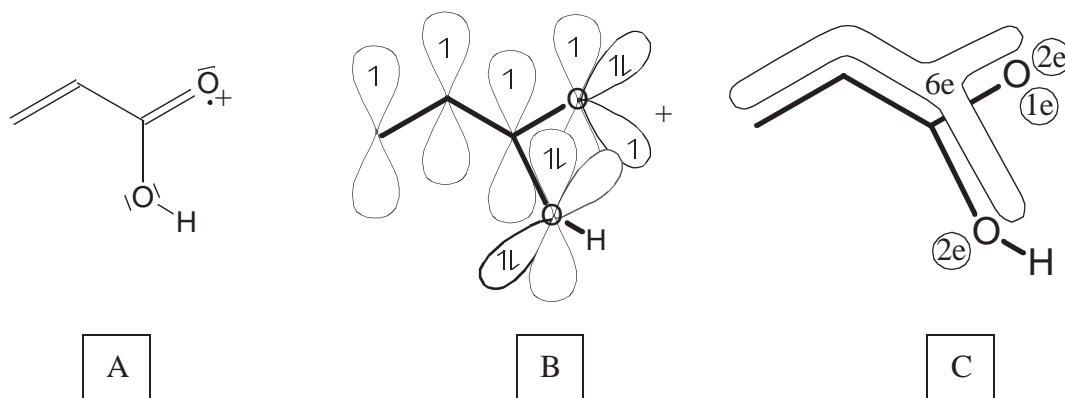


Abbildung 67: Ableitung der Kodierung einer der ionisierten Acrylsäuren. Die an Kohlenstoffatome gebundenen Wasserstoffatome sind der Übersichtlichkeit wegen weggelassen worden.

Sind zwei angrenzende π -Systeme nicht miteinander konjugiert, deutet das auf eine Verdrehung an der Stelle in der Verbindung an, an der die beiden π -Systeme aneinanderstoßen, also nur noch durch eine σ -Bindung voneinander getrennt sind. Um alle fünf Molekülorbitale für das π -System zu erhalten, kann man diese beispielsweise durch eine Hückel-Rechnung [53] bestimmen. Die Ergebnisse sollten dann als Eigenschaft des Elektronensystems gespeichert werden.

Durch die Aufteilung in einzelne Elektronensysteme mit ihren Besetzungszahlen können auch Singulett- und Triplettcarben unterschieden werden (siehe Abbildung 68).

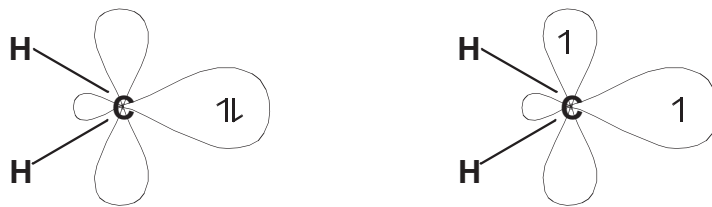


Abbildung 68: Singulett- und Triplettcarben.

Da Strukturteile, über die ein konjugiertes π -System verläuft, planar sind, hat man mit der neuen Struktur auch Teile der Information über die dreidimensionale Struktur. So lassen sich beispielsweise die beiden Formen von Tolan unterscheiden, die sich dadurch auszeichnen, daß die beiden Phenylringe in einer Ebene bzw. senkrecht zueinander sind. Bei der Darstellung nach RICOS bilden dann die beiden aromatischen Systeme ein gemeinsames bzw. zwei verschiedene π -Systeme.

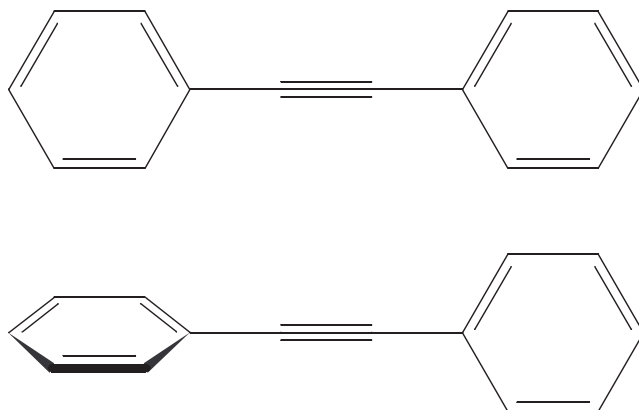


Abbildung 69: Zwei Geometrien von Tolan, die von einer Bindungsliste nicht unterschieden werden können.

5.1.2. σ -Ionisation

Durch die Kodierung von σ -Bindungen als Elektronensysteme mit zwei Zentren und zwei Elektronen können auch σ -ionisierte Verbindungen dargestellt werden, wie sie in der Massenspektroskopie auftreten und bei Alkanen die einzig mögliche Ionisationsart darstellen. Zur Ionisation einer σ -Bindung wird lediglich eines der beiden Elektronen aus dem Elektronensystem entfernt. Abbildung 70 zeigt das an der C–C- σ -Bindung ionisierte Ethan.

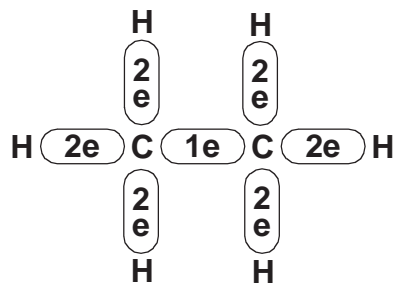


Abbildung 70: Ionisiertes Ethan.

5.1.3. Mehrzentrenbindungen

Aber auch Substanzen mit Mehrzentren- σ -Bindungen, wie Borane (siehe Abbildung 71), und auch manche Übergangszustände, wie der einer S_N2 -Reaktion, können mit RICOS kodiert werden.

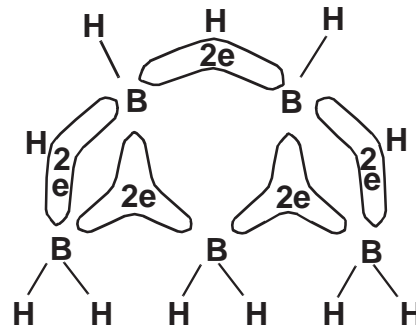


Abbildung 71: B_5H_{11} (Struktur [54]).

5.1.4. Komplexbindungen

Neben den Mehrzentren- σ -Bindungen kennt RICOS auch Komplexbindungen, die aus einem besetzten π -System (freies Elektronenpaar oder größeres π -System) und entsprechend vielen leeren Orbitalen des Metallatoms gebildet werden. So können auch die Verbindungen der gesamten metallorganischen Chemie mit RICOS dargestellt werden. In Abbildung 72 sind der High- und Lowspin-Komplex von 1,1'-Dimethylmanganocen in der Kodierung von RICOS dargestellt.

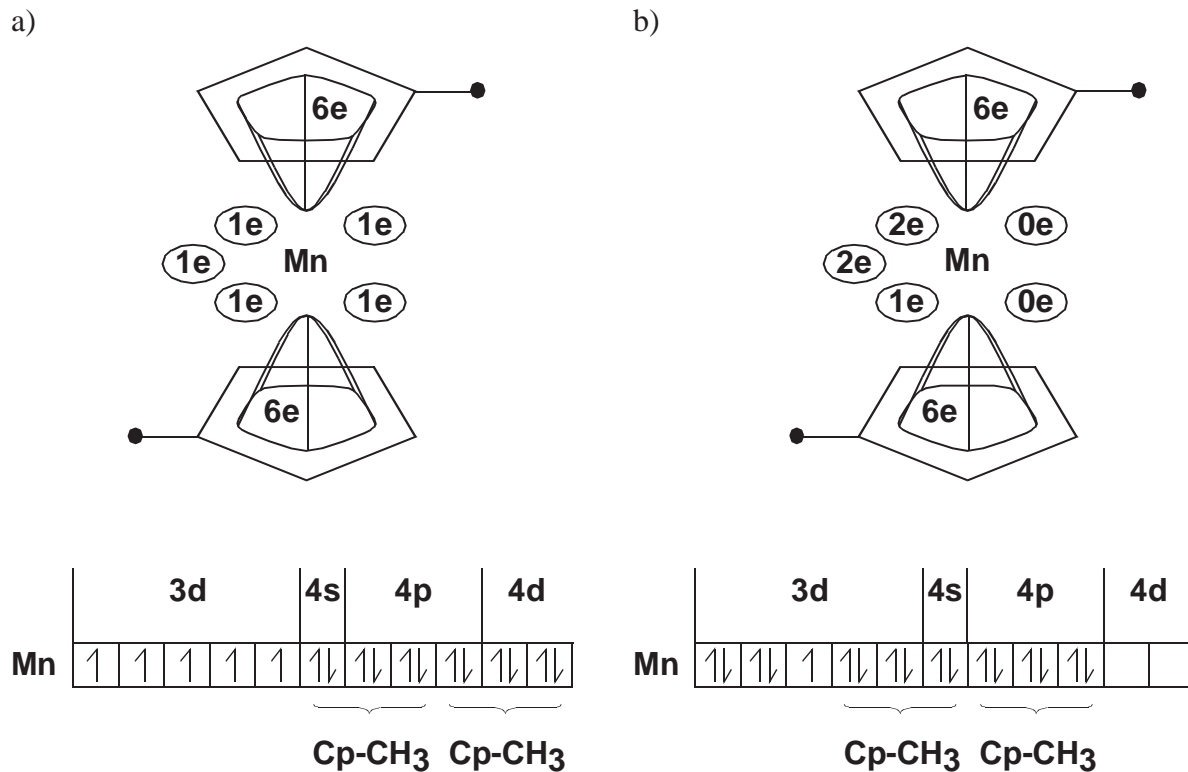


Abbildung 72: Highspin- (a) und Lowspin-Komplex (b) von 1,1'-Dimethylmanganocen. Im unteren Teil der Abbildung ist jeweils die Elektronenkonfiguration des Mangans gegeben. (aus [52])

5.1.5. Trennung von Bindungen und Nachbarschaften

Ein weiterer wichtiger Schritt wurde in Bezug auf die Weiterentwicklung durch die Trennung der Bindungen bzw. Elektronensysteme und der Nachbarschaftsbeziehungen von Atomen vorgenommen. So kann ein System, das Substrukturen sucht, wie ein Reaktionsgenerator unabhängig davon vorgehen, welche verschiedenen Bindungstypen mit der Datenstruktur kodiert werden können. Wird in Zukunft der Datenstruktur ein weiterer Bindungstyp hinzugefügt, brauchen an der Substruktursuche keine Veränderungen vorgenommen werden, da sie die Nachbarschaftsbeziehungen nutzt.

5.1.6. Gruppen und Aggregate

Desweiteren kennt RICOS Gruppen von Atomen und/oder Elektronensystemen. Diese können die Atome einer funktionellen Gruppe zusammenfassen, aber auch die Ionen eines Salzes zusammenhalten. So entsteht aus zwei Molekülen, dem Kation und dem Anion, ein Aggregat (siehe Abbildung 73). Auf die gleiche Weise können mit Gruppen auch intra- und intermolekulare Wasserstoffbrücken kodiert werden. In den meisten Fällen wird in der organischen Chemie allerdings ein Aggregat aus nur einem Molekül bestehen. Ein Reaktionsensemble besteht schließlich aus ein oder mehreren Aggregaten, den Edukten.

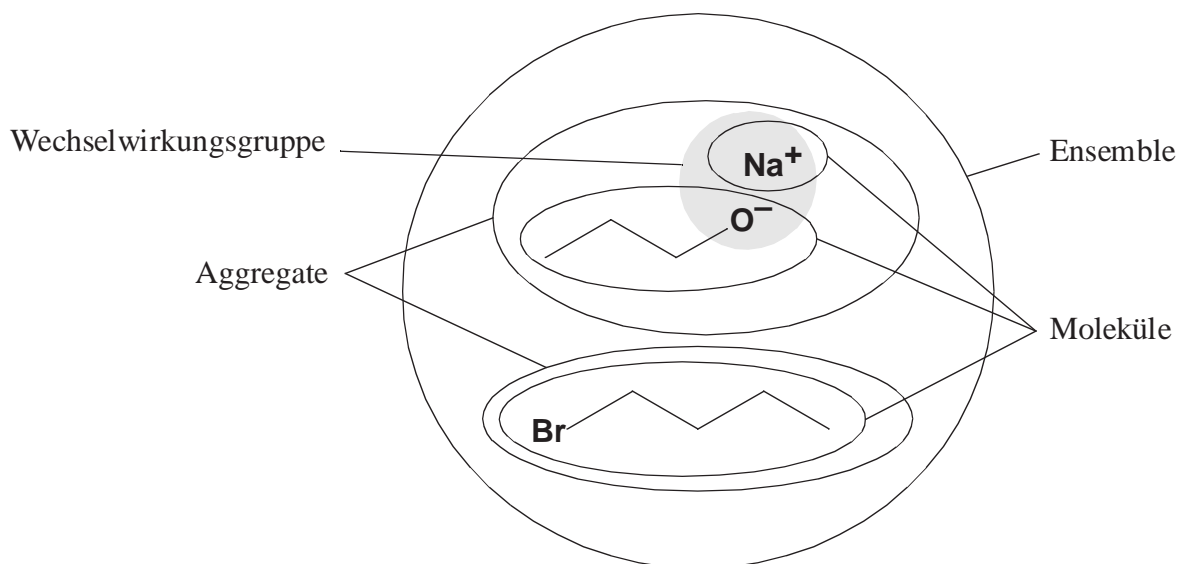


Abbildung 73: Moleküle, Gruppen, Aggregate und das Ensemble.

5.1.7. Genealogische Variablen

Für die Berechnung der physikochemischen Eigenschaften der Moleküle, Atome usw. sind zur Datenstruktur RICOS entsprechende Module zu ihrer Bestimmung dazugebunden. RICOS kennt verschiedene Typen von Eigenschaften. Einen dieser Typen besitzen Eigenschaften, die aus der Darstellung der Moleküle als Bindungslisten abgeleitet werden, die physikochemischen Eigenschaften, die schon in Kapitel 4.2.4 vorgestellt wurden. Im Bedarfsfalle wird die RICOS-Struktur dazu automatisch in die an Informationen ärmere Bindungsliste umgewandelt, die Eigenschaften berechnet und über die Datenstruktur RICOS weitergegeben. Daneben gibt es aber auch schon Eigenschaften, die aus der Struktur nach RICOS bestimmt werden. Eine wichtige Ergänzung der Eigenschaften von RICOS sind aber die Eigenschaften, die nicht aus der Struktur abgeleitet werden können und somit eine zusätzliche Information über die Moleküle darstellen. Diese Eigenschaften sind Variablen, in denen weitere Angaben zur Struktur gespeichert werden. Sie werden während der Vorbehandlung der Ausgangsmaterialien gesetzt und behalten ih-

ren Wert, bis ihnen im Laufe einer Reaktion ein neuer Wert zugewiesen bzw. der Wert modifiziert wird. Solche genealogische Variablen, deren Bezeichnung daherrührt, daß in ihnen Daten über den Entstehungsweg der Moleküle im Reaktionsnetzwerk gespeichert sind, können Angaben wie die Anregungsenergie oder Umlagerungszähler enthalten sein. Will man die Zahl der Umlagerungen begrenzen, kann man in einer solchen Variablen die Anzahl der konsekutiv durchgeführten Umlagerungen zählen und bei Überschreitung der maximalen Zahl die Umlagerung aufgrund dieses Wertes verwerfen. Da diese Variablen Einfluß darauf haben, welche Reaktionen von diesem Aggregat ausgehen können, müssen gleiche Aggregate, die sich in den Werten der genealogischen Variablen unterscheiden, differenziert werden. Dies erfolgt dadurch, daß die genealogischen Variablen bei der Berechnung des Hashcodes [55], aufgrund dessen die Identität von Aggregaten festgestellt wird, mit verwendet werden.

5.2. Vorteile von RICOS bei der Reaktionsgenerierung

5.2.1. Reaktionen von Komplexen und Boranen

Da in RICOS nun auch Borane, σ -ionisierte Verbindungen, Aggregate wie Salze und Metallkomplexe dargestellt werden können, können in EROS7, das RICOS als chemische Datenstruktur verwendet, auch solche Verbindungen als Edukte und Produkte behandelt werden.

5.2.2. Keine Erzeugung von Grenzstrukturen

Da RICOS die konjugierten π -Systeme als ganzes kodiert, müssen nicht mehr alle Grenzstrukturen einer Verbindung erzeugt werden. Damit entfallen der Aufwand der Bestimmung der Grenzstrukturen, die Probleme bei ihrer Generierung (siehe 4.5.2) und die mehrfache Durchführung von ein und derselben Reaktion inklusive der Schwierigkeiten bei der Selektion, welche Reaktion behalten werden soll. Daneben können mit RICOS auch die drei verschiedenen, n - und π -ionisierten Acrylsäuren (Strukturen A bis C in Abbildung 74) unterschieden werden und von ihnen als verschiedene Strukturen auch getrennt Reaktionen erzeugt werden.

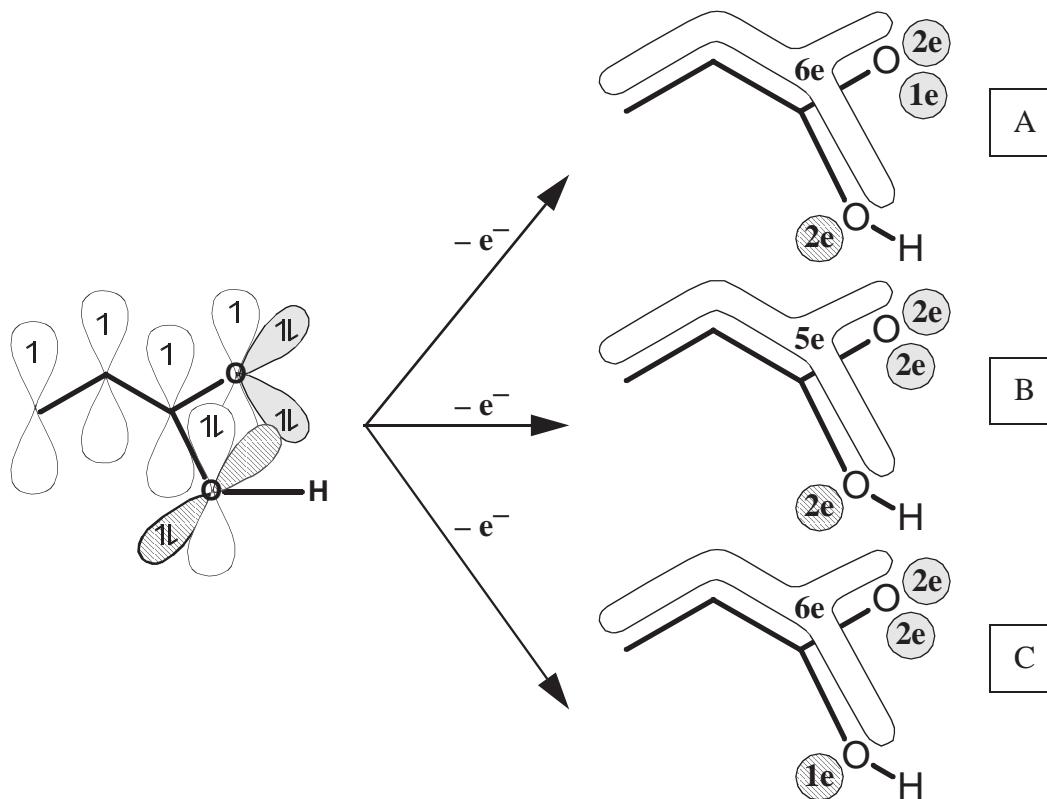


Abbildung 74: Drei Möglichkeiten der Ionisation von Acrylsäure, wenn man die Stereochemie der beiden freien Elektronenpaare am Carbonylsauerstoffatom vernachlässigt.

5.2.3. Steuerung des Reaktionsnetzwerks

Die genealogischen Variablen gestatten es, auf die Abfolge von einzelnen Reaktionstypen in einer Reaktionsfolge Einfluß zu nehmen. Speichert man bei jedem Aggregat, durch welchen Reaktionstyp es gebildet wurde, ist man in der Lage, den Reaktionstyp Y als Folgereaktion zu einer Reaktion des Typs X auszuschließen (siehe Abbildung 75).

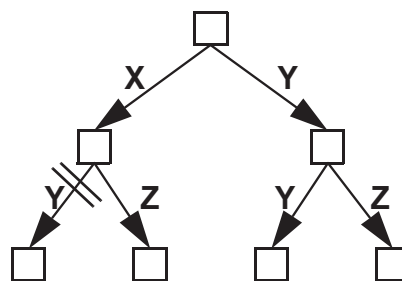


Abbildung 75: Reaktionstyp Y ist nicht als Folgereaktion des Typs X erlaubt.

So können rein formal α -Spaltungen nach einer Radikalzentren erzeugenden Reaktion wie der Halogenspaltung (Reaktionstypen siehe Abbildung 19 auf Seite 27) ausgeschlossen werden. Halogenspaltungen finden im Massenspektrometer erst statt, wenn das Kation bei der Ionisation nur wenig Anregungsenergie bekommen hat oder nach der Abspaltung größerer Fragmente durch die α -Spaltung abgekühlt ist. Ohne diesen Ausschluß würde man eine Vielzahl unnötiger, inaktiver Reaktionen erzeugen.

Aber auch die Anregungsenergie der Aggregate, zu der bei jeder Reaktion die Reaktionsenthal-

pie hinzugezählt wird und die auf die Produkte verteilt wird, kann dazu herangezogen werden, Reaktionen mit einer hohen Aktivierungsbarriere auszuschließen. In der Massenspektroskopie treten solche Reaktionen nur in den ersten Reaktionsebenen auf, da die Kationen mit den abgespaltenen Neutralfragmenten schnell an Anregungsenergie verlieren, die sie bei der Ionisation erhalten haben.

6. Das Programmsystem EROS7

6.1. Konzeption

EROS7 sollte ein Programm werden, mit dem man organische Reaktionen vorhersagen kann. Es sollte dabei so gestaltet werden, daß es in möglichst vielen Bereichen eingesetzt werden kann. So sollte es die Reaktionen simulieren können, die in einem Massenspektrometer ablaufen (siehe die Defizite einer Darstellung der chemischen Struktur in FRANZ und MASSIMO als Bindungsliste in Kapitel 4.5), wie auch Reaktionssimulationen durchführen können, wie es die Vorläuferprogramme EROS5 und EROS6 konnten. EROS5 hatte zwei interne Reaktionsgeneratoren, die entweder zwei Bindungen brechen und zwei Bindungen bilden bzw. derer drei. Zur Einschränkung der Permutationen wurde topologisch festgelegt, welche Bindungen brechbar sind und welche nicht. Die Produkte einer Reaktion wurden ähnlich einem laminaren Strömungsrohr als Ensemble zusammengehalten und nur zwischen diesen Folgereaktionen erzeugt. Um EROS5 auch zur Syntheseplanung verwenden zu können, bestand die Möglichkeit, für jede neue Reaktionsebene kleine Moleküle wie Wasser hinzuzufügen. EROS6 dagegen hatte bereits eine externe Wissensbasis mit den Reaktionstypen, die Regeln, die es auf alle Moleküle und alle möglichen Molekülpaare (inklusive zwei gleichen Molekülen) angewendet hat. Aufgrund der berechneten Geschwindigkeitskonstanten konnte EROS6 auch Konzentrationen für die einzelnen Moleküle angeben. EROS7 brauchte also verschiedene Modi für die Zusammenstellung der Edukte.

Dieses Konzept wurde noch um das Modell der Phasen und Reaktoren erweitert, damit EROS7 auch zur Simulation von Abläufen in technischen Reaktoren sowie in einem modernen Zweig, der kombinatorischen Chemie, eingesetzt werden kann. Es ist dadurch auch möglich, den Abbau von Pharmazeutika in verschiedenen Kompartimenten, wie Blutserum, Gewebe, usw., mit den Stofftransporten zwischen diesen zu simulieren. Die Reaktoren sind eingeführt worden, um auch gekoppelte Simulationen zu ermöglichen, bei denen beispielsweise in einem ersten Reaktor eine Reaktionsvorhersage durchgeführt wird und in einem zweiten für alle entstandenen Produkte die Massenspektren simuliert werden.

6.2. Aufbau

In EROS7, dem neuen Reaktionsvorhersagesystem werden die Moleküle in der MO-orientierte Datenstruktur RICOS (siehe Kapitel 5) dargestellt und mit ihnen die Reaktionen durchgeführt. Gesteuert werden die Reaktionsgenerierung, der Aufbau des Reaktionsnetzwerks und die kinetischen Berechnungen von den Regeln, der Wissensbasis, die die Reaktionstypen mit deren Bewertungsfunktion enthält. Der Aufbau des Reaktionsnetzwerks behandelt die Reaktoren und Phasen und bestimmt somit, welches bzw. welche Eduktaggregate zur Reaktionsgenerierung herangezogen werden. Durchgeführt werden alle Reaktionen, die von einem Eduktensemble ausgehen, vom Reaktionsgenerator, der über die Regelschnittstelle auf die Regeln mit der Definition der Reaktionstypen zugreift. Abhängig von diesen werden auf der chemischen Datenstruktur RICOS die Reaktionen vorgenommen. Auch die physikochemischen Eigenschaften werden über RICOS abgefragt, wodurch die entsprechenden Berechnungsverfahren angestoßen werden. Dieses Zusammenspiel ist in Abbildung 76 gezeigt.

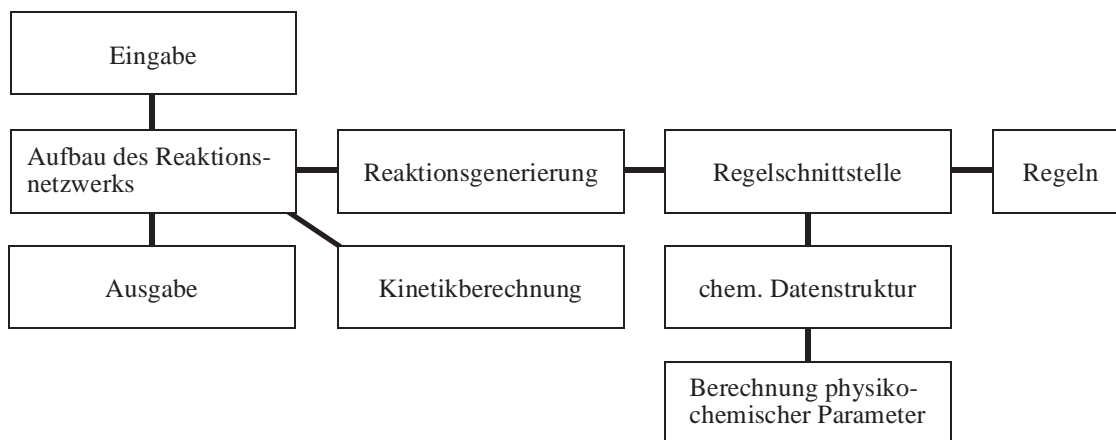


Abbildung 76: Aufbauschema von EROS7.

6.3. Funktionsweise

Der Ablauf einer Simulation mit EROS7 gliedert sich in drei Teile: das Lesen der Regeln, die Ausführung der Reaktionen für die Eduktensembles und den Abschluß der Simulationen. Diese drei Abschnitte und deren Aufgaben werden im weiteren näher erläutert.

6.3.1. Lesen der Regeln

Zu Beginn der Simulation werden die Ausgangsmaterialien eingelesen, bevor die Informationen über die Regeln gelesen werden. Zu den globalen Beschreibungen gehören die Zahl und die Art der Verschaltung der Reaktoren und Phasen, die Art der Kinetik sowie einige untergeordnete Angaben zur Reaktionsgenerierung, wie beispielsweise die Anmeldung genealogischer Variablen und der Austausch der Variablen zwischen den Regeln und EROS7 (siehe 6.7). Neben diesen allgemeinen Angaben werden dann die Namen, Attribute und die Definitionen der Reaktionssubstrukturen aller Reaktionstypen gelesen.

Sind all diese Angaben zu den Regeln gelesen, wird für jedes eingelesene Ausgangsaggregat die Vorbehandlung für Ausgangsmaterialien (siehe 6.4.1) aufgerufen. Bei diesen Vorbehandlungen der Ausgangsverbindungen wird festgelegt, welcher der definierten Phasen sie mit ihrer Anfangskonzentration und ihrem ständigen Zufluß zugeteilt werden. Neben diesen Initialisierungen werden hier auch die Werte der genealogischen Variablen (siehe 5.1.7), die nicht eingelesen wurden, für die Ausgangsmaterialien festgelegt.

6.3.2. Reaktionsvorhersage

Nachdem die Regeln gelesen wurden, werden intern in EROS7 die Reaktoren und Phasen, wie sie in den Regeln festgelegt wurden, aufgebaut. Ein Reaktor besteht dabei aus einer oder mehreren Phasen und seine Berechnungen werden komplett (inklusive der kinetischen Berechnungen) ausgeführt, ehe der nächste Reaktor beginnt, seine Phasen zur Reaktionsgenerierung anzustoßen. Für die Phasen eines Reaktors werden nacheinander je eine Reaktionsebene für jede Phase erzeugt. Dies wird solange fortgesetzt, bis entweder an den Aggregaten keine neuen Reaktionen mehr durchgeführt werden können oder die maximale Zahl der Reaktionsebenen erreicht ist.

Ausgehend von einem Reaktionsensemble wird jeweils eine Sequenz von Regelaufrufen vorgenommen, die in Abbildung 77 gezeigt ist. Den Beginn dieser Folge stellt die Vorbehandlung des Eduktensembles (siehe 6.4.2) dar, die dazu dient, für mehrere Reaktionen an diesem Ensemble oder zur sich anschließenden Suche der Reaktionssubstrukturen (siehe 6.5) für die Reaktions-

typen benötigte Werte zu bestimmen. Bei der Suche wird zunächst jedes nicht identische Atom im Ensemble für alle Reaktionstypen als erstes Atom der Reaktionssubstruktur, also als Atom eins im Reaktionszentrum für diese Reaktion, getestet. Ist dies nicht möglich, wird diese Substruktur für den entsprechenden Reaktionstyp verworfen. Danach werden die zweiten Atome in der Reaktionssubstruktur für alle verbliebenen Substrukturen gesucht und so weiter, bis ein vollständiges Reaktionszentrum gefunden wurde oder kein Reaktionstyp mehr für diese Substruktur möglich ist.

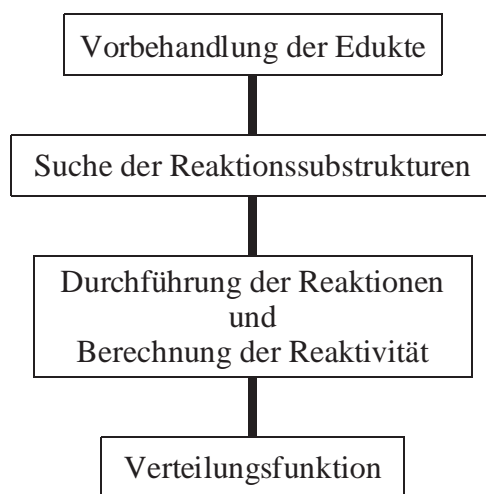


Abbildung 77: Sequenz für die Erzeugung von Reaktionen.

Für alle gefundenen, kompletten Reaktionssubstrukturen auf dem Eduktensemble wird nun die Reaktion ausgeführt, wobei gleichzeitig die Reaktivität der Reaktion aufgrund der physikochemischen Eigenschaften der Atome im Reaktionszentrum berechnet wird (siehe 6.6.1 bis 6.6.3). Sind alle konkurrierenden Reaktionen eines Eduktensembles durchgeführt worden, kann in der Verteilungsfunktion (siehe 6.6.4) noch eine Umverteilung bzw. ein Ausschluß einzelner Reaktionen oder eine Berechnung der Reaktivität unter der Berücksichtigung von Konkurrenzreaktionen vorgenommen werden. Übergibt die Verteilungsfunktion keine Werte für die konkurrierenden Reaktionen, wird eine in EROS7 eingebaute Verteilung ausgeführt. Diese dient dem Ausschluß identischer Reaktionen und Reaktionen, bei denen Edukt und Produkt identisch sind und der gleichen Phase zugeordnet wurden. In verschiedenen Phasen sind Reaktionen mit gleichem Edukt und Produkt zulässig, da es sich hierbei um einen Phasenübergang dieser Substanz handelt.

Die erzeugten Produkte werden von den Phasen gespeichert und in der nächsten Reaktionsebene als Edukte möglicher Folgereaktionen getestet, die anschließend durchgeführt werden.

6.3.3. Abschluß der Simulation

Vor Beendigung des EROS7-Programmlaufs wird noch der Abschlußteil der Regeln aufgerufen (siehe Anhang B.2.11), der nur im Fall der Nutzung von EROS7 als Schnittstelle zur chemischen Datenstruktur RICOS für andere Programme von Interesse ist (siehe 6.7). Schließlich werden optional alle Aggregate einer bestimmten Phase inklusive der kinetischen Daten in eine Datei geschrieben.

6.4. Vorbehandlungen für Ensembles

Neben der reinen Reaktionsgenerierung für die einzelnen Reaktionstypen beinhaltet die Definition der Regeln auch Teile für die Vorbehandlung der Ausgangsmaterialien und Edukte von Reaktionen. Der Funktionsumfang ist der gleiche, wie der der Reaktionsgenerierung (siehe 6.6.1).

6.4.1. Ausgangsmaterialien

In der Vorbehandlung der Ausgangsmaterialien kann beispielsweise die Initialisierung von Umlagerungszählern stattfinden, wie sie für die Einschränkung der Wasserstoffumlagerungen für die Fragmentierungsregeln zur Analyse und Vorhersage von Massenspektren verwendet wurde (siehe 4.2.2.4). Auch die Bestimmung der maximalen Anzahl von Wasserstoffumlagerungen für die verschiedenen Atome im Ausgangsmaterial kann hier durchgeführt werden. Bisher wurde dies durch ein eigenes Programm vor der Generierung des Reaktionsnetzwerks mit MASSIMO durchgeführt. Die Vorbehandlung kann aber auch dazu dienen, Teile von Molekülen zu markieren, die nicht bzw. ausschließlich reaktiv sind.

Hat man bei der Initialisierung die Zeiger auf Vektoren für die Startphase, die Startkonzentration und/oder den Zufluß der Verbindungen an EROS7 übergeben, kann man diese für jedes neue Aggregat als Ausgangsmaterial während der Vorbehandlung der Ausgangsmaterialien festlegen, wenn dies noch nicht geschehen ist. Numeriert man die Ausgangsmaterialien, wie sie der Vorbehandlung zugeführt werden, von null beginnend durch, kann durch das Setzen dieses Elements des Vektors die Startphase, Startkonzentration bzw. der Zufluß definiert werden. Dazu werden der Reihenfolge nach die Aggregat eins, zwei, usw. für alle Aufrufe der Vorbehandlung der Ausgangsmaterialien genommen, wobei mehrfach enthaltene Aggregate nur das erste Mal mitgezählt werden. Häufig werden diese Angaben in Abhängigkeit von bestimmten Strukturelementen der Aggregate festgelegt werden.

6.4.2. Eduktensembles

Die Vorbehandlung der Edukte wird durchgeführt, bevor alle konkurrierenden Reaktionen, die von einem Eduktensemble ausgehen, gesucht und vorgenommen werden. Es können hier Berechnungen angestellt werden, die dann bei der Suche der Reaktionssubstrukturen für die Entscheidungen, welche Reaktionen ablaufen und wie gut, herangezogen werden können. Es ist aber auch möglich, EROS7 mitzuteilen, daß ausgehend von diesem Ensemble keine Reaktionen erzeugt werden sollen. Dies kann dazu verwendet werden, die bei massenspektroskopischen Prozessen abgespaltenen Neutralfragmente von einer weiteren Fragmentierung auszuschließen. Dazu werden alle neutralen Moleküle von der Reaktionsgenerierung ausgenommen, bei denen die Reaktionen nicht in der ersten Reaktionsebene gemacht würden. Ohne diese Ausnahme der ersten Reaktionsebene würden auch die Ionisationsreaktionen ausgeschlossen, was dazu führte, daß überhaupt keine Reaktion erzeugt würde. Es können aber auch Teile von Molekülen mit einer bestimmten Bedeutung, z.B. funktionelle Gruppen, markiert werden, die zur Entscheidung, wo eine Reaktion stattfindet, herangezogen werden können.

Desweiteren kann man, wie in EROS5, kleine Moleküle hinzufügen, die für eine Synthesplanung benötigt werden. Günstiger ist es allerdings diese während der Reaktion selbst hinzuzufügen, da hier selektiv zum Beispiel Wasser für eine Hydrolyse addiert werden kann. Geschieht die Addition von Wasser während der Vorbehandlung des Eduktensembles, wird in jeder Reaktionsebene ein Wassermolekül hinzugefügt, ob es benötigt wird oder nicht.

6.5. Definitionsmöglichkeiten der Reaktionsstruktur

6.5.1. Lineare, verzweigte und ringförmige Reaktionsstrukturen

Eine der Forderungen an den Reaktionsgenerator von EROS7 war, zu gewährleisten, daß auch verzweigte und ringförmige Reaktionssubstrukturen behandelt werden können. Eine Reaktionsstruktur ist die Definition des Reaktionszentrums einer Reaktion. Für eine S_N2 -Reaktion besteht die Reaktionsstruktur aus zwei Teilzentren mit insgesamt drei Atomen, wie in

Abbildung 78 zeigt ist.



Abbildung 78: Reaktionssubstruktur der S_N2 -Reaktion.

Das erste Teilzentrum der S_N2 -Reaktion aus Abbildung 79 besteht nur aus einem Atom, dem nukleophilen Zentrum Atom 1. Dies kann ein angreifendes Chloridion sein, aber auch das Sauerstoffatom eines angreifenden Ethanolatansions. Atom 2 ist das Kohlenstoffatom, an dem die Substitution stattfindet und Atom 3 ist ein abgehendes Jodatom oder auch das Sauerstoffatom einer abgehenden Tosylatgruppe. Alle Atome in den Resten an den Atomen 1, 2 und 3 sind für die Beschreibung der S_N2 -Reaktion unerheblich. Die Beschreibung der S_N2 -Reaktion sieht dann folgendermaßen aus:

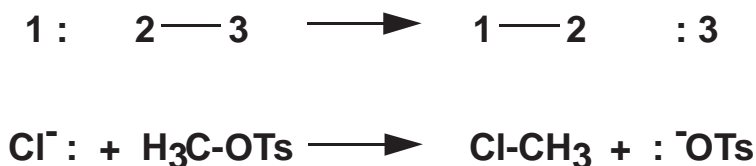


Abbildung 79: Reaktionsbeschreibung der S_N2 -Reaktion mit einem Beispiel.

Die S_N2 -Reaktion ist ein einfaches Beispiel für eine Reaktionssubstruktur. Auch die Teilschritte komplexerer Reaktionen sind in der Regel nicht viel komplizierter, man möchte aber meist das Reaktionsnetzwerk nicht aus Reaktionsteilschritten aufbauen, sondern aus ganzen Reaktionen. Für die Gesamtreaktionen werden die Reaktionssubstrukturen häufig deutlich komplexer, wie am Beispiel der Bildung von substituierten Pyrazolen aus einer 1,3-Dicarbonylverbindung und einem substituierten Hydrazin zu sehen ist [56] (siehe Abbildung 80). Diese Reaktion besitzt zwei verzweigte Teilsubstrukturen in der Reaktionssubstruktur.

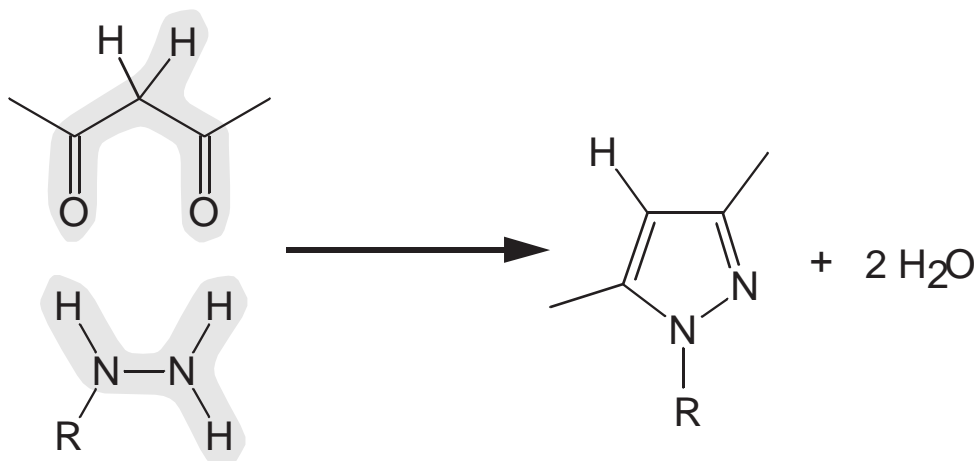


Abbildung 80: Verzweigte Reaktionssubstruktur zur Bildung des Pyrazols. Die Reaktionssubstruktur ist grau unterlegt.

Als eine logische Erweiterung der verzweigten Reaktionssubstrukturen sind auch ringförmige zugelassen.

6.5.2. Mehrere Teilsubstrukturen

Eine weitere Beschränkung bei der Wahl der Reaktionssubstruktur des Vorläuferprogramms EROS6 war, daß maximal drei Teilzentren in der Reaktionssubstruktur enthalten sein durften. Beim neuen Reaktionsgenerator gibt es keine Beschränkung der Zahl der Teilzentren mehr.

Dies ist durch die neue Methode der Kodierung der Reaktionssubstruktur (siehe 6.5.3) möglich. Da oft nur die beiden Atome an den Enden eines π -Systems Teil des Reaktionszentrums sind, wird zur Kodierung der Reaktionstypen auf der chemischen Datenstruktur RICOS teilweise eine größere Zahl von Teilzentren benötigt. Die beiden Endatome sind zwei Teilzentren, die ein gemeinsames π -System verbindet. Die Zahl der dazwischenliegenden Atome ist so ohne Bedeutung. In EROS6 mußte man für alle verschiedenen Abstände der Atome eine eigene Regel schreiben, hatte allerdings nur ein zusammenhängendes Teilzentrum.

Besitzt die Reaktionssubstruktur nur ein Teilzentrum, gibt es keine intermolekularen Reaktionen. Mit zunehmender Zahl an Teilzentren nimmt die Komplexität der Angabe, welches Teilzentrum in welchem Molekül sein muß, stark zu. Deshalb gibt es beim neuen Reaktionsgenerator auch keine Angabe mehr, ob die Reaktion intra- und/oder intermolekular zugelassen ist. Stattdessen werden bei den Bedingungen für die Reaktionssubstruktur die Moleküle überprüft, in denen die einzelnen Teilsubstrukturen gefunden wurden. Der Regelschreiber hat dann alle Möglichkeiten der Verknüpfung und teilt EROS7 nur mit, ob die gefundene Reaktionssubstruktur akzeptiert wird oder nicht.

Da nun gegenüber EROS6 beliebig viele, verzweigte und auch ringförmige Reaktionssubstrukturen (siehe 6.5.1) als Reaktionszentrum definiert werden können, gibt es diesbezüglich keine Beschränkungen mehr.

6.5.3. Suche der Reaktionssubstruktur (RSS)

Die Reaktionssubstruktur, also die Definition des Reaktionszentrums, wird schrittweise in den Strukturen der Edukte gesucht. Ist noch kein Atom in der RSS, wird mit allen Atomen im Ensemble begonnen. Für alle weiteren Atome in der RSS werden alle Atome gesucht, die die in der RSS definierten Nachbarschaften aufweisen. Damit die Definition der RSS unabhängig von der Implementation der chemischen Datenstruktur ist, ist sie über die Nachbarschaft definiert. Ist keine Nachbarschaft für das nächste Atom angegeben, werden wieder alle Atome des Ensembles verwendet.

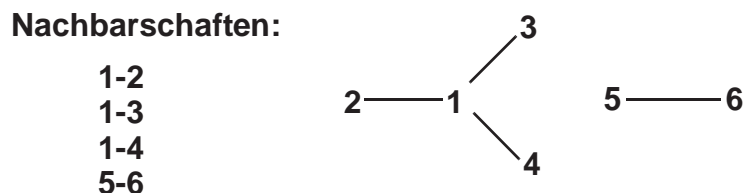


Abbildung 81: Beispiel, wie die RSS über die Nachbarschaften definiert wird.

Als nächstes werden aus der Liste der möglichen Atome diejenigen herausgenommen, die schon in der RSS enthalten sind. Die verbleibenden Atome werden dann nach ihrer Äquivalenz sortiert. Von jeder Gruppe äquivalenter Atome wird dann eines genommen und für die erweiterte RSS werden die Bedingungen überprüft. Da die Atome in der Reihenfolge 1, 2, 3, usw. gesucht werden, sollten die am meisten diskriminierenden Atome nach Möglichkeit die niedrigsten Nummern erhalten. Ist festgelegt, daß das Atom 2 in Abbildung 81 ein Chloratom sein soll, und enthalten die Ausgangsmaterialien nur vereinzelt Chloratome, empfiehlt es sich aus Geschwindigkeitsgründen die Atome der RSS bei der Erstellung der Regeln so durchzunummern, wie es in Abbildung 82 gezeigt ist.

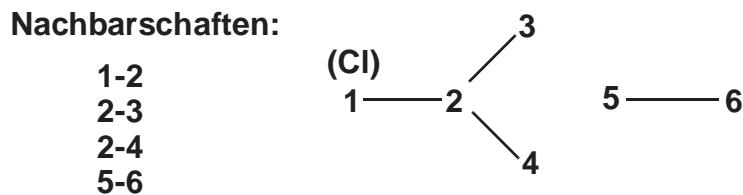


Abbildung 82: Gegenüber Abbildung 81 optimierte RSS.

6.5.4. Symmetriezahl der Reaktion

Gleichzeitig mit dem Wachsen der Zahl der gefundenen Atome in der RSS wird die Symmetriezahl der Reaktion als Produkt aller Größen der Gruppen identischer Atome berechnet.

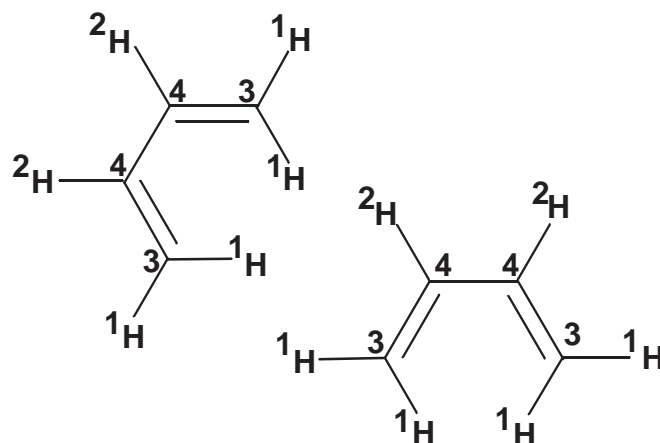


Abbildung 83: Zwei Moleküle Buta-1,3-dien für eine Diels-Alder-Reaktion mit Angabe der Äquivalenzklassen der Atome.

In Abbildung 83 sind zwei Moleküle Buta-1,3-dien mit den Äquivalenzklassen der Atome gezeigt, mit denen eine Diels-Alder-Reaktion durchgeführt werden kann. Die Suche wird an einer zufälligen Stelle begonnen. Sie soll im linken, als Dien verwendeten Molekül mit dem unteren Atom der Äquivalenzklasse 3 begonnen werden. Hiervon gibt es insgesamt vier identische Atome im Ensemble. Nur für die Atome der Äquivalenzklasse 3 werden die Bedingungen für das erste Atom in der RSS erfüllt sein, daß es sich um ein Atom am Ende eines π -Systems handelt. Bei einer MO-orientierten Reaktionsgenerierung benötigt man aus dem Dien nur die beiden Endatome des π -Systems. Es werden also für das zweite Atom in der RSS wieder alle Atome im Ensemble betrachtet. Dieses liegt am anderen Ende des π -Systems im Dien. Auch für dieses sind die Bedingungen nur für Atome der Äquivalenzklasse 3 erfüllt. Sind nun wirklich alle übrigen drei Atome der Äquivalenzklasse 3 identisch? Sie sind identisch, aber nicht gegenüber dem bereits gefundenen Teil der RSS, denn nur eines der drei Atome liegt, wie zu fordern, im gleichen Molekül. Diese Situation ist nicht darauf beschränkt, daß man zwei Moleküle besitzt. Sie kann auch in einem Molekül auftreten, wie Abbildung 84 zeigt.

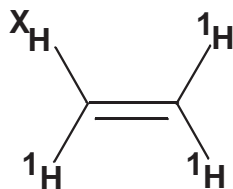


Abbildung 84: Äquivalenzbetrachtung bei Ethen.

Hat man das mit X gekennzeichnete Wasserstoffatom im ersten Teilzentrum, sind die mit 1 gekennzeichneten restlichen, eigentlich äquivalenten Wasserstoffatome für die RSS nicht gleichwertig. Würde man auch noch die Stereochemie berücksichtigen, wären sie sogar alle verschieden. Bei der Betrachtung, ob äquivalente Atome auch gleichwertig für die Suche der RSS sind, müssen also auch alle Beziehungen (Abstände) zwischen dem hinzukommenden Atom und allen bereits in der RSS enthaltenen Atomen verglichen werden.

Abschließend muß noch die Äquivalenz der Nachbarschaften abgetestet werden, was am Beispiel von Biphenylen in Abbildung 85 zu erkennen ist. Alle Atome in den Substrukturen sind äquivalent, auch bezüglich ihres Abstands, und dennoch sind die Reaktionssubstrukturen nicht gleich. Berücksichtigt man auch noch die Identität der Nachbarschaft, in die auch die Elektronensysteme zwischen den beiden Atomen und die Ringe, an denen sie beteiligt sind, eingehen, können auch diese Substrukturen unterschieden werden.

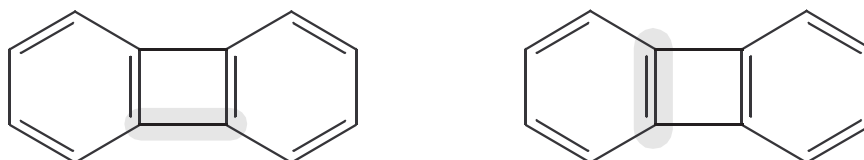


Abbildung 85: Biphenylen mit zwei unterschiedlichen Reaktionssubstrukturen.

6.5.5. Bedingungen

Durch die Entscheidung, die Schnittstelle für die Regeln in der Programmiersprache C++ abzufassen, lag es nahe, für die Überprüfung der Bedingungen eine Funktion zu verwenden. Dies eröffnet bei der Abfassung der Regeln auch die ganzen Möglichkeiten der Sprache, in der die Regeln abgefaßt sind, C++ bzw. Tcl. So ist es selbstverständlich, daß die einzelnen Bedingungen auch logisch mit *und*, *oder* und *nicht* verknüpft werden können. Die RSS wird schrittweise gesucht, wobei bei jedem neu gefundenen Atom auch immer gleich die Bedingungen für dieses Atom der RSS überprüft werden. Hierzu stehen eine ganze Reihe von Eigenschaften der chemischen Strukturen zur Verfügung. Die Überprüfung der Bedingungen wird für jedes neu gefundene Atom aufgerufen, wobei mit null gefundenen Atomen begonnen wird. Hier können Eigenschaften des gesamten Ensembles, der Aggregate und Moleküle getestet werden. Die Bedingungen werden für die zunehmende Zahl an Atomen in der RSS weiter getestet, wobei auch die Eigenschaften der schon zuvor gefundenen Atome für eine logische Verknüpfung abgefragt und zur Entscheidungsfindung herangezogen werden können.

Neben den physikochemischen Eigenschaften, die aus der chemischen Struktur berechnet werden, gibt es auch die Möglichkeit, den topologischen Abstand zweier Atome zu testen. Der topologische Abstand ist die Zahl der Bindungen, die auf dem kürzesten Weg zwischen den beiden Atomen liegen. Auch der Abstand von Teilsubstrukturen kann so festgelegt werden. Neben diesen Eigenschaften gibt es auch genealogische Variablen, deren Wert nicht aus der chemischen Struktur berechnet werden können und nur mit den Aggregaten während der Reaktionen weitergereicht werden. Sie können während der Reaktionen gegebenenfalls angepaßt bzw. gesetzt werden. Genealogische Variablen gibt es für Atome, Elektronensysteme, Gruppen, Moleküle und Aggregate. Da EROS7 intern die einzelnen Aggregate getrennt voneinander speichert, können in EROS7 keine genealogischen Variablen für das gesamte Ensemble eingesetzt werden, was allerdings keine Einschränkung sein sollte, da die Werte auch bei den Aggregaten gespeichert werden können. Die genealogischen Variablen können auch definieren, daß äquivalente Teile eines Moleküls in einer bestimmten Hinsicht doch nicht äquivalent sind, beispielsweise eine Markierung als Rest, der durch die Reaktionen nicht verändert werden darf, oder ein Zähler der Umlagerungen, an dem das Atom schon beteiligt war. Um auch EROS7 mitzuteilen, daß diese genealogische Variable die chemische Struktur bezüglich der Reaktionen

modifiziert, erklärt man bei der Anmeldung der genealogischen Variablen, daß sie für die Berechnung des Hashcodes [55] verwendet werden soll. Der Hashcode wird für die Erkennung der Äquivalenz von Strukturen und Strukturteilen eingesetzt.

Verwendet man eine genealogische Variable für die Aggregate, in der man speichert, welche Reaktion das betrachtete Aggregat erzeugt hat, kann man auch festlegen, daß nach einer Reaktion des Typs X eine Reaktion des Typs Y oder Z folgen muß, dies aber kein anderer Reaktionstyp in den Regeln sein darf. Auf diese Weise kann die Abfolge von Reaktionstypen gesteuert werden.

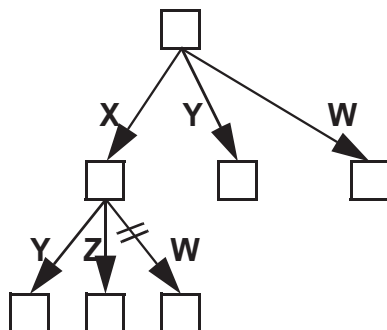


Abbildung 86: Nur die Reaktionstypen Y und Z, aber nicht W, sind nach dem Reaktionstyp X zugelassen.

6.5.6. Optionale Atome

Neben der Reaktionssubstruktur (RSS) selbst ist man oft auch an deren Nachbarschaft interessiert. Sie kann nicht Teil der RSS selbst sein kann, da die Reaktion auch bei Abwesenheit der Atome in der Nachbarschaft der RSS ablaufen soll. Holen wir uns das Beispiel der S_N2 -Reaktion noch einmal ins Gedächtnis und betrachten die Abgangsgruppe (siehe Abbildung 87). Es kann sich um Jodid oder auch Tosylat handeln. Im Fall von Jodid ist das Atom 3 in der RSS nur ein einziges Atom, während bei Tosylat ein größerer Rest am Atom 3 hängt. Da im Fall von Jodid kein weiteres Atom im Edukt an das Jodatombunden ist, die Reaktion aber ablaufen soll, kann kein weiteres Atom in die RSS aufgenommen werden. Ohne Betrachtung der Nachbarschaft der RSS läßt sich jedoch nicht ausschließen, daß auch Hydroxidionen als Abgangsgruppen angesehen werden, was man in der Regel allerdings nicht haben möchte (siehe Abbildung 87), da HO^- eine sehr schlechte Abgangsgruppe ist. Betrachtet man die Nachbaratome des Atoms 3 der Abgangsgruppe, läßt sich jedoch leicht feststellen, daß es sich um eine Hydroxidgruppe handelt, worauf die Reaktion verworfen werden kann.

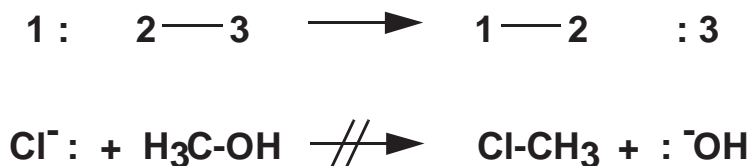


Abbildung 87: Hydroxid wird als Abgangsgruppe der S_N2 -Reaktion ausgeschlossen.

Um dies zu ermöglichen, steht das gesamte Reaktionsensemble zur Abfrage von Eigenschaften zur Verfügung. Für den Zugriff auf die Atome in der RSS fragt man zunächst die Nummer des Atoms im Ensemble für das Atom in der RSS ab. So erhält man beispielsweise, daß das Atom 3 in der RSS den Index 12 besitzt. Das Atom 12 ist also das Atom, an das der Rest der Abgangsgruppe gebunden ist. Um die Eigenschaften des Atoms 3 in der RSS zu erhalten, werden nun die Eigenschaften des Atoms 12 abgefragt. So läßt sich auch die Identität des Wasserstoffatoms (eines optionalen Atoms), das im Fall von Hydroxid dem Sauerstoffatom in der Abgangsgruppe

benachbart ist, oder eine andere Eigenschaften überprüfen.

6.6. Durchführung der Reaktionen

Nachdem die Reaktionssubstruktur gefunden wurde und auch die Bedingungen zutreffen, die an die RSS gestellt werden, kann die Reaktion selbst durchgeführt werden. Ist die Kinetik zur Bestimmung der Konzentrationen angeschaltet, wird gleichzeitig aus den physikochemischen Eigenschaften der Edukte und Produkte sowie unter Umständen von Zwischenprodukten und Übergangszuständen die Geschwindigkeitskonstante bzw. Reaktionswahrscheinlichkeit der Reaktion berechnet. Werden während der Reaktion zusätzliche Moleküle eingeschleust, muß dann noch die von EROS7 berechnete Symmetriezahl der Reaktion korrigiert werden.

6.6.1. Reaktionsgenerierung

6.6.1.1. Erzeugung von Zwischenprodukten und Übergangszuständen

Die Reaktionsgenerierung selbst besteht alternierend aus Teilen, in denen Eigenschaften der Aggregate, Moleküle, Gruppen, Atome und Elektronensysteme abgefragt werden oder das Ensemble modifiziert wird. Zu den Modifikationen gehört neben dem Bruch oder der Bildung von Elektronensystemen auch das Setzen von genealogischen Variablen. EROS7 erkennt automatisch, in welchem Teil sich der einzelne Reaktionstyp gerade befindet. Immer, wenn die Modifikationen abgeschlossen sind und begonnen wird, Eigenschaften vom Ensemble abzufragen, gilt das Ensemble als neues Produkt. Es kann sich dabei auch um ein Zwischenprodukt oder einen Übergangszustand handeln, von denen aus dann weitere Modifikationen vorgenommen werden. Als Reaktion wird nach Abschluß aller Teile der Reaktionsgenerierung die Gesamtreaktion vom Edukt zum letzten erzeugten Produkt gespeichert und ausgegeben. Während der Übergangszustand oder das Zwischenprodukt aktiv ist, können auch deren physikochemische Eigenschaften abgefragt werden und anschließend für die Berechnung der Geschwindigkeitskonstante eingesetzt werden. Läßt sich auch für den Übergangszustand die Bildungswärme berechnen, kann aus der Differenz der Bildungswärmen des Edukts und des Übergangszustands die Aktivierungsenthalpie der Reaktion ermittelt werden. Hat man gleiches auch für die Reaktionsentropie, läßt sich auch die freie Aktivierungsenergie bestimmen. Somit könnte man die Geschwindigkeitskonstante alternativ zur Berechnung aus verschiedenen physikochemischen Eigenschaften auch über die Beziehung

$$k = A \cdot e^{-\frac{E_a}{R \cdot T}} \quad (4)$$

bestimmen. Eine Beschränkung der Zahl der Zwischenprodukte und Übergangszustände gibt es nicht. Zur schnelleren Abarbeitung der Regeln von EROS7 sollten nach Möglichkeit alle Modifikationen in einem Block durchgeführt werden und zwischendurch keine Eigenschaften abgefragt werden, die auch vom Edukt abgefragt und zwischengespeichert werden können, damit möglichst wenig Zwischenprodukte erzeugt werden.

Von den Edukten über die Zwischenprodukte und Übergangszustände zu den Produkten muß gewährleistet sein, daß jedes Atom nach den einzelnen Reaktionsteilschritten auch wiedergefunden werden kann. Deshalb wird gewährleistet, daß der Index für alle Atome, Elektronensysteme und Gruppen während aller Teilreaktionen gleich bleibt. Während der Reaktionen können allerdings Elektronensysteme und Gruppen gebildet und vernichtet werden, was, wie später zu sehen sein wird (siehe 6.6.1.6), auch auf die Atome zutrifft. So kann ein Teil der anfangs durchnumerierten Indizes im Laufe der Reaktionsschritte ungültig werden, weshalb die Möglichkeit besteht zu überprüfen, welche Indizes verfügbar sind. Die Aggregat- und Molekülindizes kann man nach jeder Teilreaktion über die Atome, die sie enthalten, abfragen.

6.6.1.2. Reaktionen auf der Datenstruktur RICOS

Bruch von Elektronensystemen

Die neue Datenstruktur RICOS kodiert die Bindungen als Elektronensysteme der Typen σ und π . Dabei gilt der Teil des Moleküls als eben, über das sich ein π -System erstreckt. Sind benachbarte π -Systeme nicht miteinander konjugiert, zeigt dies eine Verdrehung des Moleküls aus der Ebene an dieser Stelle an. Neben den σ - und π -Elektronensystemen besitzen die Moleküle auch freie Elektronenpaare und unbesetzte Atomorbitale, die ebenfalls als π -Systeme kodiert sind, nur an einem Atom beteiligt sind und 2 bzw. 0 Elektronen besitzen. Die Reaktionsgenerierung besteht nun darin, die richtigen σ - und π -Systeme zu finden und sie zu verbinden bzw. zu spalten. In den Regeln werden die Elektronensysteme angegeben, die ausgehend von den Atomen in der Reaktionssubstruktur gesucht werden. So kann beispielsweise das σ -System, das die Atome 1 und 2 in der Reaktionssubstruktur verbindet, gebrochen werden. Hierzu stehen Eigenschaften, wie „Elektronensysteme an einem bzw. zwischen zwei Atomen“ zur Verfügung. Für den Bruch eines Elektronensystems wird angegeben, zwischen welchen Atomen das Elektronensystem gespalten wird (siehe Atom A und B in Abbildung 88) und wieviele Elektronen dasjenige Elektronensystem erhält, an dem das Atom A beteiligt ist.

* Bruch eines π -Systems

π -Systeme werden zwischen den beiden Atomen gespalten und es entstehen daraus zwei π -Systeme, wobei auch freie Elektronenpaare als π -Systeme kodiert sind (siehe oben). Das Elektronensystem, an dem das Atom A beteiligt ist, erhält die angegebene Anzahl an Elektronen, das andere Elektronensystem die restlichen Elektronen des gespaltenen π -Systems. Wird eine negative Elektronenzahl bzw. keine angegeben oder bleibt eine negative Elektronenzahl für das andere Elektronensystem übrig, wird die Zahl der Elektronen automatisch so verteilt, daß jedes der entstehenden Elektronensysteme möglichst so viele Elektronen erhält, wie es zum nun gespaltenen Elektronensystem beigetragen hat. Für diese automatische Verteilung der Elektronen werden intern Heuristiken (siehe 6.6.1.5) verwendet, deren Einsatz auf neutrale Moleküle oder Moleküle mit Oniumgruppen beschränkt ist, die nur Hauptgruppenelemente ohne Oktetaufweitung und keine Mehrzentren- σ -Systeme enthalten.

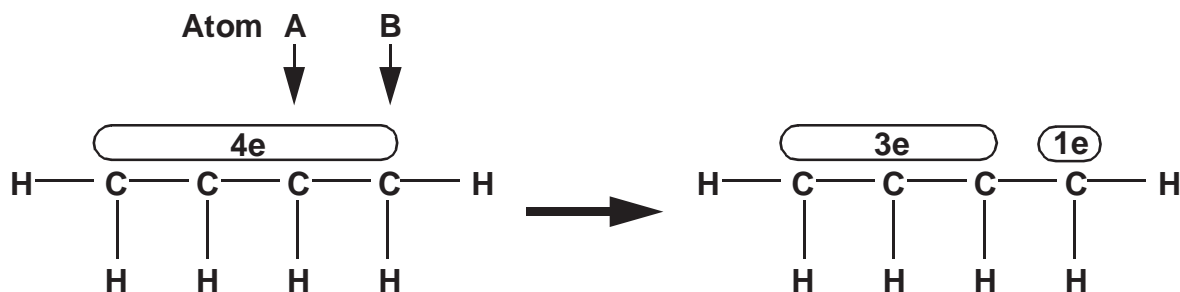


Abbildung 88: Bruch eines π -Systems.

Bei der Spaltung eines cyclischen π -Systems wird das π -System im ersten Schritt nicht gespalten, aber in der Datenstruktur als gebrochen markiert. Wird dieses π -System an einer zweiten Stelle oder alle π - und σ -Systeme zwischen den Atomen gespalten, ist das ringförmige Elektronensystem dann aufgebrochen.

* Bruch eines σ -Systems

Bei der Spaltung eines 2-Zentren- σ -Systems entstehen zwei π -Systeme, an denen je eines der beiden Atome beteiligt ist, also zwei nicht konjugierte p-Orbitale. Auch hier erhält das p-Orbital am Atom A die angegebene Zahl von Elektronen und das andere den Rest. Würde für eines der p-Orbitale eine negative Elektronenzahl resultieren, zum Beispiel dadurch, daß die Elektronenverteilung nicht angegeben wurde, bekommt das p-Orbital am Atom A ein Elektron und das am

Atom B den Rest.



Abbildung 89: Bruch eines σ -Systems, wobei das Chloratom zwei Elektronen bekommt.

Für Mehrzentren- σ -Systeme wird nur das Atom A aus dem σ -System herausgenommen und erhält ein p-Orbital mit der Elektronenzahl, die für diesen Teilschritt des Reaktionstyps in den Regeln festgelegt wurde, bzw. null Elektronen.

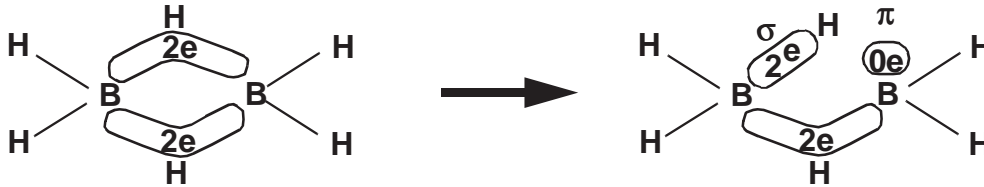


Abbildung 90: Bruch eines Mehrzentren- σ -Systems.

* Bruch einer Komplexbindung

Der Bruch einer Komplexbindung verläuft analog einem Bruch eines Mehrzentren- σ -Systems. Als Atom A wird das Metallatom angegeben, das durch den Bruch ein unbesetztes Atomorbital erhält, das aus der Komplexbindung entfernt wird. So können nach und nach alle mit dem besetzten π -System zu einer Komplexbindung vereinigten Atomorbitale herausgenommen werden, wodurch die Komplexbindung aufgehoben wird. Zurück bleiben schließlich die unbesetzten Atomorbitale am Metall und das π -System des Liganden.

Bildung von Bindungen

In umgekehrter Weise des Bruchs können die verschiedenen Elektronensysteme auch aufgebaut werden. Dazu werden die beiden Elektronensysteme angegeben, die verknüpft werden sollen, und der Typ des Elektronensystems, das entstehen soll. Der Typ kann σ , π oder *komplex* sein. Ist kein Wert oder „default“ angegeben, wird ein σ -System erzeugt, wenn zwischen den Atomen kein σ -System existiert, anderenfalls ein π -System. Die Atome, zwischen denen die Elektronensysteme verknüpft werden, werden intern bestimmt, was für die Bildung von π -Systemen keine zusätzliche Aufgabe ist.

* Bildung einer π -Bindung und Konjugation von π -Systemen

Für die Knüpfung einer π -Bindung oder die Konjugation von π -Systemen werden diese zu einem neuen, größeren vereinigt. Problemfälle, die hierbei auftreten können siehe 6.6.1.3.

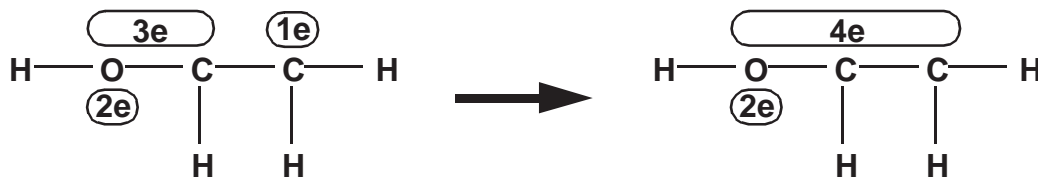


Abbildung 91: Verknüpfung zweier π -Systeme zu einem neuen.

* Bildung von σ -Systemen

Reguläre σ -Systeme werden aus zwei π -Systemen gemacht. Am besten erzeugt man sie aus zwei π -Systemen mit nur je einem Atom (p-Orbitalen). Zwischen diesen beiden Atomen wird dann das σ -System gebildet und erhält die Summe der Elektronen der beiden p-Orbitale. Handelt es sich bei den π -Systemen um Elektronensysteme mit mehr als einem Atom, werden zunächst die π -Systeme so gebrochen, daß an den Atomen ein Radikal (π -System mit einem Atom

und einem Elektron) entsteht. Da die Atome nicht übergeben werden, müssen sie intern bestimmt werden. Im Fall von zwei p-Orbitalen sind die Atome eindeutig, aber bei ausgedehnteren π -Systemen kann nicht immer gewährleistet werden, daß die richtigen Atome gefunden werden, obgleich versucht wird, die Atome mit minimaler Distanz in den beiden Elektronensystemen zu finden. Es empfiehlt sich daher in den Regeln, σ -Systeme aus p-Orbitalen zu bilden und größere π -Systeme vor der Bildung des σ -Systems selbst bis zur Bildung der benötigten p-Orbitale zu brechen.



Abbildung 92: Bildung eines σ -Systems aus zwei π -Systemen.

So wie Mehrzentren- σ -Systeme schrittweise gebrochen werden, werden sie auch schrittweise ausgehend von einem normalen 2-Elektronen-2-Zentren- σ -System aufgebaut, wobei ein p-Orbital nach dem anderen hinzugefügt wird. So entsteht nach und nach ein σ -System mit 3, 4, oder mehr Atomen.

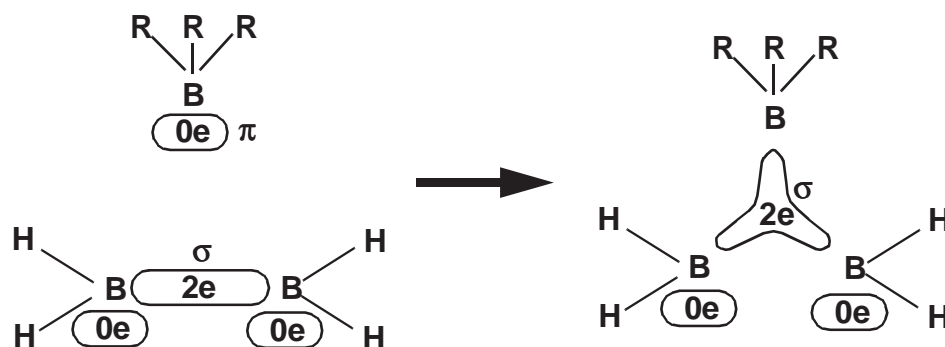


Abbildung 93: Erweiterung eines σ -Systems zum 3-Zentren- σ -System.

Zunächst entstehen alle Mehrzentren- σ -Systeme als σ -Systeme ohne Zentrum, wie die BBB-Bindung in Abbildung 94. Soll ein σ -System, wie die BHB-Bindung, gebildet werden, setzt man, nachdem man die 2-Elektronen-3-Zentren-Bindung erzeugt hat, mit der Eigenschaft EL_CENTER des Elektronensystems das Wasserstoffatom als Zentrum.

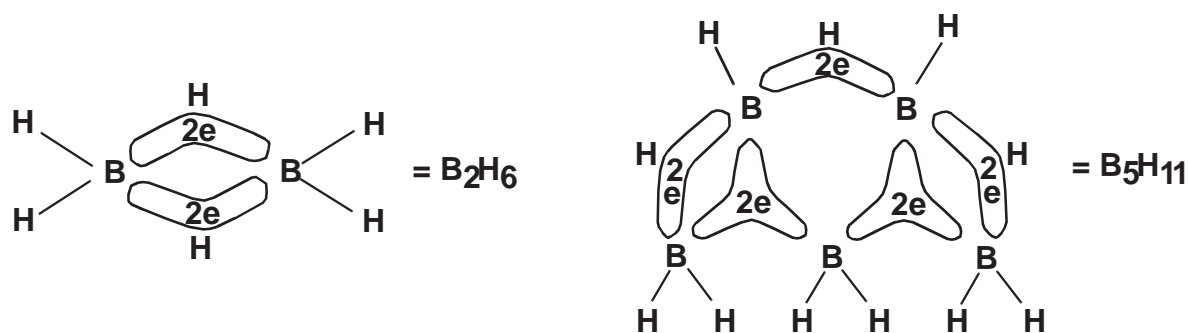


Abbildung 94: 3-Zentren- σ -System mit (BHB in B_2H_6 und B_5H_{11}) und ohne Zentrum (BBB in B_5H_{11} [54]).

* Bildung von Komplexbindungen

Analog der Bildung von Mehrzentren- σ -Bindungen werden auch Komplexbindungen ausgehend von einem π -System und der entsprechenden Zahl leerer Atomorbitale am Metallatom schrittweise aufgebaut.

Benötigte p-Orbitale und Verwaltung der Indizes

Wie schon oben erwähnt, wird teilweise ein einzelnes p-Orbital benötigt. Deshalb muß in einem Teil der Fälle dieses p-Orbital aus einem größeren π -System herausgeschnitten werden. Die Schwierigkeit liegt dabei in der Verteilung der Elektronen auf die entstehenden π -Systeme. Dazu wird der gleiche Verteilungsmechanismus verwendet, der genommen wird, wenn eine negative Elektronenzahl für das Elektronensystem am ersten Atom bei einer Spaltung angegeben wird (siehe 6.6.1.5). Diese arbeitet zuverlässig für neutrale Moleküle und Moleküle mit Oniumgruppen, in denen nur Hauptgruppenelemente enthalten sind, bei denen keine Oktetaufweitung stattgefunden hat. Mehrzentren- σ -Systeme und Komplexbindungen dürfen nicht Bestandteil der Moleküle sein.

Für alle Modifikationen muß intern auch noch mitprotokolliert werden, welche Elektronensysteme gelöscht wurden und welche neu gebildet wurden, damit von der Regel aus über die Indizes auch zugegriffen werden kann. Neue Elektronensysteme erhalten aufsteigend neue Indizes. In seltenen Fällen werden bei der Überarbeitung der chemischen Datenstruktur RICOS auch Gruppen erzeugt oder gelöscht, ohne daß die Verwaltung der Indizes etwas davon mitbekommt. Deshalb wird nach jedem neu gebildeten Produkt das gesamte Ensemble überprüft, ob sich Veränderungen daran ergeben haben.

6.6.1.3. Ausschluß der Bildung ungültiger π -Systeme

Bei der Knüpfung neuer Elektronensysteme muß sichergestellt werden, daß keine verdrillten oder anderweitig ungültigen π -Systeme erzeugt werden. In solchen Fällen werden die π -Systeme vor der Vereinigung automatisch so gespalten, daß dadurch unabhängige π -Systeme entstehen und bei der Verknüpfung keine Probleme mehr vorhanden sind.

Abbildung 95 zeigt den letzten Teilschritt der Keto-Enol-Tautomerie von der Enol- in die Keto-Form. Zuvor wurden die C-C-Doppelbindung gespalten, wobei das eine Radikal mit dem freien Elektronenpaar vom Sauerstoffatom konjugiert bleibt, die O-H-Bindung gespalten und die C-H-Bindung gebildet. Es fehlt die C-O-Bindung zur Bildung der Doppelbindung. Diese wird bevorzugt aus radikalischen, an einem Atom beteiligten Elektronensystemen erzeugt. Für das Sauerstoffatom wird dazu das Radikal und am Kohlenstoffatom das Elektronensystem mit drei Elektronen gefunden. Da zwischen den beiden Atomen schon ein σ -System existiert, werden die beiden π -Systeme miteinander verknüpft. Dies ist auch ganz richtig, aber die Konjugation mit dem freien Elektronenpaar des Sauerstoffatoms muß dabei aufgehoben werden. Daß in jedem Fall beachtet wird, daß möglicherweise Konjugationen aufgehoben werden müssen, demjenigen aufzuerlegen, der eine neue Regeln abfaßt, birgt die Gefahr, daß es vergessen wird. Deshalb wurde der Algorithmus zur Überprüfung und zum Ausschluß ungültiger π -Systeme in EROS7 aufgenommen.

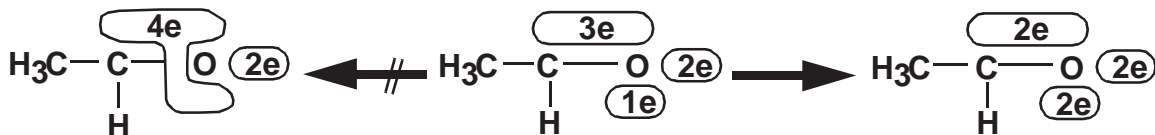


Abbildung 95: Verhinderung der Bildung eines ungültigen π -Systems aus zwei anderen. Dies ist der letzte Teilschritt vom Enol- zu Keto-Form Tautomerie.

Abbildung 96 zeigt ein rein hypothetisches Beispiel, bei dem ein verdrilltes π -System entstehen würde. Es ist eine logische Fortentwicklung der Bildung eines ungültigen π -Systems.

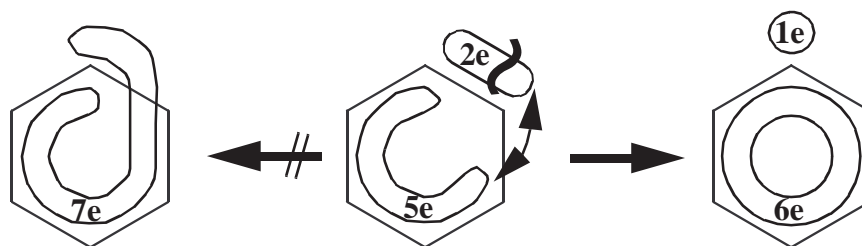


Abbildung 96: Die Entstehung eines verdrillten π -Systems (links) bei der Kombination der beiden π -Systeme wird dadurch ausgeschlossen, daß das kleinere π -System zuerst gespalten wird.

6.6.1.4. Valence-Bond-Emulation (Bindungslisten)

In vielen Fällen möchte man die Reaktionsbeschreibung wesentlich leichter als Änderung der Bindungsordnungen formulieren und nimmt in Kauf, daß nicht alle Moleküle behandelt werden können, da man nur Moleküle mit Hauptgruppenelementen ohne Oktettaufweitung behandeln möchte. Dies ist häufig der Fall, wenn man mit Reaktionsdaten aus Datenbanken arbeitet, da die Moleküle hier eigentlich immer als Bindungslisten (Strukturen nach Lewis) abgespeichert sind. Für diese Fälle gibt es die Valence-Bond-Emulation, die auf der MO-orientierten Datenstruktur RICOS arbeitet, als würde es sich um eine Bindungsliste handeln. Die Emulation kann allerdings nicht allgemein eingesetzt werden, da die automatische Verteilung der Elektronen, die hierfür benötigt wird, ihre Grenzen hat (siehe 6.6.1.5).

Soll eine Bindung zwischen zwei Atomen gebrochen werden, wird zunächst versucht, ein π -System zu spalten. Ist kein π -System zwischen den Atomen, wird das σ -System geteilt. Anschließend werden die Elektronen auf die beiden entstandenen Elektronensysteme so verteilt, daß möglichst jedes Elektronensystem so viele Elektronen erhält, wie seine Atome zum nun gespaltenen Elektronensystem beigesteuert haben (siehe 6.6.1.5). Elektronensysteme werden also homolytisch gespalten. Der Bruch eines Elektronensystems gilt dann als Bindungsbruch, wenn beide Elektronensysteme nach dem Bruch radikalisch sind. Ist dies nicht der Fall, wird angenommen, daß bei der Spaltung des Elektronensystems nur eine Konjugation aufgehoben wurde. Soll eine Bindung gebrochen werden und wird festgestellt, daß nur eine Konjugation aufgehoben wurde, werden weiter Elektronensysteme gespalten, bis ein Bindungsbruch stattgefunden hat. Soll ein polarer Bindungsbruch stattfinden und keine neue Bindungen gebildet werden, können nach dem homolytischen Bindungsbruch die Zahlen der freien Elektronen der beteiligten Atome um eins erhöht bzw. erniedrigt werden.

Existieren mehrere gleichwertige Grenzstrukturen in der Darstellung der Struktur als Bindungsliste, gibt es eine zusätzliche Betrachtungsweise. Nach dieser wird für die Spaltung eines π -Systems zunächst immer diejenige mesomere Grenzstruktur gewählt, bei der möglichst kein Bindungsbruch stattfindet (siehe Abbildung 97), um in einem weiteren Schritt dann die nächste Bindung (in Abbildung 97 die σ -Bindung) zu brechen.

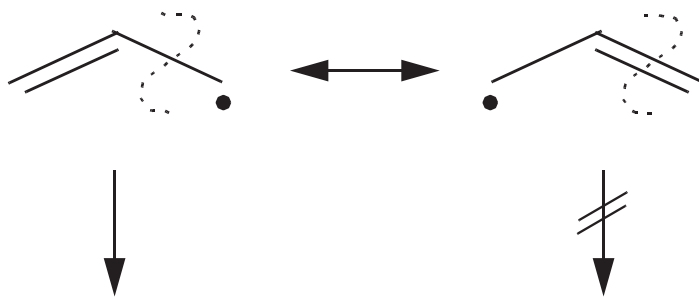


Abbildung 97: Bei der Spaltung eines π -Systems wird die Grenzstruktur gewählt, bei der nach Möglichkeit kein Bindungsbruch eintritt. Die gestrichelte Linie deutet den Ort der Spaltung des π -Systems an.

Wird ein cyclisches π -System aufgebrochen, gilt dies normalerweise als Bindungsbruch, da sonst mit einem einzigen Bruch in einem Aromaten sowohl das π -System als auch die σ -Bindung im Ring gespalten würden. Möchte man dies nicht, kann man dies beim Bindungsbruch angeben.

Für die Bildung einer Bindung werden an den Atomen bevorzugt radikalische Elektronensysteme herangezogen. Das Radikalzentrum kann dabei konjugiert oder nicht konjugiert sein. Ein σ -System wird zwischen zwei Atomen generiert, wenn zwischen diesen Atomen noch keines existiert, wobei zunächst aus den radikalischen π -Systemen falls nötig die p-Orbitale herausgenommen werden und daraus dann das σ -System gebildet wird. Ist zwischen den Atomen schon ein σ -System vorhanden, werden die beiden radikalischen π -Systeme vereinigt, wobei auch die möglichen Fälle der Bildung ungültiger π -Systeme (siehe 6.6.1.3) berücksichtigt werden.

Im Laufe der Reaktionen werden so nach und nach auch die Konjugationen aufgehoben. Um Moleküle nicht doppelt oder dreifach zu erzeugen (mit und ohne Konjugationen), müssen die π -Systeme nach jeder Reaktion wieder konjugiert werden. Der Hashcoder, der für die Erkennung identischer Moleküle und Molekülteile zuständig ist, unterscheidet Moleküle, deren π -Systeme konjugiert sind bzw. nicht. Er wurde für die MO-orientierte Datenstruktur RICOS geschrieben, bei der dieser Unterschied eine Verdrehung im Molekül anzeigt.

6.6.1.5. Automatische Verteilung der Elektronen

Für die Valence-Bond-Emulation (siehe 6.6.1.4) wird eine automatische Verteilung der Elektronen benötigt. Die hier vorgestellte, einfache Methode, aufgrund von topologischen Daten die Elektronen auf die Elektronensysteme zu verteilen, soll für die meisten organischen Verbindungen richtige Zuordnungen treffen, kann aber nicht in allen Fällen die Angabe der Elektronenverteilung ersetzen. Deshalb sollte sie außer bei der Valence-Bond-Emulation, bei der sie nicht möglich ist und daher ihr Geltungsbereich eingeschränkt ist, stets in den Regeln bestimmt und angegeben werden.

Die automatische Verteilung der Elektronen auf die beiden Elektronensysteme bei der Spaltung eines π -Systems geht von einer Tabelle aus, in der die normale Zahl von unkonjugierten Orbitalen - unbesetzt, teilweise besetzt oder besetzt - vermerkt ist. Diese Zahl ist natürlich von der Zahl der Orbitale abhängig, die ein Atom der verschiedenen Elemente in der Darstellung nach RICOS besitzt und ist für die verschiedenen Elemente in der folgenden Tabelle angegeben:

trachteten Atom ist, wird dieser Algorithmus nicht verwendet.

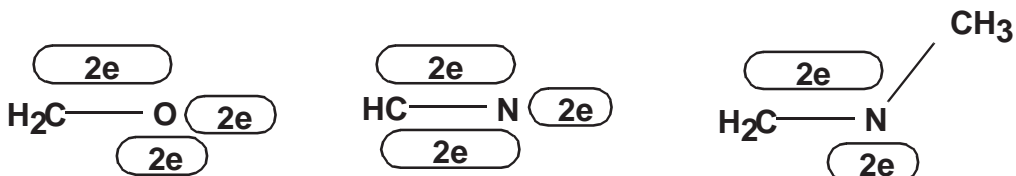


Abbildung 98: Ein Elektron wird in das konjugierte π -System doniert.

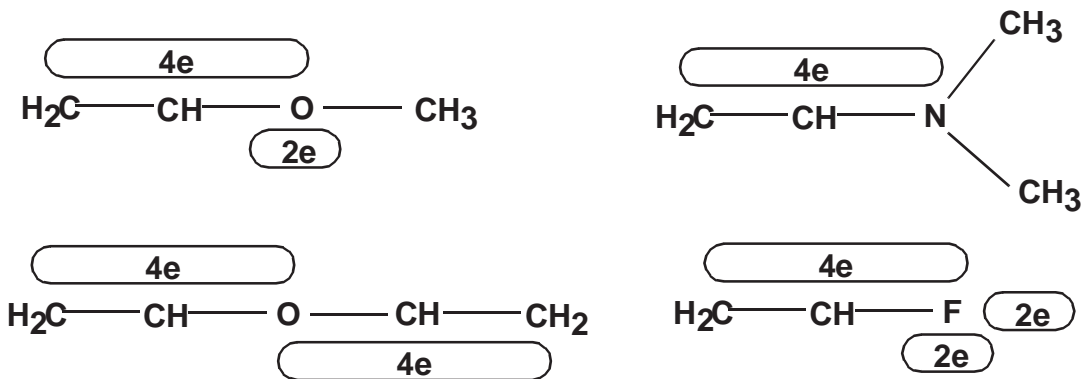


Abbildung 99: Zwei Elektronen werden in das konjugierte π -System doniert.

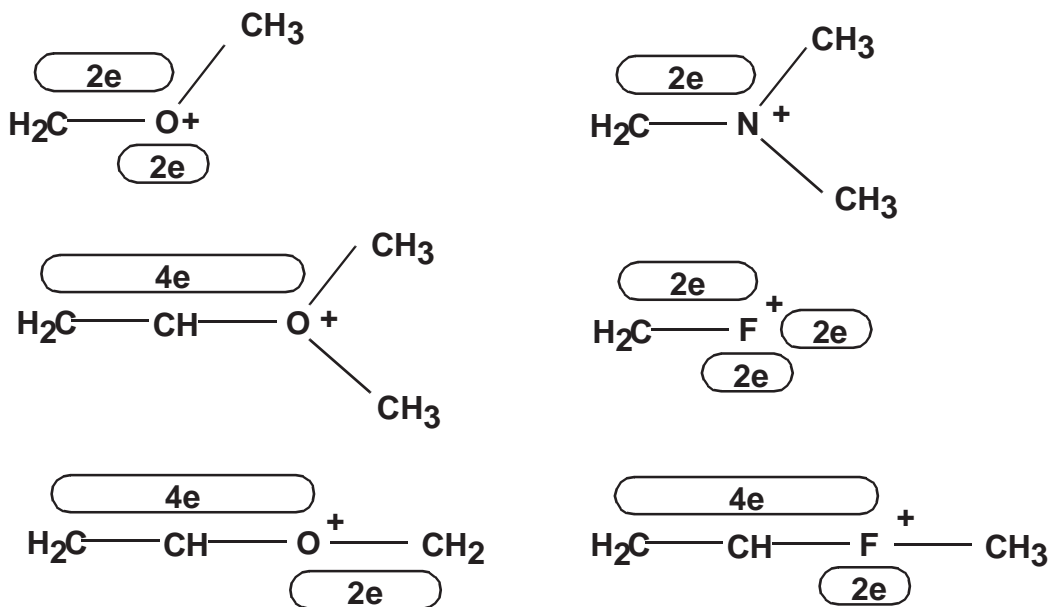


Abbildung 100: Zwei Elektronen werden in das konjugierte π -System doniert (Oniumatome; die Formalladung ist nicht Bestandteil der chemischen Datenstruktur RICOS, sondern wurde nur zur besseren Lesbarkeit angegeben.).

Hatte das gebrochene π -System mehr oder weniger Elektronen als die so bestimmte Summe der normalen Elektronenzahlen der beiden neuen Elektronensysteme, erhält die Differenz - Defizit oder Überschuß - das Elektronensystem, das mehr Elektronen zugeteilt bekommen hat.

Mit dieser Methode erhält man für neutrale Verbindungen aus Hauptgruppenelementen ohne Oktettaufweitung und Oniumverbindungen die richtige Zahl von Elektronen, wenn sie keine Mehrzentren- σ -Systeme und Komplexbindungen enthalten.

Einen Spezialfall stellt noch ein Stickstoffatom mit zwei σ -Systemen und zwei konjugierten π -

Systemen dar. In diesem Fall doniert das Stickstoffatom in das eine π -System zwei und in das andere ein Elektron. Da auch mehrere solcher Atome im Molekül enthalten sein können und sich die π -Systeme auch noch überschneiden können, kann nicht zuverlässig festgelegt werden, in welches π -System wieviele Elektronen doniert werden, zumal es sich in der Praxis auch um Radikale, Kationen und Radikalkationen handeln kann. Deshalb wurde am obigen Schema keine Korrektur für diesen Fall vorgenommen. Damit erhält das betrachtete π -System für dieses Atom zwei Elektronen. Wird dieses π -System dann gespalten, erhält das noch verbleibende π -System ein Elektron.

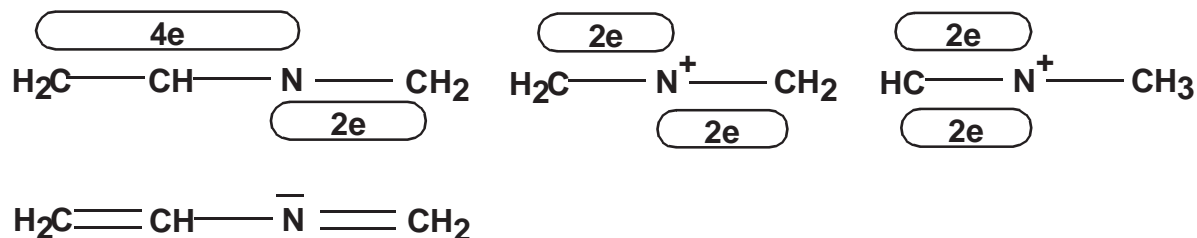


Abbildung 101: Fälle, in denen Stickstoff einmal ein und einmal zwei Elektronen doniert. Im linken Beispiel, das auch als Bindungsliste angegeben ist, doniert das Stickstoffatom in das linke π -System zwei Elektronen (freies Elektronenpaar) und eines in das rechte π -System der Doppelbindung zur Methylengruppe.

6.6.1.6. Weitere Möglichkeiten der Reaktionsgenerierung

Gruppen

Neben der genauen Angabe des Typs der einzelnen Bindungen sind die Gruppen die wohl wichtigste zusätzliche Eigenschaft der neuen Datenstruktur RICOS [51][52]. Mit ihnen lassen sich mehrere Atome und/oder Elektronensysteme zusammenfassen und Eigenschaften zuweisen. Die Gruppen haben einen Namen, der gleichzeitig auch den Typ angibt, und die Zugehörigkeit zu einem bestimmten Aggregat, Molekül, Elektronensystem, Atom oder gar einer weiteren Gruppe. Es gibt, wie bei allen anderen Eigenschaften solche Gruppen, die sich aus der Struktur selbst ableiten lassen (z.B. die Ringe, die zu den Molekülen gehören), und genealogische Variablen, die von den einzelnen Regeln gesetzt werden. Es kann mehrere Gruppen des gleichen Typs geben, wie die Gruppen des Typs Ring, die die verschiedenen Ringe im Ensemble darstellen. Es können aber auch in den Regeln Gruppen gebildet und gelöscht werden. Die so erzeugten Gruppen können beispielsweise verschiedene für die Reaktionen wichtige funktionelle Gruppen repräsentieren, die in einer Reaktion von einer funktionellen Gruppe in eine andere verwandelt werden. Daneben können auch neue Atome und/oder Elektronensysteme in die Gruppe aufgenommen bzw. aus ihr entfernt werden.

Die Gruppen dienen sowohl zum Markieren, wie im Fall der funktionellen Gruppen, als auch zur Bildung von Aggregaten. Wird eine solche Gruppe erzeugt, beschreibt sie eine spezielle Wechselwirkung, wie die Wasserstoffbrückenbindung. Faßt sie Atome aus verschiedenen Molekülen zusammen, bilden die Moleküle ein Aggregat, also ein Teilchen für EROS7. Die Quartärstruktur eines Proteins wäre ein Aggregat, das aus den Proteinsträngen, den Molekülen, besteht. Ob nun eine Gruppe in der Lage ist, ein Aggregat zusammenzuhalten oder gelöscht bzw. geteilt wird, wenn die Atome und/oder Elektronensysteme in verschiedenen Molekülen zu liegen kommen, wird angegeben, wenn diese Gruppe erzeugt wird.

Eine Möglichkeit eine Wasserstoffbrückenbindung zu kodieren wäre, die zwei Atome, das Wasserstoffatom und den Elektronendonator sowie das freie Elektronenpaar des Donoratoms in einer Gruppe zusammenzufassen und sie *h-bridge* zu nennen und diese dem Aggregat zuzuordnen.

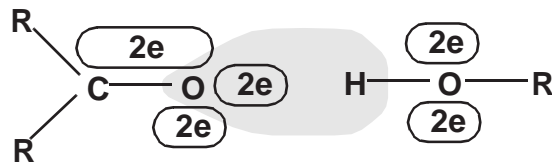


Abbildung 102: Eine Möglichkeit der Kodierung der Wasserstoffbrückenbindung.

Diese Gruppe der Wasserstoffbrückenbindung kann auch noch Eigenschaften besitzen, wie etwa ihre Bindungsstärke.

Setzen von Eigenschaften

Gesetzt werden können neben wenigen, die Struktur beschreibenden Eigenschaften, wie das Zentrum eines Mehrzentren- σ -Systems oder etwaiger Stereodeskriptoren, alle genealogischen Variablen. Wird eine genealogische Variable beim Initialisieren der Regeln mit einem Namen sowie dem Daten- und Chemietyp angemeldet, kann sie für alle Ensembles bei der Überprüfung der Bedingungen gelesen und im Funktionsteil selbst gelesen und gesetzt werden. Als Datentypen stehen ganzzahlige Werte (int), Fließkommazahlen (double), Vektoren aus diesen beiden sowie Zeichenketten und 64-Bit-Werte für Hashcodes zur Verfügung. RICOS kennt noch weitere Datentypen, die aber von der Regelschnittstelle auf diese sechs Datentypen abgebildet werden. Der Chemietyp gibt an, zu welchen Objekten diese genealogische Variable gehört. Das sind Atome, Elektronensysteme, Gruppen, Moleküle und Aggregate. Wird sie gelesen, ohne daß ihr zuvor ein Wert zugewiesen wurde, erhält man einen default-Wert, der in der Regel null ist. Die genealogischen Variablen dienen in ihrer Hauptsache dazu, die Geschichte der Entstehung eines Aggregates zu speichern. In der Massenspektroskopie werden die Moleküle bei der Ionisation gleichzeitig auch angeregt. Im Laufe der Fragmentierungen verliert das Kation meist stark an Anregungsenergie. Mit einer genealogischen Variable, in der man bei jeder Reaktion die Reaktionsenthalpie auf die Anregungsenergie nach der Ionisation addiert und die resultierende Anregungsenergie auf das Kation und das Neutralfragment aufteilt, erhält man für jedes Kation seine Anregungsenergie. Da oft große Neutralfragmente mit vielen Freiheitsgraden abgespalten werden, verliert das Kation in nur wenigen Reaktionsschritten fast seine komplette Anregungsenergie. Reaktionen mit einer hohen Aktivierungsbarriere können so für kalte Ionen ausgeschlossen werden.

Einführen von Atomen und Löschen von Aggregaten

In Fällen, wie dem Abbau von Umweltchemikalien, ist man nicht am Konzentrationsverlauf der im Überschuß vorhandenen Verbindungen wie Wasserstoff als formales Reduktionsmittel und Wasser interessiert. Man beschreibt die Kinetik als Kinetik pseudoerster Ordnung, bei der die Geschwindigkeitskonstante das Produkt aus der Geschwindigkeitskonstante zweiter Ordnung und der Konzentration des Edukts im Überschuß ist. Für jeden Reaktionstyp steht dann fest, welches zweite Edukt benötigt wird. Es müssen also nicht alle anderen möglichen Edukte auf ihre Verwendbarkeit hin überprüft werden. Für diese Art der Reaktionsgenerierung dient dann nur die Umweltchemikalie bzw. einer ihrer Metaboliten als Edukt, das zweite Edukt wird während der Reaktion eingeschleust.

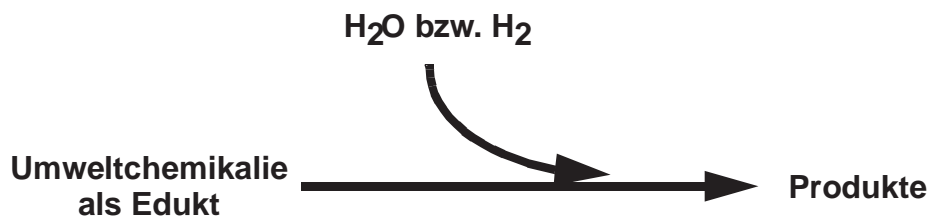


Abbildung 103: Einschleusen von Edukten während einer Reaktion.

Für eine Hydrolyse erzeugt man dazu zwei Wasserstoff- und ein Sauerstoffatom und bildet eine Bindung zwischen einem der Wasserstoffatome und dem Sauerstoffatom. Auf die zweite Bindung kann verzichtet werden, da diese Bindung für die Hydrolyse ohnehin gleich wieder gespalten würde und dieses Wasser nicht als Edukt ausgegeben wird. In der Umweltchemikalie wird anschließend eine Bindung gebrochen und an die freien Valenzen das Wasserstoffatom und die Hydroxylgruppe gebunden. Alternativ kann das zweite Edukt auch aus einer Datei eingelesen werden (siehe unten). Die ausgegebene Reaktion ist in diesen Fällen allerdings nicht stöchiometrisch ausgeglichen, da das Wasser nicht als Edukt in Erscheinung tritt.

Der gleichen Gedankenansatz kann dazu führen, daß kein Interesse an einem der Produkte besteht, nachdem eine Reaktion vollständig durchgeführt wurde. Denkbar wären hier das abgespaltene Neutralfragment bei massenspektroskopischen Prozessen, die nicht detektiert werden können, wodurch auch ihre entstandene Menge uninteressant ist. Deshalb ist es möglich, während der Reaktion auch ganze Aggregate aus dem Ensemble zu löschen. Weil über die Eigenschaften zunächst abgefragt werden muß, welche Nummer das zu löschende Aggregat hat, kann das Löschen des Aggregates erst als zweiter Teilschritt durchgeführt werden. Auch hier sind die ausgegebenen Reaktionen nicht stöchiometrisch korrekt. Im Falle der Massenspektroskopie gibt es jedoch eine bessere Möglichkeit, die Neutralfragmente von einer Weiterreaktion auszuschließen und dennoch stöchiometrisch korrekte Reaktionen zu erhalten (siehe 6.4.2).

Die Einführung von Edukten während Reaktionen pseudoerster Ordnung wurde auf diese Weise durchgeführt, damit die Ordnung der Reaktion und, von welcher Stoffkonzentration die Reaktionsgeschwindigkeit abhängt, zur Aufstellung der Differentialgleichungen automatisch abgeleitet werden können. Wird darauf bestanden, stöchiometrisch korrekte Reaktionen zu formulieren, besteht die Möglichkeit, die Reaktion als Reaktion zweiter Ordnung zu formulieren, wobei die Konzentrationen der Edukte entsprechend gesetzt werden. Die Simulation mit Reaktionen zweiter Ordnung ist langsamer, da EROS7 mehrere Eduktkombinationen durchtesten muß, bevor es die richtige gefunden hat.

Erzeugung mehrerer Reaktionen aus einer gefundenen Reaktionssubstruktur

Ausgangspunkt für die Möglichkeit mehrere Reaktionen ausgehend von einer gefundenen Reaktionssubstruktur zu erzeugen, war die Ionisationsregel für die massenspektroskopischen Fragmentierungsregeln. Hier enthält die RSS kein Atom, sondern nur ein Elektronensystem. Es gibt nur eine Reaktionssubstruktur ohne Atome, es muß aber für jedes verschiedene Elektronensystem eine eigene Ionisationsreaktion durchgeführt werden. Dazu setzt man in einer Schleife das Edukt als aktives Ensemble, entfernt das Elektron aus dem entsprechenden Elektronensystem und setzt das neue Ensemble als Produkt einer Reaktion. In der Schleife werden so nach und nach alle Reaktionen für alle unterschiedlichen Elektronensysteme ausgeführt. In diesem Fall muß die Symmetriezahl für jede Reaktion auf die Gesamtzahl identischer Elektronensysteme gesetzt werden (siehe 6.6.3).

Als Beispiel sind in Abbildung 104 die n - und π -Ionisationen von Acrylsäure angegeben. Würde der Reaktionstyp der Ionisation ein Atom im Reaktionszentrum enthalten, würde die Ionisation des konjugierten π -Systems mit sechs Elektronen für jedes der fünf an ihm beteiligten Atome erzeugt. Die Reaktionssubstruktur ohne Atome wird allerdings nur ein einziges Mal in jedem beliebigen Molekül gefunden und damit auch nur einmal für Acrylsäure. Um alle drei in Abbildung 104 gezeigten Ionisationen durchführen zu können, ist es möglich aus der einen gefundenen Reaktionssubstruktur ohne Atome mehrere Reaktionen zu generieren, nämlich alle drei Ionisationen.

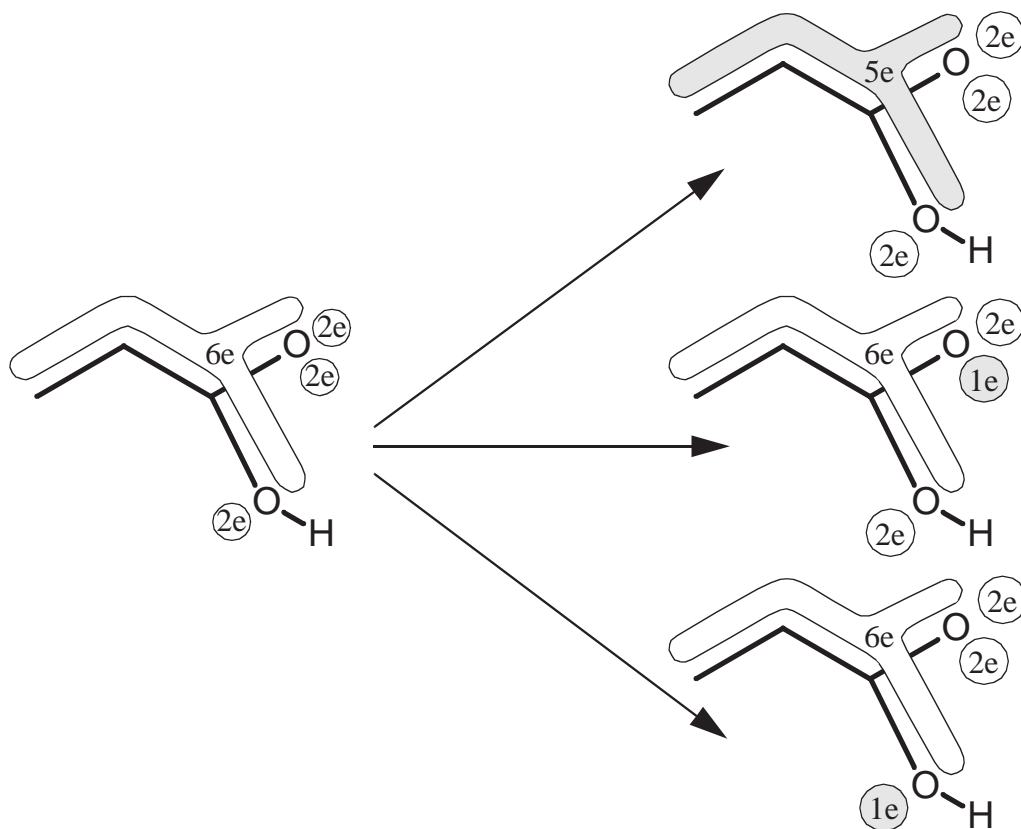


Abbildung 104: Drei Ionisationen (n und π) von Acrylsäure. Die Elektronensysteme, aus denen ein Elektron entfernt wurde, sind grau markiert.

Wird bei einer Reaktion ein Stereozentrum eingeführt und dabei Diastereomere gebildet, müssen diese Produkte mit verschiedenen Reaktionen erzeugt werden. Da auch für diese beiden Reaktionen die Reaktionssubstrukturen identisch sind, ist auch hierfür die Möglichkeit notwendig, mehrere Reaktionen aus einer gefundenen Reaktionssubstruktur zu erzeugen.

Ein- und Ausgabe innerhalb der Regeln

Zu Testzwecken einer neu geschriebenen Regel und um die Regelschnittstelle von EROS7 für andere Anwendungen verwenden zu können (siehe 6.13), kann während der Abarbeitung der Teile der Regeln, die auf einem Ensemble arbeiten, die aktuelle Struktur in verschiedenen Formaten ausgegeben werden oder in eine Datei geschrieben werden. Ebenso wie die Ensembles geschrieben werden können, kann man sie auch aus einer Datei lesen. Dies dient der Einführung von Aggregaten während der Reaktion (siehe Einfügen von Atomen auf Seite 96), wie auch für die Verwendung der Regelschnittstelle für andere Anwendungen. Da Lesevorgänge etwas länger dauern, wird jedes Ensemble, das aus einer Datei gelesen wird, nur ein einziges Mal gelesen und für alle weiteren Male intern gespeichert.

6.6.2. Berechnung der Reaktivität

Die Reaktivität einer Reaktion kann einfach gesetzt werden oder aus Eigenschaften des Edukts, der Zwischenprodukte oder Übergangszustände sowie des Produktensembles berechnet werden. Dazu können eine einfache Funktion, aber auch neuronale Netze, wie ein Backpropagationnetz und ein Counterpropagation- oder Kohonennetz verwendet werden.

Die Eigenschaften werden abgefragt, wenn das entsprechende Ensemble aktiv ist. Zu Beginn der Reaktion können die Eigenschaften des Eduktensembles abgefragt und bereits der Eduktfunktionsteil berechnet werden. Ist ein Reaktionsteilschritt bzw. die Gesamtreaktion abge-

schlossen, können die Eigenschaften von diesem Ensemble (einem Übergangszustand, Zwischenprodukt oder dem Produkt) der Reaktion, abgeholt und zur Reaktivität verrechnet werden.

6.6.3. Anpassung der Symmetriezahl einer Reaktion

Die Symmetriezahl einer Reaktion ist die Zahl der Möglichkeiten, diese Reaktion auf den Edukten durchzuführen. Sie wird als Produkt der Anzahl gleichwertiger Atome für jedes Atom in der RSS berechnet (siehe 6.5.4). Gleichartig sind die Atome dann, wenn es sich um äquivalente Atome handelt, die auch noch die gleiche Beziehung zu allen Atomen in der RSS besitzen. Als Beziehung wird sowohl der Abstand als auch der Nachbarschaftshashcode verwendet, falls das Atom in der RSS als Nachbaratom zu einem anderen Atom in der RSS definiert ist. Zur Identität von Atomen bei der Suche des Reaktionszentrums siehe 6.5.3. Hat man als Edukte zwei gleiche Moleküle und bereits ein Atom in der RSS gefunden, sind die äquivalenten Atome in den beiden Molekülen nicht mehr gleichwertig. Durch diese Bestimmung der Symmetriezahl der Reaktion ist sie auch bei mehreren Teilzentren und mehreren Molekülen korrekt. Sie wird, wie in Abschnitt 6.5.4 beschrieben, automatisch bestimmt.

Werden während der Reaktion jedoch Moleküle eingeschleust, muß die Symmetriezahl der Reaktion mit der reagierenden Substruktur im eingeschleusten Molekül multipliziert werden. Im Fall der Ionisationsreaktion (siehe Erzeugung mehrerer Reaktionen aus einer gefundenen RSS auf Seite 97) wird die Symmetriezahl auf den Wert gesetzt, der angibt wieviele gleiche Elektronensysteme im Molekül vorhanden sind. Diese Zahl kann als Eigenschaft des Elektronensystems abgefragt werden.

6.6.4. Die Verteilungsfunktion

Nachdem alle konkurrierenden Reaktionen ausgehend von einem Eduktensemble durchgeführt sind, wird die Verteilungsfunktion aufgerufen. Es gibt Situationen, in denen man an den erzeugten Reaktionen Korrekturen durchführen möchte. Ein Beispiel hierfür kann die Oniumreaktion bei der Simulation der Prozesse im Massenspektrometer sein (siehe Abbildung 105).

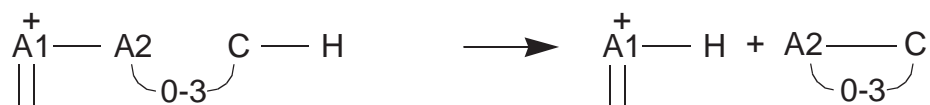


Abbildung 105: Oniumreaktion. A1 ist kein Kohlenstoff, A2 ist ein beliebiges Atom und das Kation enthält kein Radikalzentrum.

Für die Analyse von Massenspektren ist es wichtig, daß möglichst eindeutige Reaktionswege zu generieren (siehe 4.2.3.1). Die Oniumreaktion bildet unabhängig von der Länge der Atomkette (0–3) immer das gleiche Kation. Kann nun das Wasserstoffatom aus unterschiedlichen Entfernungen kommen, können diesen Reaktionen keine eindeutigen Reaktionswahrscheinlichkeiten mehr zugeteilt werden, da sie aufgrund der einzig verfügbaren Information, den Signalintensitäten im Massenspektren, nicht unterschieden werden können. Daher sollte nur die Reaktion erzeugt werden, die fast die ganze Menge des Produktkations bildet. Dies kann durch eine aufwendige Suche nach optionalen Atomen geschehen oder durch den nachträglichen Ausschluß der untergeordneten Reaktionen in der Verteilungsfunktion erfolgen, nachdem alle Reaktionen des Eduktkations generiert wurden.

Generell ist es hier möglich, die Reaktionen, die von einem Eduktensemble ausgehen, zu vergleichen und die Reaktivitätswerte der Reaktionen gegebenenfalls anzupassen oder eine Reaktion ganz zu verwerfen. Dazu stehen alle an das Kernsystem von EROS7 übergebenen Variablen der Typen *int* und *double* sowie die Symmetriezahl der Reaktionen, die Phase für die

Produkte (siehe unten) und eine Nummer für die Entscheidungen zur Verfügung, aus der die Identität der Produkte erkannt werden kann. Die Nummer 1 steht dabei für das Eduktensemble und alle anderen Nummern stehen für neue Produkte, wobei gleiche Produkte auch gleiche Nummern erhalten. Auf die Aggregate selbst kann jedoch nicht zugegriffen werden. Bei einem komplexen Satz von Reaktionstypen, wie sie für die Analyse und Simulation von Massenspektren (siehe 4.5.3) verwendet wurden, kann es vorkommen, daß eine Reaktion vom Edukt zum Produkt von zwei verschiedenen Reaktionstypen erzeugt wird (siehe Abbildung 106).

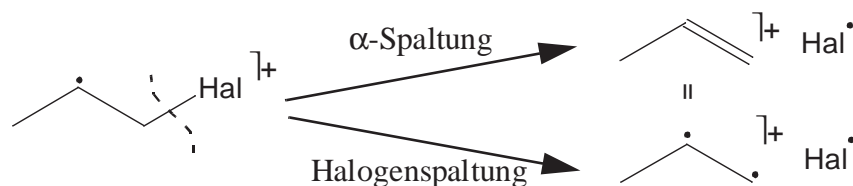


Abbildung 106: Konkurrenz zwischen der α -Spaltung und der Halogenspaltung. Die gestrichelte Linie zeigt die Spaltung der Bindung an.

In der Regel möchte man allerdings nur die Reaktion mit der höchsten Priorität behalten. Deshalb ist es möglich, die Reaktionen mit niedrigeren Prioritäten in der Verteilungsfunktion zu verwerfen. Sie ist dann nicht mehr Bestandteil des Reaktionsnetzwerks. Eine ähnliche Situation ergibt sich für die Oniumreaktion (siehe 4.5.3), in der ein Wasserstoffatom aus verschiedenen Entfernungen umgelagert und ein neutraler Rest abgespalten wird. Ist die Oniumgruppe nicht in einem Ring, entsteht in allen Fällen dasselbe Kation, nur das abgespaltene Neutralfragment unterscheidet sich, das für das Massenspektrum ohne Bedeutung ist. Unter den Oniumreaktionen mit verschiedenen Abständen zwischen dem Wasserstoffatom und dem Zentrum der positiven Ladung gibt es eine eindeutige Absenkung der Reaktivität mit steigender Entfernung. Hat man, wie im Fall der Analyse von Massenspektren, Probleme mit redundanten Reaktionen bezüglich des Kations, kann man sich durch den Ausschluß der redundanten Reaktion auf die reaktivste beschränken. Handelt es sich nicht um einen Phasenübergang, sind auch Reaktionen, bei denen das Produkt gleich dem Edukt ist, unerwünscht. Diese Reaktionen in der Verteilungsfunktion nachträglich zu verwerfen, ist wesentlich einfacher als in jedem Reaktionstyp zu überprüfen, ob dies der Fall ist.

Für organische Reaktionen ist die Verteilungsfunktion ohne Bedeutung. Hier genügt die in EROS7 eingebaute Verteilungsfunktion, die aufgerufen wird, wenn in den Regeln keine angegeben ist. Sie verwirft Reaktionsduplikate und Reaktionen, die das Edukt nicht verändern und keine Phasenübergänge sind.

Schließlich können hier zusätzliche Werte für die Reaktionen berechnet werden.

6.7. Arten der Reaktionsführung: Reaktoren und Phasen

EROS7 vereinigt alle Möglichkeiten des Aufbaus des Reaktionsnetzwerks, die die Vorläuferprogramme EROS5, EROS6 und MASSIMO gehabt haben, und erweitert diese. Das allgemeine Konzept ist das der Reaktoren und Phasen. Man kann dabei mehrere Reaktoren und Phasen verschachteln, wie in der technischen Chemie. Jeder Reaktor besteht aus einer oder mehreren Phasen. Eine Phase kann eine Phase im üblichen Sinne, wie wäßrige und organische Phase, sein.

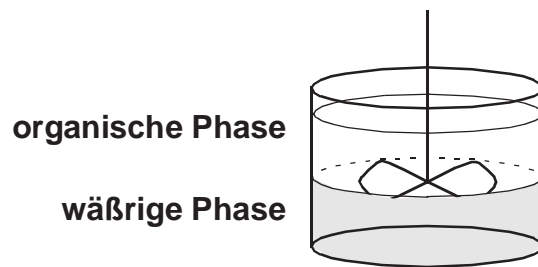


Abbildung 107: Reaktor mit einer organischen und einer wäßrigen Phase.

Die Phasen können aber auch verschiedene verschaltete Kessel darstellen, wenn man den Konzentrationsverlauf der Substanzen in den Kesseln bestimmen will.

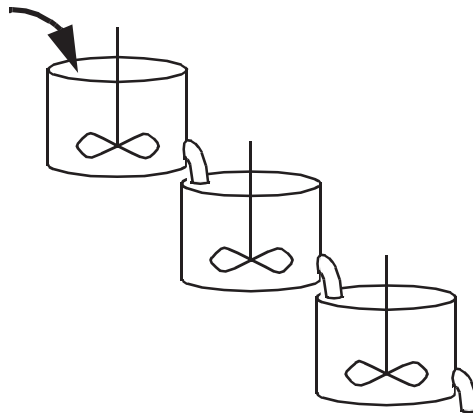


Abbildung 108: Rührkesselkaskade aus drei Kesseln.

Die Reaktoren entsprechen abgeschlossenen Prozeßeinheiten, die am Ende einer bestimmten Zeit betrachtet werden, beispielsweise die Mengenverteilung bei der Aufnahme eines Massenspektrums oder die Produktverteilung eines kontinuierlich arbeitenden Rührkessels, in dem eine Synthese durchgeführt wird und sich im stationären Zustand befindet. Die einzelnen Reaktoren werden nacheinander berechnet, wobei die Ergebnisse, also die Stoffmengen und Stoffmengenflüsse, auch wieder für die nächsten Reaktoren verwendet werden können. Beschreibt man die drei Kessel der Kaskade aus Abbildung 108 nicht als Phasen, sondern als drei Reaktoren, beginnt man mit der Berechnung des Konzentrationsverlaufs des zweiten Kessels nach Erreichen des Endzustands des ersten Kessels. Jeder Reaktor führt dabei seine eigenen kinetischen Berechnungen durch. Mit mehreren Reaktoren ist es so möglich, in einem ersten Reaktor eine Synthese zu simulieren und in einem zweiten Reaktor die Massenspektren zu berechnen. Über die Einstellungen der Phasen können die verschiedenen Typen der Reaktionsführung simuliert werden.

6.7.1. Grundtypen der Reaktionsführung

6.7.1.1. Unterscheidung kontinuierlich / absatzweise

Es gibt zwei grundsätzliche Arten der Reaktionsführung, die kontinuierliche und die absatzweise Betriebsart. Verbrennungsprozesse kommen in beiden Arten vor. Eine offene Gasflamme ist ein Beispiel für die kontinuierliche Betriebsart, da hier das Gas und der zur Verbrennung nötige Sauerstoff mit der Luft kontinuierlich zugeführt werden, wie sich auch die Verbrennungsgase vom Ort der Reaktion entfernen. Bei der Verbrennung in einem Otto-Motor (eine Zündung) oder bei einer Explosion liegt dagegen eine absatzweise Reaktionsführung vor, da in beiden Fällen zu Beginn der Reaktion eine bestimmte Menge der Reaktanden vorliegt, die während der

Reaktion verbraucht werden. Eine Zufuhr der Ausgangsmaterialien während der Reaktion findet hier nicht statt.

Da bei einer absatzweisen Betriebsart Totzeiten für das Befüllen und Entleeren der Reaktionsgefäße entstehen, in denen nichts produziert wird, bevorzugt man für großtechnische Prozesse mit einer Produktionsmenge von vielen tausend Jahrestonnen eine kontinuierliche Reaktionsführung, bei der keine Totzeiten auftreten. Eine kontinuierliche Reaktionsführung besitzen KIK, IR und Kaskade.

6.7.1.2. Absatzweise betriebener, idealer Rührkessel (AIK)

Für den idealen Rührkessel wird angenommen, daß die Reaktanden zu jeder Zeit vollständig durchmischt sind. Wird er überdies absatzweise betrieben, d.h. werden zu Beginn eine bestimmte Menge der Ausgangsmaterialien hineingegeben und die Reaktionsmischung nach einer vorgegebenen Zeit entnommen, handelt es sich um einen AIK. Eine derartige Reaktionsführung ist die übliche Arbeitsweise im Labor, wenn eine Reaktion im Dreihalskolben oder Schlenkrohr durchgeführt wird. Industrielle Chargenprozesse, die für die Synthese von kleinen Mengen - z.B. für Arzneimittel - eingesetzt werden, entsprechen ebenfalls dieser Reaktionsführung.

6.7.1.3. Ideales Strömungsrohr (IR)

Bei einem idealen Strömungsrohr, also einem Strömungsrohr mit turbulenter Pfropfströmung, gilt radial vollständige Durchmischung, während axial keine stattfindet. Deshalb verhalten sich die Konzentrationsverläufe eines idealen Strömungsrohrs genauso wie die eines absatzweise betriebenen, idealen Rührkessels. Der Quotient der Strecke vom Anfang des idealen Strömungsrohrs bis zur betrachteten Stelle im Rohr und der Strömungsgeschwindigkeit entspricht dabei der Reaktionszeit des AIK, da sich die Ausgangsmaterialien an dieser Stelle gleich lang im Reaktor befunden haben. [2]

Der Vorteil eines idealen Strömungsrohrs gegenüber einem AIK liegt darin, daß es kontinuierlich betrieben wird, was für großtechnische Prozesse von Bedeutung ist. Das ideale Strömungsrohr wird dennoch nur selten eingesetzt, da es schlecht zu regeln ist. Die Ursache dafür liegt an der Verzögerungszeit, mit der eine Veränderung der Konzentration am Ende des Reaktionsrohrs festgestellt wird.

6.7.1.4. Kontinuierlich betriebener, idealer Rührkessel (KIK)

Im Gegensatz zum AIK werden dem KIK die Ausgangsmaterialien fortlaufend zugeführt. In gleichem Maße, wie neue Substanzen zugeführt werden, wird auch ein Teil der Reaktionsmischung ständig entnommen. Bei der angenommen idealen Durchmischung verweilen allerdings nicht alle Teile der Ausgangsmaterialien die gleiche Zeit im Kessel, in der sie eine Reaktion eingehen können (siehe 6.7.1.6). Diese Art der Reaktionsführung wird großtechnisch eingesetzt.

6.7.1.5. Rührkesselkaskade

Werden mehrere kontinuierlich betriebene, ideale Rührkessel hintereinandergeschaltet, handelt es sich um eine Kaskade. Sie hat den Vorteil, daß sie kontinuierlich betrieben wird und stellt im Bezug auf Verweilzeitverhalten (siehe 6.7.1.6) und Regelbarkeit ein Zwischenglied zwischen AIK/IR und dem KIK dar. Aus je mehr Kesseln die Kaskade aufgebaut ist, um so ähnlicher ist sie einem idealen Strömungsrohr.

6.7.1.6. Vergleich der Reaktionsführungen

Gegenüber einem AIK und einem Idealrohr befinden sich beim KIK nicht alle Teile der Ausgangsmaterialien gleich lange im Kessel, in der sie miteinander reagieren können. Ein Teil der Edukte verläßt beim KIK den Kessel schon wieder, kurz nachdem sie dem Kessel zugeführt wurden. Die Rührkesselkaskade, bei der mehrere KIKs hintereinander geschaltet sind, stellt ei-

nen Kompromiß aus Idealrohr und KIK dar, wobei mit der Zahl der Kessel gewählt werden kann, ob die Kaskade sich ähnlicher einem KIK oder einem Idealrohr verhalten soll.

In Abbildung 109 sind die aufsummierten Anteile angegeben, die den Reaktor nach einer bestimmten Zeit verlassen haben. Dabei werden nur Anteile betrachtet, die dem Reaktor gleichzeitig zugegeben wurden. Es wird nicht unterschieden, ob die Edukte während ihres Aufenthalts im Reaktor eine Reaktion eingegangen sind oder nicht, weshalb Abbildung 109 das Verweilzeitverhalten einer Substanz zeigt, die keine Reaktion eingeht.

**Teil gleichzeitig zugegebener Substanzen,
der den Reaktor mit oder ohne Reaktion
wieder verlassen hat.**

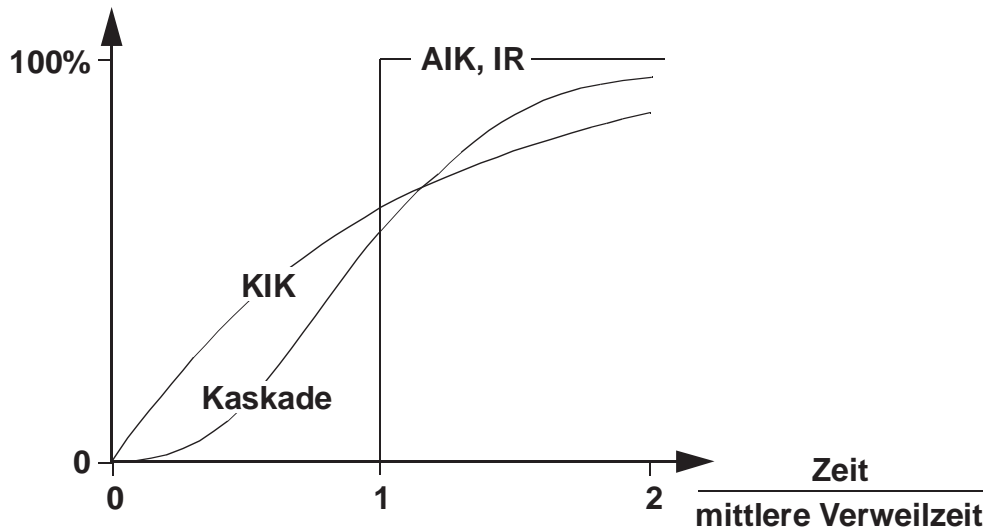


Abbildung 109: Verweilzeitverhalten von AIK/IR, KIK und Kaskade (4 Kessel). [57]

Spielt die zeitliche Kontrolle der Reaktion eine entscheidende Rolle, ist es bei einem kontinuierlichen Betrieb notwendig eine Kaskade oder ein Idealrohr zu verwenden, da der KIK mit seinen breit gestreuten Verweilzeiten hierfür nicht geeignet ist.

Die Regelbarkeit der kontinuierlichen Reaktionsführungen ist für den KIK gut und schlecht bis unmöglich für das Idealrohr. In dem Maße, in dem die Kaskade dem Rohr ähnlicher wird, sinkt auch die Regelbarkeit.

6.7.1.7. Monomolekularer Zerfall

Da beim monomolekularen Zerfall nur Reaktionen erster und pseudoerster Ordnung stattfinden, ist die Durchmischung bei dieser Reaktionsführung ohne Bedeutung. Ansonsten ist es möglich den monomolekularen Zerfall neben der üblichen, absatzweisen Betriebsart wie den idealen Rührkessel auch kontinuierlich durchzuführen. Ob bimolekulare Reaktionen erzeugt werden, ausgenommen, gleicht das restliche Verhalten dem idealen Rührkessel.

6.7.2. Die Phasenmodi und ihre Auswahl der Edukte

In diesem Kapitel werden die verschiedenen, zur Verfügung stehenden Phasenmodi und wie sie die Eduktaggregate zusammenstellen kurz vorgestellt. Wie und wann sie eingesetzt werden, wird in den nachfolgenden Kapiteln beschrieben.

Jede Phase hat ihren eigenen Satz von Aggregaten, von dem ausgehend sie ihre Reaktionen erzeugt. In welcher Weise dies geschieht, hängt vom Phasenmodus ab. Diese sind im folgenden näher beschrieben. Bei einigen Modi werden auch Aggregate aus zwei unterschiedlichen Phasen als Eduktensemble verwendet. Die Produkte einer Reaktion können in einer beliebigen an-

deren Phase gespeichert werden. Da alle Phasen eines Reaktors miteinander verbunden sind und für sie eine gemeinsame kinetische Berechnung durchgeführt wird, müssen alle Phasen, die der Edukte und die der Produkte einer Reaktion, zu einem Reaktor gehören, wenn die Konzentrationen der Produkte bestimmt werden. Ein zweiter Reaktor kann mit den erzeugten Verbindungen inklusive deren abschließenden Konzentrationen und Stoffflüssen des vorhergehenden Reaktors fortfahren.

Generell werden alle Reaktionstypen auf die Eduktensembles aller Phasen angewendet, sofern eine entsprechende Reaktionssubstruktur gefunden werden kann. Während der Substruktursuche steht allerdings auch die Information zur Verfügung, welche Phase gerade Reaktionen generieren möchte, also zu welcher Phase das erste Eduktaggregat gehört. So kann festgelegt werden, welche Reaktionstypen für die einzelnen Phasen erlaubt sind.

6.7.2.1. Der Phasenmodus Rührkessel

Der Modus „Rührkessel“ ist der Modus von EROS6, bei dem alle Aggregate alleine und zusammen mit allen anderen in einer Phase reagieren. Dabei werden zwei Fälle unterschieden: einerseits dürfen Eduktaggregate auch mit sich selbst reagieren, andererseits ist dies verboten (siehe auch MIX und MIX_NO_A_A im Anhang B auf Seite 222). Neben diesen Reaktionen werden auch noch die Grenzflächenreaktionen (siehe Seite 105) erzeugt, wenn für diese Phase Kontakte zu anderen Phasen angegeben sind.

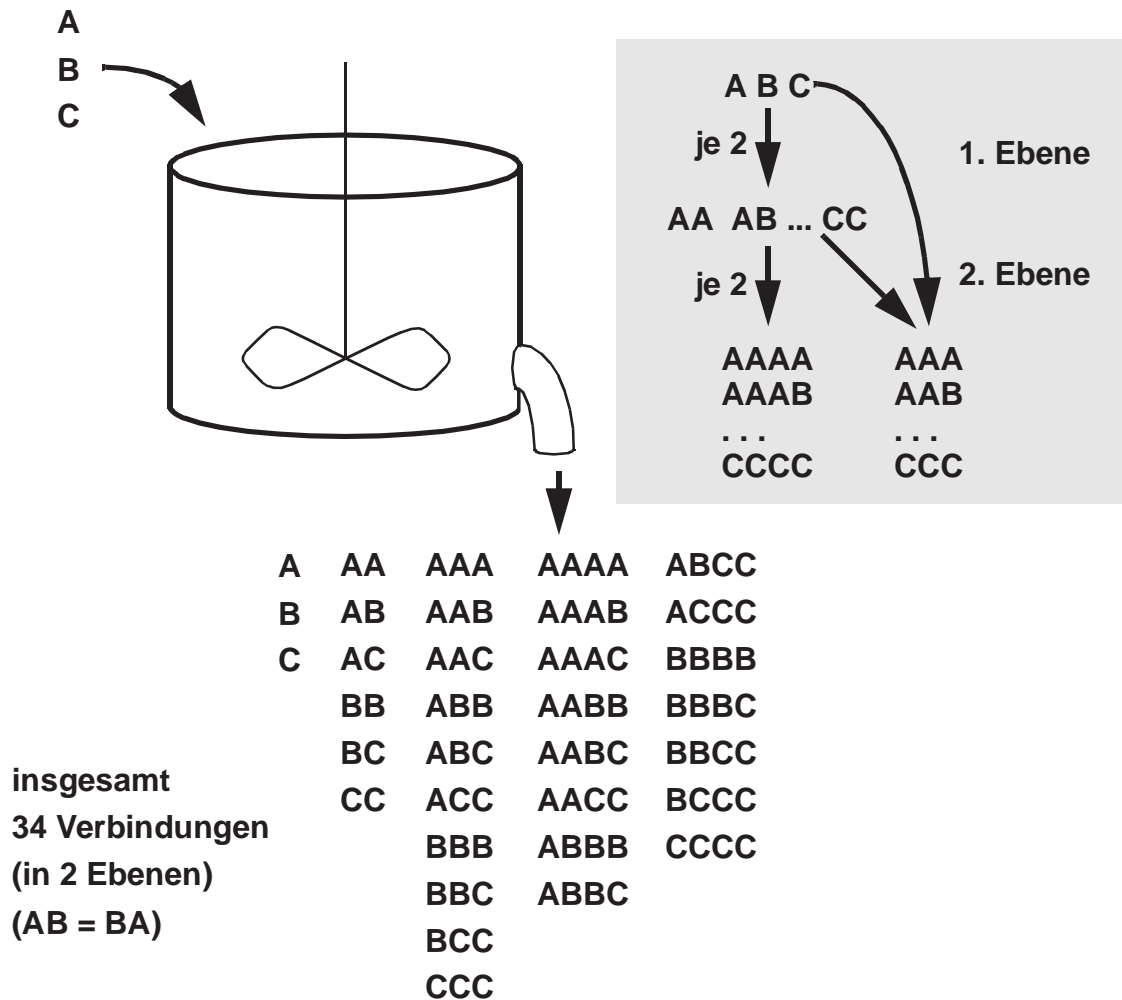


Abbildung 110: Phasenmodus „Rührkessel“ (MIX), zwei Reaktionsebenen. Grau unterlegt ist gezeigt, wie die Produkte in den zwei Reaktionsebenen gebildet werden, wobei jeweils zwei Moleküle miteinander reagieren.

Bezüglich der Kinetik kann der ideale Rührkessel absatzweise (siehe 6.7.4.2) oder kontinuierlich (siehe 6.7.4.4) betrieben werden.

6.7.2.2. Der Phasenmodus Monomolekular

Dieser Modus ist eine etwas abgewandelte Form des Modus, wie er in MASSIMO verwendet wurde. Er erzeugt ausschließlich Reaktionen erster und pseudoerster Ordnung. Pseudoerster Ordnung sind sie dann, wenn während der Reaktion Reagenzien hinzugefügt werden, die im Überschuß vorliegen. Für EROS7 sind die reagierenden Spezies Aggregate. Da ein Aggregat in den allermeisten Fällen aus nur einem Molekül bestehen wird, es sich also tatsächlich um monomolekulare Reaktionen handelt, und der Begriff monomolekularer Zerfall allgemein gebräuchlich ist, wurde dieser Modus „monomolekular“ genannt. Besteht ein Aggregat aus mehreren Molekülen, handelt es sich beispielsweise um ein Protein mit Quartärstruktur, bei dem die verschiedenen Proteinstränge, die Moleküle, über spezielle Wechselwirkungsgruppen zusammengehalten werden. Die Moleküle eines Aggregates können sich also nicht unabhängig voneinander bewegen, wodurch sich das kinetische Verhalten der Reaktionen eines solchen Aggregates um nichts von monomolekularen Reaktionen unterscheidet. EROS7 generiert in diesem Modus Reaktionen, die von einem Aggregat ausgehen und wendet alle möglichen Reaktionstypen auf dieses an. Werden während der Reaktion Moleküle oder Aggregate eingeschleust, handelt es sich um Reaktionen pseudoerster Ordnung. Entstehen in einer Reaktion mehrere Produktaggregate, werden diese voneinander getrennt und jeweils als separate Eduktensembles eingesetzt.

Dieser Modus eignet sich für die Simulation der massenspektroskopischen Fragmentierung, wie auch für den Abbau verdünnter Chemikalien in der Umwelt und die Metabolisierung von Arzneimitteln.

(siehe auch MONOMOLEC im Anhang B auf Seite 222)

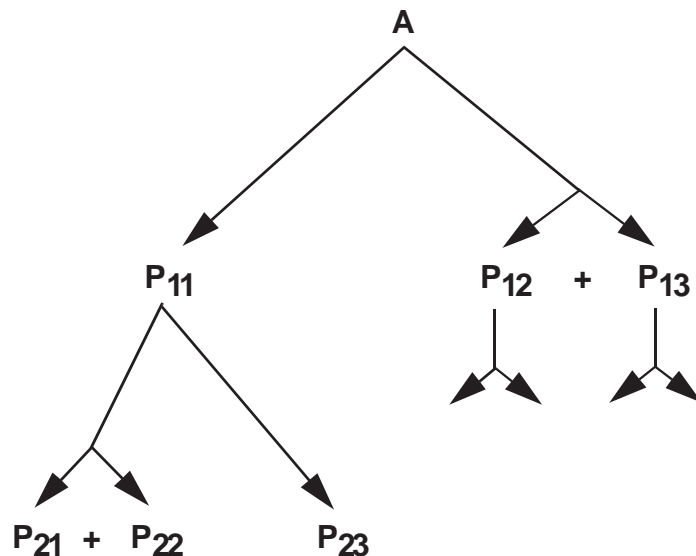


Abbildung 111: Monomolekularer Abbau von A.

6.7.2.3. Der Phasenmodus Grenzflächenreaktionen

Mit diesem Modus werden Grenzflächenreaktionen erzeugt, wobei ein Aggregat mit einem Aggregat einer angrenzenden Phase reagiert. Reaktionen, bei denen nur ein Aggregat beteiligt ist, werden in diesem Modus nicht durchgeführt. Benachbart sind die Phasen, für die die Nachbarschaft angegeben ist. Die Nachbarschaft der Phasen ist gerichtet, damit Reaktionen nicht doppelt erzeugt werden. Es wird angegeben, zu welchen Phasen die aktuelle Phase im

Phasenmodus „Grenzflächenreaktionen“ Kontakte hat; z.B. Phase 1 hat Kontakt zu Phase 2. Dann erzeugt die Phase 1 die Reaktionen und kombiniert als Eduktenssembles alle ihre Aggregate mit allen Aggregaten der Phase 2. Die Phase 2 kann im Modus „Inert“ (siehe 6.7.2.4) sein und selbst keine Reaktionen erzeugen oder auch im Modus „Grenzflächenreaktionen“. Wird während der Reaktion keine andere Phase für die Produkte angegeben, werden sie der Phase 1 zugeteilt. Sind Kontakte zu anderen Phasen angegeben, werden diese Reaktionen auch im Modus „Rührkessel“ durchgeführt.

(siehe auch SURFACE im Anhang B auf Seite 222)

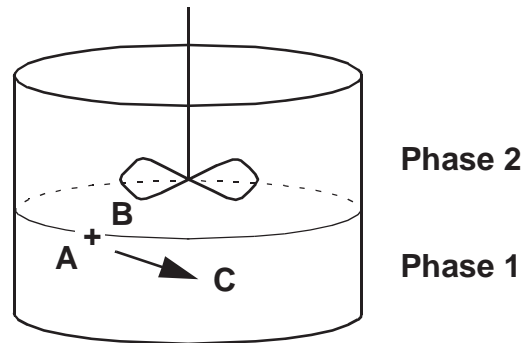


Abbildung 112: Grenzflächenreaktion.

6.7.2.4. Der Phasenmodus Inert

Dieser Modus erzeugt selbst keine Reaktionen, sondern dient nur der Speicherung von Aggregaten, die dann an Grenzflächenreaktionen teilnehmen können.

(siehe auch INERT im Anhang B auf Seite 222)

6.7.2.5. Der Phasenmodus laminares Strömungsrohr

Der Phasenmodus des „laminaren Strömungsrohrs“ ist der Modus von EROS5, bei dem das Reaktionsensemble während aller nacheinander ablaufender Reaktionen zusammengehalten wird. Die Produkte einer Reaktion können dabei untereinander weiterreagieren, jedoch nicht mit den Edukten oder den Produkten parallel ablaufender Reaktionen. Dies entspricht einem Strömungsrohr mit laminarer Strömung, bei dem keine Durchmischung stattfindet. Werden einfache Moleküle im Reaktionsverlauf öfters benötigt, können diese, wie in EROS5, jedesmal in der Vorbehandlungsfunktion für die Edukte, aber auch selektiv für einen Reaktionstyp, eingeschleust werden.

(Siehe auch TUBE im Anhang B auf Seite 223)

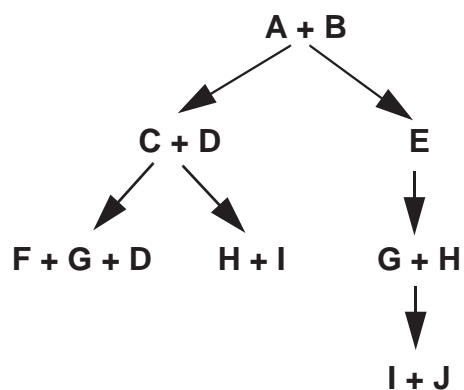


Abbildung 113: Phasenmodus „laminares Strömungsrohr“.

Da ein Molekül / Aggregat in mehreren Reaktionsebenen auftreten kann und dort auch in unter-

schiedlichen Konzentrationen vorliegen kann, die Kinetik aber für vollständige Durchmischung definiert ist, kann für diesen Modus keine Kinetik berechnet werden, wie auch in EROS5 keine Kinetik enthalten war. Dies schränkt den Einsatz dieses Phasenmodus stark ein.

6.7.3. Phasenübergänge und Phasentransferreaktionen

Zur Überführung von Aggregaten von einer in eine andere Phase werden Reaktionen eingesetzt, die den Übergang in eine andere Phase beschreiben (Phasenübergänge) oder eine Reaktion an den Edukten zusammen mit dem Übergang durchführen (Phasentransferreaktionen). Die Eduktaggregate können, wenn bimolekulare Reaktionen erzeugt werden, aus verschiedenen Phasen stammen und die Produkte in eine dritte Phase gehen. Aus welchen Phasen die Edukte kommen, wird über die Einstellungen der Phasen gesteuert, welcher Phase die Produkte zugeordnet werden, wird in der Reaktionsgenerierung des Reaktionstyps festgelegt. Anfänglich ist die Phase für die Produkte identisch mit der Phase des ersten Edukts, von der aus die Reaktion erzeugt wird. Wird die Phase für die Produkte während der Reaktion auf eine andere Phase gesetzt, wird die Reaktion zum Phasenübergang bzw. zur Phasentransferreaktion. In den meisten Fällen werden die Reaktionen innerhalb einer Phase durchgeführt und es existiert ein spezieller Reaktionstyp, der nur für die Phasenübergänge zuständig ist, aber keine Veränderungen am Ensemble vornimmt. Bei den speziellen Reaktionen für die kombinatorische Chemie werden häufig auch Kombinationen aus der Modifikation des Ensembles und einem Phasentransfer (Phasentransferreaktionen) verwendet (siehe 6.7.4.10).

6.7.4. Aufbau der Reaktoren mit Phasen in verschiedenen Modi

6.7.4.1. Wann werden welche Phasenmodi eingesetzt?

Abbau von Umweltchemikalie und Pharmakokinetik

Hierfür wird ein Reaktor mit eine Phase im Modus „monomolekular“ eingesetzt, der Reaktionen erster und pseudoerster Ordnung generiert.

Kombinatorische Chemie

Abhängig von der Zahl der reagierenden, funktionellen Gruppen und der Zahl der verschiedenen Stoffgruppen für die Reaktion wird ein Reaktor mit mehreren Phasen in den Modi „Grenzflächenreaktionen“ und „Inert“ benötigt.

Laborsynthesen und Chargenprozesse

Für die Simulation dieser Prozesse wird der absatzweise betriebene, ideale Rührkessel (AIK) verwendet. Dieser kann je nach Art der Synthese eine (siehe absatzweise betriebene, ideale Rührkessel) oder mehrere Phasen (siehe Rührkessel mit organischer und wässriger Phase, wenn dieser absatzweise betrieben wird) besitzen.

Simulation von massenspektroskopischen Prozessen

Auch für die Simulation der Reaktionen im Massenspektrometer eignet sich ein Reaktor mit einer Phase im Modus „monomolekular“.

Großtechnische Prozesse

Für die Vorhersage großtechnischer Prozesse kann keine eindeutige Aussage getroffen werden, da diese mit ganz unterschiedlichen Reaktionsführungen durchgeführt werden. Da der überwiegende Teil der Synthesen kontinuierlich erfolgt, kommen hier der kontinuierliche betriebene, ideale Rührkessel, die Rührkesselkaskade, die Rührkesselkaskade im Gegenstrom, ein Rührkessel mit organischer und wässriger Phase (kontinuierlich) und das ideale Strömungsrohr zum Einsatz.

Syntheseplanung

Zur Nutzung von EROS7 in der Syntheseplanung können die Phasenmodi „laminares Strö-

mungsrohr“ bzw. „monomolekular“ eingesetzt werden.

6.7.4.2. Absatzweise betriebener, idealer Rührkessel (AIK)

Ausgehend von den verschiedenen Modi der Phasen lassen sich die einzelnen technischen Reaktoren aufbauen. Der absatzweise betriebene, ideale Rührkessel ist ein Reaktor mit einer Phase im Modus „Rührkessel“. Anwendung findet diese Art der Reaktionsführung bei der Simulation von Chargenprozessen, wie sie für industrielle Synthesen kleiner Mengen, zum Beispiel von Arzneimitteln und im Labor eingesetzt werden.

Für die Ausgangsmaterialien sind bei dieser Reaktionsführung Anfangskonzentrationen größer null gegeben, aber keine Stoffmengenzuflüsse. Die Phase hat ein bestimmtes Volumen und einen Fluß (Zu- und Abfluß) von null. Zudem wird beim idealen Rührkessel angenommen, daß die Substanzen immer vollständig durchmischt sind.

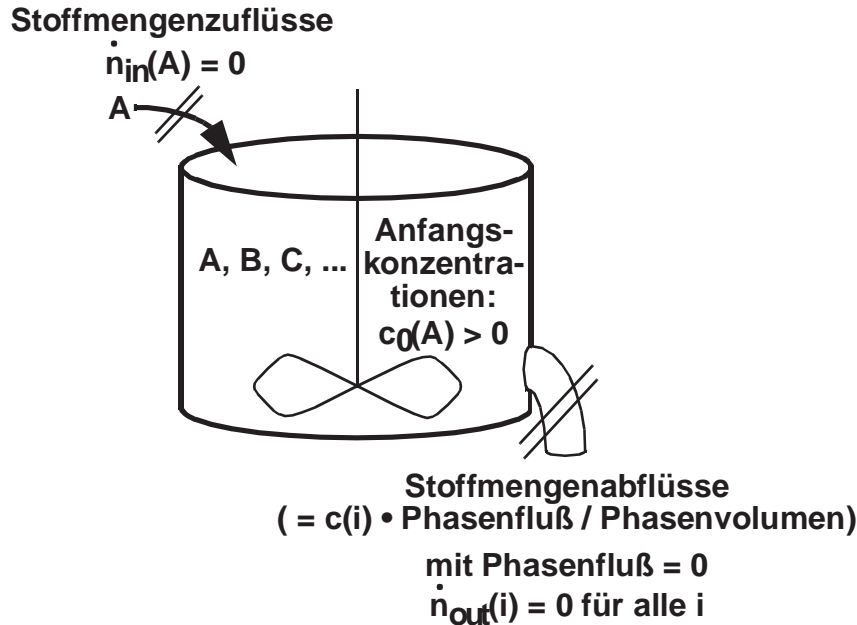


Abbildung 114: Einstellungen für einen AIK.

6.7.4.3. Ideales Strömungsrohr (IR)

Da das Ideale Strömungsrohr das gleiche kinetische Verhalten besitzt wie der AIK, sind die Einstellungen für EROS7 identisch (siehe 6.7.4.2). Die Reaktionszeit steht in diesem Fall für einer entsprechenden Strecke des Reaktionsrohrs.

6.7.4.4. Kontinuierlicher, idealer Rührkessel (KIK)

Abweichend vom AIK sind beim kontinuierlich betriebenen, idealen Rührkessel die Stoffmengenzuflüsse der Ausgangsmaterialien sowie der Phasenfluß, der Fluß des Lösungsmittels durch die Phase, größer null. Die Anfangskonzentrationen der Ausgangsmaterialien können sowohl festgelegt werden als auch null sein, wenn man auch das Befüllen des Reaktors simulieren möchte.

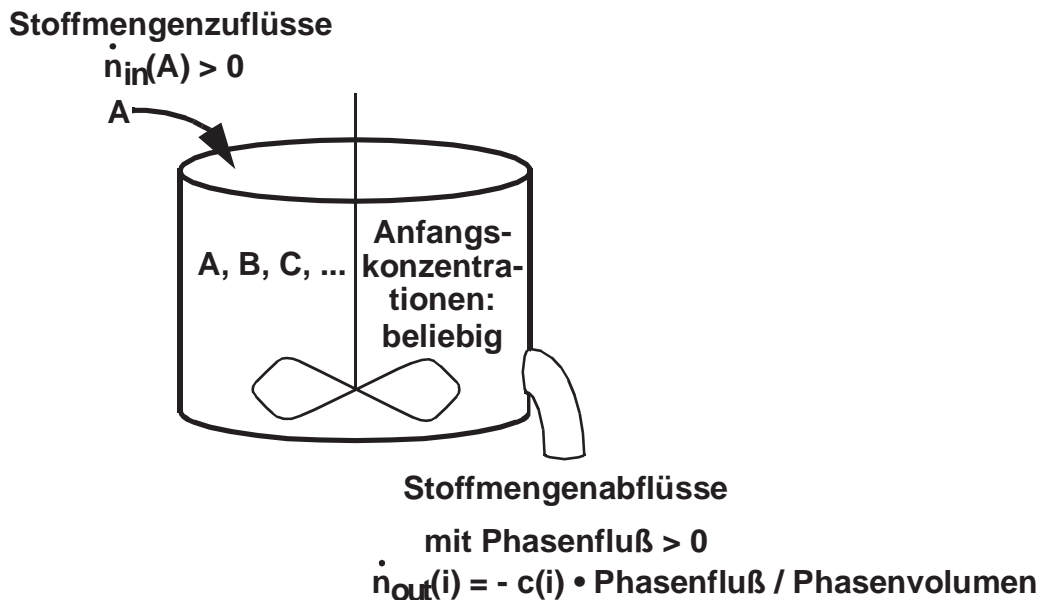


Abbildung 115: Einstellungen für einen KIK.

6.7.4.5. Rührkesselkaskade

Ausgehend vom KIK kann man auch Reaktoren mit mehreren Phasen und mehreren Reaktorteilen aufbauen. Mehrere kontinuierlich betriebene, ideale Rührkessel lassen sich zu einer Rührkesselkaskade zusammenschalten. Für eine Kaskade aus drei Kesseln hat man einen Reaktor mit drei Phasen im Modus „Rührkessel“, die die einzelnen Kessel darstellen. Der Ausfluß der ersten beiden Rührkessel fließt jeweils in den nächsten, die Ausgangsmaterialien fließen in den ersten Kessel und die Konzentrationen der Substanzen im letzten Kessel sind gleich den Konzentrationen im Abfluß der Kaskade. Jede Verbindung erhält für jeden Kessel eine eigene Aggregatnummer, damit die unterschiedlichen Konzentrationsverläufe in den Kesseln berechnet werden können. Intern werden die Aggregate dennoch nur einmal gespeichert.

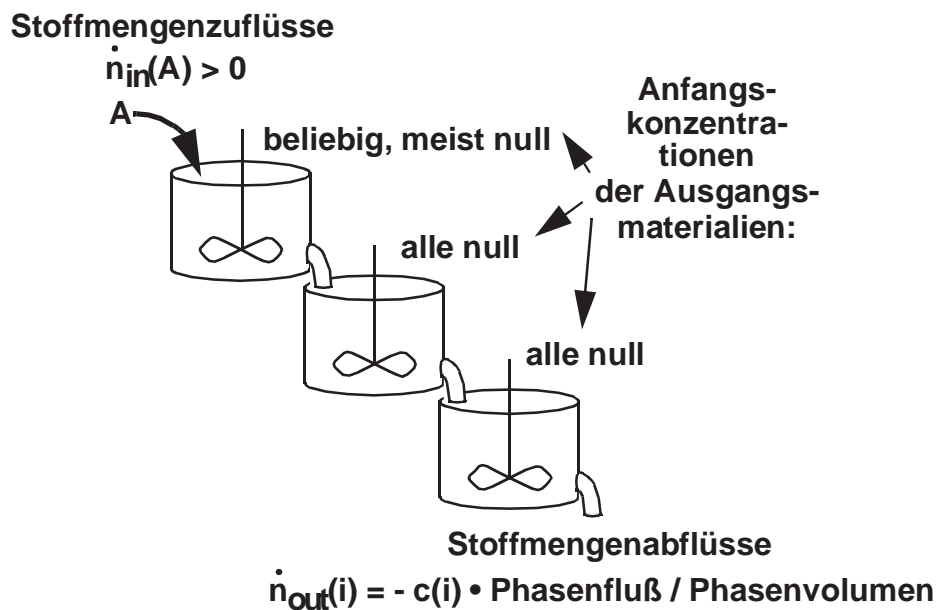


Abbildung 116: Rührkesselkaskade aus drei Kesseln.

Die Reaktionen werden für jeden Kessel einzeln generiert, da die Reaktionen in den verschiedenen Phasen (Kesseln) unterschiedlich gut bzw. gar nicht ablaufen können. Dies liegt an der

Möglichkeit, daß man die Rührkessel in gleicher Weise für andere Zwecke verschalten kann. So kann man im ersten Kessel eine Reaktion in trockenem Alkohol durchführen und im zweiten Kessel die Lösung mit Wasser versetzen und wäßrig aufarbeiten. Im zweiten, wäßrigen Kessel laufen dann natürlich bevorzugt andere Reaktionen als im ersten Kessel ab.

Der Abfluß der letzten Phase einer Kaskade gilt als Ergebnis des Reaktors. Dient diese Phase als Eingabe für einen weiteren Reaktor, ist ihre Endkonzentration die Anfangskonzentration und ihr Stoffabfluß gleich dem Stoffzufluß für den zweiten Reaktor.

In den meisten Fällen werden alle Rührkessel das gleiche Volumen und den gleichen Fluß durch die Kessel besitzen, es ist aber theoretisch auch möglich, verschiedene Werte zu verwenden. Sind die Flüsse aus einer und in eine andere Phase unterschiedlich, sind die Stoffströme für die einzelnen Substanzen dennoch gleich. Alle Substanzen, die aus einem Kessel herausfließen, kommen also im nächsten Kessel der Kaskade an, was bei unterschiedlichen Flüssen nicht für das Lösungsmittel gilt.

6.7.4.6. Rührkessel mit organischer und wäßriger Phase

Zur Simulation eines Rührkessels mit einer organischen und einer wäßrigen Phase nimmt man einen Reaktor mit zwei Phasen im Modus „Rührkessel“. Die Ausgangsmaterialien werden zusammen mit ihren Anfangskonzentrationen je einer der beiden Phasen zugeteilt. Werden für sie Zuflüsse angegeben geht man vom absatzweisen Betrieb auf eine kontinuierliche Reaktionsführung über. Bisher würde es sich um zwei unabhängige absatzweise betriebene bzw. kontinuierliche Rührkessel handeln. Ein Rührkessel mit zwei Phasen entsteht daraus durch das Erzeugen von reinen Phasenübergängen für alle Substanzen. Am Ensemble werden dabei keine Modifikationen durchgeführt, sondern nur die Phase für die Produkte gesetzt. Die Geschwindigkeitskonstante des Transfers von der Phasengrenzfläche, den Diffusionskoeffizienten durch die Grenzschicht, die Grenzschichtdicke, dem Verteilungskoeffizienten und den Phasenvolumina ab (siehe 6.9.1). Die Geschwindigkeitskonstanten der beiden Phasenübergänge für eine Substanz haben in beiden Richtungen unterschiedliche Werte und der Quotient deren Produkte mit den Phasenvolumina ergibt den Verteilungskoeffizienten N für die Substanz in den beiden Phasen.

$$N = \frac{c_1}{c_2} = \frac{k_2 \cdot V_2}{k_1 \cdot V_1} \quad (5)$$

6.7.4.7. Rührkesselkaskade im Gegenstrom

Kombiniert man nun kontinuierliche Rührkessel mit zwei Phasen zu einer Kaskade, kann man die Rührkesselkaskade im Gleich- oder Gegenstrom schalten, indem man die Reihenfolge der Zu- und Abflüsse der einzelnen Phasen variiert. Hier soll nun eine Rührkesselkaskade aus drei Kesseln im Gegenstrom gezeigt werden. Dazu werden sechs Phasen in einem Reaktor benötigt.

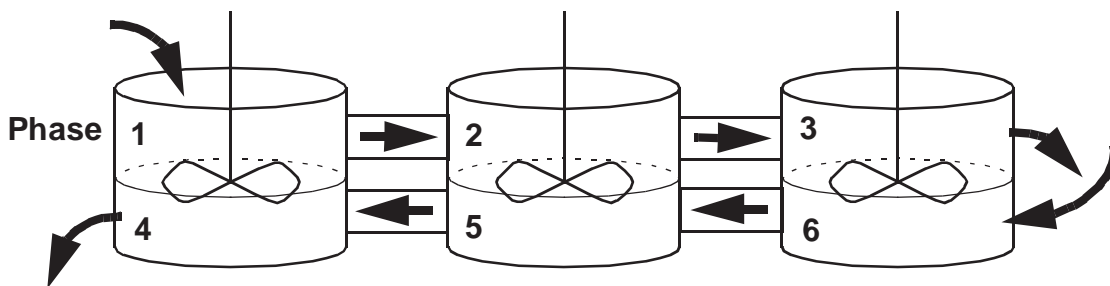


Abbildung 117: Rührkesselkaskade aus drei Kesseln im Gegenstrom.

Die Ausgangsmaterialien werden der Phase 1 und möglicherweise der Phase 6 zugeteilt. Die Phasenflüsse gehen von Phase 1 in Phase 2, von 2 in 3, von 6 in 5 und von 5 in 4, wobei die

Substanzen der Phase 3 (oder 4) mit ihren Konzentrationen als Ergebnis der Gegenstromkaskade ausgegeben werden. Phasenübergänge werden zwischen den Phasen 1 und 4, 2 und 5 sowie 3 und 6 erzeugt.

6.7.4.8. (Pseudo-)Monomolekularer Zerfall

Unter dem Begriff (pseudo-)monomolekularer Zerfall werden Reaktionsnetzwerke zusammengefaßt, die aus Reaktionen erster und pseudoerster Ordnung aufgebaut sind. Dazu gehört auch der monomolekulare Zerfall durch die Fragmentierungen und Umlagerungen einer Verbindung nach einer Elektronenstoßionisation in einem Massenspektrometer. Bei Umlagerungen entstehen ein, bei Fragmentierungen zwei Aggregate als Produkte einer Reaktion, die getrennt voneinander weiterreagieren. Die gebildeten Neutralfragmente reagieren teilweise weiter zu anderen Neutralfragmenten. Da Neutralfragmente nicht detektiert werden können, sind diese Reaktionen sowohl für die Analyse als auch die Simulation von Massenspektren ohne Bedeutung, weshalb Reaktionen von Neutralfragmenten ausgeschlossen werden. Die neutralen Ausgangsverbindungen werden in der ersten Reaktionsebene vor den Fragmentierungen und Umlagerungen zunächst ionisiert.

Ein weiteres Einsatzgebiet für den Phasenmodus „monomolekular“ liegt bei Reaktionen pseudoerster Ordnung vor, die beim Abbau von Chemikalien in der Umwelt Anwendung finden. Die Reaktanden Wasser und Wasserstoff, der formal für Reduktionen verwendet wird, liegen im großen Überschuß vor. Für die Reaktionen werden dann nur die Umweltchemikalien bzw. die schon gebildeten Metaboliten verwendet. Die Reaktanden Wasser oder Wasserstoff werden während der Reaktion in das Reaktionsensemble eingeschleust. So erreicht man eine Geschwindigkeitssteigerung bei der Simulation des Abbaus, nimmt aber in Kauf, daß die erzeugten Reaktionen stöchiometrisch nicht korrekt sind, da das Wasser bzw. der Wasserstoff auf der Eduktseite fehlen. Für eine korrekte Symmetriezahl, die Zahl der Anordnungsmöglichkeiten der Edukte, bei der dieselbe Reaktion erfolgt, muß die von EROS7 berechnete Zahl noch mit der Symmetriezahl der Teilreaktionssubstruktur im während der Reaktion eingeführten Molekül multipliziert werden. Für die Hydrolyse und Reduktion mit Wasserstoff wird die Symmetriezahl verdoppelt. Die Geschwindigkeitskonstante für die Reaktion ist pseudoerster Ordnung mit

$$k' = k \cdot c(\text{H}_2\text{O}) \quad \text{bzw.} \quad k' = k \cdot c(\text{H}_2) \quad (6)$$

Will man stöchiometrisch richtige Reaktionen und nicht nur die Metaboliten mit ihren Konzentrationsverläufen modellieren, nimmt man einen Rührkessel und benötigt neben den Umweltchemikalien auch Wasser und Wasserstoff als Ausgangsmaterialien. Wasser und Wasserstoff müssen nicht von der Datei mit den Edukten eingelesen werden, sondern können auch bei der Vorbehandlung der Ausgangsmaterialien eingeschleust werden. Man will gewöhnlich allerdings nur wissen, wie die Umweltchemikalie abgebaut wird. Hat man Eigenschaften des Bodens, in dem die Chemikalie abgebaut wird, und berücksichtigt sie, dann stehen sie in den Regeln und dazu gehört auch, daß der Boden wasserhaltig ist und Reduktionsäquivalente enthält.

EROS7 kennt aber nicht nur einzelne Moleküle, sondern auch Aggregate aus mehreren Molekülen, wie die Ionen eines Salzes oder die Proteinstränge eines Proteins mit Quartärstruktur. Da sich die in einem Aggregat zusammengehaltenen Moleküle nicht unabhängig voneinander bewegen können, sind auch Reaktionen, die von einem Aggregat ausgehen, Reaktionen erster Ordnung und verhalten sich damit genauso wie monomolekulare Reaktionen. Dies ist der Grund, weshalb als Edukte für den (pseudo-)monomolekularen Zerfall neben Aggregaten mit nur einem Molekül auch Aggregate aus mehreren Molekülen zugelassen sind.

In all diesen Fällen verwendet man einen Reaktor mit einer Phase im Modus „monomolekular“.

6.7.4.9. Laminares Strömungsrohr

Der Modus des „laminaren Strömungsrohrs“ ist der Modus von EROS5, bei dem die Produkte einer Reaktion zusammengehalten werden und ausgehend von ihnen die weiteren Reaktionen vorgenommen werden. In diesem Modus werden keine Aggregate neu miteinander kombiniert. Da während der Reaktionen auch beispielsweise kleine Moleküle, wie Wasser, hinzugefügt werden können, eignet sich dieser Modus sowohl für die Reaktionsvorhersage in verdünnten Lösungen als auch für die Syntheseplanung, wenn die Reaktionstypen retrosynthetischen Schritten entsprechen.

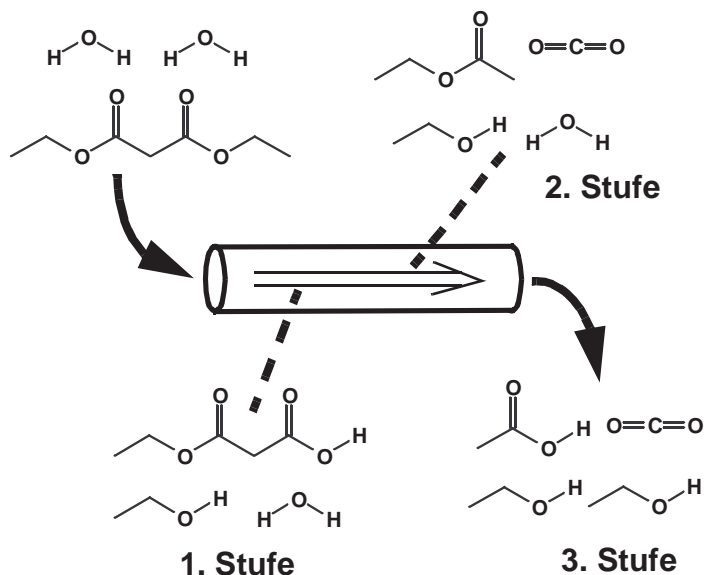


Abbildung 118: Laminares Strömungsrohr.

Da in diesem Modus die Kinetik nicht angewendet werden kann, sind die Ergebnisse rein qualitativ. Wie [9] zeigt, kann diese Art der Reaktionsgenerierung für die Auffindung der möglichen Produkte und die Syntheseplanung in vielen Fällen erfolgreich eingesetzt werden.

6.7.4.10. Kombinatorische Chemie

Eine weitere Anwendungsmöglichkeit, bei der kinetische Berechnungen keine Rolle spielen, ist der Einsatz von EROS7 in der kombinatorischen Chemie. In der kombinatorischen Chemie ist man daran interessiert, den vollständigen Satz aller Produkte zu erzeugen, der bei den Reaktionen einer definierten Menge ein- und/oder multifunktionaler Moleküle bei vollständigem Umsatz entstehen. In diesem Zusammenhang erhalten die Phasen eines Reaktors eine neue Bedeutung. Sie stellen hier die einzelnen Reaktionsebenen und Substanzgruppen dar. Dies wird an der Esterbildung aus Säurechloriden und Alkoholen gezeigt.

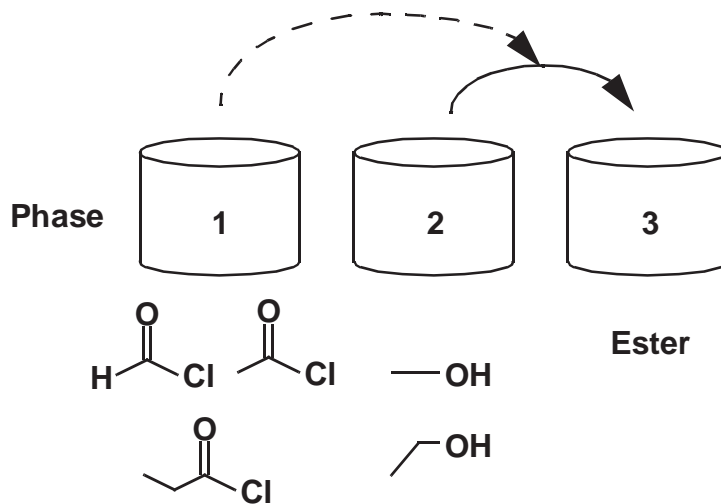


Abbildung 119: Phasen für die Bildung aller möglichen Ester.

Die Phasen eins und drei sind inert und die Phase zwei erzeugt Grenzflächenreaktionen. Durch den Kontakt der Phase zwei zur Phase eins werden je ein Alkohol und ein Säurechlorid kombiniert und zur Reaktion gebracht, wobei die Produkte der Reaktion der Phase drei zugeteilt werden. In der Phase drei erhält man so neben der gebildeten Salzsäure alle Esterkombinationen (siehe Abbildung 120).

<chem>COC=O</chem>	<chem>HCl</chem>	<chem>CCOC=O</chem>	<chem>CC(=O)OC</chem>
<chem>CCOC=O</chem>	<chem>CCOC=O</chem>	<chem>CCCOC=O</chem>	

Abbildung 120: Esterkombinationen.

Aber auch die möglichen Produkte von Reaktionen, deren Verbindungen mehrere funktionelle Gruppen besitzen, können mit EROS7 bestimmt werden. Verestert man Glycerin gleichzeitig mit drei verschiedenen Säurechloriden, benötigt man einen Reaktor mit fünf Phasen. In die erste kommen zu Beginn die Säurechloride und in die zweite das Glycerin. Man definiert nun die Phasen so, daß die erste und letzte Phase inert sind und alle anderen Phasen Grenzflächenreaktionen erzeugen und Kontakt zur ersten Phase besitzen. Die ersten Veresterungsreaktionen werden so von der Phase zwei erzeugt, indem sie das eigene Aggregat Glycerin mit allen Aggregaten aus der Phase eins reagieren läßt. Setzt die Reaktion die Phase für die Produkte immer um eins höher als die Nummer der Phase, die die Reaktion generiert, kommen alle Monoacylglyceride in die Phase drei, die Diacylglyceride in die Phase vier und alle möglichen Triacylglyceride befinden sich schließlich nach einer Reaktionsebene für alle Phasen in der Phase fünf.

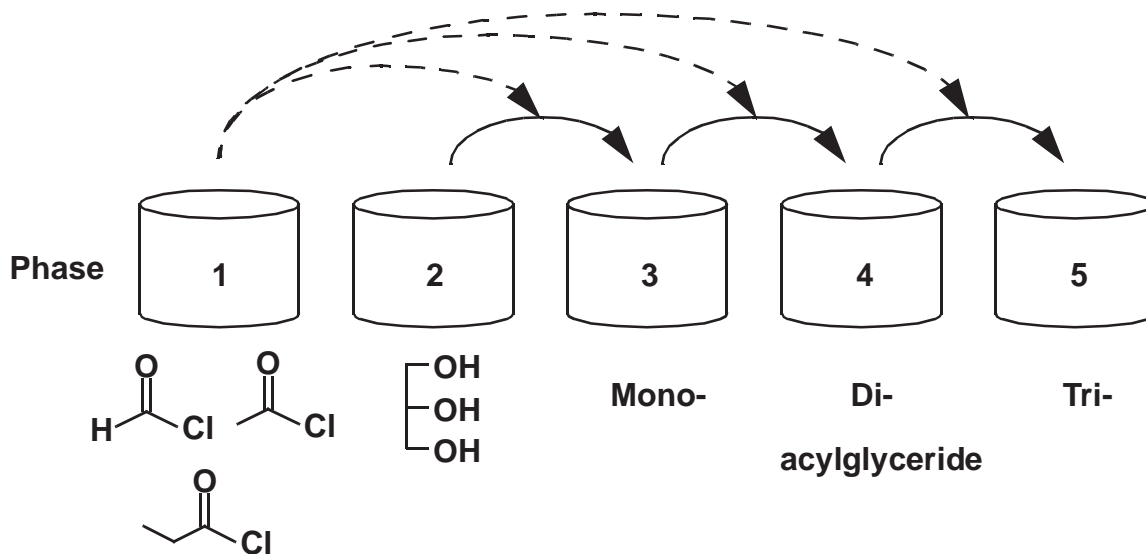


Abbildung 121: Phasen für die Bildung aller möglicher Triacylglyceride.

Neben den 18 Triacylglyceriden, die unter Vernachlässigung der Stereochemie gebildet wurden, ist auch die bei den Reaktionen entstehende Salzsäure in den Phasen drei bis fünf enthalten.

<chem>HCOOCH2CH(OOCC)OCHO</chem>	<chem>H-Cl</chem>	<chem>CH3COOCH2CH(OOCC)OCHO</chem>	<chem>HCOOCH2CH(OOCC)OCC(=O)CH3</chem>	<chem>HCOOCH2CH(OOCC)OCC(=O)CH2CH3</chem>
<chem>HCOOCH2CH(OOCC)OCC(=O)CH3</chem>	<chem>CH3COOCH2CH(OOCC)OCHO</chem>	<chem>CH3COOCH2CH(OOCC)OCC(=O)CH3</chem>	<chem>HCOOCH2CH(OOCC)OCC(=O)CH2CH3</chem>	<chem>CH3COOCH2CH(OOCC)OCC(=O)CH2CH3</chem>
<chem>CH3COOCH2CH(OOCC)OCHO</chem>	<chem>CH3COOCH2CH(OOCC)OCC(=O)CH2CH3</chem>	<chem>CH3COOCH2CH(OOCC)OCC(=O)CH2CH3</chem>	<chem>CH3COOCH2CH(OOCC)OCC(=O)CH2CH3</chem>	<chem>CH3COOCH2CH(OOCC)OCC(=O)CH2CH3</chem>
<chem>CH3COOCH2CH(OOCC)OCC(=O)CH2CH3</chem>	<chem>CH3COOCH2CH(OOCC)OCC(=O)CH2CH3</chem>	<chem>CH3COOCH2CH(OOCC)OCC(=O)CH2CH3</chem>	<chem>CH3COOCH2CH(OOCC)OCC(=O)CH2CH3</chem>	

Abbildung 122: Gebildete Triacylglyceride inklusive Salzsäure.

Da man an der Salzsäure in der Regel kein Interesse hat und sie möglicherweise für die weitere Verarbeitung gar entfernen müsste, kann die gebildete Salzsäure während der Reaktionsgenerierung aus dem Produktensemble gelöscht werden. Auch in diesem Fall sind die Reaktionen stöchiometrisch nicht einwandfrei, da die Salzsäure auf der Produktseite fehlt.

6.7.4.11. Weitere Aufbauschemata für Reaktoren

Bei der Verschaltung der einzelnen Phasen in den verschiedenen Modi ist der Phantasie wie bei den technischen Reaktoren keine Grenze gesetzt. So können auch viele andere Reaktortypen aufgebaut werden.

6.8. Steuerung der Reaktionsfolge

Wie schon in 6.5.5 angesprochen, kann mit Hilfe der genealogischen Variablen gesteuert werden, welche Reaktionstypen nach einem bestimmten Reaktionstyp folgen dürfen. In der Massenspektroskopie stellt die radikalische Wasserstoffumlagerung durch die Erzeugung einer Fülle von Reaktionen und damit einer großen Zahl von Ionen ein gewisses Problem dar. Schließt man induktive Fragmentierungen nach einer radikalischen Wasserstoffumlagerung aus, verliert man fast keine Information, da die Ionen, die eine Wasserstoffumlagerung eingehen, sich in einem angeregten Zustand befinden und durch ihre Anregung und das radikalische Zentrum mit großer Wahrscheinlichkeit nach einer oder mehreren Umlagerungen eine α -Spaltung eingehen. Dazu verwendet man eine genealogische Variable für die Aggregate vom Typ *int* und initialisiert sie während der Vorbehandlung der Ausgangsmaterialien beispielsweise mit null. Für das Reaktionsprodukt aller Wasserstoffumlagerungen setzt man sie dann auf eins und schließt alle Reaktionstypen außer der Wasserstoffumlagerung und der α -Spaltung aus, wenn diese genealogische Variable für das Edukt der Reaktion auf eins gesetzt ist. Eine α -Spaltungsreaktion setzt die genealogische Variable wieder zurück auf null.

In gleicher Weise kann man auch abschnittsweise bestimmte Reaktionstypen vorschreiben. So kann man festlegen, daß induktive Spaltungen erst durchgeführt werden, nachdem alle möglichen radikalischen Reaktionen stattgefunden haben und ab dann keine anderen radikalischen Reaktionen mehr durchgeführt werden können, auch wenn etwa durch eine Abspaltung eines Halogenatoms wieder ein Radikalzentrum in das Kation eingeführt wird. Dazu erzeugt man eine weitere genealogische Variable für die Aggregate, initialisiert sie mit null und setzt sie für das Produktkation auf eins, wenn es kein Radikalzentrum mehr enthält. Werden nun alle radikalischen Reaktionstypen, wie die radikalische Wasserstoffumlagerung oder die α -Spaltung, unterdrückt, wenn diese Variable gesetzt ist, werden keine Radikalreaktionen mehr durchgeführt, wenn ein Kation auf seinem Entstehungsweg einmal kein Radikalzentrum mehr enthalten hat.

Eine andere Möglichkeit ist, die Anregungsenergie der Kationen zu bestimmen und abhängig davon Reaktionen mit höheren Aktivierungsenergien auszuschließen. Daneben sind auch die Geschwindigkeitskonstanten nicht unabhängig von der Anregungsenergie des Kations. Betrachten wir hierzu die beiden Reaktionsfolgen in Abbildung 123.

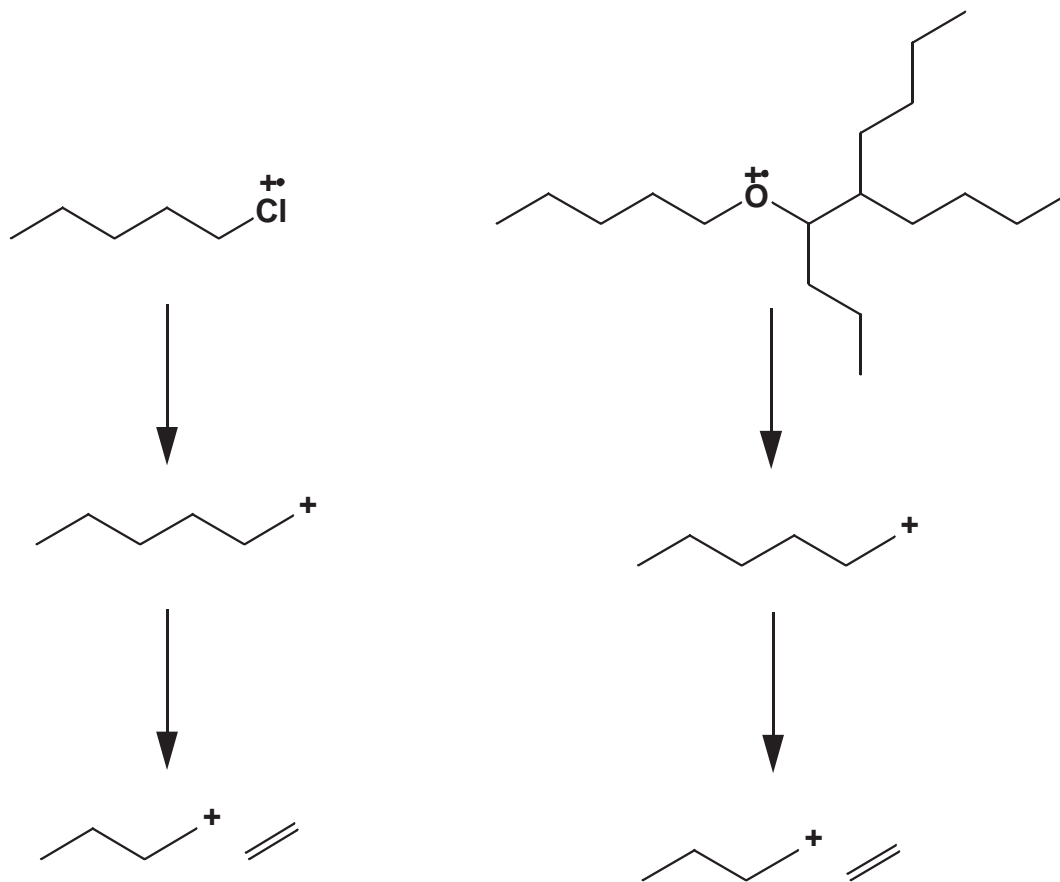


Abbildung 123: Die Reaktivitäten der beiden Folgereaktionen sind unterschiedlich, da die Anregungsenergien der Edukte der Folgereaktion verschieden sind.

Die 1-Hexylkationen haben eine unterschiedliche Anregungsenergie, da die Neutralfragmente, die im ersten Reaktionsschritt abgespalten werden, durch ihre unterschiedliche Zahl an Atomen und damit an Freiheitsgraden einen verschiedenen Teil der anfänglich gleichen Anregungsenergie erhalten. Erhalten die abgespaltenen Neutralfragmente unterschiedliche Energien, bleibt in beiden Fällen auch eine verschiedene Anregungsenergie für die 1-Hexylkationen übrig. Dies gilt nicht nur für die Abspaltung unterschiedlich großer, sondern auch für unterschiedlich schwere Neutralfragmente. In den Massenspektren von 1-Hexylchlorid (siehe Abbildung 124) und 1-Hexyliodid (siehe Abbildung 125) ist im Massenbereich von 25 bis 85 Masseneinheiten (m/z) deutlich zu erkennen, daß die Weiterfragmentierung des 1-Hexylkations im Falle des 1-Hexyliodids durch seine geringere Anregungsenergie schwächer ist. Das einzige Ion, das ein Signal mit einer großen Intensität besitzt, den Basepeak, ist das Isopropylkation (m/z 43).

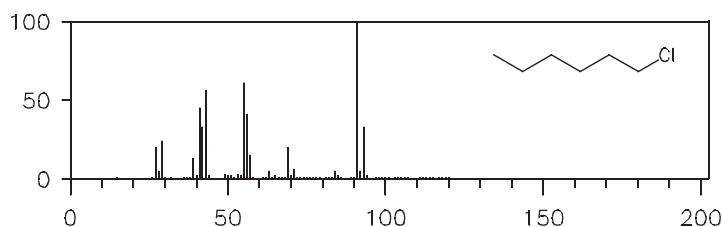


Abbildung 124: Massenspektrum von 1-Hexylchlorid.

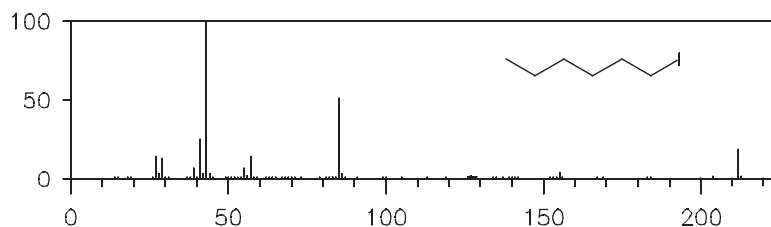


Abbildung 125: Massenspektrum von 1-Hexyliodid.

Ebenso, wie die Anregungsenergie der Kationen, kann die Zahl der radikalischen Wasserstoffumlagerungen auf eine funktionelle Gruppe zur Beschränkung der Reaktionsgenerierung herangezogen werden, wie es in Abschnitt 4.2.2.4 beschrieben ist.

Wurden alle Atome eines Aromaten in der Ausgangsverbindung als zum aromatischen System gehörend markiert, kann so verhindert werden, daß aromatische Ringe fragmentiert werden, bevor alle Seitenketten abgespalten wurden. Die Möglichkeit der Steuerung der Reaktionsfolge durch genealogische Variablen ist dabei nicht auf den Schutz aromatischer Systeme beschränkt.

6.9. Kinetik

Möchte man zusätzlich zur Generierung der Reaktionen auch die Konzentrationen der Edukte und Produkte berechnen, wird von den Regeln eine Variable vom Typ *double* unter dem Namen *reactivity* an das Kernsystem von EROS7 übergeben, die für die kinetischen Berechnungen verwendet wird. Ist eine Reaktion abgeschlossen, steht in ihr die Geschwindigkeitskonstante bzw. die Reaktionswahrscheinlichkeit. Man kann zwischen zwei verschiedenen Arten der kinetischer Berechnungen wählen. Mit den Werten der Reaktivität werden automatisch die Differentialgleichungen bzw. die Gleichungen mit den Reaktionswahrscheinlichkeiten für die Bildung und den Abbau der Verbindungen aufgestellt. Über die Integration dieser bzw. Lösung der Gleichungen für die Reaktionswahrscheinlichkeiten werden die Konzentrationen der Substanzen bestimmt. Für den vollen Umfang der kinetischen, mit EROS7 möglichen Berechnungen benötigt man für alle Reaktionen die Geschwindigkeitskonstanten. Stehen für die Bewertung der Reaktionen nur Reaktionswahrscheinlichkeiten zur Verfügung, können nur noch die Endkonzentrationen der Aggregate bestimmt werden. Für beide möglichen Kinetikmodelle wird dabei angenommen, daß das Volumen und die Temperatur der Reaktionsmischung sich nicht ändern. Zusätzlich können auch Vorgänge simuliert werden, bei denen die verschiedenen Produkte unterschiedliche Temperaturen besitzen, ihre Anregungsenergie aber nicht an andere Moleküle übertragen. Dies trifft beispielsweise auf die Moleküle in der Ionisationskammer eines Massenspektrometers zu, da hier ein Stoß mit einem zweiten Teilchen im betrachteten Zeitraum nahezu abgeschlossen ist.

6.9.1. Kinetik mit Geschwindigkeitskonstanten

Hat man für alle Reaktionen eine Geschwindigkeitskonstante, kann man eine der drei verfügbaren Methoden zur Integration der Differentialgleichungen wählen: Gear-Algorithmus [58], Methode nach Runge-Kutta oder nach Runge-Kutta-Merson [59][60]. Der Gear-Algorithmus ist deutlich langsamer, aber robuster (siehe 7.1.2.1 und 7.1.3.1). Wird eine dieser Methoden gewählt, werden gleichzeitig mit der Erzeugung der Reaktionen auch die Differentialgleichungen für das Reaktionsnetzwerk aufgestellt. Ist das Aggregat A ein Edukt, gilt:

$$\frac{dc_A}{dt} = -U \quad (7)$$

Mit dem Umsatz U pro Volumen

$$U = z \cdot k \cdot \prod_{i = \text{Edukt}} c_i \quad (8)$$

wobei z die Symmetriezahl und k die Geschwindigkeitskonstante der Reaktion sind. Handelt es sich bei der Reaktion um einen Phasenübergang, gibt die Geschwindigkeitskonstante die Diffusion in die andere Phase an.

$$\frac{dn_i}{dt} = \frac{D_i \cdot A}{d} \cdot c_i \quad (9)$$

Ausgehend vom Stofftransport dn_i/dt für die Substanz i (siehe Formel 9) wird die Geschwindigkeitskonstante k berechnet (siehe Formel 10). Um die normalen Reaktionen und die Phasenübergänge gleich behandeln zu können, wird auch für die Phasenübergänge die Konzentrationsänderung $dc_{i,1}/dt$ angegeben (V_1 ist das Volumen der Phase, aus der die Substanz kommt, D_i die Diffusionskonstante der Verbindung i , A die Größe der Phasengrenzfläche, d die Dicke der laminaren Grenzschicht und z die Symmetriezahl der Reaktion.):

$$\frac{dc_{i,1}}{dt} = \frac{1}{V_1} \cdot \frac{dn_i}{dt} = k \cdot c_i \quad \text{mit} \quad k = \frac{D_i \cdot A}{d \cdot V_1} \quad \text{und} \quad z = 1 \quad (10)$$

Für die Diffusion durch beide Grenzschichten ergibt sich mit dem Verteilungskoeffizienten $N=c_{i,2}/c_{i,1}$ für k :

$$k = \frac{1}{V_1} \cdot \frac{1}{\frac{d_1}{D_1 \cdot A} + \frac{d_2}{D_2 \cdot A \cdot N}} \quad (11)$$

Für die Modellierung der Diffusion zwischen zwei Phasen wird für jede der beiden Diffusionsrichtungen ein Phasenübergang generiert.

Für die Flüsse in die und aus den Phasen werden die Stofftransporte automatisch den Differentialgleichungen hinzugefügt.

Die Ergebnisse der Integration sind bei allen drei Methoden der Konzentrationsverlauf der Aggregate in den Phasen eines Reaktors, das Integral der Konzentration über die gesamte Integrationszeit (Persistenz) sowie die Auf- und Abbauraten erster Ordnung der Verbindungen. Die Abbaurrate k_d erhält man, wenn man für eine Substanz nur deren Abbaureaktionen betrachtet und den Quotienten aus ihrer Konzentrationsänderung und der Anfangskonzentration berechnet (siehe Formel 12). Für k_d ergibt sich dann bei Reaktionen erster Ordnung die Summe der Produkte von Symmetriefaktor und Geschwindigkeitskonstante erster bzw. pseudoerster Ordnung.

$$k_{d,i} = \frac{dc_i}{c_i} \quad \text{mit} \quad c_j = 0 \quad \text{für} \quad j \neq i \quad (12)$$

Etwas schwieriger gestaltet sich die Aufbaurrate erster Ordnung k_p . Sie ist bei der Betrachtung des Konzentrationsverlaufs der Verbindung die maximale Konzentrationssteigerung in Bezug auf das Produkt der Konzentrationen der Ausgangsmaterialien (siehe Formel 13). Hat man, wie bei der Simulation des Abbaus von Umweltchemikalien, nur ein Aggregat als Ausgangsmaterial, ist die Aufbaurrate der Quotient der maximalen Konzentrationssteigerung und der Anfangskonzentration der Umweltchemikalie.

$$k_{p,i} = \frac{\max\left(\frac{dc_i}{dt}\right)}{\prod_{j = \text{Startmaterialien}} c_{j,0}} \quad (13)$$

6.9.1.1. Ausschluß unbedeutender Reaktionen

Ist eine Kinetik angeschaltet, wird vor der Durchführung der Reaktionen eines oder mehrerer Edukte zunächst überprüft, ob ihre Konzentrationen einen festgelegten Schwellwert überschreiten. Um dies testen zu können, werden nach jeder Reaktionsebene die Konzentrationen aller Verbindungen berechnet. Die Konzentrationen werden so zu einem Zeitpunkt bestimmt, zu dem im Reaktionsnetzwerk noch keine Abbaureaktionen der Substanz enthalten sind. Die Konzentration, die zur Überprüfung herangezogen wird, ist also die maximal mögliche innerhalb der betrachteten Reaktionszeit. Um unbedeutende Reaktionen auszuschließen, werden nur für solche Edukte Reaktionen generiert, die die minimale Konzentration überschreiten.

6.9.1.2. Bestimmung der Integrationszeit

Für die Begrenzung der Integrationszeit, also die betrachtete Zeit, gibt es zwei Methoden: ist 1) die voreingestellte, maximale Reaktionszeit erreicht oder 2) eines der Ausgangsmaterialien zum eingestellten Prozentsatz umgesetzt ist. Da alle eingebauten Integrationsverfahren mit gleichmäßigen Intervallen arbeiten, muß vor der Integration die Reaktionszeit abgeschätzt werden, wenn sie durch den Abbau eines der Ausgangsmaterialien begrenzt ist. Deshalb ist die Genauigkeit, mit der die Reaktionszeit bis zum gewünschten Umsatz erreicht wird, von der Abschätzung der Reaktionszeit abhängig. Die Güte ist für die verschiedenen Methoden unterschiedlich.

6.9.1.3. Vergleich der Integrationsmethoden

Die Genauigkeit, mit der bei der Integration die Zeit getroffen wird, wenn sie durch den Abbau eines Ausgangsmaterials zu einem vorgegebenen Prozentsatz begrenzt ist, ist bei der Methode nach Runge-Kutta vierter Ordnung am größten. Der Gear-Algorithmus und die Methode nach Runge-Kutta-Merson erzielen eine leicht geringere Genauigkeit. Für alle drei Methoden sind aber die Abweichungen hinsichtlich des Integrationsabbruchs, wie auch der bestimmten Konzentrationsverläufe, viel kleiner als die, welche durch die Ungenauigkeit der angegebenen Geschwindigkeitskonstanten hervorgerufen werden. Normalerweise verwendet man eine Funktion, die für die verschiedenen erzeugten Reaktionen die Geschwindigkeitskonstanten liefert. Da diese oft mit einer multilinenen Regressionsanalyse oder ähnlichen, mittelnden Methoden gewonnen werden, sind die so bestimmten Werte auch mit Fehlern für einzelne Reaktionen behaftet.

Daß die beiden Methoden Runge-Kutta und Runge-Kutta-Merson neben Differentialgleichungen erster Ordnung auch Gleichungen zweiter Ordnung lösen können, spielt keine Rolle, da bei EROS7 nur Gleichungen erster Ordnung auftreten. (Gleichungen wie Formel 14 können nicht Bestandteil der Differentialgleichungen sein.)

$$\frac{d^2 c}{dt^2} + \dots \cdot \frac{dc}{dt} = \dots \cdot c \quad (14)$$

Die Zeiten für die Lösung der Differentialgleichungen sind in der Praxis von erheblicher Bedeutung. In dieser Beziehung wird die Methode nach Runge-Kutta von den beiden anderen mit einer Berechnungszeit von nur einem Drittel deutlich geschlagen, wobei die Methode nach Runge-Kutta-Merson die schnellste ist. Dies wurde am Abbau von verschiedenen s-Triazinen bestimmt [61].

6.9.1.4. Ergebnisse am Beispiel vom Abbau des Herbizids Atrazin

Mit den Regeln zum Abbau von s-Triazinen, der Hydrolyse und der reduktiven Dealkylierung, die formal mit Wasserstoff durchgeführt wird, kann der Abbau von Atrazin simuliert werden [62]. Im folgenden sind die Konzentrationsverläufe der wichtigsten Verbindungen bis zu einer Halbwertszeit in Abbildung 126 und bis zu einem Umsatz von 99.9% in Abbildung 127 gezeigt. Die Anfangskonzentration von Atrazin betrug dabei 0.1 mol/l, die von Wasser und Wasserstoff je 1 mol/l.

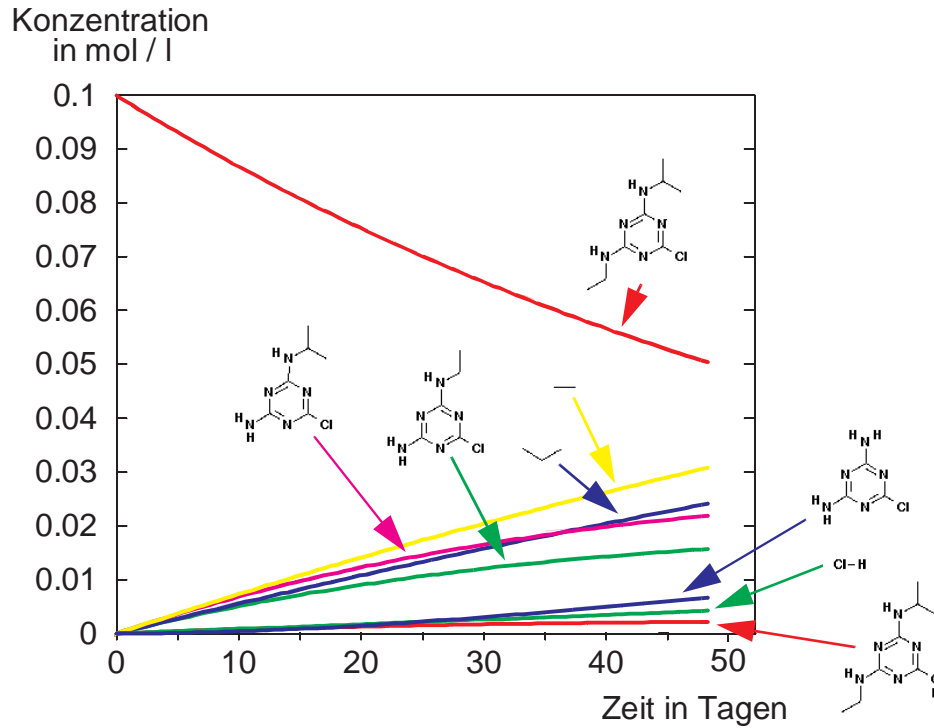


Abbildung 126: Metabolisierung von Atrazin innerhalb einer Halbwertszeit.

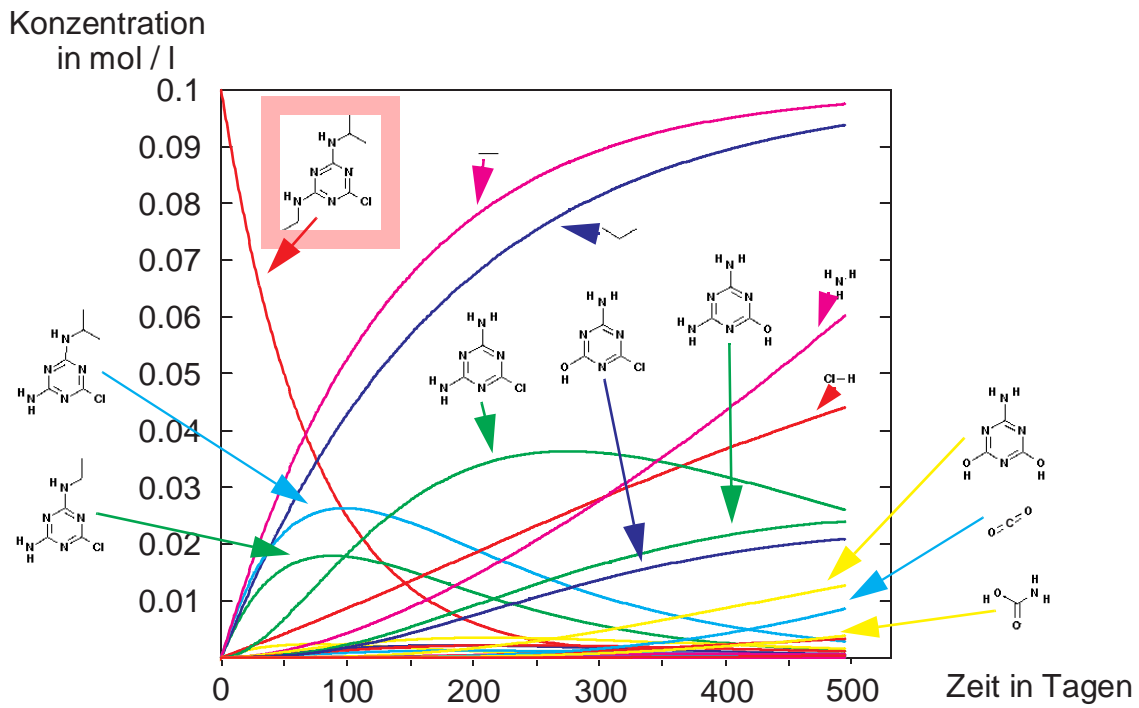


Abbildung 127: Konzentrationsverläufe von Atrazin und seinen Metaboliten über den Zeitraum von 480 Tagen, bis 99.9% des Atrazins metabolisiert sind.

Abbildung 128 zeigt neben dem Konzentrationsverlauf eines Metaboliten auch die Werte für die Persistenz, das Integral der Konzentration über die betrachtete Zeit, und die Auf- und Abbauraten erster Ordnung.

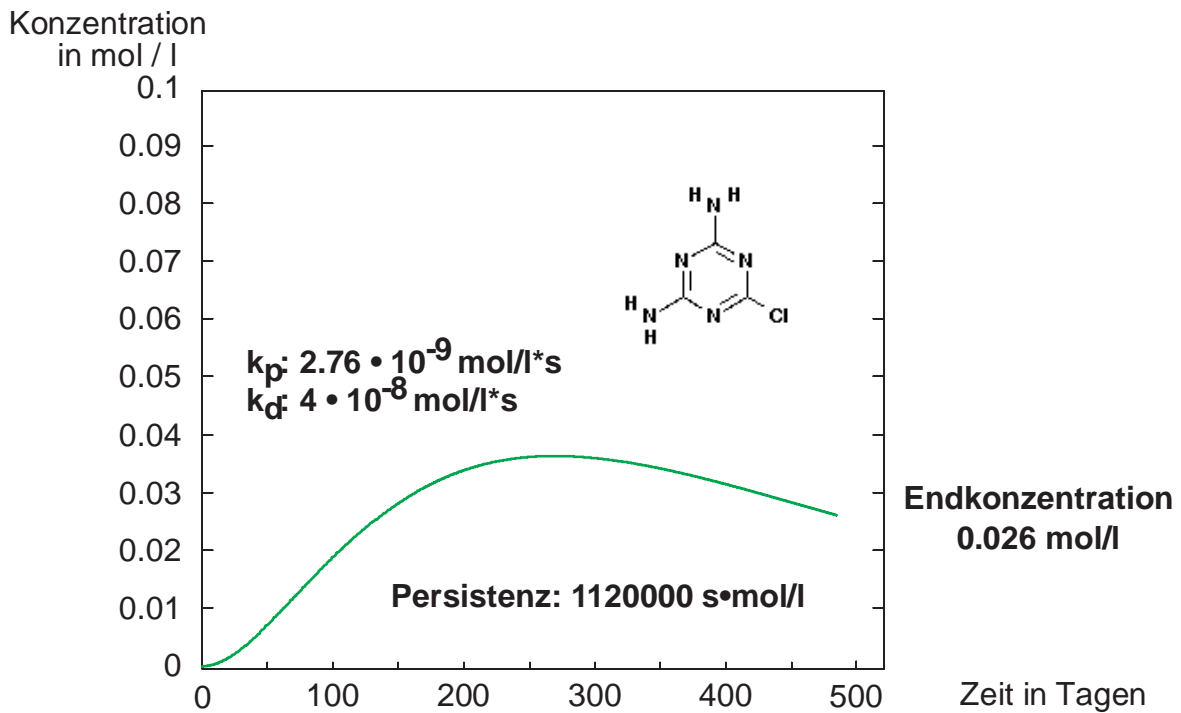


Abbildung 128: Werte einer Verbindung beim Abbau von Atrazin bis zu 99.9%. Die Persistenz ist das Integral des gezeigten Konzentrationsverlaufs über die gesamte dargestellte Zeit.

6.9.1.5. Michaelis-Menten-Kinetik und Reaktionen nullter Ordnung

Für enzymkatalysierte Reaktionen kann statt der Reaktionen erster, pseudoerster und zweiter Ordnung auch eine Michaelis-Menten-Kinetik für einzelne Reaktionstypen angenommen werden. Dies wird dadurch erreicht, daß für diesen Reaktionstyp in den Regeln ein entsprechendes Attribut gesetzt wird und zusätzlich zur Reaktivität die Michaelis-Konstante an EROS7 übergeben wird (siehe Anhang B.3.1). Diese Kinetik ist eine Vereinfachung der Enzymreaktion mit vorgelagertem Gleichgewicht, wie es in Abbildung 129 gezeigt ist.



Abbildung 129: Teilschritte einer enzymatischen Reaktion: Gleichgewicht zur Bildung des Enzym-Substrat-Komplexes und die enzymkatalysierte Reaktion.

Mit Hilfe der Michaelis-Menten-Kinetik kann diese Reaktion vereinfacht gesehen und von EROS7 erzeugt werden (siehe Abbildung 130).



Abbildung 130: Vereinfachte Enzymreaktion.

Für diese vereinfachte Reaktion gilt folgende Bildungsgeschwindigkeit von P [53]:

$$\frac{dP}{dt} = \frac{k_2 \cdot [E]_0 \cdot [S]}{K_m + [S]} \quad \text{mit} \quad K_m = \frac{k_2 + k_{-1}}{k_1} \quad (15)$$

An EROS7 werden dabei das Produkt $k_2 \cdot [E]_0$ als Reaktivität und K_m übergeben. Befindet sich die Reaktion ständig im Sättigungsbereich des Enzyms, kann über die Attribute des Reaktionstyps auch bestimmt werden, daß es sich um eine Reaktion nullter Ordnung handelt, wobei die Geschwindigkeitskonstante nullter Ordnung als Reaktivität übergeben wird.

6.9.2. Wahrscheinlichkeitskinetik

Hat man lediglich Reaktionswahrscheinlichkeiten, läßt sich als Ergebnis der Kinetik der Konzentrationsverlauf für die Verbindungen nicht berechnen und damit auch keine Auf- und Abbauraten oder die Persistenz. Mit dieser Kinetik werden nur die Endkonzentrationen der Substanzen ermittelt. In der von den Regeln übergebenen Variablen der Reaktivität wird in diesem Modus die Reaktionswahrscheinlichkeit im Bereich von null bis eins erwartet. Die Umsätze der konkurrierenden Reaktionen werden folgendermaßen berechnet:

$$U_i = \frac{\ln(1 - P_i)}{\ln\left(\prod_k (1 - P_k)^{z_k}\right)} \cdot z_i \cdot c_0 \cdot \left(1 - \prod_k (1 - P_k)^{z_k}\right) \quad (16)$$

U_i ist der Umsatz der Reaktion i , P_i die Wahrscheinlichkeit, z die Symmetriezahl und c_0 die Anfangskonzentration des Edukts. Dabei werden nacheinander alle Reaktionen i dieses Edukts betrachtet. k ist eine Laufvariable über alle Reaktionen dieses Edukts. Die Endkonzentration des Edukts ergibt sich als Differenz der Anfangskonzentration und der Summe aller Umsätze.

Auch in EROS7 ist eine Methode für die Behandlung von Umlagerungsreaktionen eingebaut. Sie unterscheidet sich von der in MASSIMO enthaltenen dadurch, daß die Variable *reactivity* den Prozentsatz des Edukts angibt, der mit dieser Reaktion umgelagert wird (z.B. 0.15 für 15%). Ergeben alle vom Edukt ausgehenden Umlagerungsreaktionen einen Umsatz von mehr als

100%, werden die Umsätze reskaliert, sodaß die Reaktionen 100% umsetzen. Ein Abbruchkriterium, daß Produkte im Umlagerungssystem nur einmal gebildet werden, gibt es nicht. Diese Methode hat die Vorzüge, daß sie immer eindeutig ist und auch näher an der Realität liegt (vergleiche 4.3.3.1). EROS7 wertet wie MASSIMO zur Berechnung der Umsätze alle Umlagerungen aus, bevor die Umsätze der Fragmentierungen bestimmt werden. Diese Berechnung mit den angegebenen Prozentsätzen wird dann durchgeführt, wenn für den Reaktionstyp das Attribut *rearrangement* angegeben wurde.

Diese Kinetik wird für die Simulation von Elektronenstoßmassenspektren verwendet, kann aber auch angewendet werden, wenn man lediglich Reaktionswahrscheinlichkeiten zur Verfügung hat, die beispielweise aus einer Reaktionsklassifizierung stammen. In diesem Fall empfiehlt es sich, Reaktionen mit einer zu geringen Wahrscheinlichkeit gänzlich zu verwerfen. Zu bedenken ist noch, daß die Implementation der Wahrscheinlichkeitskinetik nur für monomolekulare Reaktionen Werte liefern kann.

6.10. Einsatz von EROS7 zur Simulation massenspektroskopischer Prozesse

Mit EROS7 können neben der Vorhersage von Reaktionen auch Massenspektren simuliert werden. Dazu benötigt EROS7 Regeln, die die Fragmentierungen und Umlagerungen im Massenspektrometer, inklusive der Ionisationen, simulieren. Analog der Reaktionsvorhersage mit angeschalteter Kinetik berechnet EROS7 die Endkonzentrationen der einzelnen Kationen und Neutralfragmente. Zur Berechnung des Massenspektrums werden für alle Kationen die Isotopensignale bestimmt und mit der Konzentration des Fragments multipliziert. Hat das Fragment eine höhere Ladung als eins, werden die Massen der Signale durch die Ladung geteilt. Summiert man schließlich all diese so vorbereiteten Signale auf, erhält man das Massenspektrum, dessen Signale noch in der Weise skaliert werden, daß das größte Signal den Wert 1000 für ein Promillspektrum erhält. Dies wird bei EROS7 dadurch initiiert, daß beim Aufruf die entsprechende Option angegeben wird. Die Ausgangsverbindung wird dann zusammen mit dem vorhergesagten Massenspektrum in eine gesonderte Datei geschrieben.

6.10.1. Einstellungen in den Regeln, Unterschiede zu MASSIMO

Für die Simulation von Massenspektren verwendet man einen Reaktor mit einer Phase, die Reaktionen im „monomolekularen“ Modus erzeugt. In der Vorbehandlung der Edukte werden alle Neutralfragmente von der Weiterreaktion ausgeschlossen. Eine Ausnahme stellt hier die Ausgangsverbindung dar, da diese in einem ersten Schritt zunächst ionisiert werden muß. Aus diesem Grund werden in der ersten Reaktionsebene alle Edukte zugelassen. Zu welcher Reaktionsebene die aktuelle Reaktion gehört, steht in einer Variablen, die von EROS7 an die Regeln übergeben wird.

Die Regeln, die für die in 6.10.2 gezeigten simulierten Massenspektren verwendet wurden, basieren auf den gleichen neuronalen Netzen für die Bewertung, die auch mit MASSIMO eingesetzt wurden (siehe 4.2.5.2). Für die Berechnung der Konzentrationen wird, wie in MASSIMO, die gleiche Wahrscheinlichkeitskinetik verwendet, die sich allerdings bei der Behandlung der Umlagerungen unterscheidet. Wie in Kapitel 4.3.3.1 gesagt, gilt die Annahme nicht ganz, daß sich die Wasserstoffumlagerungen im Gleichgewicht befinden, die von MASSIMO für die Konzentrationsberechnungen für die Wasserstoffumlagerung eingesetzt wird. Deshalb wird für diese Reaktionen bei EROS7 statt der Gleichgewichtskonstante der Prozentsatz angegeben, der bei einer Symmetriezahl von eins von dieser Umlagerung umgesetzt wird. EROS7 berücksichtigt dabei auch, wenn das Produkt wieder zurücklagern kann und eine entsprechende Reaktion durchgeführt wurde.

Die Regeln enthalten sechs Reaktionstypen: die Ionisation von freien Elektronenpaaren und π -Systemen, denen kein freies Elektronenpaar benachbart ist (in Anlehnung an die Reaktionsgenerierung mit MASSIMO), die α -Spaltung, die Oniumreaktion, die Carbonyleliminierung, die McLafferty-Spaltung und die Wasserstoffumlagerung. Die Bewertungen wurden ebenso aus den Regeln von MASSIMO übernommen, wie die Reaktionstypen. Die Ionisation und die Wasserstoffumlagerung konnten nicht direkt übertragen werden und wurden deshalb an die Version von MASSIMO angelehnt. Ohne diese Anlehnung hätte man nicht mit einer sinnvollen Bewertung der Reaktionen rechnen können. Eine Besonderheit stellt dabei die Ionisation dar. Sie ist unabhängig von jeglichem Atom und hat damit kein Atom in der Definition der Reaktionsstruktur. Es gibt dennoch mehrere verschiedene Ionisationsmöglichkeiten in einem Molekül, nämlich so viele, wie es unterschiedliche, besetzte Elektronensysteme im Molekül gibt. Die Ionisation sucht diese, unter der oben angegebenen Beschränkung und führt sie durch. Für jede Reaktion wird zunächst das Ensemble eins, das Edukt, als aktives Ensemble gesetzt und von ihm ausgehend alle Reaktionen durchgeführt.

An den verwendeten Regeln zur Simulation von Massenspektren, die im Anhang F.2 gezeigt sind, sieht man neben der Variablenübergabe von EROS7 an die Regeln (Reaktionsebene) und der Erzeugung mehrerer Reaktionen aus einer gefundenen Reaktionsstruktur (Ionisationsregel) weitere Möglichkeiten der Programmierung der Regeln. So wird das radikalische Elektronensystem nur einmal für jedes Edukt gesucht und seine Nummer in einer statischen Variablen gespeichert. Bei der Überprüfung der Bedingungen am Reaktionszentrum und während der Reaktion selbst wird dann auf diese zugegriffen. Desweiteren setzen die Regeln Backpropagationnetzwerke zur Ermittlung der Reaktionswahrscheinlichkeit ein. Für das Edukt und das Produkt werden die Funktionen zum Setzen der Eingabewerte des Netzes aufgerufen, um schließlich beim Abholen des Ergebnisses das neuronale Netz auszuwerten. Hat man aufwendigere, klar umrissene Aufgaben, die für mehrere Reaktionstypen benötigt werden, oder, um die Regeln leichter lesen zu können, kann man sich für diese Aufgaben eigene Funktionen dafür schreiben. Dies geschah für die Dekonjugation von π -Systemen ohne richtigen Bindungsbruch, die Zählung von freien Elektronenpaaren an Atomen, die an einem bestimmten π -System beteiligt sind, die Abfrage von ganzzahligen Eigenschaften, die als Funktionswert zurückgegeben werden und damit manche Stelle der Regeln übersichtlicher gestalten, und die Berechnung des Ionisationspotentials. Die Berechnung des Ionisationspotentials nach [26][46], die in MASSIMO eingebaut war, ist nicht so leicht an die Ionisation aus Elektronensystemen anzupassen, weshalb die Funktion in den Regeln immer den Wert 10.0 eV zurückgibt. Mit dem Vorfaktor und der Kennzeichnung als Umlagerungsreaktion werden die Ionisationsprodukte proportional zur Zahl gleicher Elektronensysteme gebildet. Diese Regeln können also nur für Verbindungen angewendet werden, deren verschiedene Ionisationsorte sehr ähnliche Ionisationspotentiale besitzen. Überdies sind die Regeln so gestaltet, daß man über eine Option steuern kann, ob Wasserstoffumlagerungen stattfinden sollen oder nicht. Zur Aktivierung wird EROS7 mit dem entsprechenden Aufrufparameter mitgeteilt, daß es eine Variable vom Typ *int* unter dem Namen *huml* und dem Wert 1 an die Regeln übergeben soll. Wird diese Option beim Aufruf von EROS7 weggelassen, werden keine Wasserstoffumlagerungen durchgeführt, da dies in den Regeln so definiert ist.

Die Weise, in der EROS7 das Reaktionsnetzwerk aufbaut, unterscheidet sich von der von MASSIMO speziell für die Umlagerungen. MASSIMO hat diese für jedes Umlagerungssystem im ganzen vor den Fragmentierungen durchgeführt und bewertet. In EROS7 wird dies alles von der Kinetik übernommen und auch in Fällen, bei denen ein Reaktionsprodukt erneut über eine andere Reaktionssequenz gebildet wird, werden von EROS7 im Gegensatz zu MASSIMO die Reaktionen, die von diesem Kation ausgehen, nicht erneut generiert.

Da die Datenstruktur RICOS auch die Isotopenmarkierung von einzelnen Atomen kennt, ist es mit EROS7 möglich, für solche Verbindungen das Massenspektrum zu berechnen.

6.10.2. Simulation von Massenspektren

Wie in Kapitel 6.10.1 erwähnt, wurden für die folgenden simulierten Massenspektren die neuronalen Netze genommen, die für MASSIMO entwickelt wurden. Einige der Werte, die als Eingabewerte für die Netze genutzt werden, basieren auf der Strukturdarstellung als Bindungsliste und beinhalten damit auch die Probleme der Bindungslisten. Zu ihrer Berechnung wird die chemische Struktur von RICOS in eine Bindungsliste umgewandelt und die berechneten Werte werden an RICOS zur Speicherung zurückgegeben. Dies wurde so implementiert, daß mit der neuen Datenstruktur RICOS auch ältere Funktionen und neuronale Netze verwendet werden können und da für die meisten der physikochemischen Eigenschaften noch keine Berechnungsalgorithmen für die neu Strukturdarstellung bereitstehen.

Die simulierten Massenspektren von (1-Ethyl-1-methyl-butyl)-methyl-ether in Abbildung 131 und 3-Hexanon in Abbildung 133 wurden unter Ausschluß der Wasserstoffumlagerung berechnet. Die Absenkung der Abspaltung von Wasserstoffradikalen, wie sie bei der Verwendung der Wasserstoffumlagerung benötigt wird, war jedoch in den Regeln enthalten. Zum Vergleich zeigen Abbildung 132 und Abbildung 134 die von MASSIMO simulierten Spektren.

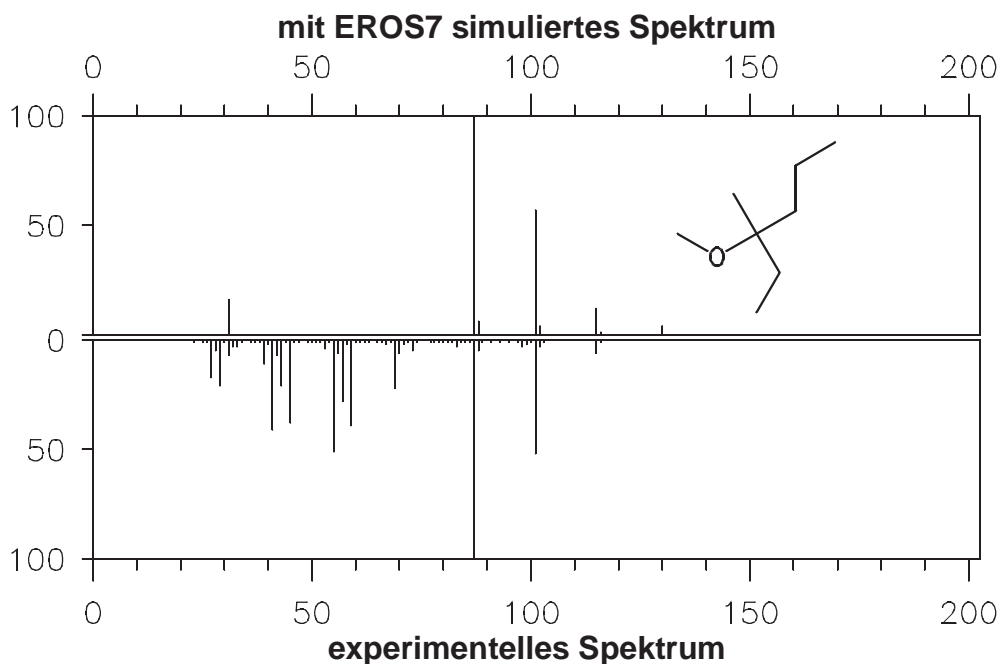


Abbildung 131: Spektrenvergleich von (1-Ethyl-1-methyl-butyl)-methyl-ether; experimentelles Spektrum aus [7].

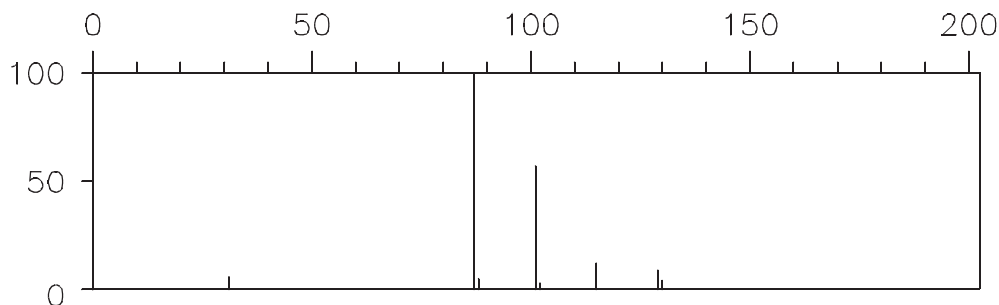


Abbildung 132: Von MASSIMO simuliertes Spektrum von (1-Ethyl-1-methyl-butyl)-methyl-ether.

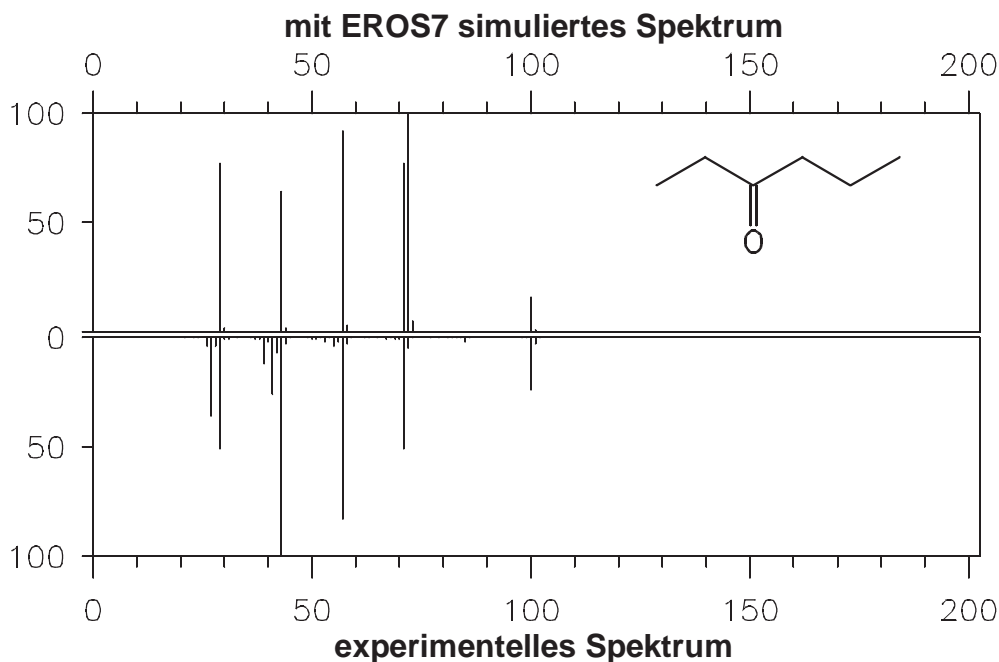


Abbildung 133: Spektrenvergleich von 3-Hexanon; experimentelles Spektrum aus [22].

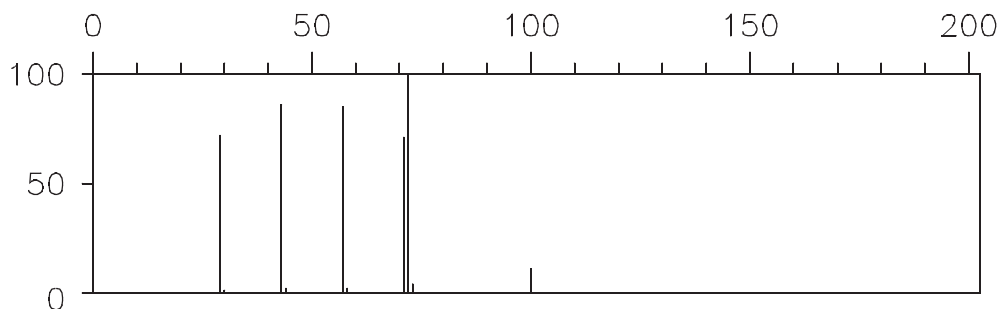


Abbildung 134: Von MASSIMO simuliertes Spektrum von 3-Hexanon.

Man sieht, daß auch EROS7 in der Lage ist, mit den Bewertungen der neuronalen Netze von MASSIMO den Teil der Massenspektren gut zu simulieren, deren Signale von Ionen hervorgerufen werden, die durch in den Regeln enthaltene Fragmentierungstypen gebildet werden. Im simulierten Spektrum von 3-Hexanon (siehe Abbildung 133) drehen sich die Intensitäten der Signale bei m/z 29 und 43 gegenüber der Simulation mit MASSIMO (siehe Abbildung 134) um. Da sowohl bei MASSIMO als auch bei EROS7 die neuronalen Netze identische Werte liefern, kann der Unterschied nicht daran liegen, daß nach der Umwandlung der RICOS-Datenstruktur in eine Bindungsliste, damit die physikochemische Eigenschaften berechnet werden können, EROS7 eine andere Struktur liefert als die, die MASSIMO generiert hat. Da die Reaktionswahrscheinlichkeiten für die beiden Carbonyleliminierungen (m/z 71→43 und 57→29) mit 0.454 und 0.453 auch fast gleich sind und auch die Symmetriezahlen in beiden Fällen eins sind, entspricht das Spektrum von EROS7 den Wahrscheinlichkeiten, obwohl das nicht ganz korrekt berechnete Spektrum von MASSIMO näher am experimentellen Massenspektrum liegt. Für die Wasserstoffumlagerungen wurden wegen der geänderten Kinetik für die Umlagerungen neue Reaktivitäten bestimmt. Abbildung 135 zeigt den Spektrenvergleich: simuliertes gegen das experimentelle Massenspektrum von Methylstearat. Zum Vergleich ist das mit MASSIMO simulierte Spektrum von Methylstearat in Abbildung 136 dargestellt.

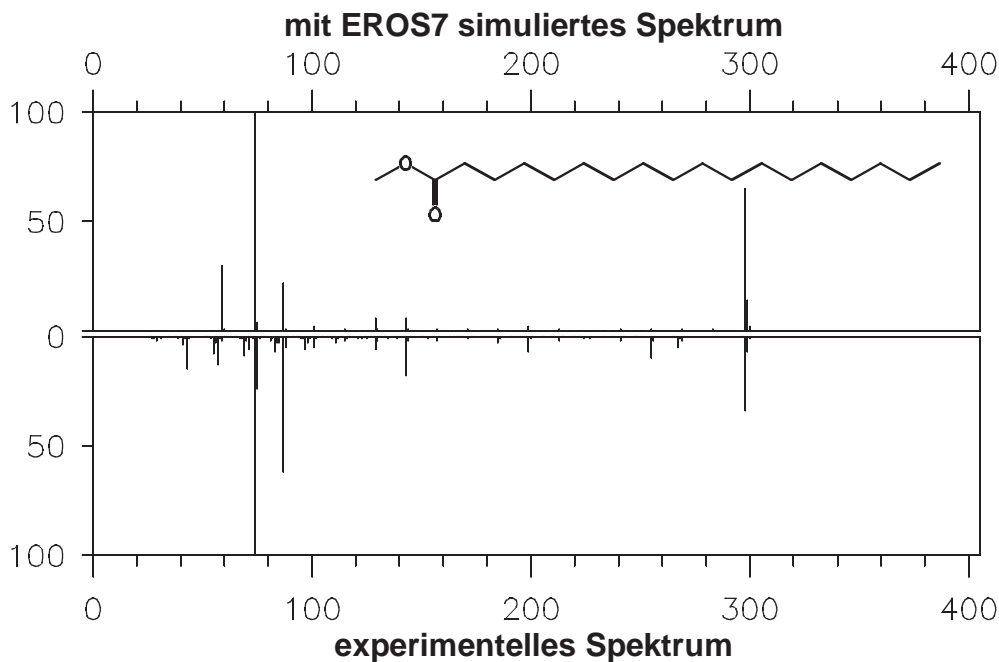


Abbildung 135: Spektrenvergleich von Methylstearat; experimentelles Spektrum: [22].

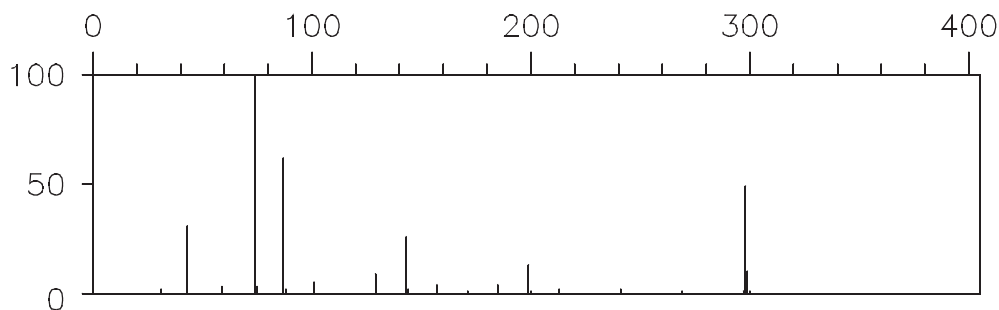


Abbildung 136: Von MASSIMO simuliertes Spektrum von Methylstearat.

Die von EROS7 simulierten Massenspektren entsprechen noch nicht ganz den Erwartungen, es konnte jedoch gezeigt werden, daß es auch mit dem auf der MO-orientierten Datenstruktur RICOS aufbauenden Programm EROS7 möglich ist, Massenspektren zu simulieren. Mit EROS7 hat man nun all die Möglichkeiten, die während der Erstellung der Regeln für MASSIMO vermißt wurden (siehe 4.5). Man kann nun darangehen, Fragmentierungs- und Umlagerungsregeln zu schreiben, die diese Möglichkeiten von RICOS und dem neuen Reaktionsgenerator von EROS7 nutzen. Bei der Wasserstoffumlagerung kann auch daran gedacht werden, diese deutlich zu vereinfachen. Bis jetzt wurden nacheinander alle möglichen Umlagerungen in den verschiedenen Entfernungen hin und zurück durchgeführt, was aufwendig und damit zeitraubend bei der Berechnung des Massenspektrums ist. Es wäre möglich, die nacheinander folgenden Umlagerungen zusammenzufassen und gleich die zusammengefaßte Wahrscheinlichkeit anzugeben. Dies ist möglich, da EROS7 aus einer gefundenen Reaktionssubstruktur auch mehrere Reaktionen erzeugen kann. So würde der Berechnungsaufwand und damit die Berechnungszeit für die Umlagerungen auf einen Bruchteil reduziert. Daneben hätte man auf diese Weise auch eine bessere Möglichkeit auf spezielle Situationen in der Folge der Umlagerungen einzugehen.

6.11. Implementationsstrategien und Anbindung der Regeln

Bei einer Umstellung des Reaktionsvorhersagesystems EROS6, das mit einer Darstellung der chemischen Strukturen als Bindungsliste arbeitet, auf eine MO-orientierte Datenstruktur hätte fast das gesamte System umgeschrieben werden müssen, da an vielen Stellen im Programm direkte Zugriffe auf die Datenstruktur enthalten waren. Deshalb wurde EROS7 auf der MO-orientierten Datenstruktur RICOS aufbauend von Grund auf neu geschrieben. Um in Zukunft nicht wieder in eine solche Situation zu kommen, wurde beschlossen, die einzelnen Teile des Systems durch Datenabstraktion voneinander zu trennen und nur Zugriffe über entsprechende Funktionen zuzulassen. Möchte man später beispielsweise erneut die Datenstruktur ändern, sind lediglich Zugriffsfunktionen nötig, die auf der neuen Datenstruktur die angeforderten Aktionen des Reaktionsgenerierungsteils ausführen. Die Reaktionsgenerierung selbst ist von der Umstellung dann nicht betroffen. Als Programmiersprache für das neue System hat sich C++ angeboten, da sie als objektorientierte Sprache die Datenabstraktion unterstützt. Desweiteren hat sie den Vorteil, daß ein sanfter Umstieg auf das neue System möglich ist. Es können Teile, wie die Berechnung der physikochemischen Parameter von EROS6, das in Fortran77 geschrieben ist, eingebunden werden. Darüber hinaus bietet C++ die Möglichkeit eine Reihe weiterer Programmiersprachen zu Programmsystem zu binden, da sie sowohl den objektorientierten als auch den algorithmischen Programmierstil zuläßt. So wurde für das neue System EROS7 für die Kodierung der Regeln auch ein Anschluß an einen Tcl-Interpreter erstellt. Tcl ist eine Programmiersprache [63], die nur den algorithmischen Programmierstil zuläßt und nur Variablen vom Typ String kennt. Tcl ist bezüglich der Funktionalität deutlich ärmer als C++ und stellt somit etwa den Fall größter anzunehmender Schwierigkeiten für den Anschluß dar.

Dadurch, daß EROS7 von Grund auf neu programmiert wurde, konnten auch neue Überlegungen bezüglich der Anbindung der Regeln angestellt werden. Die Regeln sind die Wissensbasis für EROS7, in denen die einzelnen Reaktionstypen inklusive der Beschränkungen, Bewertungen der Reaktionen und die zu verwendende Kinetik kodiert sind. Für EROS6 wurden die Regeln in einer eigenen Sprache (VERGIL, Versatile EROS Reactivity functions Generation and Interpretation Language [41]) geschrieben, die dann von selbstgeschriebenen Programmteilen ausgewertet werden mußten. Dies hat den Nachteil, daß man den Compiler bzw. Interpreter für die Regeln selbst erstellen muß, der dann alle gewünschten Funktionalitäten enthält, was durchaus aufwendig sein kann. Da liegt der Gedanke nahe, die Programmiersprache, in der das Programm selbst geschrieben ist, auch für die Formulierung der Regeln zu verwenden. So erhält man für die Regeln von ROS7 alle Möglichkeiten der mächtigen Programmiersprache C++. Wird die Sprache C++ erweitert, so ist im schlimmsten Fall das ganze Programm neu zu übersetzen und zu binden, um auch die Erweiterungen nutzen zu können. Der große Aufwand für die Implementation eines Compilers bzw. Interpreters für die Regeln bei Verwendung einer eigenen Sprache und die Aussicht auf die Möglichkeit einer flexiblen Weiterentwicklung der Regeln durch den Einsatz der Sprache C++ haben den Ausschlag gegeben, die Regelschnittstelle in C++ zu definieren. Nachteile hiervon sind, daß die Regeln in einer komplexeren, nicht unbedingt den chemischen Problemen angepaßten Sprache abgefaßt werden, sowie der Aufwand für eine Erklärkomponente, die für die Beurteilung, warum eine Reaktion stattfindet oder nicht, die Regeln analysieren muß. Auswege daraus sind die Erstellung eines Regeleditors mit grafischer Oberfläche, der für einen bestimmten Zweck immer nur eine Untermenge der möglichen Einstellungen in den Regeln verwendet und dies in immer gleicher Weise tut, oder der Anschluß einer geeigneten anderen Programmiersprache, was der Grund dafür ist, daß bei der Definition der Regelschnittstelle auf objektorientierte Ansätze weitestgehend verzichtet wurde. Zur Zeit existiert für EROS7 der Anschluß an den Tcl-Interpreter. Ob die Abfassung der Regeln in Tcl

oder in C++ gewählt wird, macht einen geringen Unterschied. Tcl hat den Vorteil, daß die Regeln interpretiert werden können. Es wird, nachdem man die Regeln abgeändert hat, für einen erneuten Programmlauf kein zusätzliches Programm, wie etwa ein Compiler, benötigt. Die Abarbeitung der Regeln in Tcl und damit die Reaktionssimulation durch EROS7 dauert mit compilierten und nicht compilierten Regeln länger als mit den entsprechenden C++-Regeln. Interpretierte Tcl-Regeln sind dagegen flexibler, da sie geändert direkt ohne Compilation erneut getestet werden können. Insgesamt ist der Aufbau der Regeln gegenüber denen von EROS6 wegen der größeren Funktionalität um einiges komplexer. Deshalb sollte für einen breiteren Einsatz ein Regeleditor erstellt werden.

6.12. Implementation des Programmsystems EROS7

EROS7 ist wie die verwendete, neue Datenstruktur RICOS aus Gründen der Datenabstraktion in C++ geschrieben. Wie schon in Abschnitt 3.1 gesagt, besteht ein Programm zur Vorhersage von Reaktionen aus mehreren Teilen. Diese sind noch einmal in Abbildung 137 gezeigt, wobei die Regelschnittstelle als Bindeglied zwischen der Reaktionsgenerierung, der chemischen Datenstruktur und den Regeln eingefügt wurde. Die Regeln fassen die Kodierung der Reaktionstypen und die Bewertung der Reaktionen sowie sämtliche anderen Einstellmöglichkeiten für EROS7 zusammen.

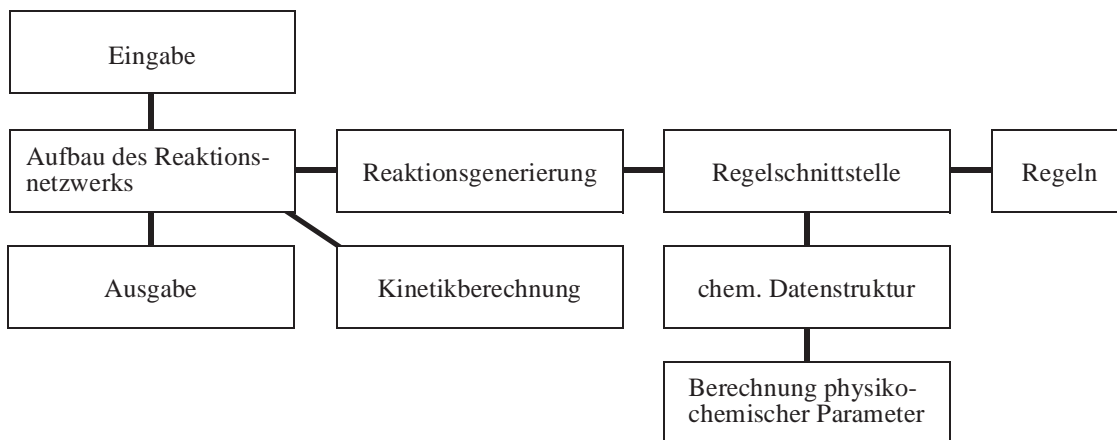


Abbildung 137: Aufbauschema von EROS7.

6.12.1. Aufbau der Reaktionsgenerierung

Analog zum logischen Aufbau von EROS7 in Abbildung 137 gliedern sich auch die Klassen von EROS7. Die Hauptwege, wie die verschiedenen Klassen aufeinander aufbauen, sind in Abbildung 138 dargestellt. Durchgezogene Pfeile zeigen an, daß die Klasse an der Spitze in der Klasse am Ende enthalten ist. Ist der Pfeil gestrichelt, hat die Klasse nur einen Zeiger auf die Klasse und greift über diesen auf deren Elementfunktionen zu. Neben dem Hauptprogramm *eros7*, der chemischen Datenstruktur RICOS, deren Klassen der Vereinfachung halber zusammengefaßt wurden und die in [52] genauer beschrieben sind, und den wahlweise aufgerufenen Regelfunktionen *rules_cc* und *rules_tcl* gliedern sich die Klassen, angezeigt durch ihr Präfix, in drei Gruppen: *gg_* (graph generation) Aufbau des Reaktionsnetzwerks und Kinetik, *rx_* Reaktionsgenerierung und *rul_* (Teil der Regel) ein Reaktionstyp.

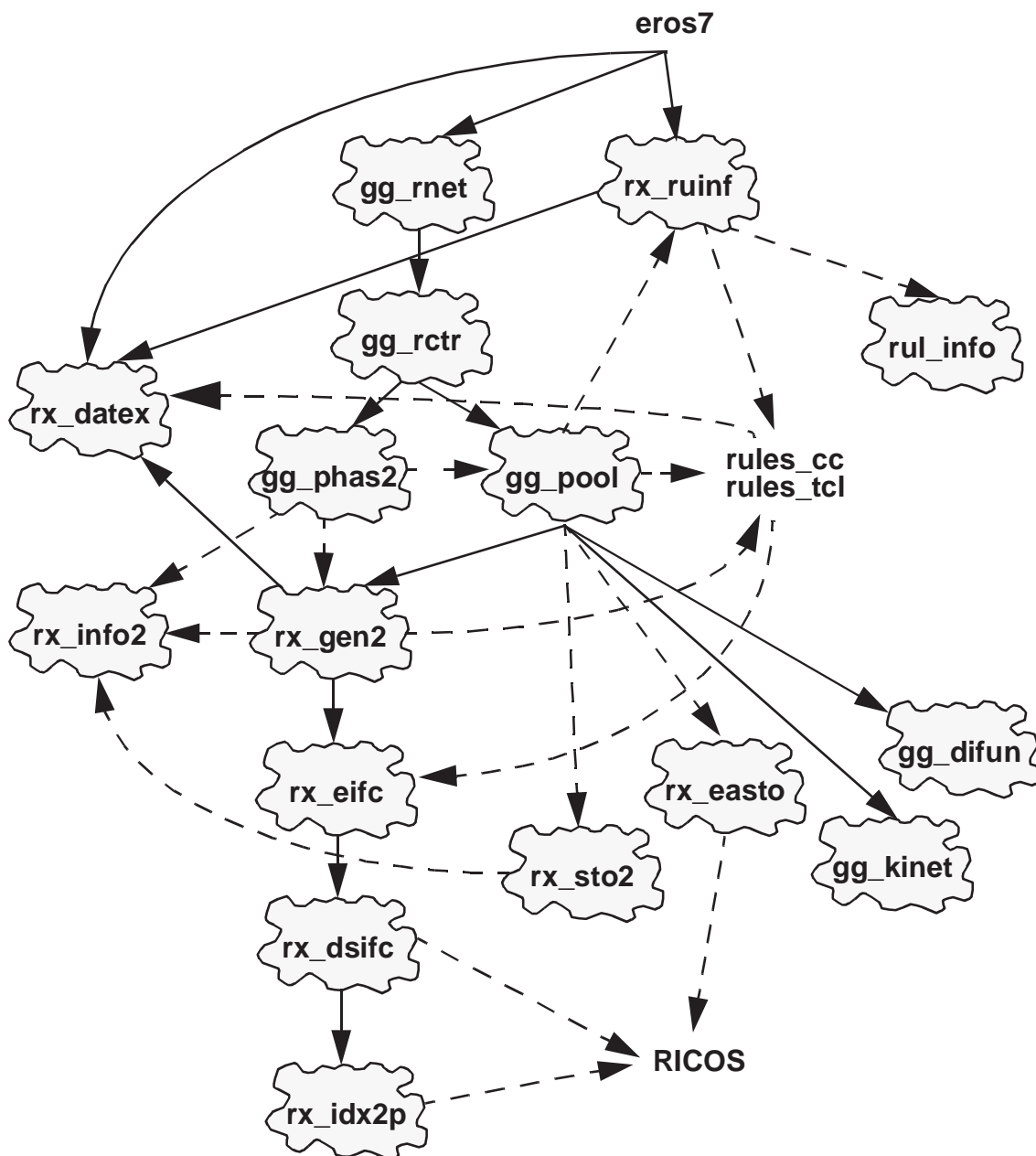


Abbildung 138: Hauptbeziehungen der Klassen von EROS7.

Der reine Aufbau des Reaktionsnetzwerks gliedert sich, wie die Funktionalität von EROS7 in das Reaktionsnetzwerk (*gg_rnet*), die Reaktoren (*gg_rctr*) und die Phasen (*gg_phas2*). Da die Reaktoren, Phasen und der Reaktionsgenerator (*rx_gen2*) auf mehrere gemeinsame Objekte zugreifen müssen, sind diese in einer Klasse zusammengefaßt (*gg_pool*). Sie enthält die Speicherung der Aggregate bzw. Ensembles (*rx_easto*) und die der Reaktionen (*rx_sto2*) sowie die Kinetik (*gg_kinet*) und davon abgetrennt die aufgestellten Differentialgleichungen (*gg_difun*). Ein Objekt der Klasse *rx_info2* speichert genau eine erzeugte Reaktion. Die Informationen über die verwendeten Regeln sind in den Klassen *rx_ruinf* (globale Information, wie Zahl der Reaktoren und Phase, usw.) und *rul_info* (Angabe zu einem Reaktionstyp) gespeichert.

Neben dem Reaktionsgenerator (*rx_gen2*), der die Regelfunktionen *rules_cc* bzw. *rules_tcl* für die Durchführung der Reaktionen aufruft, verbleiben noch die beiden wichtigen Klassen *rx_datex* (Variablen austausch zwischen EROS7 und den Regeln) sowie *rx_eifc* (Zugriff auf die Moleküle und Modifikation der chemischen Struktur), über die die Regeln auf die entsprechen-

den Teile von EROS7 zugreifen. Hierzu werden die Regelfunktionen jeweils mit einem Zeiger auf die aktuell gültigen Objekte dieser Klassen aufgerufen. Der direkte Zugriff auf die chemische Datenstruktur erfolgt über ein Objekt der Klasse *rx_dsifc*, das sich die Arbeit der Abbildung der in den Regeln verwendeten Indizes für die Atome, Elektronensysteme, etc. auf die entsprechenden Zeiger der RICOS-Objekte von *rx_idx2p* abnehmen läßt.

Der Aufbau des Reaktionsnetzwerks wird also von den Klassen *gg_rnet*, *gg_rctr*, *gg_phas2*, *gg_pool*, *rx_sto2*, *rx_easto* und *rx_info2* übernommen (vergleiche Abbildung 137). Die Regelschnittstelle gliedert sich in die Klassen *rx_ruinf*, *rul_info*, *rx_eifc*, *rx_dsifc*, *rx_idx2p* und *rx_datex*. Da die Klasse *rx_datex* den Variablenaustausch mit den Regeln bewerkstelligt und die Regeln auch in Tcl geschrieben werden können, muß diese Klasse auch den Anschluß an die Variablen des Tcl-Teils enthalten. Zu diesem Zweck entspricht diese Klasse im Fall der C++-Regeln der Klasse *rul_datx* bzw. im Fall der Tcl-Regeln der um die Anbindung der Tcl-Variablen erweiterten Klasse *dat_trans*. Diese Umschaltung und einige weitere kleine Änderungen werden durchgeführt, wenn bei der Compilation TCL definiert wird. Ist dies der Fall, können zusätzlich zu den C++-Regeln auch compilierte und interpretierte Tcl-Regeln verwendet werden.

6.12.2. Anbindung der Wissensbasis (Regeln) an EROS7

Um keinen eigenen Regelinterpreter schreiben zu müssen, die Konstrukte der mächtigen Sprache C++ nutzen zu können und da bereits bei der Planung an den Anschluß von Tcl gedacht wurde, sind die Regeln als Funktionsaufruf an EROS7 angebunden. EROS7 kann mehrere solcher Funktionen verwalten und ruft standardmäßig entweder die C++- bzw. Tcl-Regeln auf. Als Übergabeparameter erhält sie drei Zeiger auf Klassen und mehrere Schaltervariablen, die festlegen, welcher Teil der Regeln gerade ausgeführt werden soll. Diese drei Klassen sind ein Stream zur Ausgabe von Kommentaren, *rx_datex*, die Klasse für den Variablenaustausch, und *rx_eifc* für die Zugriffe und Modifikationen auf der Datenstruktur RICOS.

6.12.2.1. Teile der Regelfunktion

Zunächst wird über die Variable, die die Regelnummer angibt, unterschieden, ob der Regeldateikopf (GLOBAL) gefragt ist oder einer der Reaktionstypen, die Nummern von 1 bis 1000 haben können (siehe Abbildung 139). Die Nummern müssen dabei nicht zusammenhängend sein. Ob der entsprechende Aufruf erfolgreich beendet werden konnte, zeigt die Regelfunktion dadurch an, daß sie den Wert OK (erfolgreich) bzw. BAD als Rückgabewert liefert.

Regelkopf
Reaktionstyp 1
Reaktionstyp 12
Reaktionstyp 13
Reaktionstyp 100

Abbildung 139: Grundgliederung der Regeln.

Der Regelkopf und jeder von den Reaktionstypen sind noch weiter untergliedert. Dies wird von der Variable vom Typ OP (wie operation) selektiert. Für den Regelkopf ist dies in Abbildung 140 gezeigt.

INIT_RULES
PREP_ROOT
PREP_EDUCT
DISTRIBFUNC
MODFUNC
FINISH

Abbildung 140: Teile des Regelkopfes.

In *INIT_RULES* werden die Einstellungen für die Reaktoren, Phasen und die Kinetik sowie der Variablen austausch neben ein paar weiteren globalen Einstellungen, zu denen beispielsweise der Modus für die Valenzprüfung der Produkte gehört, durchgeführt (siehe 6.7 und Anhang B.2). *PREP_ROOT*, *PREP_EDUCT* und *DISTRIBFUNC* sind die Vorbehandlungsfunktionen für die Ausgangsmaterialien (siehe 6.4.1) und die Edukte (siehe 6.4.2) sowie die Verteilungsfunktion (siehe 6.6.4). *FINISH* (siehe Anhang B.2.11) ist der Teil, der als letztes vor Beendigung eines EROS7-Laufs aufgerufen wird. Er ist für abschließende Tätigkeiten vorgesehen (z.B. Schließen von Dateien), wenn EROS7 als Schnittstelle für andere Programme zur chemischen Datenstruktur RICOS verwendet wird. *MODFUNC* (siehe Anhang B.2.10) ist ein Teil, der für Erweiterungen von EROS7 vorgesehen ist. Hier können auch andere Funktionen, die nichts mit der Funktion der Regeln zu tun haben, EROS7 bereitgestellt werden. Der Name von *MODFUNC* kommt vom englischen Ausdruck model function.

Jeder der Reaktionstypen gliedert sich dagegen in nur drei Teile, *RULE_INFO* (siehe 6.5), der die Definition der Reaktionssubstruktur, den Namen und die Attribute des Reaktionstyps enthält, *CONSTR* (constraints, siehe 6.5.5), die Überprüfung der Bedingungen am Reaktionszentrum und *FUNC* (function, siehe 6.6), die Durchführung der Reaktion sowie die Berechnung der Reaktivität.

RULE_INFO
CONSTR
FUNC

Abbildung 141: Teile eines Reaktionstyps.

Detaillierte Angaben, wie die Regeln in C++ bzw. Tcl abgefaßt werden, stehen im Anhang B.

6.12.2.2. Datenaustausch zwischen den Regeln und dem Kernsystem

Der Datenaustausch zwischen den Regeln und dem Kernsystem von EROS7 wurde so konstruiert, daß EROS7 leicht erweitert werden kann. Deshalb wurde darauf verzichtet, einen festen Satz von Variablen zu übergeben, über den EROS7 gesteuert wird. Bei Erweiterungen erfordern erfahrungsgemäß auch zusätzliche Einstellungsmöglichkeiten, für die zusätzliche Variablen benötigt werden. Bei einem festen Satz von Variablen müßte auch die Variablenübergabe angepaßt werden. Um dies zu vermeiden, erfolgt die Variablenübergabe zwischen den Regeln und EROS7 über ein Objekt der Klasse *rx_datex*, bei der die beiden miteinander kommunizierenden

Teile die Variablen typischer unter einem frei wählbaren Namen austauschen. Das Objekt speichert alle von einem Teil angebotenen Variablen, während der andere Teil in diesem Objekt nachsieht, ob die ihm bekannten Variablen enthalten sind. Dieser Teil kann dann dynamisch darauf reagieren, ob eine bestimmte Variable zur Verfügung steht oder nicht. Als mögliche Reaktionen bei Abwesenheit sind denkbar, eine alternative Quelle der Information zu nutzen, beispielsweise Standardwerte zu nehmen, oder bei einer essentiellen Variablen mit einer Fehlermeldung den entsprechenden Programmteil zu verlassen bzw. EROS7 gänzlich zu beenden.

Programmiertechnisch werden in jeweils zwei parallelen Listen für die verschiedenen Variablentypen die Zeiger auf den Namen und auf die Variable selbst gespeichert. Es wird der Zeiger auf die Variable und nicht deren Inhalt übergeben, damit sie nur einmal übergeben werden muß, während sie zu mehreren Zeitpunkten des Programmlaufs gelesen werden kann und man immer den aktuellen Wert erhält. Wichtig ist noch, daß die Variable zum Zeitpunkt des Lesezugriffs noch nicht deallociert wurde. Da die Programmiersprache C++ wie auch C ihre Variablen standardmäßig dynamisch allociert und deallociert, hätte das zur Folge, daß die Variable beim Verlassen eines Programmteils gelöscht würde und somit der später folgende Lesezugriff fehlschlagen würde. Deshalb müssen alle übergebenen Variablen als *static* deklariert werden, damit sie nach der ersten Allocation erst wieder zum Programmende gelöscht werden. Das gleiche gilt im Grunde auch für die Namen, unter denen die Variablen übergeben werden. Da für sie in der Regel Konstanten verwendet werden, die automatisch *static* sind, muß für sie nichts besonderes berücksichtigt werden.

Da fast alle Variablen bei der Initialisierung der Regeln ausgetauscht werden, muß auch der erhaltene Zeiger auf eine Variable dauerhaft gespeichert werden, damit er beim nächsten Aufruf noch seinen Wert besitzt. Deshalb müssen auch die gespeicherten Zeiger als *static* deklariert werden. Da der lesende Teil einen *const*-Zeiger erhält, kann dieser den Inhalt der Variablen nicht ändern und der setzende Teil kann sich immer darauf verlassen, daß die Variable immer ihren vom setzenden Teil zuletzt zugewiesenen Wert enthält.

Übergeben werden können Zeichenketten, ganzzahlige Werte und Fließkommazahlen sowie Vektoren und Matrizen der beiden letzteren. Konstante Werte werden getrennt übergeben, damit man sich bei ihrer Übergabe eine Referenzierung sparen kann. Für sie werden die Inhalte der Variablen statt der Zeiger auf sie weitergereicht. Die Konstanten werden fast ausschließlich für die Einstellungen von EROS7 verwendet. Da es für Vektoren und Matrizen in C++ schwierig ist, Konstanten anzugeben, können statt Vektoren und Matrizen Konstantenarrays vermittelt werden. Für Konstanten stehen folgende Typen zur Verfügung: ganze Zahlen, Arrays ganzer Zahlen, Fließkommazahlen, Zeichenketten und Arrays von Zeichenketten.

Das folgende Beispiel zeigt die Übergabe der Variablen, die die Geschwindigkeitskonstante für die einzelnen Reaktionen enthält. Drei Punkte stehen im unten angegebenen Code für Teile, die der Übersichtlichkeit halber weggelassen wurden. Die eingerückten Teile zeigen die angesprochenen Abschnitte der Regeln.

```

rx_datex vars;
RULE_OK res;
...
// Übergabe des Zeigers auf die Variable mit
// der Geschwindigkeitskonstante an die Klasse
// rx_datex
res = rules_cc(..., &vars, GLOBAL, INIT_RULES, ...);
                                RULE_OK rules_cc(..., rx_datex *val, ...) {
                                    static double k;
                                    ...
                                    case INIT_RULES:
                                        ...
                                        val->put("reactivity", &k);
                                        return OK;
                                    ...
                                }

// Abholen des Zeigers aus dem
// Objekt der Klasse rx_datex
static const double *reactivity;
vars.get("reactivity", &reactivity);
...

// Aufruf zur Durchführung einer Reaktion
// und Setzen der Geschwindigkeitskonstante
res = rules_cc(..., 0, 1, FUNC, ...);
                                RULE_OK rules_cc(...) {
                                    ...
                                    case FUNC:
                                        ...
                                        k = 1e-13;
                                        return OK;
                                    ...
                                }

// Lesezugriff auf die Variable mit der
// Geschwindigkeitskonstante und Ausgabe
cout << "Die Geschwindigkeitskonstante ist: " << *reactivity << endl;

```

6.12.2.3. Anbindung der Regelfunktion

Da die Regeln als Funktion aufgerufen werden, verhält sich der Aufruf der Regelfunktion wie jeder andere Funktions- oder Subroutinenaufruf. Dementsprechend gibt es für die Anbindung zwei Wege, sie statisch oder dynamisch zum ausführbaren Programm zu binden. Bei der statischen Bindung sind die übersetzten Regeln Bestandteil des Programms. Sollen sie geändert werden, müssen die neuen Regeln übersetzt werden und das gesamte Programm, EROS7, neu gebunden werden. Bei den modernen Compilern und Linkern ist es neben dieser Methode auch möglich, einen Satz von Funktionen in einer Bibliothek dynamisch zum Programm zu binden. Dies bedeutet, daß beim Binden des Programms zwar die Anwesenheit der einzelnen Funktionen in der Bibliothek überprüft wird, die Funktionen selbst aber nicht zum Programm gebunden werden. Es wird lediglich im Programm vermerkt, daß die eine oder andere Funktion aus dieser Bibliothek für das Programm benötigt wird. Wird dieses Programm aufgerufen, werden das Programm und die dynamische Bibliothek in den Hauptspeicher geladen und fertig gebunden. Es ist also möglich, ein EROS7 ohne Regeln zu erzeugen und die Regeln in einer zweiten Datei, einer dynamischen Bibliothek zu halten. Neben diesen beiden Möglichkeiten, die bei jeder modernen Entwicklungsumgebung besteht, gibt es unter Solaris 2 auch einen inkrementellen Linker (ild) [64], der beim ersten Binden etwas Platz im Programm reserviert, wodurch er auch

beim fertigen Programm in der Lage ist, einzelnen Module (Funktionen) nachträglich auszutauschen. Wurden zu viele Module ausgewechselt, bindet auch der inkrementelle Linker das gesamte Programm neu.

6.12.2.4. Anbindung anderer Sprachen, wie zum Beispiel Tcl

Soll zur Kodierung der Regeln eine andere Sprache verwendet werden, die gegenüber C++ als geeigneter angesehen wird oder andere Vorteile bietet, liegt die Vorgehensweise auf der Hand. Man erstellt eine Funktion in C++, die beim Aufruf der verschiedenen Teile der Regeln die entsprechenden Abschnitte in der anderen Sprache ausführt. Damit EROS7 neben den Regeln in C++ wahlweise auch Regeln in der anderen Sprache ausführen kann, hat EROS7 standardmäßig zwei gleiche Regelschnittstellen für die Regeln in C++ und Tcl. Da schon während der Entwicklung von EROS7 der Anschluß der Sprache Tcl geplant wurde, heißen die beiden Regelfunktionen *rules_cc* und *rules_tcl*. Welche dieser Funktionen aufgerufen wird, wird mit der Option `-r` beim Starten von EROS7 festgelegt. Fehlt diese Option oder wird `-rcompiled_c++` angegeben, werden die C++-Regeln verwendet, anderenfalls wird die Funktion *rules_tcl* aufgerufen. Notwendige Initialisierungen können beim ersten Aufruf der Regeln (*INIT_RULES*) vorgenommen werden, wobei auch der Wert der Option `-r` zur Verfügung steht. Sind in einer Spezialversion weitere Regeln in C++ zu EROS7 gebunden worden, können auch diese mit der Option `-r` angesprochen werden.

Die Schnittstelle zu Tcl, die von Frau L. Steinhauer erstellt wurde, ruft intern den Tcl-Interpreter auf. Dieser liest die Regeln beim Initialisieren aus der Datei, die für die Regeln angegeben wurde. Durch die Verwendung eines Interpreters kann man auch Regeln verwenden, die erst nach dem Start des Programms eingelesen werden und nicht zum System gebunden sind – eine andere Möglichkeit von dynamischen Regeln. Da eine Datei auch mehrere Tcl-Prozeduren enthalten kann, wird optional auch noch der Name der zu verwendenden Prozedur angegeben. Standardmäßig wird die Prozedur mit dem Namen *Tcl_rule* genommen. Alternativ dazu kann EROS7 auch so gestartet werden, daß der Tcl-Interpreter eine compilierte und zum System gebundene Tcl-Prozedur aktiviert. Dies erfolgt durch die Angabe der Option `-rcompiled_tcl`. Die Selektion der einzelnen Abschnitte der Regeln erfolgt in der Tcl-Prozedur über den Inhalt von vordefinierten globalen Variablen, denen die Werte der Schaltervariablen zugewiesen werden, mit denen die Regelfunktion von EROS7 aufgerufen wurde. Für den Variablenaustausch sowie für die Zugriffe auf die chemische Datenstruktur RICOS (Modifikationen und Lesen von Eigenschaften) gibt es Prozeduren, die die Zugriffe auf die Objekte der Klassen *rx_datex* und *rx_eifc* bewerkstelligen, über die EROS7 mit den Regeln kommuniziert und die beim Aufruf der Regeln angegeben wurden.

Eine besondere Rolle spielt die Übergabe der Variablen von und zu EROS7, da Tcl nur Variablen vom Typ Zeichenkette sowie Arrays und Listen von Zeichenketten kennt. Bei jedem Austausch müssen die Werte deshalb konvertiert werden. Für die Übergabe von Variablen durch *put* oder *putco*, die über ein Objekt der Klasse *rx_datex* an EROS7 übergeben werden, muß eine C++-Variable vom richtigen Typ allociert werden. Damit der Typ automatisch festgestellt werden kann, muß die Tcl-Variable zum Zeitpunkt der Übergabe einen Wert besitzen, aus dem der Typ abgeleitet werden kann. Damit bei jedem Aufruf und Verlassen des Tcl-Teils die übergebenen Variablen die richtigen Werte enthalten, kann Tcl solche Variablen linken, was bedeutet, daß sie am Anfang und Ende der Tcl-Teile hin- bzw. herkonvertiert werden. Dieser Mechanismus wird auch verwendet, wenn eine Variable mit *get* in den Regeln von EROS7 abgeholt wird. Diese Funktionalität, das Linken und Konvertieren von Tcl-Variablen, ist der Klasse *dat_trans* zu der der Klasse *rul_datx* hinzugefügt worden. Ist EROS7 ausschließlich für C++-Regeln übersetzt worden, verbirgt sich hinter der Klasse *rx_datex* die Klasse *rul_datx*, ist auch der Tcl-Teil in EROS7 enthalten, ist sie die Klasse *dat_trans*. Da die an EROS7 übergebenen Tcl-Variablen bei jedem Verlassen in die C++-Variablen konvertiert werden, also gelesen werden, dürfen die

Variablen nicht dynamisch allociert werden, sondern müssen statisch verfügbar sein, was durch die *global*-Deklaration erreicht wird. Übergebene Konstanten werden dagegen nicht gelinkt, sondern nur ein einziges Mal konvertiert.

Für die Manipulation und den Zugriff auf die Datenstruktur sind entsprechende Prozeduren vorhanden, die die übergebenen Werte konvertieren und anschließend die Elementfunktionen (Zugriffsfunktionen auf Objekte der Klasse) der Klasse *rx_eifc* aufrufen.

6.12.3. Zugriffe auf die chemische Datenstruktur RICOS

Der Zugriff auf die Datenstruktur RICOS erfolgt über Elementfunktionen der Klasse *rx_eifc*. In den Regeln werden die Atome, Elektronensysteme usw. über Indizes selektiert, damit es auch möglich ist, andere Sprachen wie Tcl zur Kodierung der Regeln verwenden zu können. Die chemische Datenstruktur RICOS arbeitet allerdings mit Zeigern auf die Bestandteile der Moleküle, weshalb bei jedem Zugriff zunächst die Indizes in die entsprechenden Zeiger umgewandelt werden müssen. Erst danach kann die eigentliche Aktion ausgeführt werden.

6.12.3.1. Indizes der Atome, Elektronensysteme usw.

Alle Zugriffe auf die chemische Datenstruktur RICOS erfolgen von den Regeln aus über Indizes der Bestandteile des Ensembles: Atome, Elektronensysteme usw.. Sowohl die Abfrage von Eigenschaften, wie auch die Reaktionsgenerierung, werden unter Verwendung dieser Indizes ausgeführt. Während der Reaktionen werden die Elektronen und damit meist auch die Elektronensysteme umgeordnet. Dies hat zur Folge, wenn zwei oder mehrere Elektronensysteme zu einem vereinigt werden, daß manche Elektronensysteme nach einer Reaktion nicht mehr existieren bzw. nach einem Bruch eines Elektronensystems neu gebildet werden. Neu gebildete Elektronensysteme erhalten neue höhere Nummern.

EROS7 ist in der Lage, auch mehrere Reaktionsteilschritte nacheinander durchzuführen. Um das Edukt bzw. Zwischenprodukt nicht zu verändern und damit weiterhin zugriffsbereit zu halten und dennoch die Reaktion an der Verbindung durchzuführen, kopiert EROS7 vor jedem Abschnitt von Reaktionsteilschritten das Reaktionsensemble und führt diese auf der Kopie durch. Zu welchem Abschnitt die gerade ausgeführte Aktion gehört, wird von EROS7 automatisch aus dem Wechsel zwischen der Abfrage von Eigenschaften und Reaktionsteilschritten erkannt. Um Eigenschaften des Reaktionszentrums im Produkt abfragen zu können und anschließend weitere Reaktionsteilschritte vorzunehmen, müssen die Atome, Elektronensysteme und Gruppen während der gesamten Reaktionsschritte ihre Nummern behalten. Da durch die Kombination von Elektronensystemen und das Löschen von Aggregaten Elektronensysteme und Atome gelöscht werden können und somit nach Reaktionen die Indizes nicht mehr konsekutiv durchnummeriert sind, die Datenstruktur RICOS für ihre Indizes aber darauf besteht, können diese nicht verwendet werden. Die Regelschnittstelle generiert deshalb eigene Indizes, über die von den Regeln aus auf die Datenstruktur zugegriffen wird. Für die Zuordnung der Atome, Elektronensysteme und Gruppen im Original und in der Kopie werden die beim Cleanup der Datenstruktur neu zugeweilten Indizes von RICOS eingesetzt. Danach werden die Indizes der Regelschnittstelle so gesetzt, daß gleiche Atome usw. gleiche Nummern erhalten. Bei den meisten Reaktionen ändern sich die Zuordnungen von Atomen, Elektronensystemen und Gruppen zu Molekülen und Aggregaten. Da in einigen Fällen die Zuordnung der Moleküle im Eduktensemble zu denen im Produktensemble nicht eindeutig ist und oft nur interessant ist, zu welchem Molekül ein bestimmtes Atom gehört, werden Moleküle und Aggregate nach der Reaktion neu durchnummeriert. Aggregat- und Molekülnummern für die Produkte werden über die Atome, die sie enthalten, abgefragt.

Diese Abbildung und Aktualisierung der Indizes auf die richtigen Zeiger der Datenstruktur RICOS übernimmt die Klasse *rx_idx2p*. Während des Kopiervorgangs wird die Zuordnung der In-

dizes von RICOS auf die Indizes der Reaktionsschnittstelle in einem Objekt der Klasse namens *rx_numbr* gespeichert. Diese wird bis zum Abschluß aller Teilschritte aufbewahrt, damit es auch möglich ist, zu einem späteren Zeitpunkt auf ein früheres Zwischenprodukt oder das Edukt zugreifen zu können. Dies ist beispielsweise für die Ionisationsreaktionen in der Massenspektroskopie notwendig (siehe Erzeugung mehrerer Reaktionen aus einer Reaktionsstruktur auf Seite 97).

6.12.3.2. Eigenschaften

Die Funktionen zur Abfrage von Eigenschaften bilden eine übersichtliche Schnittstelle, bei der leicht neue Eigenschaften hinzugefügt werden können. Sie wird deshalb auch für alle Zugriffe auf die Datenstruktur RICOS herangezogen. Bedingt durch die Umsetzung der Zeiger der Datenstruktur auf Indizes gliedern sich die Eigenschaften in zwei Gruppen: diejenigen Eigenschaften, die von der Regelschnittstelle erzeugt werden, und jene, die die Datenstruktur zur Verfügung stellt. Auf Seiten der Regeln ist von dieser Aufspaltung nichts festzustellen.

Zu den Eigenschaften der Schnittstelle gehören all diejenigen, die Indizes auf Teile der Struktur liefern. RICOS stellt die Nachbaratome eines Atoms beispielweise als Liste bereit, die die Zeiger auf die Nachbaratome beinhaltet. Die Eigenschaft *A_NEIGHBORS* holt sich die Zeiger auf die Nachbaratome aus der Liste, bestimmt die Indizes der Regelschnittstelle und gibt sie den Regeln als Vektor mit diesen Indizes zurück. Eigenschaften dieser Art gibt es eine ganze Reihe. Es gehören dazu die Nummern des Aggregates und Moleküls, zu dem ein Atom gehört, die Elektronensysteme, an denen ein Atom beteiligt ist, und umgekehrt die Atome, die sich an einem Elektronensystem beteiligen usw.. Daneben gibt es auch noch Eigenschaften des Ensembles, wie viele Atome, Elektronensysteme usw. in diesem Ensemble sind und welcher der höchste Index ist, sowie eine Eigenschaft, die sagt, ob das Elektronensystem mit diesem Index existiert oder nicht (siehe 6.12.3.1). Aber auch das Lesen und Setzen des Zentralatoms eines σ -Elektronensystems mit Zentrum ist als setzbare Eigenschaft implementiert. RICOS hat hierfür spezielle Funktionen, auf die diese Eigenschaft zugreift. Historisch bedingt, sind auch noch einige weitere Eigenschaften Teil der Regelschnittstelle (siehe Produktfunktion auf Seite 246). Neben den Eigenschaften der Regelschnittstelle stellt RICOS den überwiegenden Teil der Eigenschaften bereit. Zu diesen gehören die physikochemischen Eigenschaften, topologische Deskriptoren und auch die setzbaren, genealogischen Variablen, die zur Steuerung des Aufbaus des Reaktionsnetzwerks verwendet werden können (siehe 5.1.7). RICOS kennt eine Vielzahl von verschiedenen Datentypen, die in der Regelschnittstelle zur Vereinfachung auf sechs Typen reduziert werden. So werden Fließkommazahlen mit hoher und niedriger Genauigkeit beide als Fließkommazahlen-Eigenschaften mit hoher Genauigkeit den Regeln bekannt gegeben und weitergereicht. Die sechs Datentypen, auf die alle RICOS-Datentypen abgebildet werden, sind: *int*, *double*, *vector<int>*, *vector<double>*, *e_bit64* und *const char**.

6.12.3.3. Umordnung der Elektronensysteme (Reaktionen)

Die Umordnung der Elektronensysteme, also die Durchführung der Reaktionen, ist eine Modifikation der chemischen Datenstruktur RICOS. Für einen Reaktionsteilschritt werden nicht immer alle Änderungen an der Datenstruktur durchgeführt, die mit dem Teilschritt einhergehen. Dies liegt einerseits an der Geschwindigkeit der Abarbeitung, aber auch daran, daß manchmal erst nach einem weiteren Teilschritt klar wird, was passieren soll. Man denke hierbei an einen einfachen Bruch eines aromatischen Rings. Erst durch die Spaltung des Rings an einer zweiten Stelle bzw. des darunterliegenden σ -Systems wird klar, was gemacht wird. Deshalb erfolgen die Veränderungen von RICOS über eine eigene Klasse namens *ds_modify*, die sich die noch auszuführenden Aktionen auf der Datenstruktur gegebenenfalls merkt und dann bei der abschließenden Bereinigung durchführt. Dieses Cleanup wird vom Reaktionsgenerator aufgerufen, wenn ein Abschnitt von Reaktionsteilschritten beendet ist und Eigenschaften der Produkte ab-

gefragt werden bzw. die Reaktion im ganzen beendet ist.

Was genau mit den intern gespeicherten Zeigern usw. in RICOS für die Kombination oder die Spaltung eines Elektronensystems geschehen muß, kennen die Elementfunktionen der Klasse *ds_modify*. Die Schnittstelle wurde in dieser Weise definiert, damit die Datenabstraktion gewahrt ist und der Reaktionsgenerator nichts darüber wissen muß, wie die Elektronensysteme in RICOS implementiert sind. Er kennt nur ihre Typen, was man mit ihnen machen kann, und kann die Zahl der Elektronen im System ermitteln, ändern sowie abfragen, welche Atome an ihm beteiligt sind. All dies geschieht mittels Zugriffsfunktionen.

Was mit welchem Elektronensystem zu geschehen hat, wird dagegen vom Reaktionsgenerator entschieden. Bei der MO-orientierten Reaktionsgenerierung gehören hierzu die Überprüfung der Situation und gegebenenfalls die vorherige Aufhebung von Konjugationen. Soll beispielsweise eine 2-Zentren-2-Elektronenbindung gebildet werden, benötigt man eigentlich zwei einzelne Orbitale (kodierte als π -Systeme an nur einem Atom) mit zusammen zwei Elektronen. Hat man dem Reaktionsgenerator dagegen größere π -Systeme angegeben, müssen diese zunächst gespalten werden. Daneben wird hier auch verhindert, daß verdrehte Strukturen oder unmögliche Elektronenkonfigurationen erzeugt werden, wie es in 6.6.1.3 beschrieben ist.

Für die eingebaute Valence-Bond-Emulation, also der Behandlung der Datenstruktur RICOS als würde es sich um eine Bindungsliste handeln, ist wesentlich mehr erforderlich. Zunächst müssen für einen Bindungsbruch alle Elektronensysteme zwischen den beiden Atomen gefunden werden. Danach wird entschieden, ob es sich bei einem π -System nur um die Konjugation zweier Doppelbindungen bzw. einer Doppelbindung mit einem freien Elektronenpaar usw. handelt. In solche Fällen wird zunächst die Konjugation aufgehoben und dann wieder ein Elektronensystem zum Spalten gesucht. Für die Erhöhung der Bindungsordnung werden mögliche π -Systeme gesucht, wobei bevorzugt kleine, radikalische verwendet werden. Eine besondere Behandlung erfahren aromatische Systeme, bei denen ein einfacher Bruch dennoch als Bindungsbruch angesehen wird. Würde dies nicht auf diese Weise durchgeführt, würde bei einer Erniedrigung der Bindungsordnung um eins im Ring eines Aromaten sowohl das cyclische π -System als auch die darunterliegende σ -Bindung gespalten. Eine Reaktion am aromatischen System wäre also unmöglich, da ein Aromat grundsätzlich ganz aufgespalten würde.

All dies wird von den entsprechenden Elementfunktionen der Klasse *rx_dsifc* unter Zuhilfenahme der Klasse *rx_dsvb* durchgeführt. Der Name *rx_dsvb* stammt vom Zugriff auf die Datenstruktur für die Valence-Bond-Emulation (Bindungslisten). Inzwischen hilft die Klasse aber auch bei der MO-orientierten Reaktionsgenerierung.

6.13. Verwendung der Reaktionsgenerierung auch in anderen Programmen

Die Regelschnittstelle von EROS7 kann auch von anderen Programmen, die beispielsweise in Tcl geschrieben sind, als Schnittstelle zur neuen, chemischen Datenstruktur RICOS genutzt werden. Startet man EROS7 ohne Eingabedatei, wird kein Ensemble eingelesen, sondern EROS7 startet mit einem leeren Ensemble und die Regeldatei enthält den Code für das Programm. Es werden dazu denkbar einfache Regeln geschrieben: ein Reaktor mit einer Phase, keine Kinetik, kein Variablenaustausch mit dem Kernsystem von EROS7 und nur eine Regel ohne Atome in der Reaktionssubstruktur. Die Vorbehandlungsfunktionen für die Startmaterialien (siehe Anhang B.2.7) und das Edukt (siehe Anhang B.2.8) sowie der Teil zur Überprüfung der Bedingungen für die einzige Regel (siehe Anhang B.3.3) sind leer (return OK). Der Funktionsteil FUNC (siehe Anhang B.3.4) der einzigen Regel enthält dann das Programm, das die Schnittstelle zu RICOS von EROS7 nutzt. Wird dieser Teil der Regeln mit der Fehlerindikation (return BAD) verlassen, sind alle Bemühungen von EROS7 weitere Reaktionen zu generieren gebannt.

Innerhalb des Teils FUNC können neben der Reaktionsgenerierung über die in der Schnittstelle enthaltenen Funktionen auch Aggregate gelesen und geschrieben werden. Bindet man zu EROS7 zusätzlich zu Tcl auch noch den Tk-Interpreter hinzu, lassen sich so auch Programme, wie Moleküleditoren oder Browser, mit grafischer Oberfläche schreiben. Eine Einschränkung der Nutzung besteht lediglich dahingehend, daß bei einer großen Zahl von behandelten Aggregaten der Hauptspeicherbedarf steigt, da die Aggregate innerhalb des Funktionsteils FUNC nicht dynamisch in eine Datei ausgelagert werden.

7. Anwendungen von EROS7

7.1. Auf organisch-chemische Reaktionen

7.1.1. Laborchemie und Chargenprozesse

Im Labor werden Reaktionen in der Regel in einem Dreihalskolben, Schlenkrohr oder ähnlichen Geräten durchgeführt. Zu Beginn werden die Edukte in den Kolben gefüllt und die gesamte Reaktionsmischung nach Ablauf der Reaktionszeit aufgearbeitet. Da die Reaktionsgefäße im Labor keine übermäßigen Dimensionen annehmen, ist die vollständige Durchmischung auch sehr gut gegeben. Analog geht man bei industriellen Chargenprozessen vor, wie sie beispielweise bei der Synthese kleinerer Mengen eines Medikaments eingesetzt werden. (Nach Beendigung der Reaktion und Entfernung der Reaktionsmischung kann das Reaktionsgefäß wieder neu mit Edukten für die gleiche oder auch eine andere Synthese befüllt werden.) Es werden immer wieder Synthesen vom Befüllen bis zur Entnahme der Produkte vorgenommen, um anschließend wieder mit dem Befüllen zu beginnen. Der idealisierte Reaktor einer solchen Reaktionsführung ist der absatzweise betriebene, ideale Rührkessel (AIK, siehe 6.7.4.2).

7.1.1.1. Bromierung von Phenol

Als Beispiel der Reaktionsführung eines absatzweise betriebenen, idealen Rührkessels wird nachfolgend die Bromierung von Phenol gezeigt. Ausgangsmaterialien sind elementares Brom und Phenol. EROS7 benötigt zur Simulation dieser elektrophilen, aromatischen Substitution einen Reaktor mit einer Phase im Modus „Rührkessel“, die keine Zu- und Abflüsse besitzt. Die Anfangskonzentrationen der Edukte werden mit 0.1 mol/l für Phenol und 0.4 mol/l für Brom festgelegt. (siehe Abbildung 142)

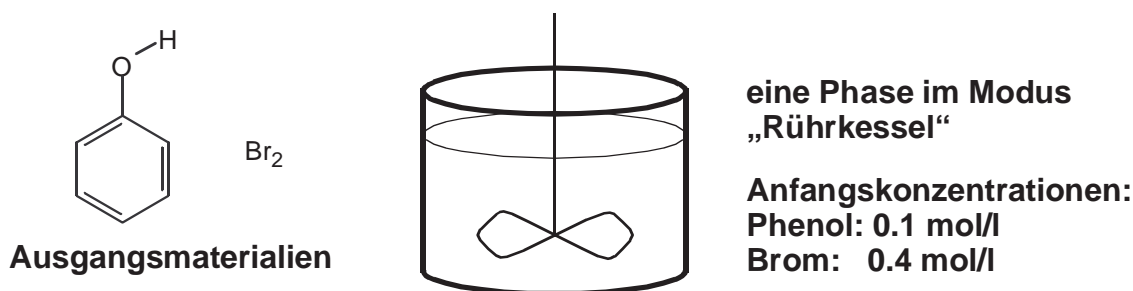


Abbildung 142: Ausgangsmaterialien und Einstellungen der Reaktoren und Phasen für die Bromierung von Phenol als Chargenprozeß.

Der folgende Ausschnitt der Tcl-Regeln zeigt die Einstellungen im Teil *INIT_RULES*. Bei Kommentaren in Tcl (# ...) ist zu beachten, daß sie nicht an beliebigen Stellen stehen dürfen (siehe Anhang B.1).

```
proc Tcl_rule {} {  
  
    global irule op sub_op rule_trace  
    global_name  
    global k  
    global attrib_1 center_1 rule_1  
    global rname rdate inphase phasepr phaseprop pcontacts in2gether  
    global outphase kinmode minconc rxtime convlim sconc  
    switch $irule {  
        GLOBAL {  
            switch $op {
```

```

INIT_RULES {
  # globale Einstellungen
  set rname „elektrophile aromatische Substitution“
  set rdate „29.09.97“
  # Einstellungen der Phasen und Reaktoren
  # Inputphasen der Reaktoren
  set inphase(0) 0
  # 1 Phase im ersten Reaktor
  set phasepr {1,0}
  # Phasenmodi
  set phaseprop(0) „$MIX“
  # keine Phasenkontakte
  set pcontacts(0) 0
  # alle Aggregate kommen zusammen in den Reaktor 1
  set in2gether(0) 1
  # Ausgabephase
  set outphase 1
  # Uebergabe der Variablen an EROS7
  putco input_phase_for_reactors    inphase
  putco phases_per_reactor          phasepr
  putco phase_property              phaseprop
  putco phase_contacts              pcontacts
  putco use_all_educts_together     in2gether
  putco output_phase                outphase
  # kinetische Einstellungen
  # Reaktor 1: Gear-Algorithmus
  set kinmode(0) gear
  # minimale Konzentration fuer ein reagierendes Aggregat
  set minconc 1.e-5
  # maximale Reaktionszeit in Sekunden
  set rxtime 5000.
  # minimaler Umsatz; 1.0=100%
  # es gibt also keinen Umsatzbedingten Abbruch
  set convlim 1.0
  # Anfangskonzentration fuer die Edukte in mol/l
  set sconc(0) 0.1
  set sconc(1) 0.4
  # Anfangswert fuer die Reaktivitaet
  set k 0.0
  # Uebergabe der Variablen an EROS7
  putco kinetic_model                kinmode
  putco minimal_concentration         minconc
  putco reaction_time                 rxtime
  putco conversion_limit              convlim
  putco multi_dose                    multid
  put reactivity                      k
  put start_conc                      sconc
}
...

```

Desweiteren ist in den Regeln der Reaktionstyp der elektrophilen, aromatischen Substitution enthalten, die nur in ortho- und para-Position zur OH-Gruppe gestattet ist. Mit diesem Reaktionstyp baut EROS7 in drei Reaktionsebenen das Reaktionsnetzwerk auf, das in Abbildung 143 dargestellt ist.

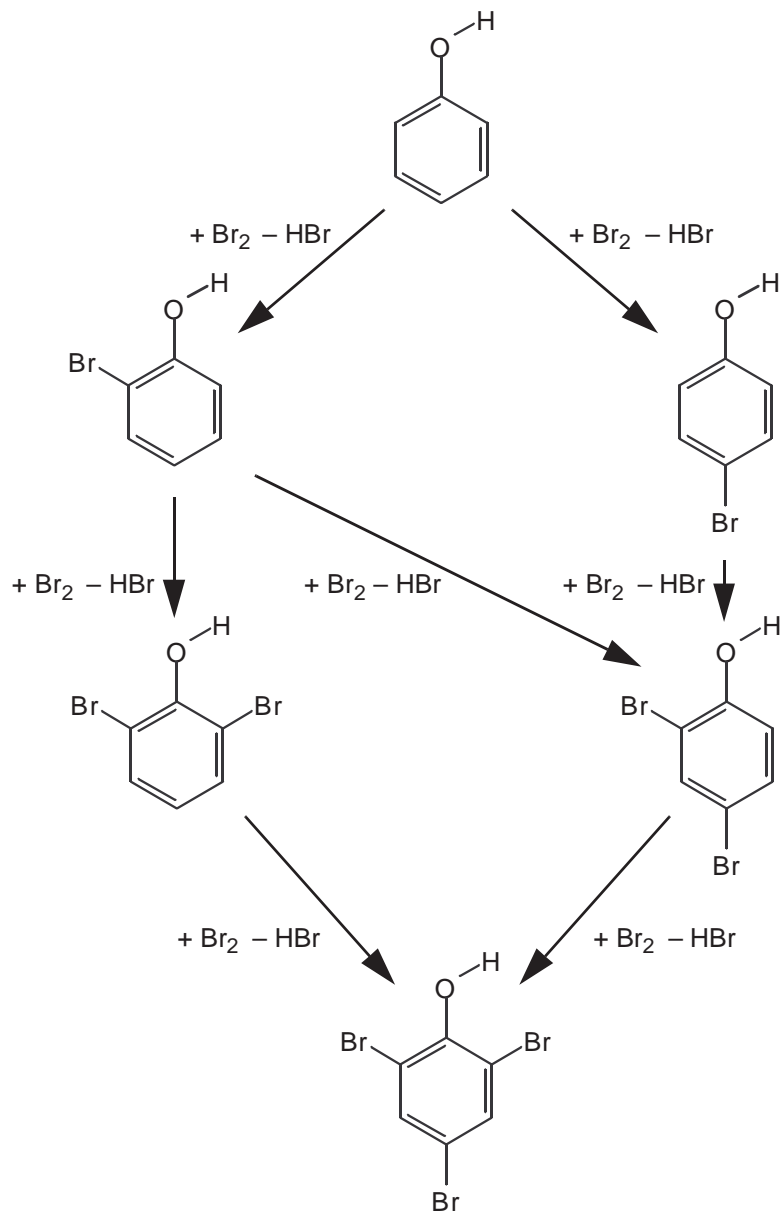


Abbildung 143: Reaktionsnetzwerk der Bromierung von Phenol.

Zur Ableitung der Geschwindigkeitskonstanten konnten in der Literatur folgenden Daten gefunden werden: in [65] steht, daß die Nitrierungsgeschwindigkeit von Brombenzol nur 3% der von Benzol beträgt. Dies wird verallgemeinert und angenommen, daß die Bromierungsgeschwindigkeit je zusätzlichem Bromsubstituenten am Phenol auf 3% des Wertes sinkt, der sich mit einem Bromatom weniger ergeben würde. [56] kann entnommen werden, daß p-Bromphenol aus Phenol mit einer Ausbeute von 80-84% gewonnen werden kann. Da es zwei gleichwertige ortho-Positionen bei Phenol gibt und etwa 80% p-Bromphenol entstehen, ist die Bromierung in para-Position etwa achtmal so schnell wie in ortho-Position. Da keine absoluten Geschwindigkeitskonstanten zweiter Ordnung bekannt waren, wurden sie auf 0.08 für die Para- und 0.01 l/s•mol für die ortho-Substituierung von Phenol festgelegt. Mit diesen Geschwindigkeitskonstanten für die einzelnen Reaktionen kann EROS7 die Konzentrationsverläufe der entstehenden Produkte vorhersagen. In Abbildung 144 ist zu sehen, daß zunächst o- und p-Bromphenol etwa im Verhältnis 1:4 entstehen, die anschließend zu o,p-Dibromphenol umgesetzt werden.

Da für die absolute Größe der Geschwindigkeitskonstanten ein Faktor von $0.01 \text{ l/s}\cdot\text{mol}$ angenommen wurde und kein experimenteller Wert verwendet wurde, ist die Zeitskala in den Abbildungen 144 und 145 nur relativ anzusehen, weshalb keine Werte angegeben wurden. Werden die Geschwindigkeitskonstanten, wie oben gesagt, verwendet, entspricht der dargestellte Zeitraum 5000 Sekunden.

Konzentration in mol/l

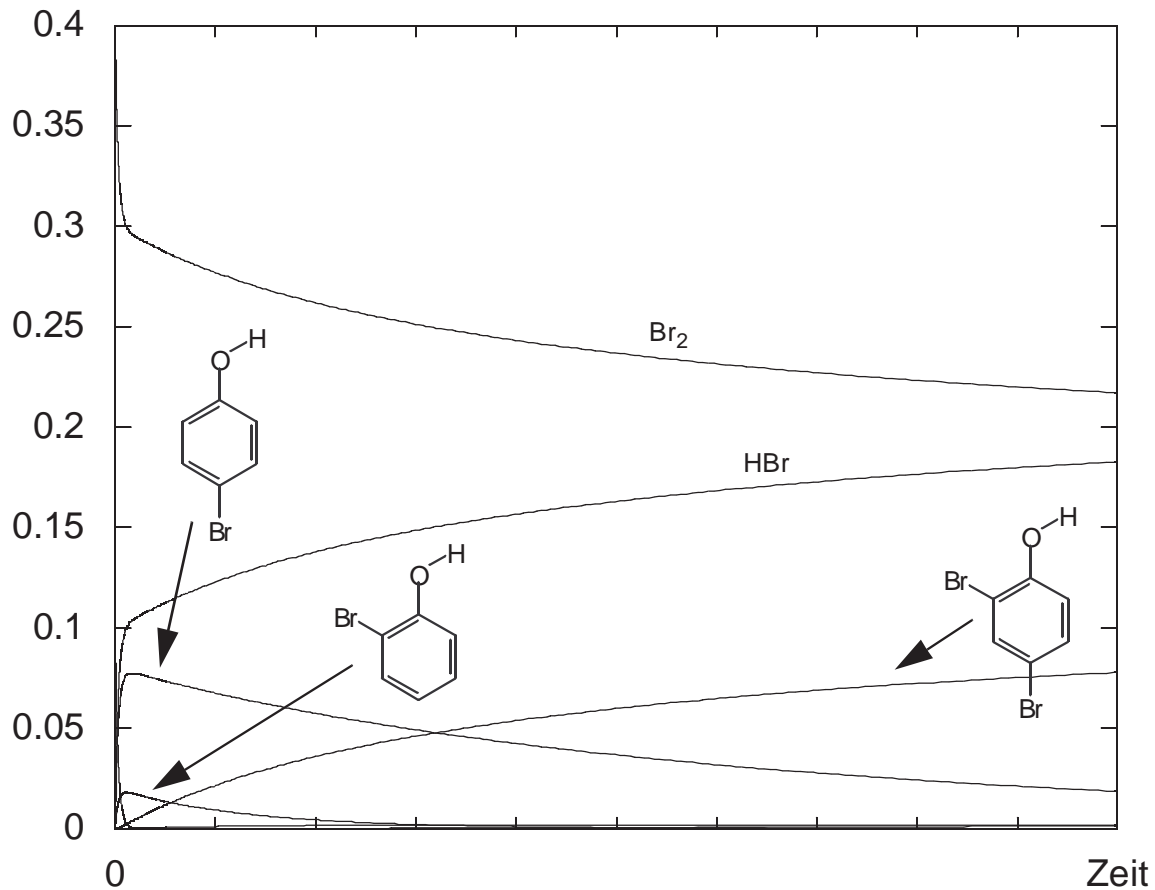


Abbildung 144: Konzentrationsverläufe der Edukte und Produkte der Bromierung von Phenol.

Um den Konzentrationsverlauf der nur langsam und in geringen Mengen auftretenden Verbindungen besser sehen zu können, zeigt Abbildung 145 die Konzentrationen bis 0.1 mol/l über den gleichen Zeitraum als vergrößerten Ausschnitt.

Konzentration in mol/l

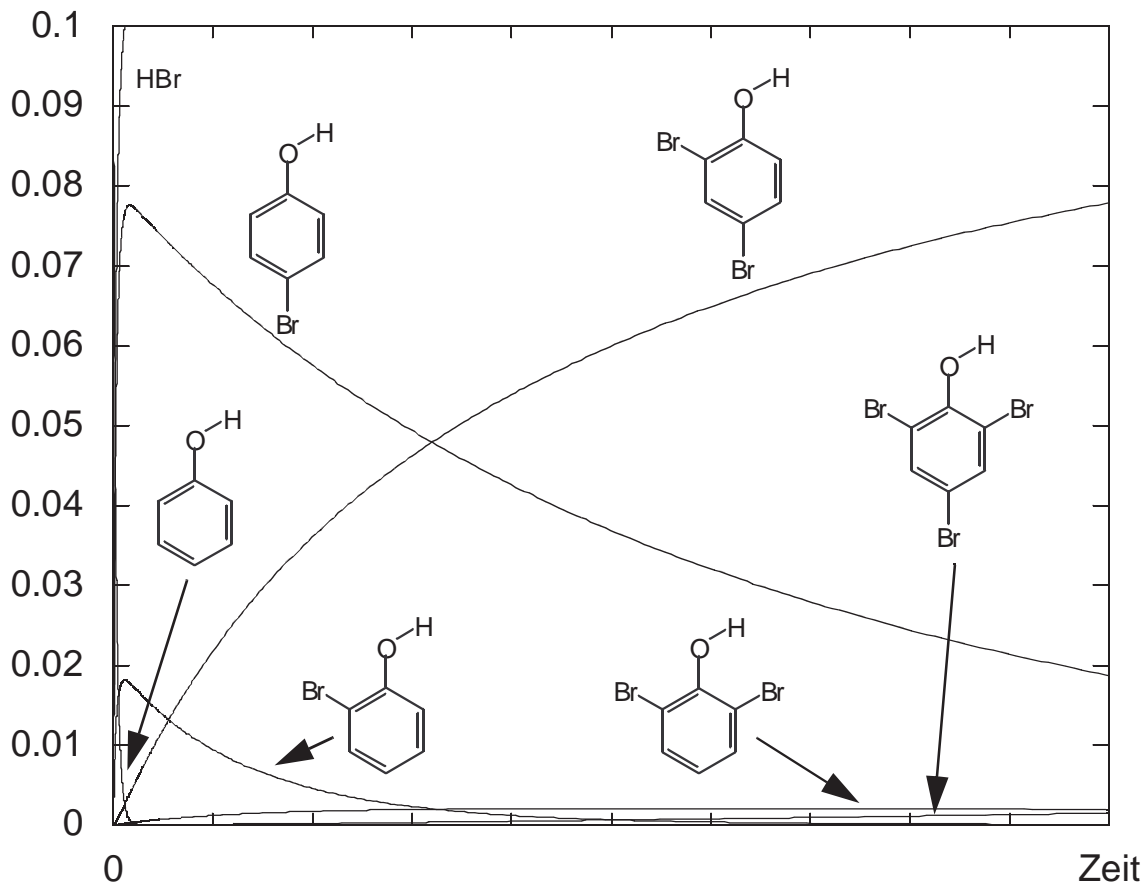


Abbildung 145: Konzentrationsverläufe der Edukte und Produkte der Bromierung von Phenol. Ausschnitt bis zu einer Konzentration von 0.1 mol/l.

In Abbildung 145 ist auch der Verlauf des nur wenig gebildeten *o*-,*o'*-Dibromphenols sowie des Tribromphenols gegen Ende des betrachteten Zeitraums zu erkennen.

Berechnungszeit auf einer SparcStation 10/50 mit 32 MB in Min.:Sek.: 1:37

7.1.2. Kontinuierliche Prozesse, wie sie großtechnisch durchgeführt werden

Soll großtechnisch ein Produkt im Maßstab von mehreren Tonnen hergestellt werden, möchte man die bei Chargenprozessen anfallenden Totzeiten des Reaktors, in denen er entleert und wieder neu befüllt wird, vermeiden. Deshalb werden die Reaktionen kontinuierlich geführt, wobei fortlaufend die Edukte in das Reaktionsgefäß zugegeben und die Produkte entnommen werden. Strömungsrohre zeigen das gleiche kinetische Verhalten wie Chargenprozessen und erzeugen damit auch das gleiche Produktverhältnis. Sie werden allerdings nur selten eingesetzt, da sie durch ihre Trägheit bei der Erkennung von Konzentrationsabweichungen schlecht zu regeln sind. Andere kontinuierlich betriebene Reaktortypen sind der kontinuierliche, ideale Rührkessel (KIK) und die Rührkesselkaskade (siehe 6.7.1).

7.1.2.1. Estersynthese mit gleichzeitiger Extraktion

Im folgenden wird ein kontinuierlicher Rührkessel mit einer wässrigen und einer organischen Phase vorgestellt, in dem aus kontinuierlich zugeführtem Ethanol und Essigsäure Essigester synthetisiert und gleichzeitig in die organische Phase extrahiert wird. Die Ausgangsmaterialien

werden dabei in die wäßrige Phase zugegeben, in der die Reaktion der Estersynthese abläuft. Die nicht mischbare, organische Phase, in der keine Reaktionen stattfinden, wird mit dem größten Teil des Essigesters kontinuierlich entnommen. (Siehe Abbildung 146). Die Volumina der beiden Phasen betragen je 5 Liter. Pro Sekunde fließen 5 ml Lösungsmittel zu jeder Phase zu und auch wieder ab, wobei im Zufluß der wäßrigen Phase pro Sekunde 2 mmol Essigsäure und 2.2 mmol Ethanol zugehen. Zu Beginn der Simulation ist der Reaktor mit reinen Lösungsmitteln gefüllt, d.h. alle Anfangskonzentrationen sind null.

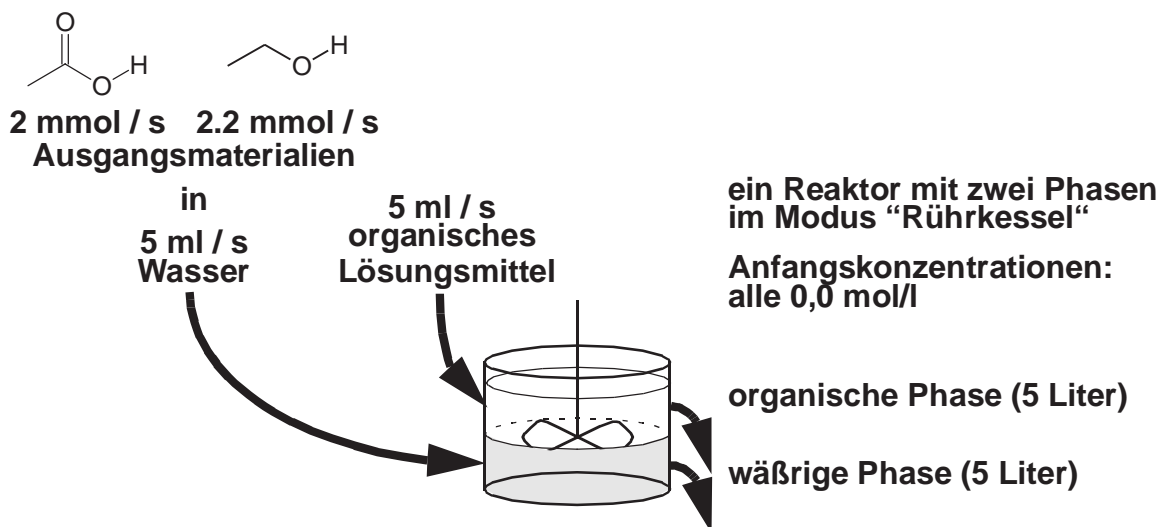


Abbildung 146: Reaktor mit zwei Phasen für die kontinuierliche Synthese von Essigester.

Das Beispiel der kontinuierlichen Synthese von Essigester soll nur dazu dienen, zu zeigen, daß mit EROS7 solche Berechnungen durchgeführt werden können, weshalb für die einzelnen Geschwindigkeitskonstanten Werte angenommen wurden und keine experimentellen Werte verwendet wurden. Die Geschwindigkeitskonstanten wurden mit $0.05 \text{ l}/(\text{s}\cdot\text{mol})$ für die Esterbildung und für die Hydrolyse, die nach pseudoerster Ordnung verläuft, mit $0.002 \text{ l}/\text{s}$ (gleich $k\cdot[\text{H}_2\text{O}]$) angenommen. Diese Reaktionen finden nur in der wäßrigen Phase statt. Für die Diffusionen in die andere Phase wurden die Werte $0.05 \text{ mol}/((\text{mol/l})\cdot\text{s})$ in beiden Richtungen für Ethanol und Essigsäure angenommen sowie 0.5 für den Ester von der wäßrigen in die organische Phase und $0.025 \text{ mol}/((\text{mol/l})\cdot\text{s})$ in umgekehrter Richtung. Diese Werte sind in Abbildung 147 zusammengefaßt.

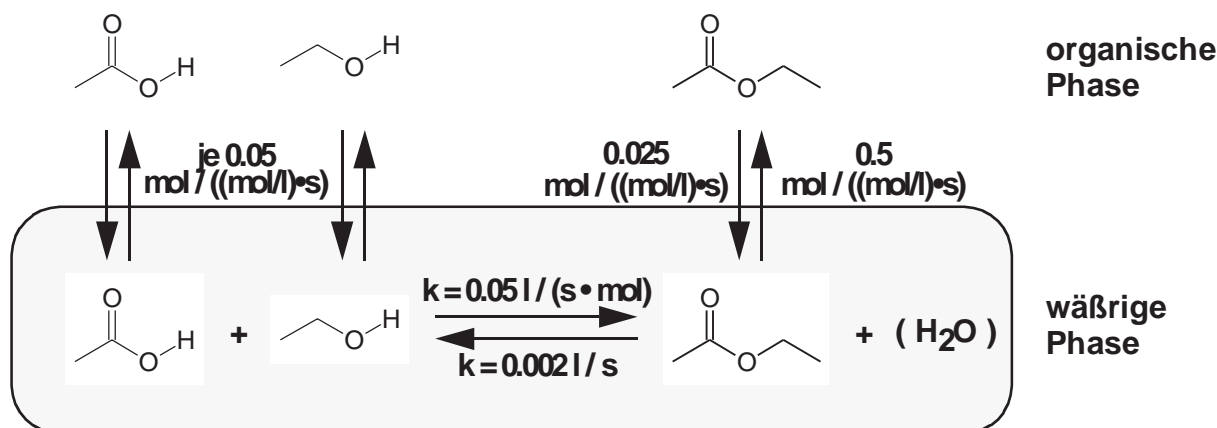


Abbildung 147: Angenommene Geschwindigkeitskonstanten für die Prozesse der kon-

tinuierlichen Essigestersynthese mit gleichzeitiger Extraktion.

In den Regeln werden für die Diffusion die in Abbildung 147 angegebenen Werte geteilt durch das Volumen der Ausgangsphase von fünf Litern (0.01, 0.005 und 0.1 • 1/s) als Reaktivität für die Phasenübergänge angegeben, da EROS7 die Konzentrationsänderung und nicht den Stoffstrom benötigt. Die Konzentrationsänderung in der Zielphase wird von EROS7 automatisch aus dem Verhältnis der Volumina der Phasen ermittelt.

Die sich ergebenden Konzentrationsverläufe der Substanzen in den beiden Phasen sind in Abbildung 148 gezeigt. Die schwarzen Linien sind die Konzentrationen in der wäßrigen Phase und die grauen die in der organischen. Die Konzentrationen der Verbindungen im abfließenden Wasser bzw. organischen Lösungsmittel entsprechen denen in den beiden Phasen.

Konzentration
in mol / l

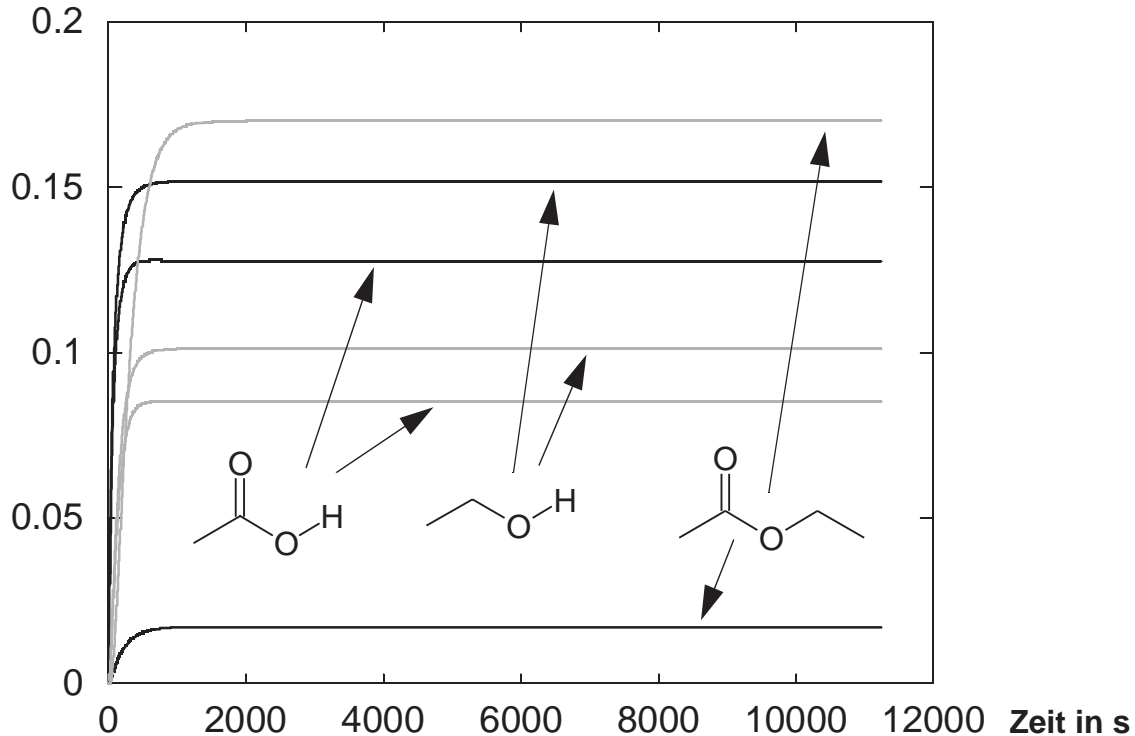


Abbildung 148: Konzentrationsverläufe für die kontinuierliche Bildung von Essigester mit gleichzeitiger Extraktion. Die schwarzen Kurven sind die Konzentrationen in der wäßrigen und die grauen die in der organischen Phase.

Berechnungszeit auf einer SparcStation 10/50 mit 32 MB in Min.:Sek.: 18:29 (Gear)

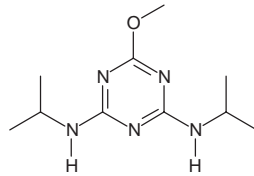
Die Methoden Runge-Kutta und Runge-Kutta-Merson beendeten ihre Berechnungen sehr rasch, führen jedoch keine Integration durch.

7.1.3. Abbau von Umweltchemikalien

Beim Abbau von Umweltchemikalien handelt es sich vorwiegend um Reaktionen pseudoerster Ordnung. Von T. Kostka [62] wurden Regeln entwickelt, die den Abbau von s-Triazinen, die zu den Herbiziden gehören, durch reduktive Dealkylierung und Hydrolyse simulieren. Ein Beispiel hierfür wurde mit dem Abbau von Atrazin bereits im Kapitel 6.9.1 gezeigt. Ein weiteres Beispiel für den Abbau von s-Triazinen ist Prometon.

7.1.3.1. Abbau von Prometon

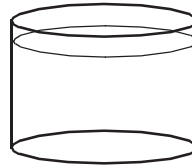
Für den Abbau von Umweltchemikalien durch Reaktionen pseudoerster Ordnung setzt man einen Reaktor mit einer Phase im Modus „monomolekular“ ein. Wird die Chemikalie, wie bei den s-Triazinen, zu einer bestimmten Zeit ausgebracht (nicht ständig), wird für sie eine Anfangskonzentration angegeben, aber kein kontinuierlicher Zufluß. Als Reaktionstypen werden die reduktive Dealkylierung und die Hydrolyse eingesetzt, die die im Überschuß vorliegenden Reaktionspartner Wasser und den formal für Reduktionsäquivalente verwendeten Wasserstoff während der Reaktion einschleusen.



(Prometon)

Ausgangsmaterial
Anfangskonzentration:
0.1 mol / l

eine Phase im Modus
„monomolekular“



zwei Reaktionstypen pseudoerster Ordnung:
reduktive Dealkylierung
Hydrolyse

Abbildung 149: Phasen und Anfangsbedingungen für den Abbau von Prometon.

Ausgehend von Prometon werden mit den beiden Reaktionstypen Hydrolyse und reduktive Dealkylierung von der Phase im Modus „monomolekular“ die Reaktionen erzeugt, wie sie in Abbildung 150 gezeigt sind.

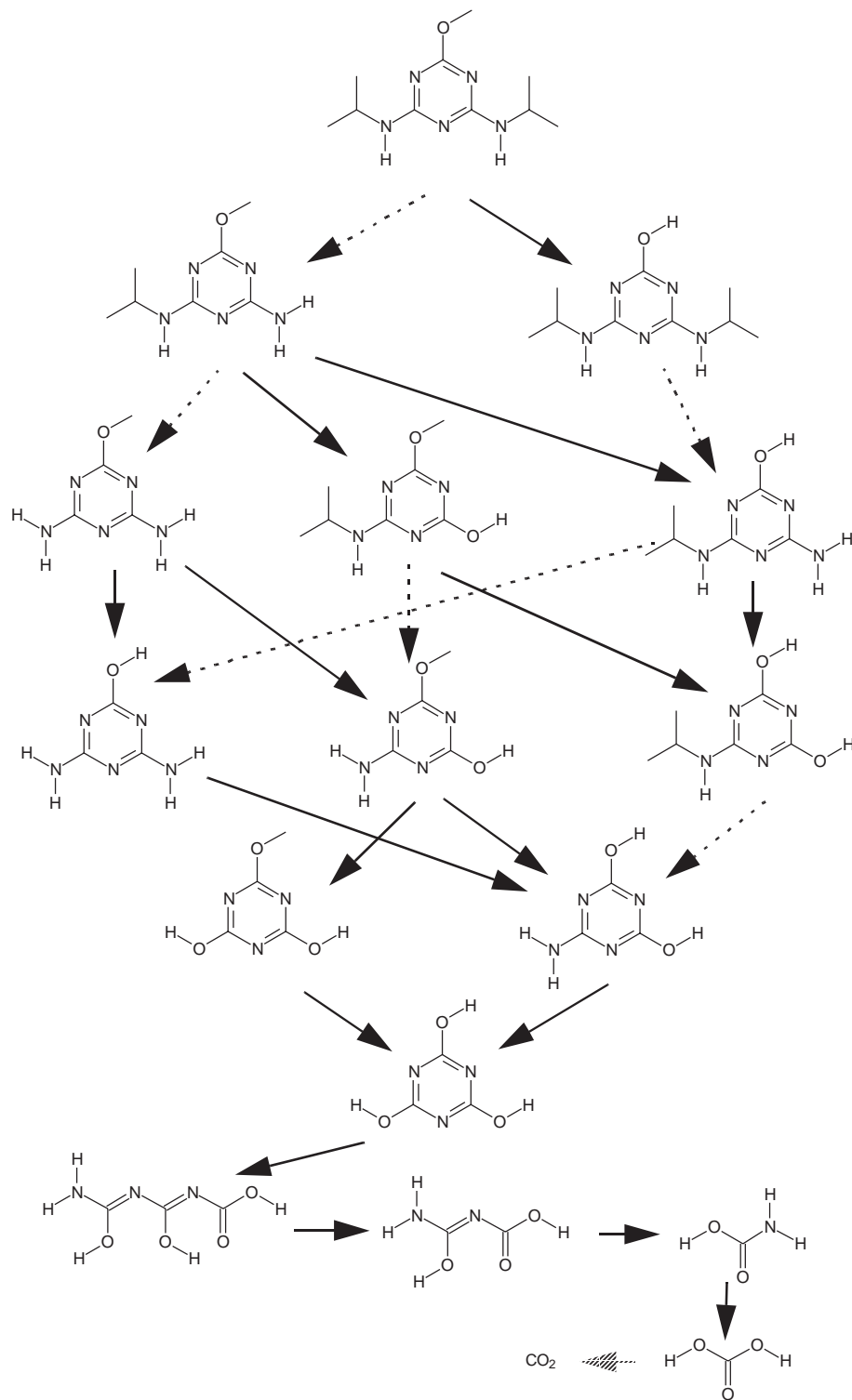


Abbildung 150: Abbau von Prometon durch reduktive Dealkylierung (gestrichelt) und Hydrolyse (durchgezogen). Die Reaktionen werden als Reaktionen pseudoerster Ordnung durchgeführt. Der formal für die Dealkylierung verwendete Wasserstoff und das Wasser für die Hydrolyse, die in großem Überschuss vorliegen, werden während der Reaktion hinzugefügt. Bei der Reduktion wird Propan abgespalten, während bei der Hydrolyse Methanol, Ammoniak und Kohlendioxid abgespalten werden können.

Unter Berücksichtigung der Geschwindigkeitskonstanten für die Reaktionen ergeben sich Konzentrationsverläufe für Prometon und seine Abbauprodukte, wie sie in Abbildung 151 gezeigt sind.

**Konzentration
in mol / l**

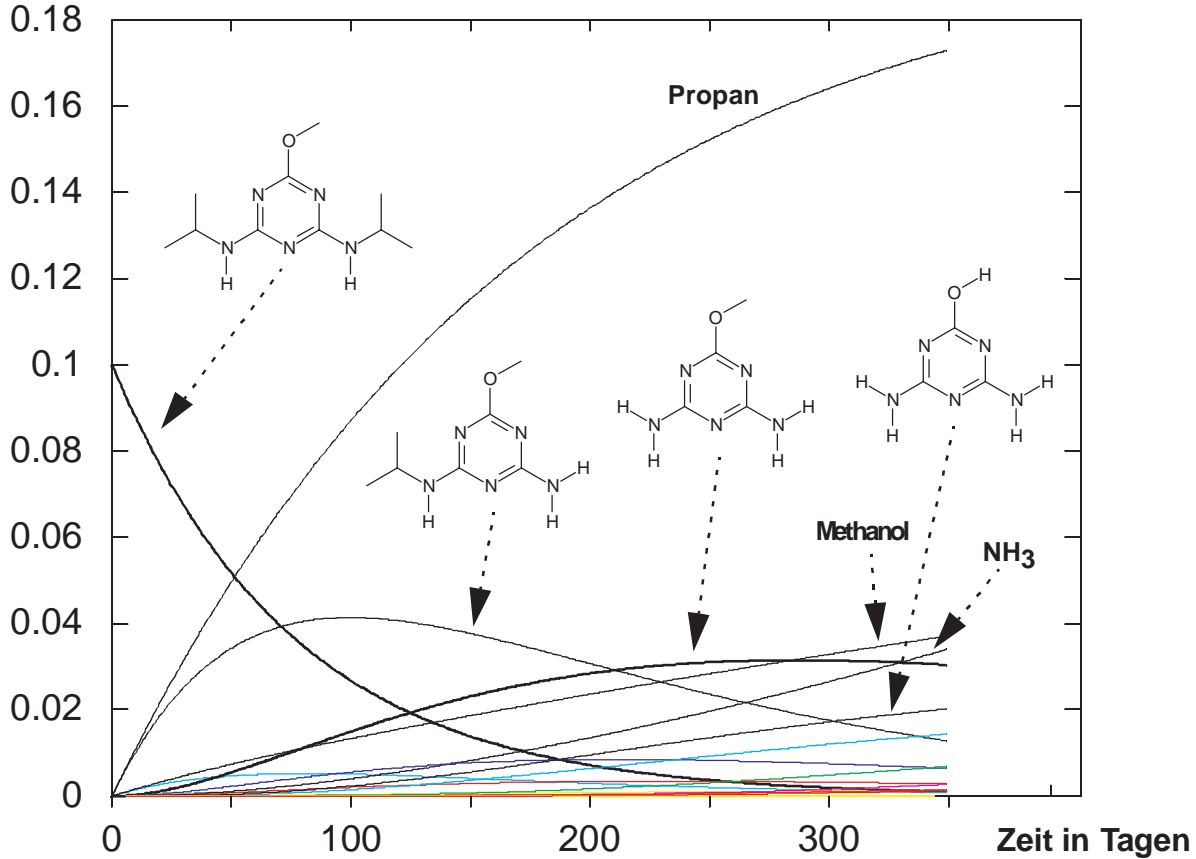


Abbildung 151: Konzentrationsverläufe für Prometon und seine Abbauprodukte.

In Abbildung 151 sind die Konzentrationsverläufe für den Abbau von Prometon angegeben. Die Reaktionszeit wurde dabei so gewählt, daß am Ende 99% des Prometons reagiert haben. Alternativ dazu wurde die Berechnung auch nur für eine Halbwertszeit (Umsatz von 50%) durchgeführt. Die Berechnungszeiten für diese beiden Umsätze und die beiden Integrationsverfahren Gear-Algorithmus (langsamstes) und Methode nach Runge-Kutta-Merson (schnellstes) sind in der nachfolgenden Tabelle zusammengefaßt. Es wurden dabei alle drei Möglichkeiten der Regeln getestet: in C++ geschriebene und gelinkte Regeln, Tcl-Regeln, die interpretiert wurden, und compilierte Tcl-Regeln.

Methode	Gear		Runge-Kutta-Merson	
Umsatz	0.5	0.99	0.5	0.99
C++	1:20	6:38	0:47	1:57
Tcl compiliert	2:55	8:15	2:30	3:31
Tcl interpretiert	2:54	8:21	2:25	3:29

Tabelle 1: Berechnungszeiten auf einer SparcStation 10/50 mit 32 MB in Min.:Sek.

Es zeigt sich deutlich die höhere Geschwindigkeit der Methode nach Runge-Kutta-Merson, die bei der längeren Berechnungszeit für den Umsatz von 99% im Verhältnis noch größer ist.

Schreibt man die Regeln in der Programmiersprache C++, ist die benötigte Zeit immer etwa um 1:30 Min.:Sek. kürzer. Vergleicht man schließlich die Zeiten der compilierten mit denen der interpretierten Tcl-Regeln, so kann man den erwarteten Geschwindigkeitsvorteil nicht erkennen. Es ist sogar so, daß der EROS7-Lauf mit compilierten Tcl-Regeln meist etwas länger gedauert hat. Diese kleinen Unterschiede liegen allerdings innerhalb der Genauigkeit der Zeitmessung. Alle Zeiten wurden als benötigte User- und System-Time für den Prozeß unter UNIX bestimmt. Die Abrechnung dieser Zeiten ist allerdings nicht ganz unabhängig von der Systemauslastung, weshalb es von Messung zu Messung zu Schwankungen kommt. Da beim Übergang von interpretierten zu compilierten Tcl-Regeln die flexible Anbindung und damit die Änderungsmöglichkeit verloren geht und dem kein Geschwindigkeitsgewinn gegenübersteht, kann auf die Möglichkeit, compilierte Tcl-Regeln zum System zu binden, verzichtet werden, zumal dafür ein entsprechender Compiler mit dazugehöriger Bibliothek Voraussetzung sind.

7.1.3.2. Jährliche Ausbringung von s-Triazinen

Neben der Simulation der einmaligen Ausbringung können mit EROS7 auch die Konzentrationsverläufe bei einer jährlichen Ausbringung von Prometon studiert werden. Es ist bekannt, daß es nach der Ausbringung bestimmter s-Triazinherbizide bezüglich mancher Pflanzen im darauffolgenden Jahr Unverträglichkeiten gibt. Da das Herbizid selbst nach einem Jahr zu etwa 99% abgebaut ist, dürfte diese an einem Abbauprodukt liegen, das sich noch im Boden befindet. Bei einer jährlichen Ausbringung würden sich diese Produkte im Boden anreichern. Dies läßt sich mit EROS7 vorhersagen und ist für den Zeitraum von gut drei Jahren in Abbildung 152 gezeigt.

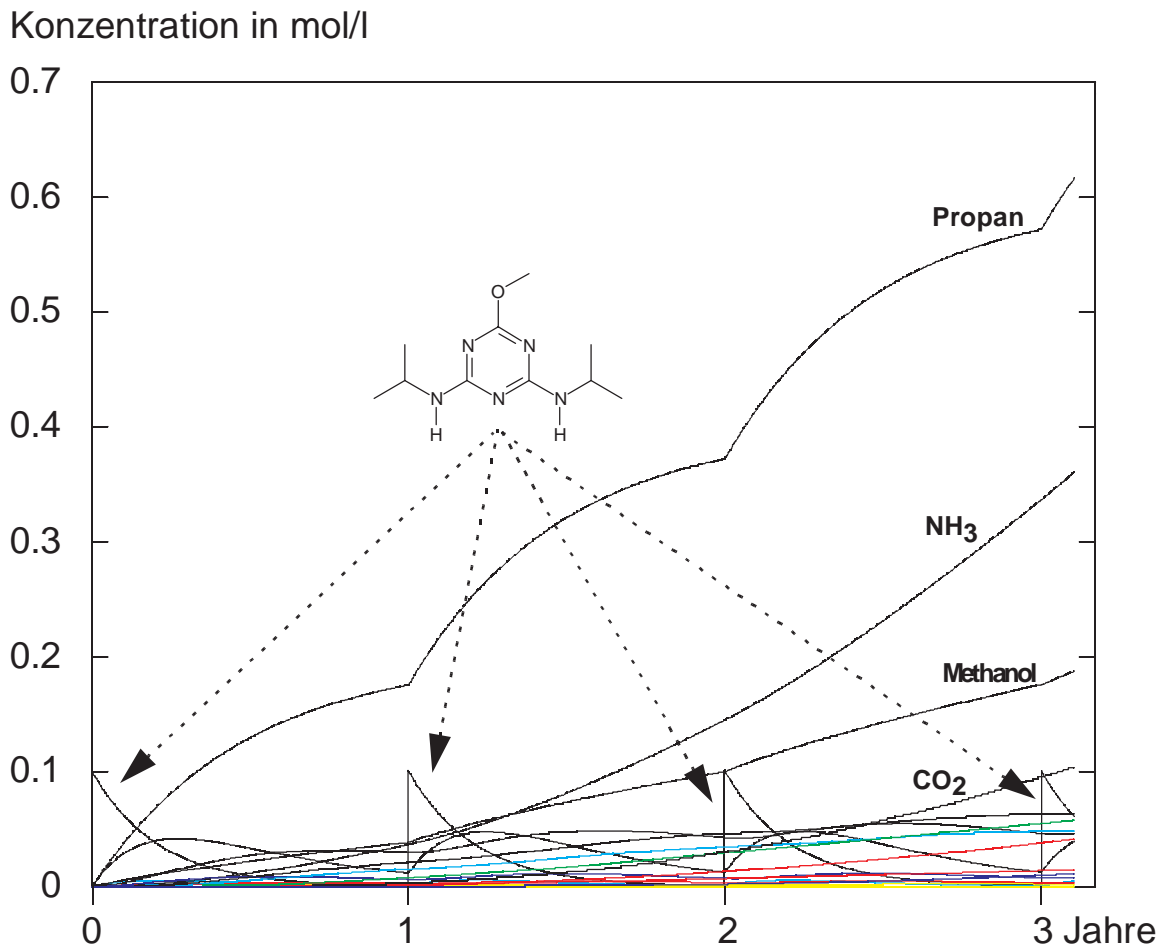


Abbildung 152: Jährliche Ausbringung der gleichen Menge Prometon.

Da man in Abbildung 152 vorwiegend die Konzentrationsverläufe der Endabbauprodukte Propan, Ammoniak, Methanol und Kohlendioxid sieht, an Zwischenprodukten jedoch wenig zu erkennen ist, wir in Abbildung 153 der Konzentrationsbereich bis 0.125 mol/l herausvergrößert.

Konzentration in mol/l

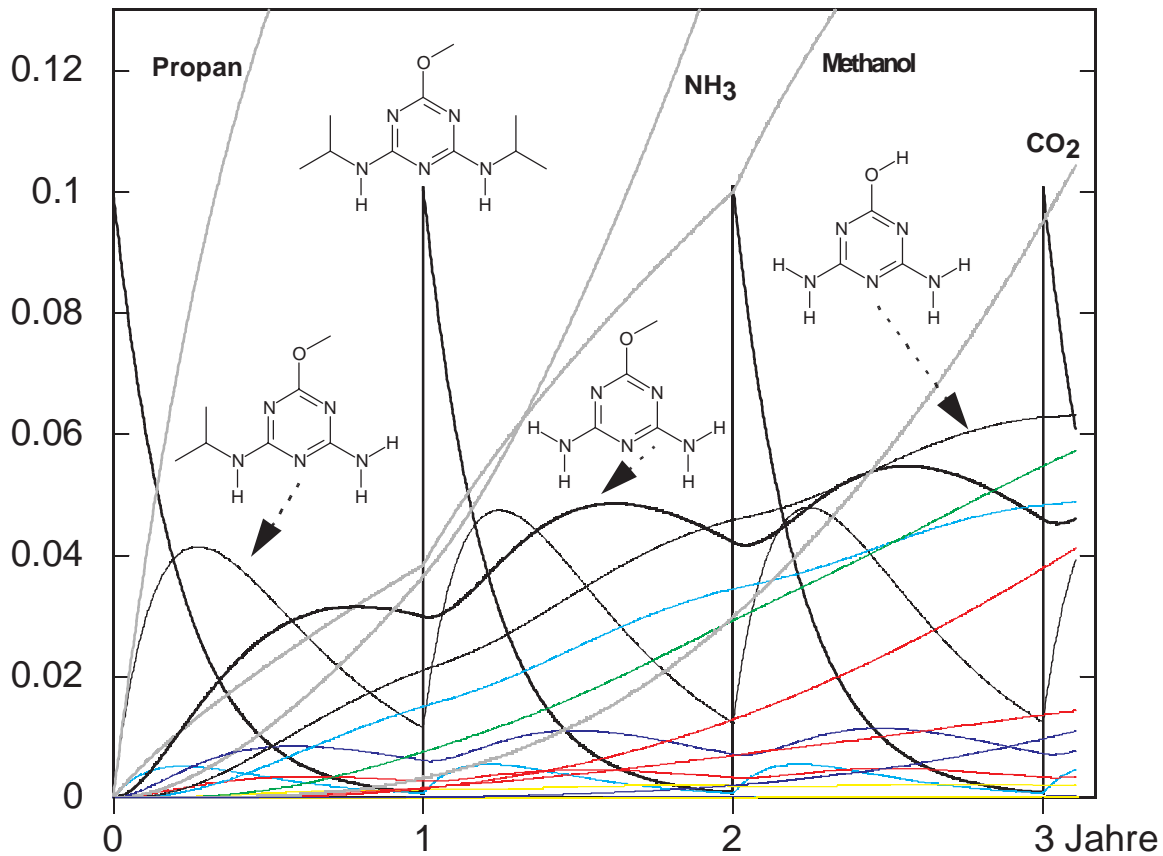


Abbildung 153: Jährliche Ausbringung der gleichen Menge Prometon. Ausschnitt für Konzentrationen bis 0.125 mol/l.

Berechnungszeit auf einer SparcStation 10/50 mit 32 MB in Min.:Sek.: 42:55 (Gear, C++)

In Abbildung 153 kann man erkennen, daß sich ein Teil der Abbauprodukte von Prometon über die Jahre hin im Boden anreichern.

7.1.4. Pharmakokinetik

Analog zum Abbau von Umweltchemikalien kann auch die Metabolisierung von Arzneimitteln mit EROS7 simuliert werden. Da in den folgenden Abschnitten gezeigt werden soll, wie mehrere Phasen für die verschiedenen Kompartimente eingesetzt werden können, dient ein sehr einfacher Weg der zweistufigen Metabolisierung als Beispiel. Es handelt sich dabei um ein hypothetisches Beispiel.

Stehen die entsprechenden kinetischen Daten eines Medikaments zur Verfügung, kann diese Simulation inklusive der in den nächsten Abschnitten folgenden Erweiterungen des Modells natürlich auch für jede in der Praxis eingesetzte Arznei mit EROS7 durchgeführt werden. Dies gilt auch für komplexere Abbaumechanismen und Kompartimentierungen.

7.1.4.1. Abbaureaktionen

In Abbildung 154 ist die einfache, zweistufige Metabolisierung gezeigt, die als Abbauweg des Medikaments angenommen wurde. Da die Strukturen, die für die Berechnung verwendet wur-

den, keinen Bezug zu Arzneimitteln haben, sondern nur die angenommenen Reaktionen pseudoerster Ordnung eingehen können, sind ihre Strukturen nicht angegeben. Sie würden nur verwirren. Alle Phasen, auch die der später hinzugefügten Kompartimente, erzeugen nur Reaktionen erster oder pseudoerster Ordnung und sind deshalb im Modus „monomolekular“.

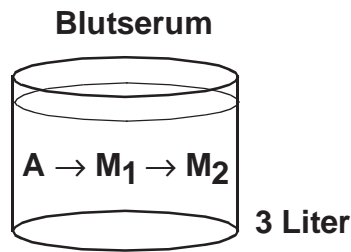


Abbildung 154: Schrittweise Metabolisierung eines Arzneimittels A zu den Metaboliten M_1 und M_2 .

Dieser zweistufige Abbau wurde gewählt, da Arzneimittel häufig durch Einführung einer polaren Gruppe und anschließende Bindung z.B. an Glutathion metabolisiert werden [68]. Für die Abbaugeschwindigkeiten wurde angenommen, daß der erste Schritt mit einer Halbwertszeit von 3.5 Stunden und der zweite von 10 Stunden abläuft. Daraus ergeben sich Geschwindigkeitskonstanten von $5.5 \cdot 10^{-5}$ und $1.9 \cdot 10^{-5}$ 1/s. Bei einer Anfangskonzentration von 0.01 mol/l folgen daraus die in Abbildung 155 dargestellten Konzentrationsverläufe.

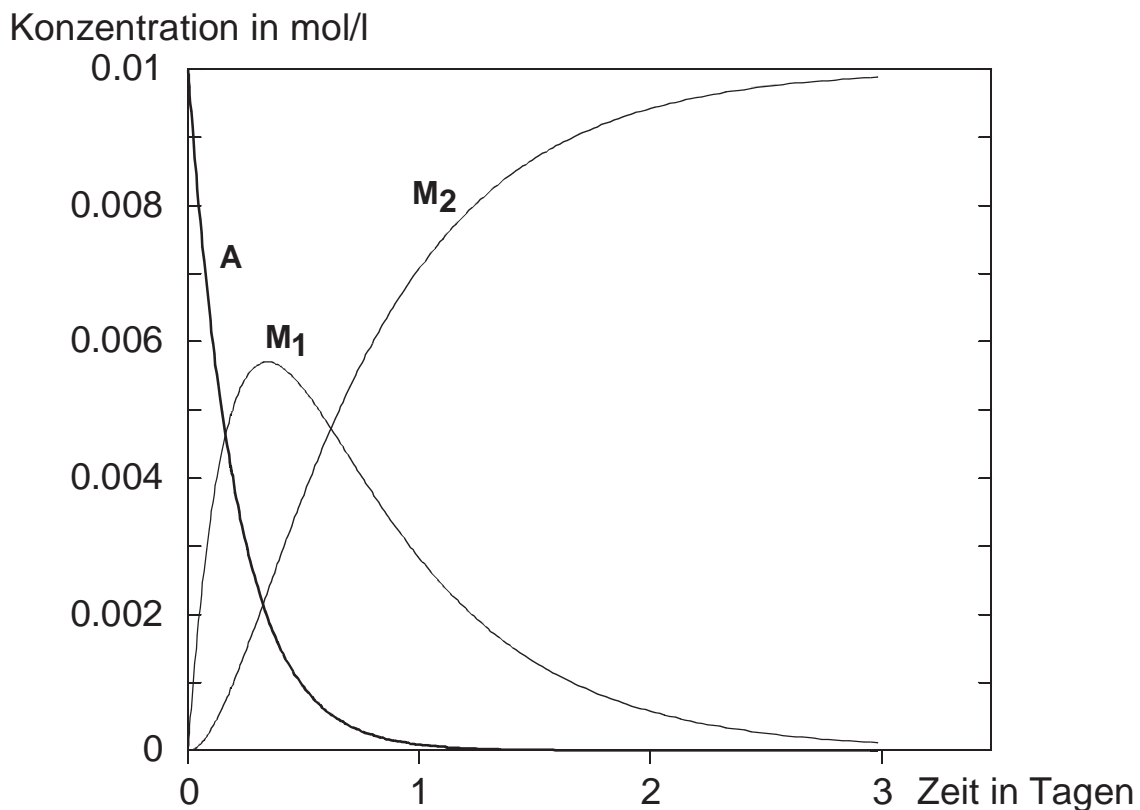


Abbildung 155: Zweistufige Metabolisierung des Arzneimittels A.

7.1.4.2. Mehrfachdosierung

Wird die Arznei A nun alle acht Stunden mit der gleichen Menge verabreicht, ergeben sich die Konzentrationsverläufe, die in Abbildung 156 gezeigt sind.

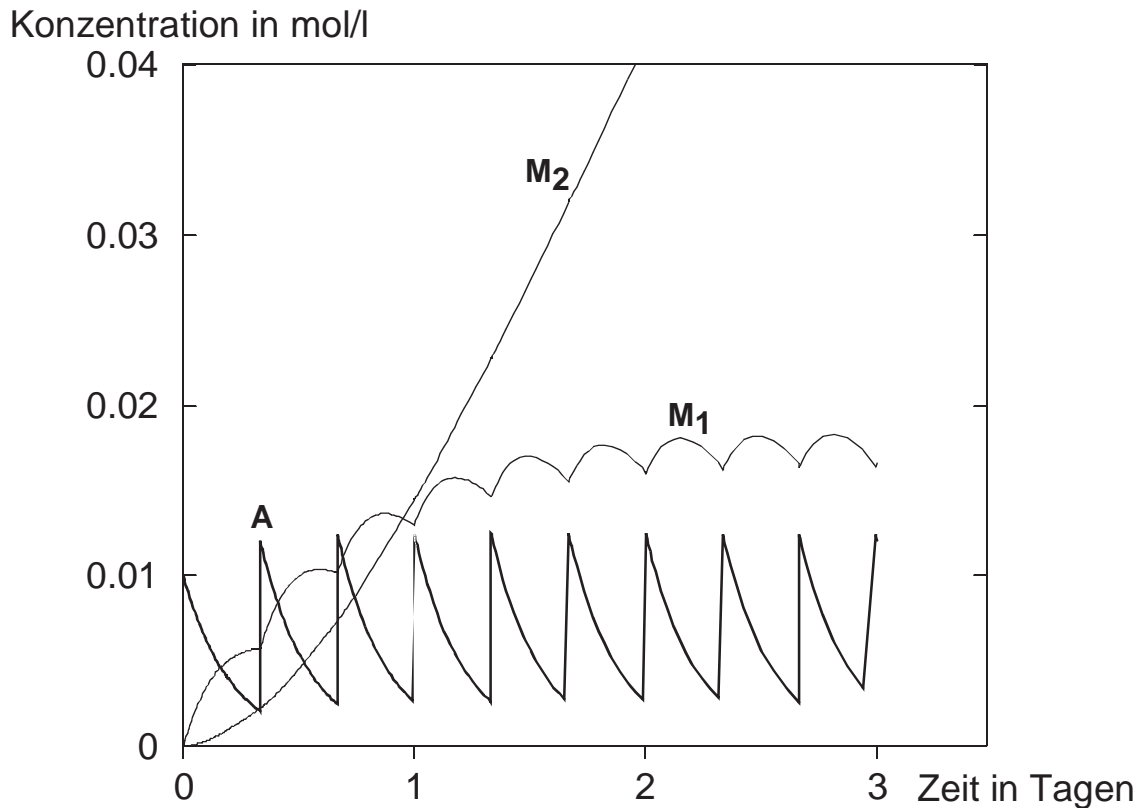


Abbildung 156: Konzentrationsverläufe der Arznei A und ihrer Metaboliten bei einer Verabreichung der gleichen Menge alle acht Stunden.

Abbildung 156 zeigt einen steilen Anstieg der Konzentration des Metaboliten M_2 , der in Realität nicht eintritt, da die Verbindung ausgeschieden wird. Damit auch die Konzentration für den Metaboliten M_2 richtig wiedergegeben wird, soll das Modell nun verbessert werden.

7.1.4.3. Ausscheidung

Zur Simulation der Ausscheidung der verschiedenen Verbindungen wird zum Reaktor mit einer Phase eine zweite hinzugefügt, die die ausgeschiedenen Substanzen aufnimmt. Das Volumen der Phase für die Ausscheidungsprodukte wurde ebenfalls auf drei Liter gesetzt, damit man an den Konzentrationen sieht, welche Konzentrationen erzeugt worden wären, wenn die Substanzen nicht ausgeschieden würden. Darüber hinaus wird für alle Verbindungen, die ausgeschieden werden, ein Phasenübergang generiert, deren Geschwindigkeitskonstante die Ausscheidungsgeschwindigkeit angibt (siehe Abbildung 157). Für das Arzneimittel A und seine Metaboliten M_1 und M_2 werden verschiedene Ausscheidungsgeschwindigkeiten angenommen: Die Arznei selbst soll nicht ausgeschieden werden und die Metaboliten M_1 und M_2 sollen mit Halbwertszeiten von 12 und 1.5 Stunden das Serum des Patienten verlassen. Das Abbauprodukt M_1 wird im Gegensatz zu M_2 nur langsam ausgeschieden.

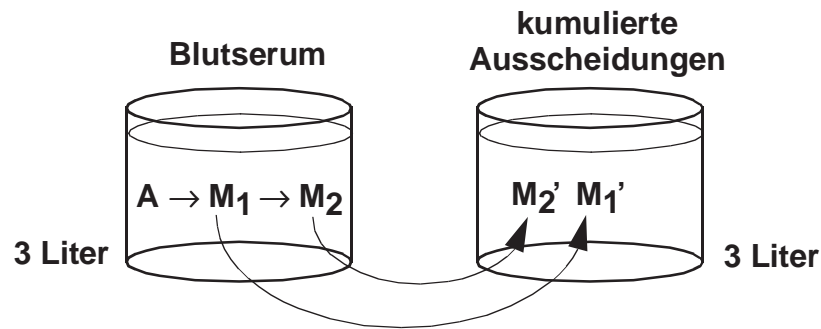


Abbildung 157: Phasenübergänge für die Ausscheidung der Verbindungen.

Mit diesem erweiterten Modell ergibt sich für den Metaboliten M_2 nach einer Anfangsphase eine nahezu konstante Konzentration (siehe Abbildung 157). M_1' und M_2' sind die kumulierten, ausgeschiedenen Metaboliten.

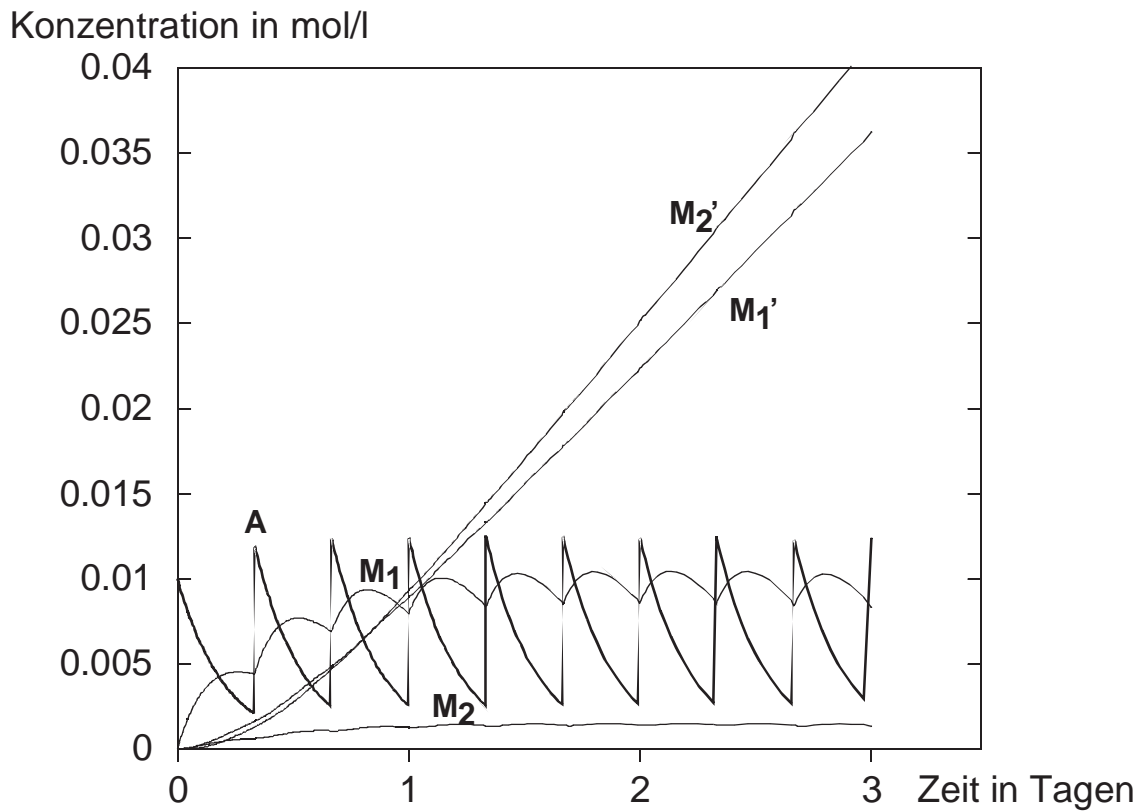


Abbildung 158: Konzentrationsverläufe des Abbaus der Arznei A unter Berücksichtigung der Ausscheidung der Metaboliten.

Trotz der wesentlich kleineren Ausscheidungsgeschwindigkeit des Metaboliten M_1 , sind die ausgeschiedenen Mengen M_1' und M_2' nahezu identisch. Dies liegt an der mit der Halbwertszeit von 10 Stunden langsameren Reaktion von M_1 zu M_2 , gegenüber einer Halbwertszeit von 3.5 Stunden für die Reaktion A zu M_1 . Daraus ergibt sich, wie in Abbildung 158 zu sehen ist, für M_1 auch eine deutlich höhere Konzentration im Blut als für M_2 .

7.1.4.4. Absorption des Arzneimittels

Die bisherigen Simulationen haben die intravenöse Applikation, also die Verabreichung durch eine Spritze, simuliert. Die häufigste Darreichungsform ist allerdings die Tablette oder auch ein

Dragee, bei denen der Wirkstoff nicht sofort in das Blut aufgenommen wird. Zur Simulation des Vorgangs der Absorption wird dem Reaktor eine weitere Phase hinzugefügt, sowie eine Phasentransfer für die Absorption des Medikaments erzeugt. (Siehe Abbildung 159) Auch das Volumen dieser neuen Phase wird auf drei Liter gesetzt, damit man erkennen kann, welche Konzentration eine vollständige Absorption bringen würde.

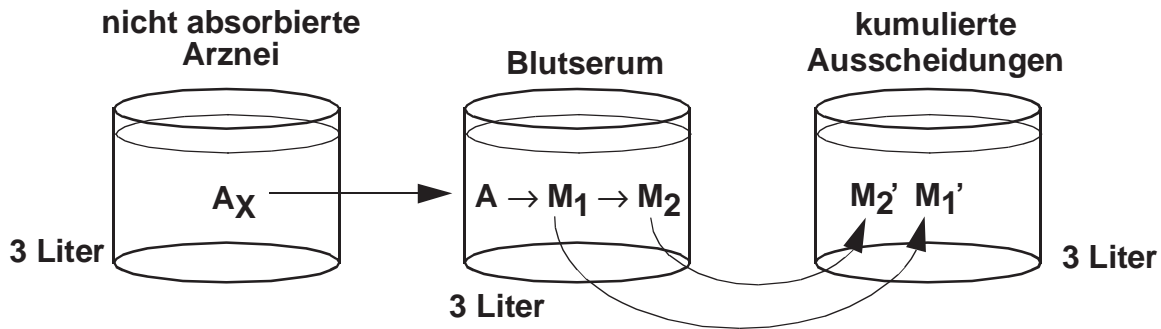


Abbildung 159: Absorption und Abbau der Arznei A mit Ausscheidung der Metaboliten.

Die Geschwindigkeitskonstante für die Absorption wurde mit $1.93 \cdot 10^{-4} \text{ 1/s}$ so gesetzt, daß sich für die Absorption eine Halbwertszeit von einer Stunde ergibt. Für die Arznei A im Blutserum erhält man dadurch etwas niedrigere, aber nicht so stark schwankende Konzentrationen (siehe Abbildung 160).

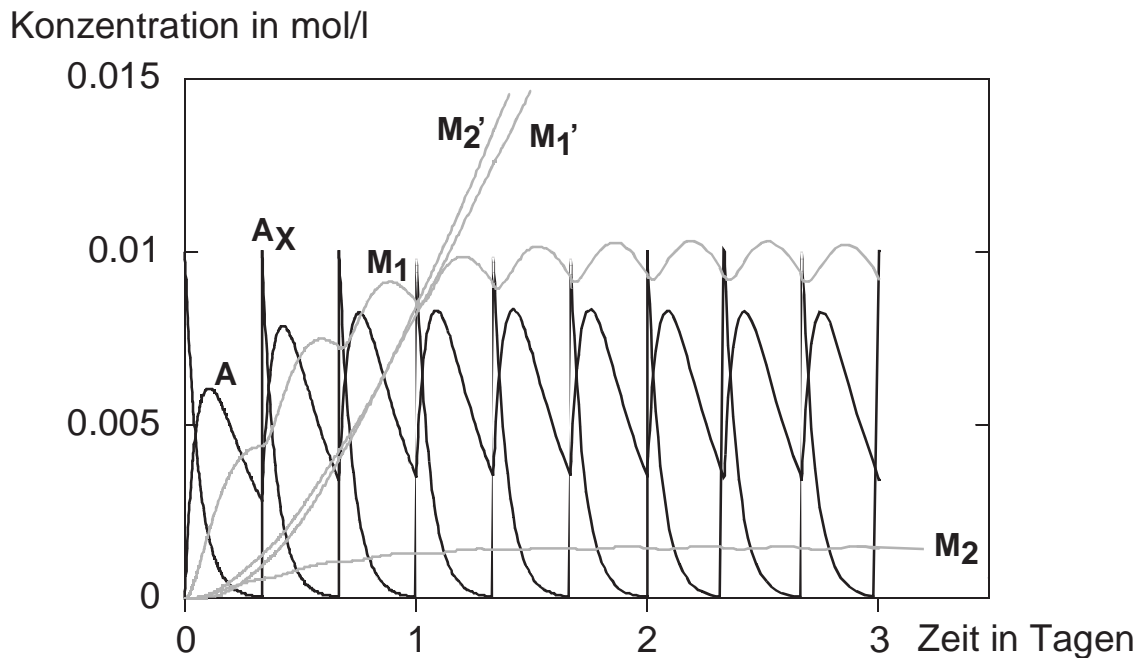


Abbildung 160: Konzentrationsverläufe mit Absorption und Ausscheidung. A_X ist der Teil der noch nicht ins Blutserum übergegangenen Arznei. A ist die Konzentration im Serum.

Die Erweiterung um die Absorption sowie die nachfolgenden Ergänzungen um das Gewebe als Medikamentenspeicher und ein Wirkkompartiment sind [66] und [67] entnommen. Die Werte der Geschwindigkeitskonstanten sind unabhängig von dieser Literatur angenommen worden.

7.1.4.5. Das Gewebe als Arzneimittelspeicher

Für eine noch genauere Vorhersage kann auch das Gewebe berücksichtigt werden, in das die

Arznei diffundieren kann. Finden im Gewebe keine Abbaureaktionen statt, dient es lediglich als Arzneimittelspeicher. Für die Vorhersage mit EROS7 wird dazu eine weitere Phase im Modus „monomolekular“ ergänzt und Phasenübergänge von der Phase, die das Serum repräsentiert, zur Gewebephase und zurück erzeugt. Die Größe des Gewebes wurde mit sieben Litern angenommen.

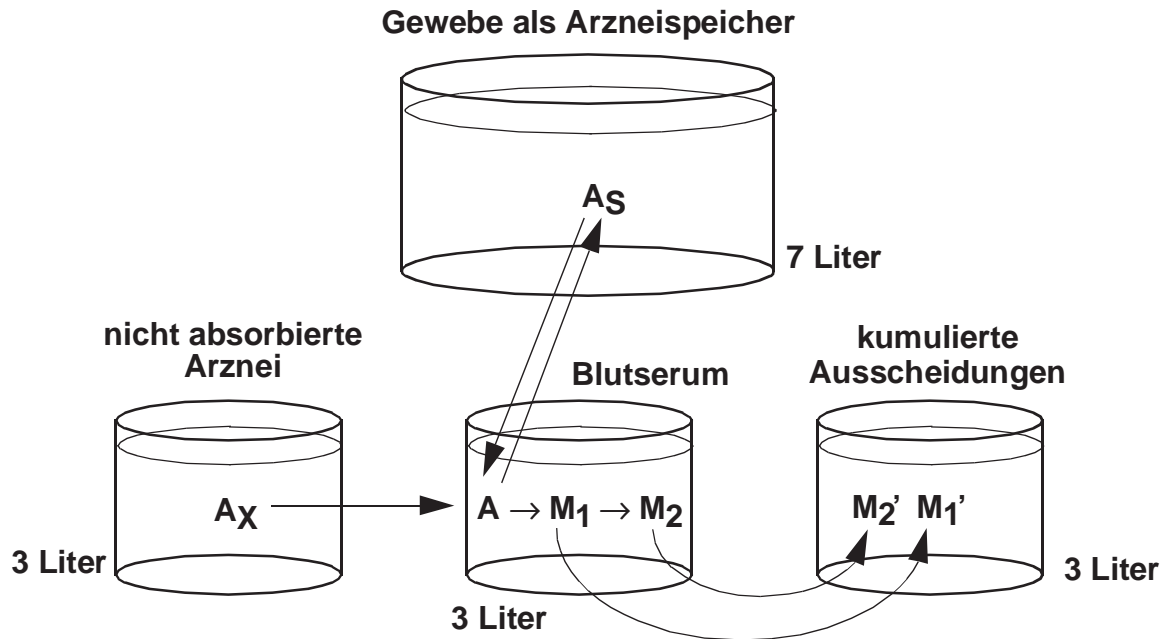


Abbildung 161: Phasen und generierte Reaktionen bei Berücksichtigung des Gewebes als Arzneimittelspeicher. A_S ist die Konzentration des Medikaments im Gewebe.

Für beide Richtungen des Transfers des Medikaments A in und aus dem Gewebe wurde die Geschwindigkeitskonstante mit $9.63 \cdot 10^{-5} \text{ 1/s}$ angenommen (entspricht einer Halbwertszeit für die Diffusion in die andere Phase von zwei Stunden). Da bei Phasenübergängen die Volumina der Ausgangsphasen in die Geschwindigkeitskonstante eingehen (siehe 6.9.1), entsprechen hier gleiche Geschwindigkeitskonstanten einem Verteilungskoeffizienten N von $0.43 = 3 \text{ l} / 7 \text{ l}$. In Abbildung 162 sind die sich unter Berücksichtigung des Gewebes als Arzneimittelspeicher ergebenden Konzentrationsverläufe dargestellt. A_S ist der Konzentrationsverlauf der Arznei im Serum A durch das Gewebe als Arzneimittelspeicher etwas gedämpft werden.

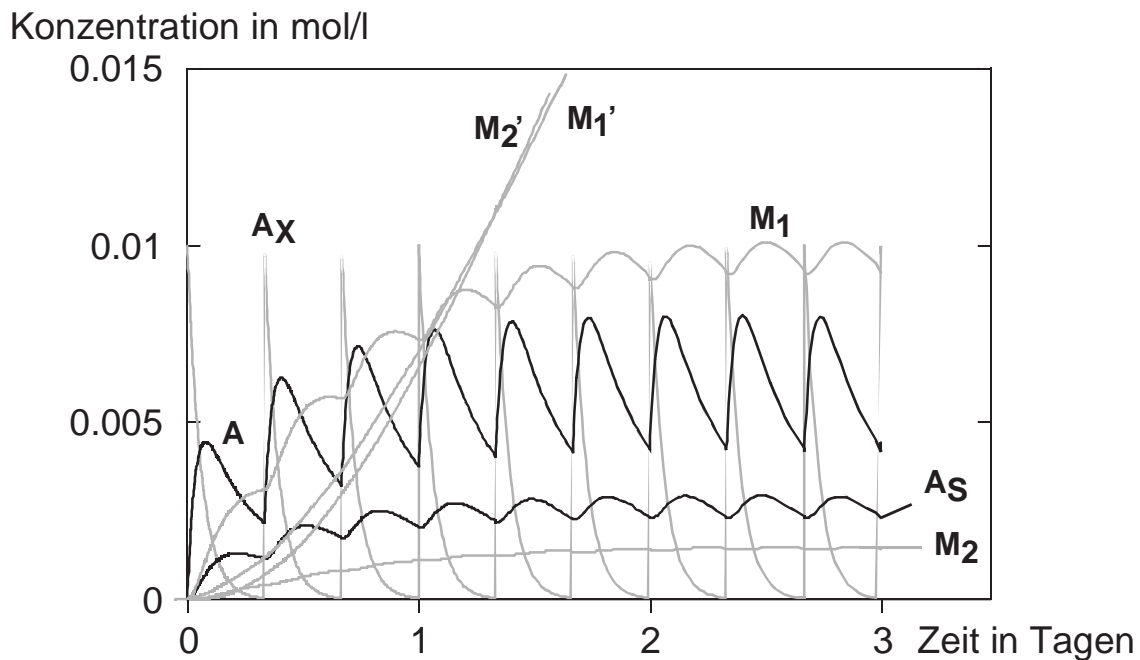


Abbildung 162: Konzentrationsverläufe unter Berücksichtigung des Gewebes als Arzneimittelspeicher.

7.1.4.6. Wirkkompartiment

Sehr häufig ist das Blutserum nicht der Ort, an dem das Medikament seine Wirkung entfaltet. Die Arzneikonzentration im Serum kann dann nur als Anhaltspunkt für die Wirkung genommen werden. Für eine genauere Simulation wurde deshalb eine weitere Phase mit nur einem Volumen von 0.01 Litern hinzugenommen, die das Wirkkompartiment darstellt. Die Konzentration A_W am Wirkort kann dann direkt für die Vorhersage der Wirkung eingesetzt werden. Für die Diffusion vom Blutserum an den Wirkort und zurück wurden für die Geschwindigkeitskonstanten Werte von $1.6 \cdot 10^{-7}$ und $2.4 \cdot 10^{-5}$ 1/s eingesetzt. Daraus folgt ein Verteilungskoeffizient N von 0.5 (siehe 7.1.4.5). Mit einer Halbwertszeit von 48 Minuten geht die Arznei am Wirkort A_W die Reaktion ein, die die Wirkung verursacht und das Produkt E bildet, das wiederum sofort ausgeschieden wird. Auf diese Weise kann man beispielsweise die „blood brain barrier“ simulieren. Das Zusammenspiel der Reaktionen und Phasen ist in Abbildung 163 gezeigt.

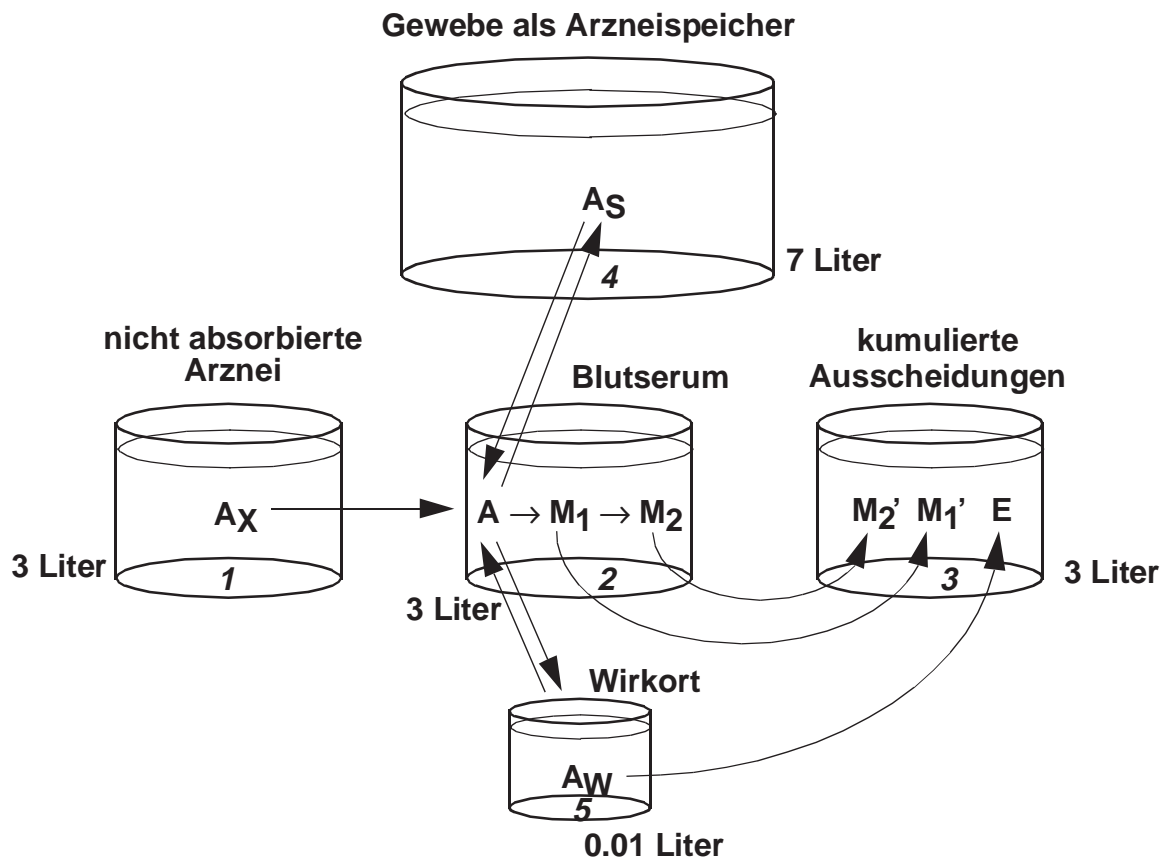


Abbildung 163: Reaktionen in und zwischen den Phasen mit Simulation des Gewebes als Arzneispeicher und Wirkort. Die kursiven Ziffern geben die Phasennummern an, wie sie für die folgenden Regelausschnitte verwendet wurden.

In Abbildung 164 sind die sich daraus ergebenden Konzentrationsverläufe dargestellt.

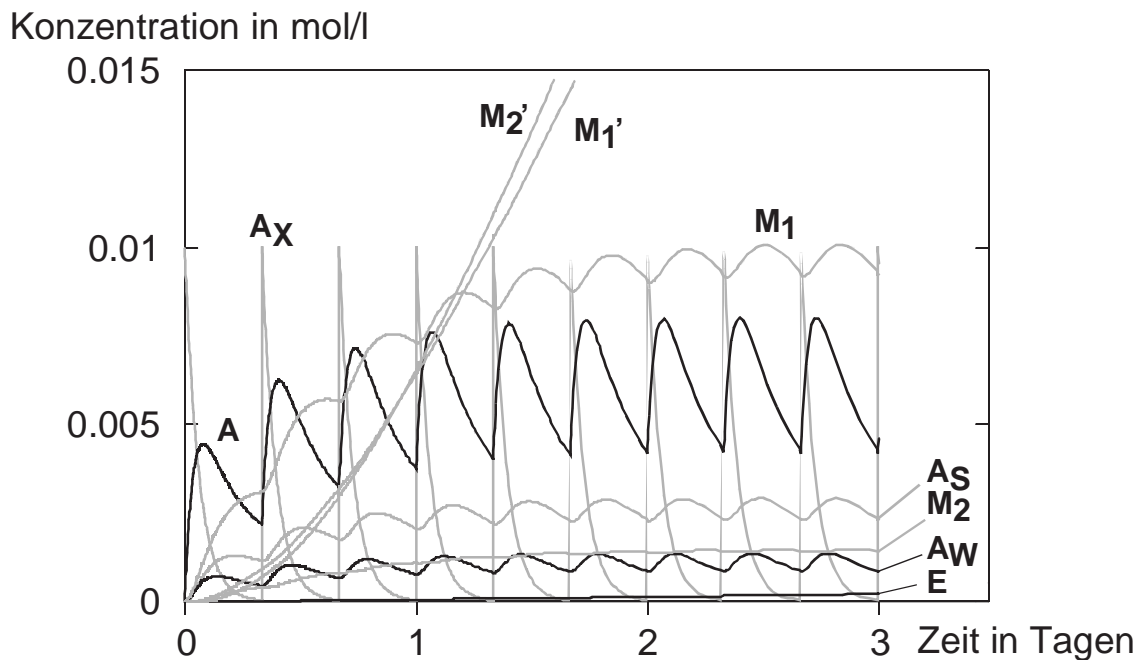


Abbildung 164: Konzentrationsverläufe der Arznei im Serum A und am Wirkort A_W

sowie die Summe der Abbauprodukte des Arzneimittels E, die eine Wirkung verursacht haben.

Berechnungszeit auf einer SparcStation 10/50 mit 32 MB in Min.:Sek.: 1:30 C++, 1:53 Tcl

Die nachfolgenden Ausschnitt der Regeln (Tcl) zeigen, wie das System der Phasen (Kompartimente) und die Reaktionstypen kodiert wurden.

System der Phasen:

```
proc Tcl_rule {} {

    global irule op sub_op rule_trace
    global_name
    global k
    global attrib_1 center_1 rule_1
    global attrib_2 center_2 rule_2
    global attrib_3 center_3 rule_3
    global rname rdate inphase phasepr phaseprop pcontacts in2gether
    global outphase kinmode minconc rxtime convlim multid sconc pvol
    switch $irule {
    GLOBAL {
        switch $op {
        INIT_RULES {
            # globale Einstellungen
            set rname „pharmakokinetik test“
            set rdate „29.09.97“
            # Einstellungen der Phasen und Reaktoren
            # Inputphasen der Reaktoren
            set inphase(0) 0
            # 5 Phasen im ersten Reaktor
            set phasepr {5,0}
            # Phasenmodi
            set phaseprop(0) „$MONOMOLEC“
            set phaseprop(1) „$MONOMOLEC“
            set phaseprop(2) „$INERT“
            set phaseprop(3) „$MONOMOLEC“
            set phaseprop(4) „$MONOMOLEC“
            # keine Phasenkontakte
            set pcontacts(0) 0
            # alle Aggregate kommen getrennt in den Reaktor 1
            set in2gether(0) 0
            # Ausgabephase
            set outphase 1
            # Uebergabe der Variablen an EROS7
            putco input_phase_for_reactors    inphase
            putco phases_per_reactor          phasepr
            putco phase_property              phaseprop
            putco phase_contacts              pcontacts
            putco use_all_educts_together     in2gether
            putco output_phase                outphase
            # kinetische Einstellungen
            # Reaktor 1: Gear-Algorithmus
            set kinmode(0) gear
            # minimale Konzentration fuer ein reagierendes Aggr.
            set minconc 1.e-5
            # maximale Reaktionszeit in Sekunden
            set rxtime 2.592e5
            # minimaler Umsatz; 1.0=100%
            # es gibt also keinen Umsatzbedingten Abbruch
            set convlim 1.0
            # Dosierung alle 2.88e4 Sekunden = 8 Stunden
            set multid 2.88e4
            # Anfangskonzentration fuer das erste Edukt in mol/l
```

```

set sconc(0) 0.01
# Phasenvolumina
set pvol(0) 3.0
set pvol(1) 3.0
set pvol(2) 3.0
set pvol(3) 7.0
set pvol(4) 0.01
# Anfangswert fuer die Reaktivitaet
set k      0.0
# Uebergabe der Variablen an EROS7
putco kinetic_model      kinmode
putco minimal_concentration minconc
putco reaction_time      rxtime
putco conversion_limit   convlim
putco multi_dose         multid
put reactivity           k
put start_conc           sconc
put phase_volume         pvol
# Holen der Variable von EROS7, die die aktuelle Phase enhaelt
get phase                 phase
}

```

...

Die Regeln enthalten drei verschiedene Reaktionstypen: für reine Phasenübergänge, für die Abbaureaktionen A zu M₁ und von M₁ zu M₂ sowie die Reaktion, die die Wirkung hervorruft (A zu E). Der Reaktionstyp, für die Phasenübergänge, sieht dabei folgendermaßen aus:

```

RULE_2 {
  switch $op {
  RULE_INFO {
    # Name des Reaktionstyps
    set rule_2 „Phasentransfer“
    putco rule_name rule_2
    # Attribute
    set attrib_2(0) NULL
    putco attributes attrib_2
    # Reaktionssubstruktur
    set center_2(0) 0
    putco center_connectivity center_2
    return OK
  }
  CONSTR {
    # Bedingungen fuer den Reaktionstyp
    # Die Reaktionssubstruktur enthaelt keine Atome
    switch $sub_op {
    0 {
      return OK
    }
    default {
      return BAD
    }
  }
}
FUNC {
  # Festlegung, um welche Verbindung es sich handelt.
  # n=0  A, AS, AW oder AX (in Abhaengigkeit der Phase)
  # n=2  M1
  # n=4  M2
  ...
  set n ...
  # Setzen der Phase fuer die Produkte auf aktuelle Phase+1
  set result_phase [expr $phase+1]
  if {$phase == 1} {
    # A: 1 -> 2

```



```

        # Diese Reaktion laeuft nur in Phase 2 ab.
        if {$phase != 2} {return BAD}
        return OK
    }
    1 {
...
    }
}
FUNC {
    # Festlegung, um welche Verbindung es sich handelt.
    # n=0  A, AS, AW oder AX (in Abhaengigkeit der Phase)
    # n=2  M1
    # n=4  M2
    ...
    # Hier steht die Abbaureaktion
    ...
    # Produktfunktion
    set k 5.5e-5
    if {$n > 0} {set k 1.9e-5}
    set symmetry_factor 1
    return OK
}
default {
    return BAD
}
}
}

```

Und die Wirkungsreaktion:

```

RULE_3 {
    switch $op {
    RULE_INFO {
        set rule_3 „Wirkung“
        putco rule_name rule_3
        set attrib_3(0) NULL
        putco attributes attrib_3
        set center_3 {1,2,0,0}
        putco center_connectivity center_3
        return OK
    }
    CONSTR {
        switch $sub_op {
            0 {
                # Diese Reaktion laeuft nur in Phase 5 ab.
                if {$phase != 5} {return BAD}
                return OK
            }
            1 {
...
            }
        }
    }
    FUNC {
        # Hier steht die Wirkungsreaktion
        ...
        # Produktfunktion
        set k 2.4e-4
        set result_phase 3
        return OK
    }
    default {
        return BAD
    }
}
}

```

}
}

All diese Berechnungen zeigen die Konzentrationsverläufe für die Arznei und ihre Abbauprodukte für einen sehr verantwortungsbewußten Patienten, der das verordnete Medikament exakt alle acht Stunden einnimmt. EROS7 kann die Konzentrationskurven aber auch bei einer unregelmäßigen Dosierung bestimmen.

7.1.4.7. Unregelmäßige Dosierung

In der Praxis kommt es vor, daß die verordneten Medikamente nicht ganz regelmäßig eingenommen werden. In Abbildung 165 sind die Konzentrationsverläufe angegeben, die sich für die in Kapitel 7.1.4.6 vorgestellt Pharmakokinetik ergeben, wenn:

- eine Dosis ausgelassen wird (nach 1 Tag und 16 Stunden),
- zu einem Zeitpunkt die Dosis vergessen wurde und dafür zum nächsten Zeitpunkt die doppelte Menge eingenommen wird (nach 3 Tagen und 8 Stunden) und
- eine Dosis verspätet verabreicht wurde (nach 4 Tagen und 18 Stunden).

Konzentration in mol/l

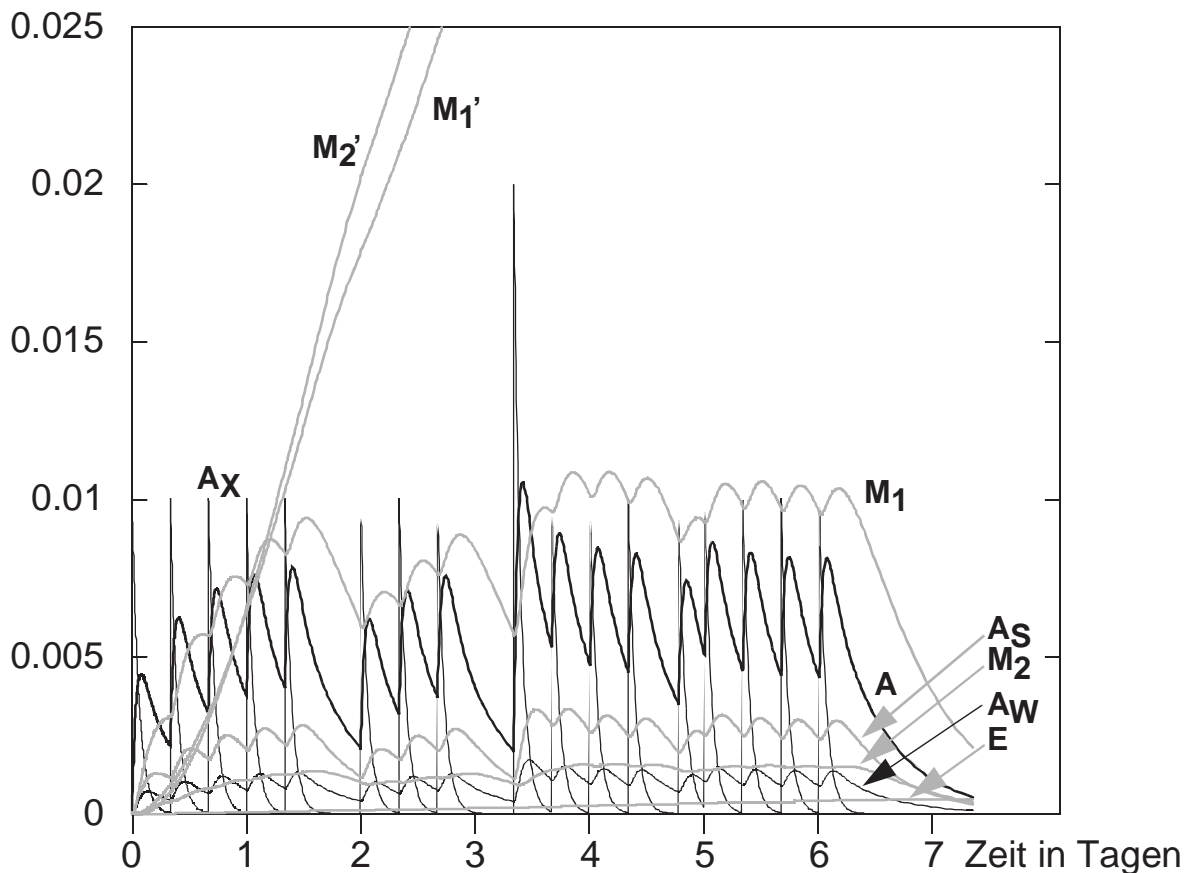


Abbildung 165: Unregelmäßige Medikamentierung.

Berechnungszeit auf einer SparcStation 10/50 mit 32 MB in Min.:Sek.: 3:35 (C++)

7.1.4.8. Kinetik nach Michaelis-Menten und nullter Ordnung

In den Kapiteln 7.1.4.1 bis 7.1.4.7 wurden für die Reaktionen nur Reaktionen erster und pseudoerster Ordnung verwendet. Biochemische Prozesse sind aber häufig enzymkatalysierte Reaktionen. Will man das Enzym und die Bildung des Enzym-Substrat-Komplexes nicht explizit

formulieren, benötigt man für die kinetische Behandlung solcher Reaktionen zusätzliche Methoden zur Bestimmung der Reaktionsgeschwindigkeit dieser Reaktionen. Um auch solche Reaktionen mit EROS7 simulieren zu können, ist es möglich, für einzelne Reaktionstypen eine Kinetik nach Michaelis-Menten oder einer nullter Ordnung zu definieren. Die Michaelis-Menten-Kinetik [53] faßt die Gleichgewichtsreaktion von Enzym und Substrat zum Komplex und die Reaktion des Komplexes zu Enzym und Produkt zusammen (siehe 6.9.1.5). EROS7 werden in diesem Fall die Michaelis-Konstante K_M und das Produkt der Geschwindigkeitskonstante zur Bildung des Produkts aus dem Komplex mit der Enzymkonzentration als Reaktivität übergeben. Eine Reaktion mit einer Kinetik nullter Ordnung liegt dann vor, wenn sich die Enzymreaktion immer im Sättigungsbereich der Michaelis-Menten-Kinetik befindet. Abbildung 166 stellt die verschiedenen Möglichkeiten der Kinetik gegenüber.

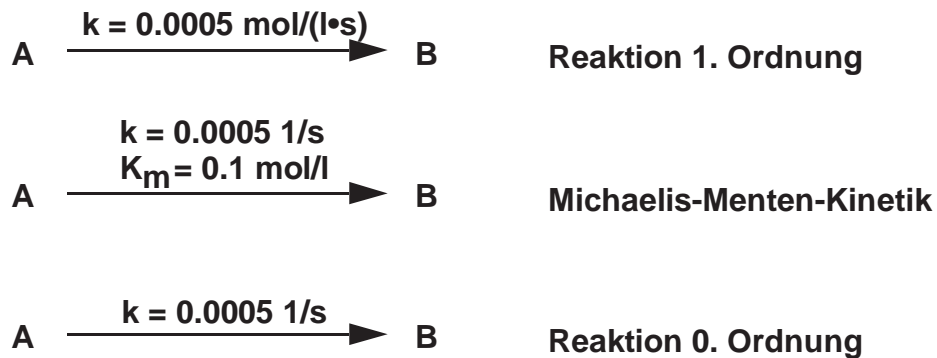


Abbildung 166: Gegenüberstellung der möglichen Kinetiken einer Reaktion für die Bestimmung der Konzentrationsverläufe mit Geschwindigkeitskonstanten. K_M ist die Michaelis-Konstante und k entspricht für die Michaelis-Menten-Kinetik $k_2 \cdot [E]_0$.

In den folgenden drei Abbildungen (167 bis 169) sind die Konzentrationsverläufe für die in Abbildung 166 angegebenen Reaktionen für die Substanzen A und B gezeigt.

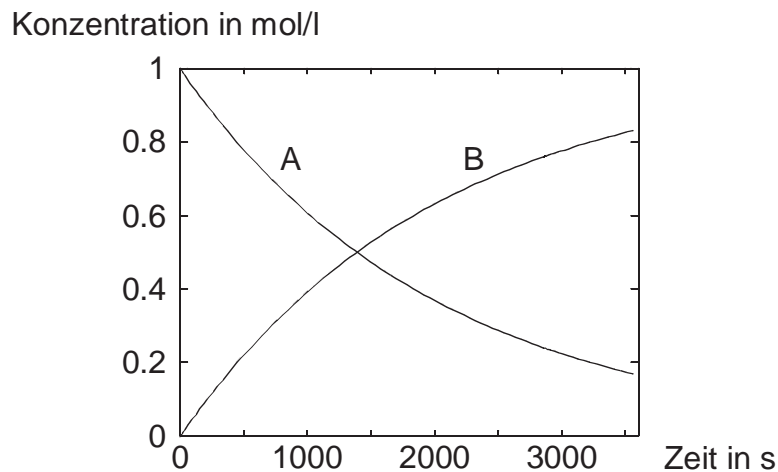


Abbildung 167: Konzentrationsverläufe für eine Reaktion erster Ordnung.

Konzentration in mol/l

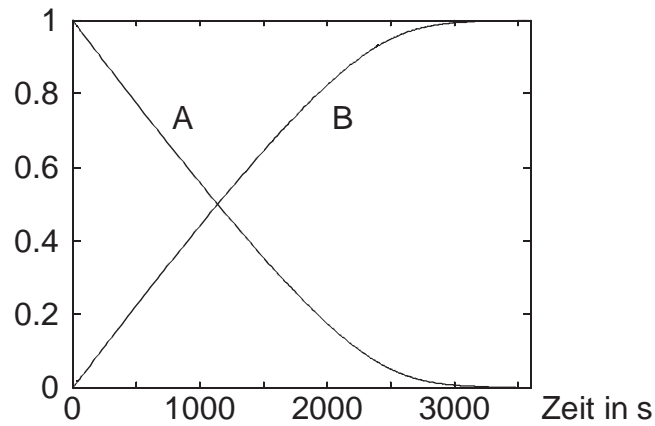


Abbildung 168: Konzentrationsverläufe für eine Reaktion mit Michaelis-Menten-Kinetik.

Konzentration in mol/l

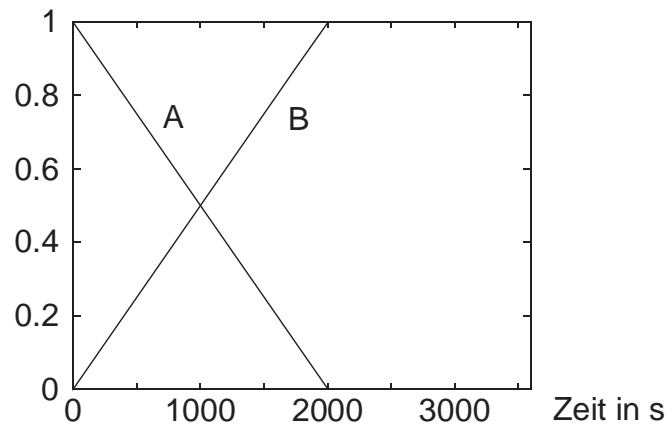
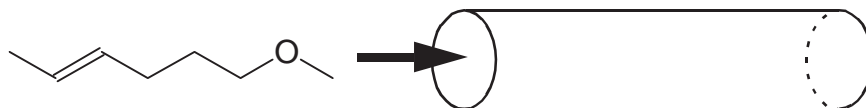


Abbildung 169: Konzentrationsverläufe für eine Reaktion nullter Ordnung.

7.1.5. Syntheseplanung

Da bei EROS7 die einzelnen Reaktionsschritte nicht notwendigerweise stöchiometrisch sein müssen, sondern während der Reaktionen auch kleine Moleküle oder Atome ergänzt werden können, eignet sich EROS7 auch zur Syntheseplanung. Will man alle Ausgangsmaterialien zusammen sehen, aus denen auf einem der möglichen Synthesewege die Zielverbindung aufgebaut wird, eignet sich dazu der Phasenmodus des „laminaren Strömungsrohrs“, bei dem die Produkte der Reaktionen zusammengehalten werden. Die Reaktionstypen in den Regeln beschreiben hierbei die retrosynthetischen Schritte.

eine Phase im Modus
„laminares Strömungsrohr“



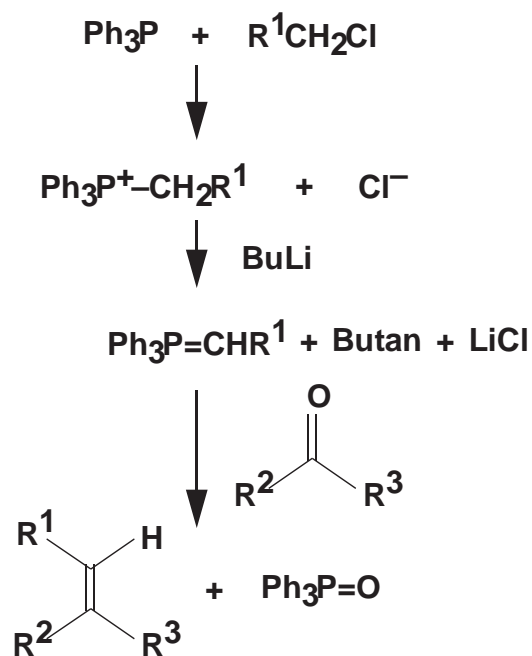
Ausgangsmaterial

zwei Reaktionstypen:
Retro-Wittig-Reaktion
Hydrolyse

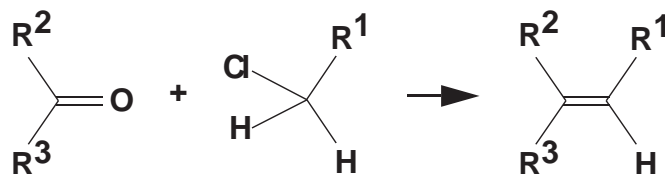
Abbildung 170: Ausgangsmaterial und Phasen für das folgende Beispiel der Syntheseplanung.

Für das nachfolgende Beispiel wurden als Reaktionstypen die Hydrolyse von Ethern und Estern sowie die Umkehrung der Wittig-Reaktion verwendet. Die Wittig-Reaktion ist eine mehrstufige Reaktion. Um sie für die Syntheseplanung in einer geeigneten Weise zu kodieren, wurde zunächst die Gesamtreaktionsgleichung über alle Schritte der Wittig-Reaktion bestimmt (siehe Abbildung 171). Die Reagenzien und die Produkte, in denen die abgespaltenen Atome (Carbonylsauerstoff-, Wasserstoff- und Chloratom) gebunden wurden, sind dabei weggelassen worden. Die Gesamtreaktion stellt also eine Transformation der Ausgangsverbindungen dar, gibt jedoch keine Reagenzien an. Diese können daraus abgeleitet werden, daß es sich um eine Wittig-Reaktion handelt. Die Umkehrung dieser Gesamtreaktion, die Retro-Wittig-Reaktion, wurde dann als retrosynthetischer Schritt eingesetzt.

Wittig-Reaktion:



Gesamtreaktion:



Retro-Wittig-Reaktion:

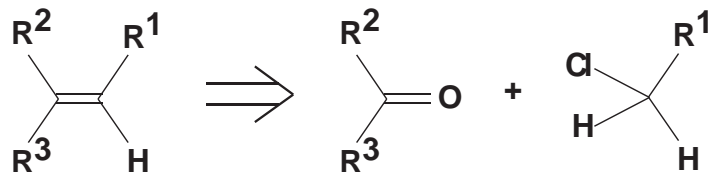


Abbildung 171: Ableitung des retrosynthetischen Schritts Retro-Wittig-Reaktion.

Bei der Retro-Wittig-Reaktion wird das Reaktionsensemble um ein Wasserstoff-, ein Chlor- und ein Sauerstoffatom erweitert. Diese werden an die beiden Seiten der gebrochenen Doppelbindung gebunden. Für die Hydrolyse wird während der Reaktion das notwendige Wassermolekül eingeführt. Das Ergebnis der Anwendung dieser retrosynthetischen Schritte ohne jegliche Bewertung der verschiedenen Wege auf ein Beispielmolekül ist in Abbildung 172 zu sehen.

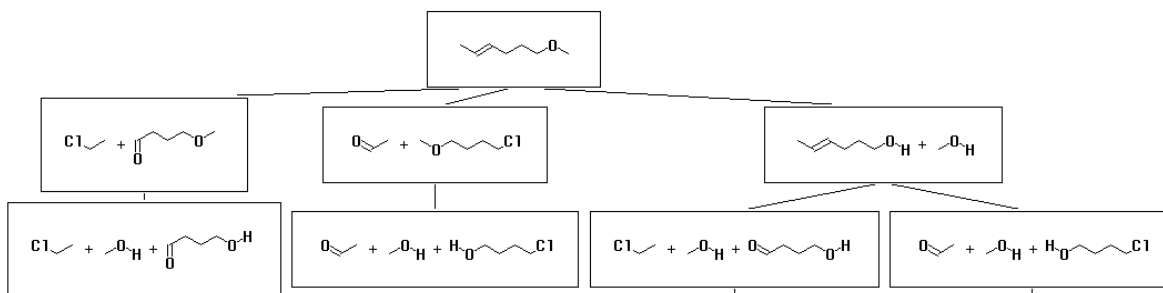


Abbildung 172: Anwendung der retrosynthetischen Schritte Hydrolyse und Retro-Wittig-Reaktion auf ein Beispielmolekül. Die kleinen Striche unter den rechten beiden Ensembles deuten an, daß diese im Netzwerk schon enthalten sind (nämlich die linken beiden) und deshalb an dieser Stelle ausgehend von diesen keine Reaktionen gezeichnet sind. Auf der linken Seite (einmal pro Netzwerk) würden weitere retrosynthetische Schritte angeben.

Berechnungszeit auf einer SparcStation 10/50 mit 32 MB in Min.:Sek.: 0:06 (C++)

Mit geeigneten Kriterien für die Bewertungsfunktionen für die retrosynthetischen Schritte kann EROS7 den einzelnen Synthesewegen Güten zuordnen. Wie bei den hier verwendeten retrosynthetischen Schritten zu sehen ist, ist es nicht notwendig, jeden Teilschritt als vollständige Reaktion im Detail zu formulieren. Bei der Retro-Wittig-Reaktion tritt beispielsweise keinerlei Phosphorverbindung in Erscheinung. Dies läßt sich auch noch weiter bis hin zur Formulierung von Transformationen fortsetzen, die beispielsweise nur noch das Kohlenstoffgerüst der Verbindung zurücklassen. Hierbei handelt es sich um Techniken, möglichst schnell zu den Ausgangsverbindungen zu gelangen [69].

Will man eine Syntheseplanung durch vorherige Markierung von strategischen Bindungen vornehmen [69], findet diese während der Vorbehandlung des Eduktensembles statt. Hier werden auch die Vorstufen von der weiteren Syntheseplanung ausgeschlossen, von denen man durch einen Blick in eine Datenbank herausgefunden hat, daß sie käuflich sind.

Für eine aufwendigere Syntheseplanung mit EROS7 eignet sich jedoch der Phasenmodus „monomolekularer“ besser, da hier die schon gefundenen Ausgangssubstanzen nicht bei jedem Schritt von einer weiteren Anwendung der retrosynthetischen Schritte ausgeschlossen werden müssen.

7.1.6. Erzeugung eines vollständigen Satzes von Verbindungen

7.1.6.1. Mögliche Produkte einer Diels-Alder-Reaktion

Zur Durchführung einer Diels-Alder-Reaktion gibt man ein Dien und ein Dienophil in einen Kolben und rührt die Reaktionsmischung. Um dies mit EROS7 zu simulieren, verwendet man eine Phase im Modus „Rührkessel“ (siehe Abbildung 173). Als einfaches Beispiel zur Erzeugung aller möglichen Diels-Alder-Produkte wird die Diels-Alder-Reaktion von 1,3-Pentadien mit 1-Buten verwendet. Diese läuft durch die Abwesenheit einer elektronischen Aktivierung nur unter hohem Druck und erhöhter Temperatur ab (vergleiche die Reaktionen von 1,3-Butadien mit Ethen [70], 1,3-Butadien mit Propen [71], 1,3-Pentadien mit Ethen [72] und Terpen mit Ethen [73]), zeigt aber in einfacher Weise die Zahl der möglichen Kombinationen der Edukte. Da die einzelnen, unterschiedlich schnell ablaufenden Reaktionen nicht bewertet wurden, werden nur die Strukturen aller Verbindungen angegeben.

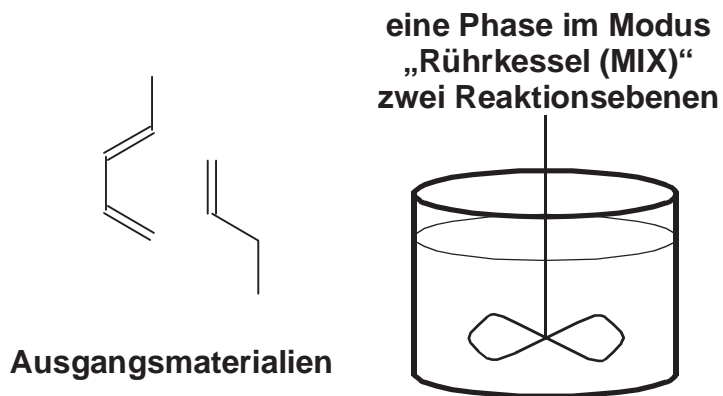


Abbildung 173: Phasen und Ausgangsmaterialien für die Diels-Alder-Reaktion.

Ausgehend von den Ausgangsmaterialien (siehe Abbildung 174) werden zwei verschiedene Reaktionen der beiden Verbindungen untereinander gefunden. Ihre Produkte sind in Abbildung 175 gezeigt.

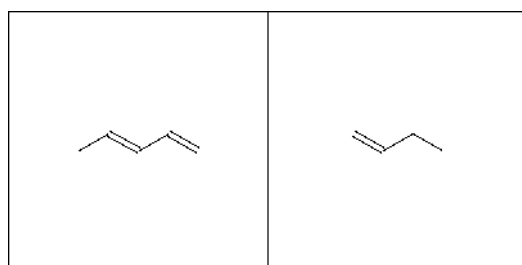


Abbildung 174: Ausgangsmaterialien 1,3-Pentadien und 1-Buten.

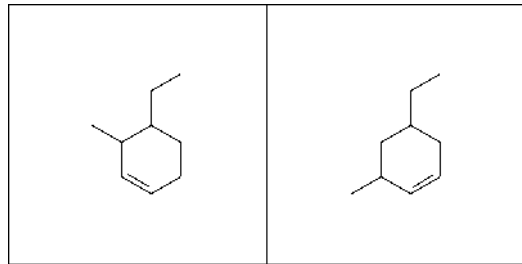


Abbildung 175: Produkte der Reaktion von 1,3-Pentadien mit 1-Buten.

Das Dien kann aber auch als Dienophil und damit mit sich selbst reagieren. Die Produkte dieser Reaktionen sind in Abbildung 176 gezeigt.

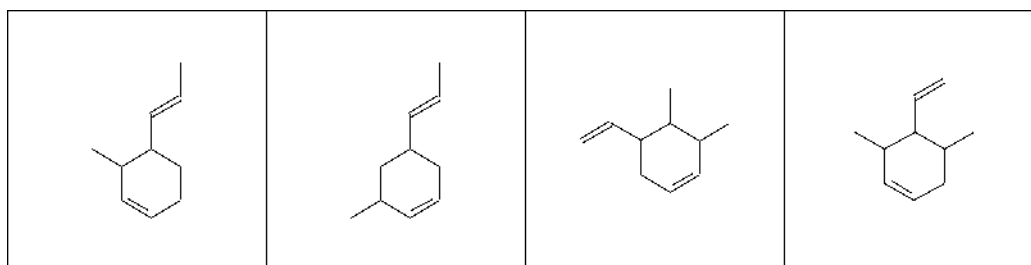


Abbildung 176: Produkte der Reaktion von 1,3-Pentadien mit sich selbst.

Da alle Produkte der Diels-Alder-Reaktionen mindestens eine Doppelbindung enthalten, können sie wieder als Dienophil auftreten und mit noch nicht umgesetztem 1,3-Pentadien reagieren.

Die Verbindungen, die sich in einer zweiten Reaktionsebene durch die Reaktionen der Moleküle aus Abbildung 175 mit 1,3-Pentadien ergeben, zeigt Abbildung 177.

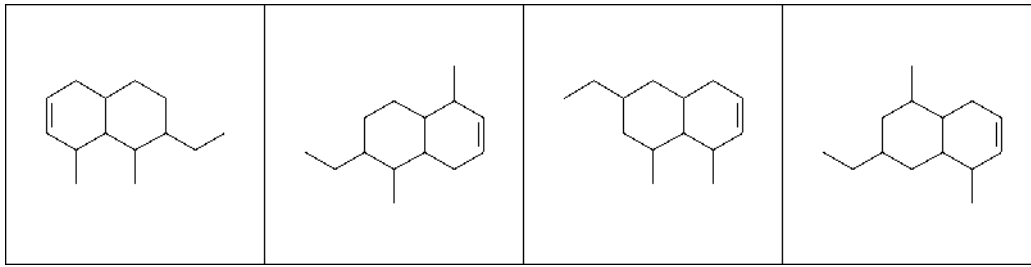


Abbildung 177: Reaktionsprodukte der zweiten Reaktionsebene, bei denen 1-Buten zweimal hintereinander mit 1,3-Pentadien umgesetzt wurde.

Aber auch die zwei Doppelbindungen enthaltenden Reaktionsprodukte, bei denen 1,3-Pentadien mit sich selbst reagiert hat (siehe Abbildung 176), können ein weiteres Mal mit 1,3-Pentadien umgesetzt werden. Diese zwölf Substanzen sind in Abbildung 178 dargestellt.

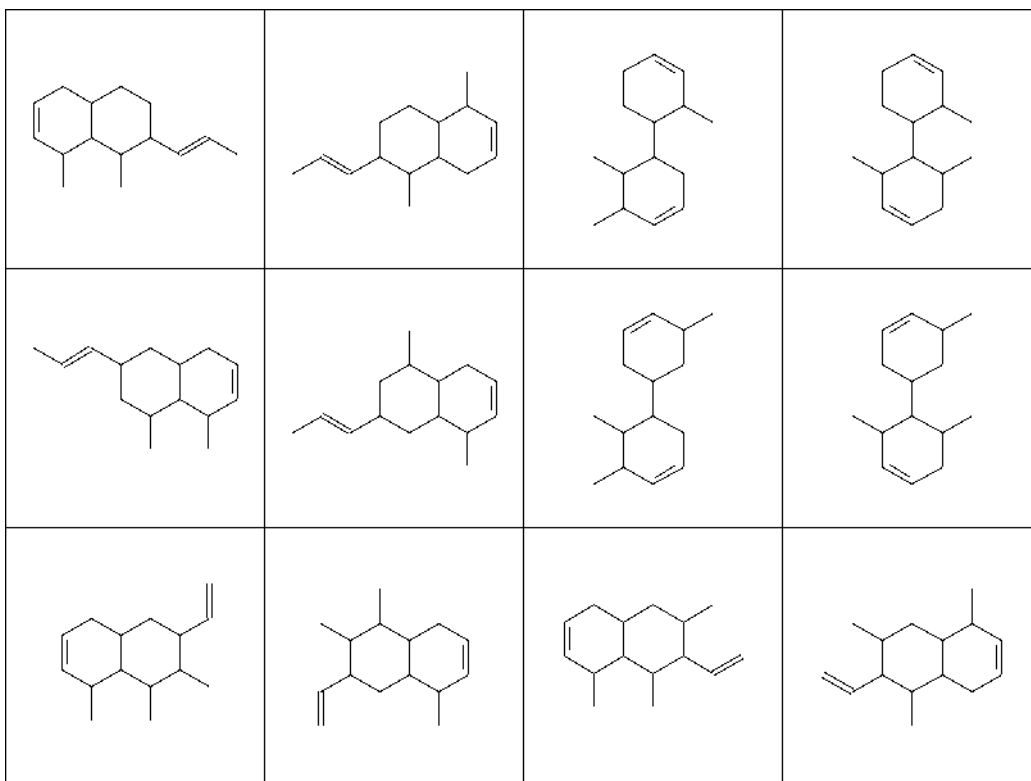


Abbildung 178: Diels-Alder-Produkte aus drei Molekülen 1,3-Pentadien.

Berechnungszeit auf einer SparcStation 10/50 mit 32 MB in Min.:Sek.: 1:15 (C++)

Dies könnte für eine dritte, vierte, usw. Reaktionsebene mit der Erzeugung immer komplexerer Strukturen fortgeführt werden. Da die Produkte mit zunehmender Reaktionsebene in immer geringeren Mengen gebildet werden, wurde auf die Angabe weiterer Strukturen verzichtet.

7.1.6.2. Chlorierte Dioxine

So wie nach und nach die möglichen Produkte der Diels-Alder-Reaktion erzeugt wurden, können auch sukzessive alle chlorierten Dioxine gebildet werden. Dazu wird ein Reaktor mit zwei Phasen in den Modi „inert“ und „Grenzflächenreaktionen“ mit Kontakt zur „inerten“ Phase ein-

gesetzt (siehe Abbildung 179). Es wäre auch möglich, die chlorierten Dioxine mit einer Phase im Modus „Rührkessel“ zu erzeugen. Diese Methode ist allerdings deutlich langsamer, da EROS7 dann versucht, je zwei der schon chlorierten Dioxine miteinander zur Reaktion zu bringen. Da hierbei keine Reaktionen gefunden werden, führt die Reaktionsgenerierung mit den Phasen aus Abbildung 179 zu einer schnelleren Bearbeitung.

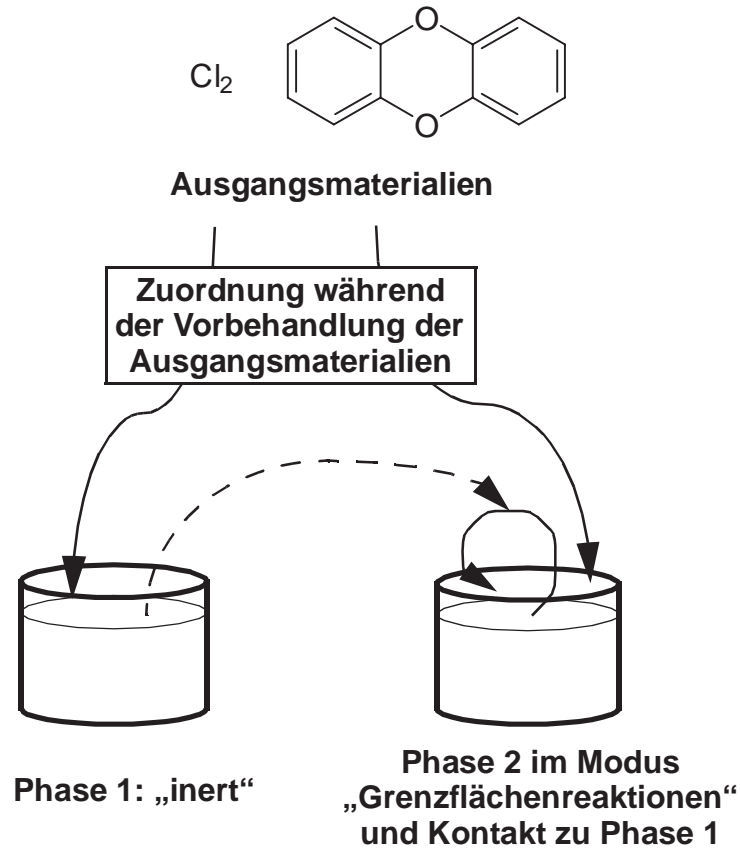


Abbildung 179: Erzeugung aller möglichen, unterschiedlich chlorierten Dioxine.

Der Regelausschnitt (Tcl) zur Festlegung der Phasen sieht dabei folgendermaßen aus:

```

...
static vector<int> phase
static int nedagrs=0;

proc Tcl_rule {} {

    global irule op sub_op rule_trace
    global_name
    global k
    global attrib_1 center_1 rule_1
    global rname rdate inphase phasepr phaseprop pcontacts in2gether
    global outphase kinmode startphase nedagrs
    switch $irule {
    GLOBAL {
        switch $op {
        INIT_RULES {
            # globale Einstellungen
            set rname „Dioxine1“
            set rdate „29.09.97“
            # Einstellungen der Phasen und Reaktoren
            # Inputphasen der Reaktoren
            set inphase(0) 0
            # 2 Phasen im ersten Reaktor

```

```

set phasepr {2,0}
# Phasenmodi
set phaseprop(0) „$INERT“
set phaseprop(1) „$SURFACE“
# Phasenkontakt 2 -> 1
set pcontacts(0) 2
set pcontacts(1) 1
set pcontacts(2) 0
# alle Aggregate kommen zusammen den Reaktoren zugeteilt
set in2gether(0) 1
# Ausgabephase
set outphase 2
# Uebergabe der Variabeln an EROS7
putco input_phase_for_reactors    inphase
putco phases_per_reactor          phasepr
putco phase_property              phaseprop
putco phase_contacts              pcontacts
putco use_all_educts_together     in2gether
putco output_phase                outphase
# kinetische Einstellungen: keine Kinetik
set kinmode(0) none
# Erzeugung der Variable fuer die Startphase
set startphase(0) 1
# Initialisierung des Aggregatzaehlers
set nedagrs 0
# Uebergabe der Variablen an EROS7
putco kinetic_model               kinmode
put start_phase                   startphase
}

```

...

Während der Vorbehandlung der Ausgangsmaterialien werden diese den Phasen zugeteilt. Für den hier gezeigten Algorithmus darf in der Ausgangsdatei keine Verbindung mehrfach enthalten sein.

```

PREP_ROOT {
  prop E_N_AGGREGATES n
  for {set i 1} {$i <= $n} {incr i} {
    # chlorhaltig Aggregate kommen in die Phase 1,
    # alle übrigen in die Phase 2
    set phs 2
    set ncl [dioxin_agr_cl $i]
    if {$ncl > 0} {set phs 1}
    set startphase($nedagrs) $phs
    incr nedagrs
  }
  return OK
}

```

Zur Erweiterung, daß die Zuweisung zu den Phasen auch für mehrfach enthaltene Aggregate eingesetzt werden könnte, werden Ausgangsmaterialien, die schon einmal vorbehandelt wurden, ignoriert.

Die aufgerufene Prozedur *dioxin_agr_cl* zählt die Chloratome im angegebenen Aggregat und liefert sie als Funktionswert zurück. Sie ist Teil der Regeln:

```

proc dioxin_agr_cl {iagr} {
  # in der Variablen ncl werden die Chloratome gezaehlt.
  set ncl 0
  prop E_LAST_ATOM il
  # Schleife ueber alle Atome im Ensemble
  for {set i 1} {$i <= $il} {incr i} {
    prop A_AGGREGATE $i ii
    if {$ii == $iagr} {
      # das Atom gehoert zum Aggregat, das betrachtet werden soll
      prop A_ELEMENT $i elem
    }
  }
}

```

```

        if {$selem == 17} {incr ncl}
    }
}
# Rueckgabe der Zahl der Chloratome im Aggregat $iagr
return $ncl
}

```

Der einzige, benötigte Reaktionstyp ist die elektrophile aromatische Substitution, für die elementares Chlor als Reagens erforderlich ist. Die Phase zwei mit Kontakt zur Phase eins kombiniert nun alle ihre Substanzen, beginnend mit Dioxin, mit Chlor, das das einzige Molekül der Phase eins darstellt. Da auch die Produkte der Phase zwei zugeteilt werden, entstehen so in der Phase zwei neben Chlorwasserstoff in acht Reaktionsebenen nach und nach alle chlorierten Dioxine. Man erhält:

2	Monochlordioxine
10	Dichlordioxine
14	Trichlordioxine
22	Tetrachlordioxine
14	Pentachlordioxine
10	Hexachlordioxine
2	Heptachlordioxine
1	Octachlordioxin

Der vollständige Satz der Strukturen ist, gegliedert nach der Zahl der Chloratome, in Anhang E angegeben.

Berechnungszeit auf einer SparcStation 10/50 mit 32 MB in Min.:Sek.: 1:44 C++, 3:05 Tcl

Hier wurden jetzt alle chlorierten Dioxine zusammen erzeugt. Mit der gleichen Technik, die für die kombinatorische Chemie eingesetzt wird (siehe 7.1.7), können beispielsweise selektiv alle pentachlorierten Dioxine erzeugt werden. Dazu wird ein Reaktor mit den in Abbildung 180 gezeigten Phasen eingesetzt.

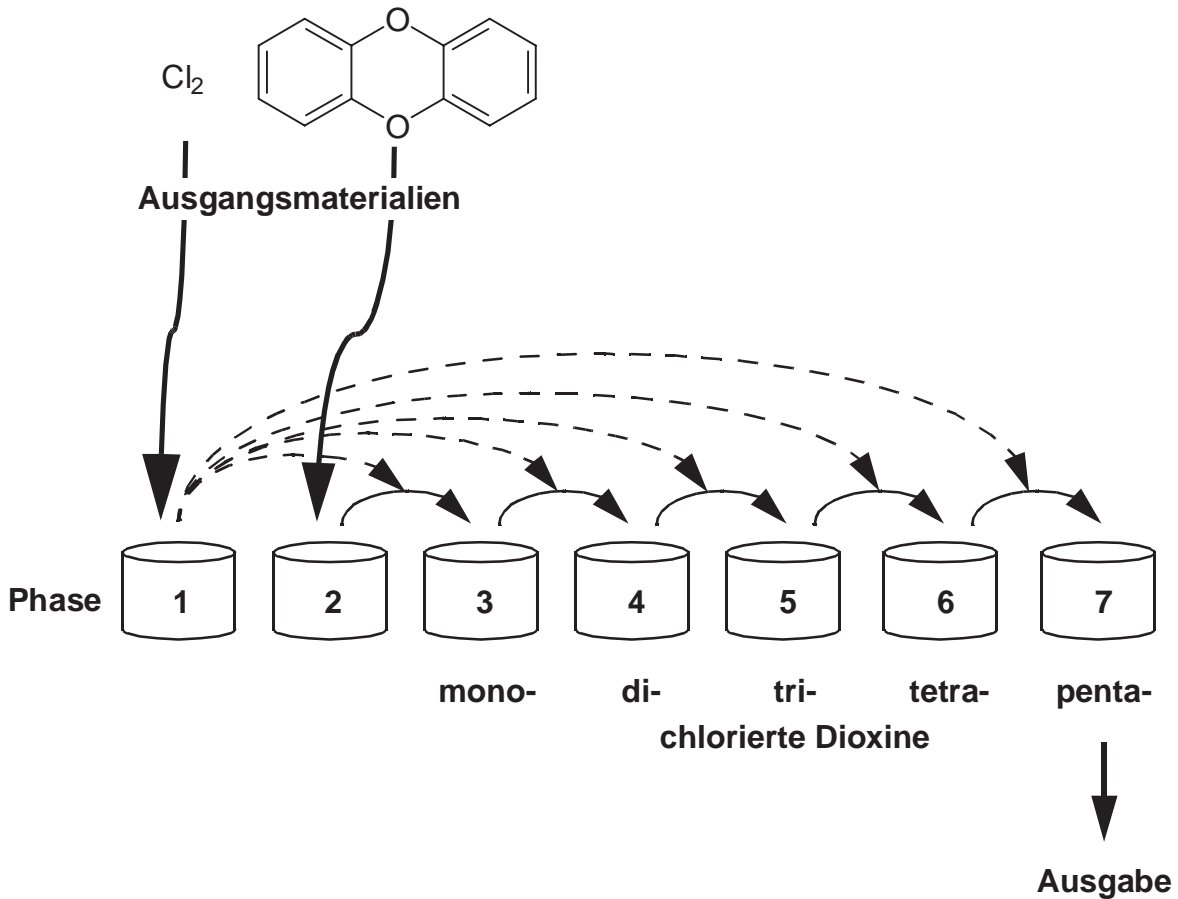


Abbildung 180: Erzeugung aller pentachlorierten Dioxine durch elektrophile, aromatische Substitution mit Chlor. Die Phasen 1 und 7 sind inert. Alle anderen Phasen erzeugen Grenzflächenreaktionen und haben Kontakt zur Phase 1.

Der Regelausschnitt zur Festlegung der Phasen sieht dabei folgendermaßen aus:

```

proc Tcl_rule {} {
  global irule op sub_op rule_trace
  global name
  global name date sphase kin_mode nedagrs
  global phases_p_r phase_contacts phprop comp_edus in_phase ophase
  global attrib_1 center_1 rule_1
  global phase_inc
  switch $irule {
  GLOBAL {
    switch $op {
    INIT_RULES {
      set name „dioxin2.tcl“
      putco name name
      set date „30.09.1997“
      putco date date
      # reactors and phases
      set in_phase(0) 0
      set comp_edus(0) 1
      # 7 Phasen
      set phases_p_r {7,0}
      # Phasenkontakte
      set phase_contacts {2,1,3,1,4,1,5,1,6,1,0}
      # Phasenmodi
    }
  }
}

```

```

set phprop(0) "$INERT"
set phprop(1) "$SURFACE"
set phprop(2) "$SURFACE"
set phprop(3) "$SURFACE"
set phprop(4) "$SURFACE"
set phprop(5) "$SURFACE"
set phprop(6) "$INERT"
putco phases_per_reactor phases_p_r
putco phase_contacts phase_contacts
putco phase_property phprop
putco use_all_educts_together comp_edus
putco input_phase_for_reactors in_phase
set ophase 2
putco output_phase ophase
set kin_mode(0) none
putco kinetic_model kin_mode
set sphase(0) 1
set nedagrs 0
put start_phase sphase
return OK
}
...

```

Werden die Ausgangsmaterialien wie zuvor in die Phasen eins und zwei gegeben und die Produkte der Substitutionsreaktion gegenüber der Phase, die die Reaktion erzeugt, der jeweils um eins höheren Phase zugeteilt, sammeln sich die Dioxine mit der gleichen Anzahl an Chloratomen in je einer Phase, deren Strukturen dann ausgegeben werden.

```

FUNC {
# Durchfuehrung der Reaktion
...
set result_phase [expr $result_phase+1]
return OK
}

```

Da bei jeder Substitutionsreaktion auch immer Chlorwasserstoff entsteht, enthalten die Phasen drei bis sieben auch diesen. Dieser kann während der Reaktion auch gelöscht werden, wodurch nur noch die pentachlorierten Dioxine in die Phase sieben gelangen (siehe Abbildung 181).

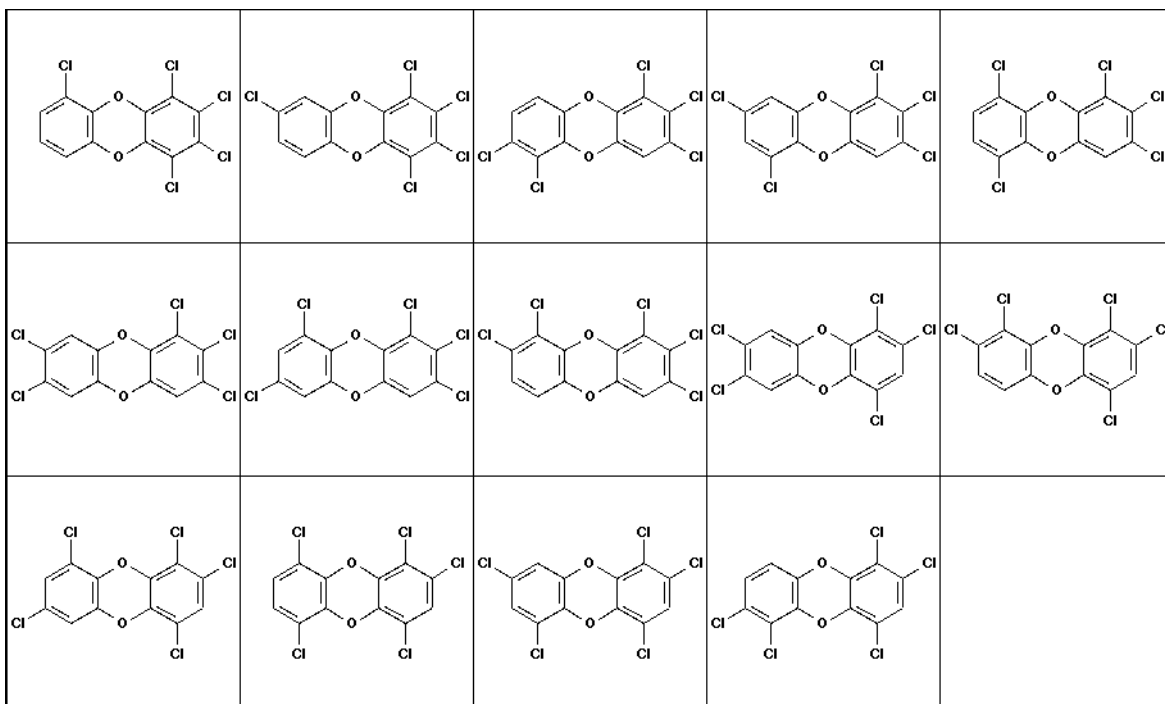


Abbildung 181: Pentachlorierte Dioxine.

Berechnungszeit auf einer SparcStation 10/50 mit 32 MB in Min.:Sek.: 1:04 (C++)

7.1.7. Kombinatorische Chemie

In der kombinatorischen Chemie werden aus zwei oder mehr Sätzen von Verbindungen Reaktionen mit allen möglichen Substanzkombinationen durchgeführt, wobei je ein Molekül aus jeder Gruppe von Verbindungen verwendet wird. Beispiele für den Einsatz von EROS7 in der kombinatorischen Chemie befinden sich auch im Abschnitt 6.7.4.10, in dem kombinatorische Esterbildungen und die entsprechenden Einstellungen für EROS7 beschrieben sind. In den folgenden Abschnitten werden weitere Beispiele gezeigt.

7.1.7.1. 1,4-Benzodiazepine

1,4-Benzodiazepine zeigen in einem breiten Bereich biologische Aktivitäten und werden in vivo nicht so schnell abgebaut. Deshalb sind die verschiedenen 1,4-Benzodiazepine von breitem pharmazeutischen Interesse. Dargestellt werden sie aus einem substituierten 2-Aminobenzophenon, einer Aminosäure und einem Alkylierungsreagens. Zu ihrer Festphasensynthese wird zunächst das gebundene 2-Aminobenzophenon mit dem Aminosäurefluorid, deren Aminogruppe geschützt ist, umgesetzt. Es schließen sich die Abspaltung der Schutzgruppe, die Cyclisierung und die Alkylierung an (siehe Abbildung 182). [74]

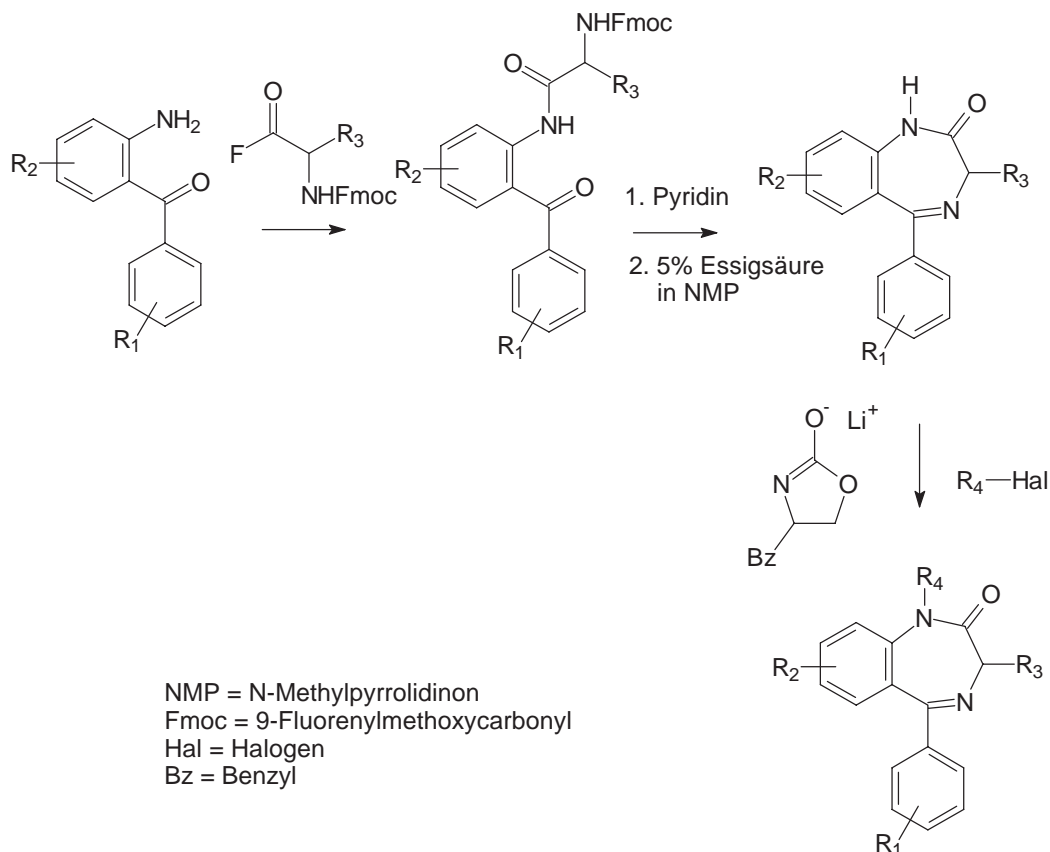


Abbildung 182: Synthese von 1,4-Benzodiazepinen [74].

Für die Erzeugung der 1,4-Benzodiazepine mit EROS7 kann man die Reaktionen mit dem geschützten Säurefluorid generieren, wie sie tatsächlich durchgeführt werden. Es ist jedoch auch möglich mehrere Reaktionsschritte zusammenzufassen. So wurden der Schutz und die Aktivierung der Aminosäure, die Abspaltung der Schutzgruppe und die Cyclisierung vereinigt. EROS7 generiert also in einem Schritt aus der Aminosäure und dem 2-Aminobenzophenon direkt das 1,4-Benzodiazepingerüst, das im zweiten Schritt alkyliert wird. Die 2-Aminobenzophenone (siehe Abbildung 183), Aminosäuren (siehe Abbildung 184) und Alkylierungsmittel (siehe Abbildung 185) stehen gemeinsam in der Eingabedatei und werden während der Vorbehandlung der Ausgangsmaterialien den Phasen eins bis drei zugewiesen. Als Entscheidungskriterien, in welche Phase eine Verbindung kommt, wurden dabei die Existenz eines π -Elektronensystems mit mindestens 15 Atomen und die Anwesenheit von Halogenatomen im Molekül verwendet.

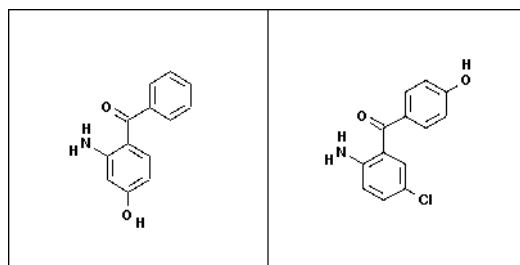


Abbildung 183: 2-Aminobenzophenone.

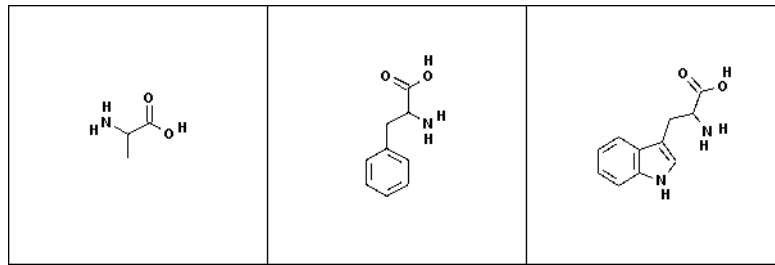


Abbildung 184: Aminosäuren.

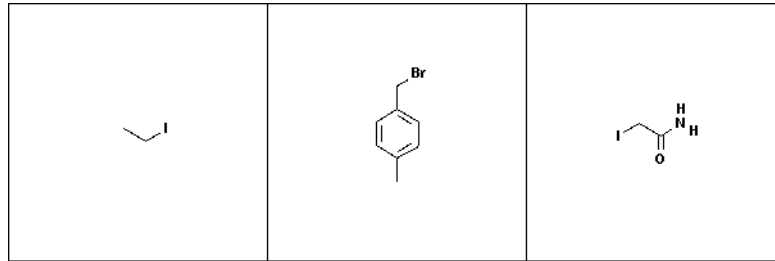


Abbildung 185: Alkylierungsreagenzien.

Ausgehend von den drei verschiedenen Gruppen von Ausgangsmaterialien in den Phasen eins bis drei werden die beiden Reaktionen von den Phasen zwei und vier erzeugt, wie es in Abbildung 186 gezeigt ist.

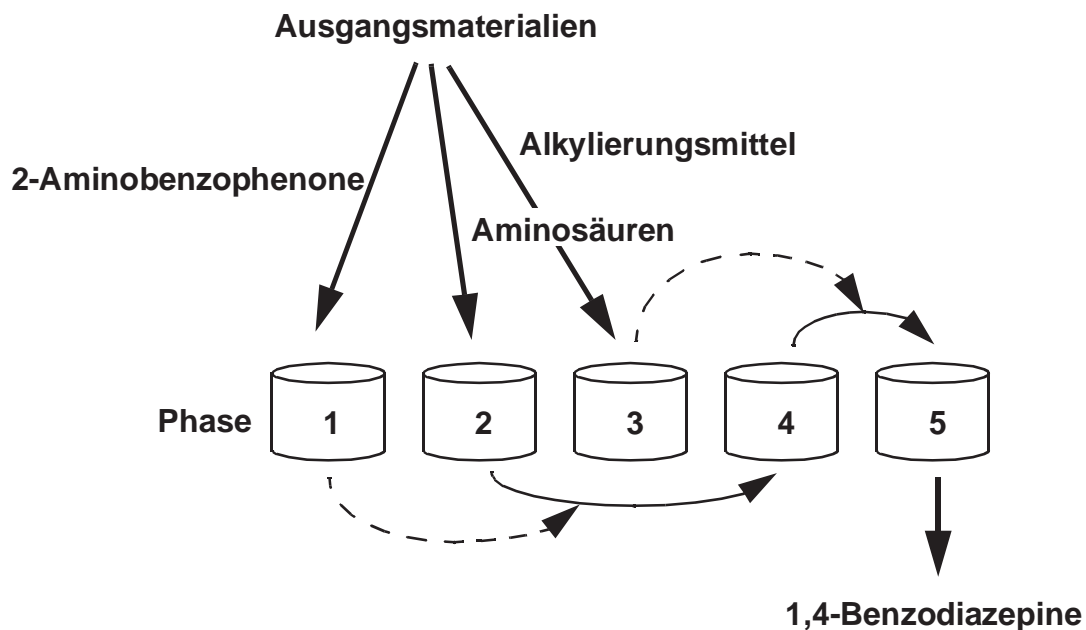


Abbildung 186: Phasenaufbau zur Erzeugung der 1,4-Benzodiazepine. Die Phasen 1, 3, und 5 sind inert und die Phasen 2 und 4 generieren Grenzflächenreaktionen und haben Kontakt zur Phase 1 bzw. zur Phase 3.

Nachdem das Proton und die Halogenidionen aus den Molekülen der Phase fünf entfernt wurden, erhält man die 1,4-Benzodiazepine, wie sie in Abbildung 187 gezeigt sind. Die Entfernung des Proton und des Halogenids, die durch die zugegebene Base gebunden werden, können auch während der Reaktion entfernt werden. In diesem Fall sind nur noch die 1,4-Benzodiazepine in der Phase fünf.

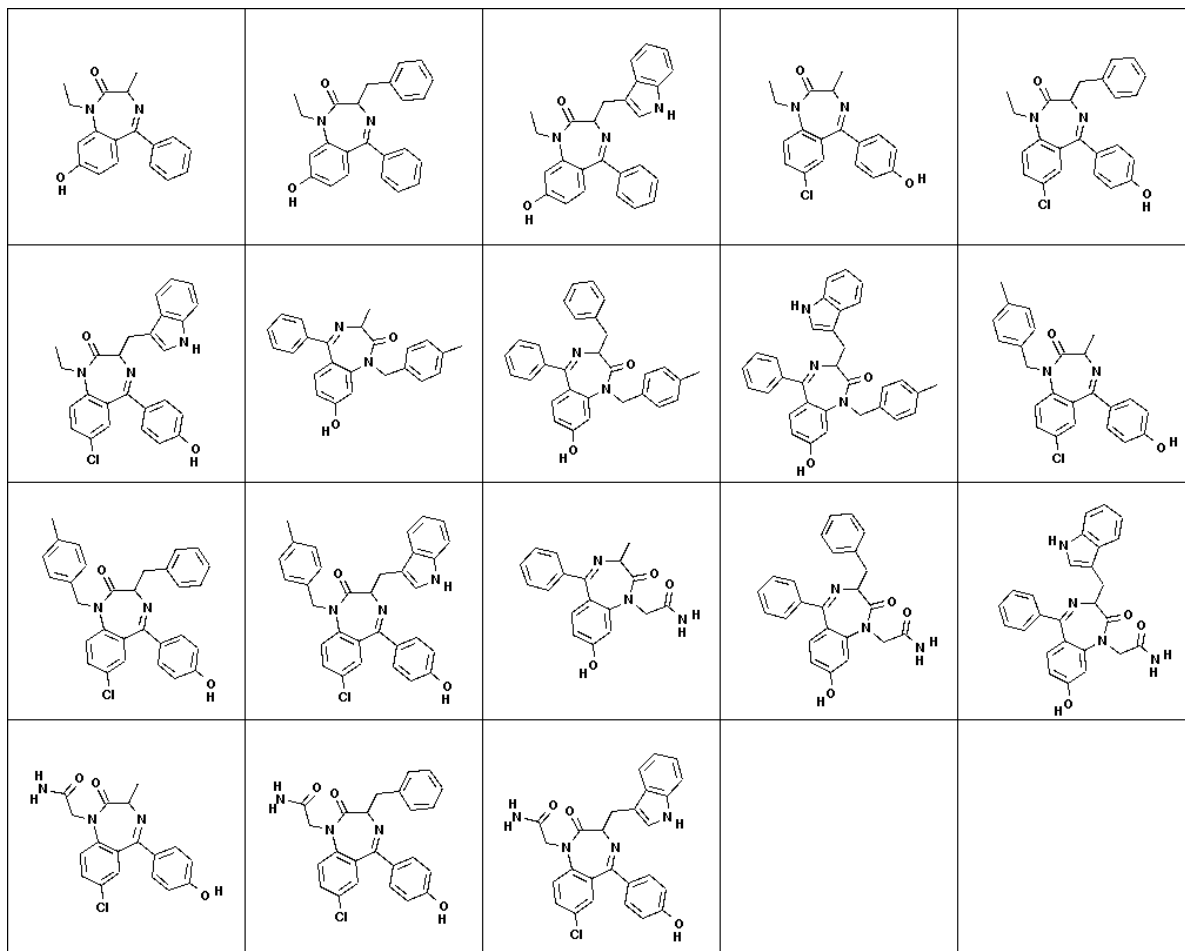


Abbildung 187: Gebildete 1,4-Benzodiazepine.

Berechnungszeit auf einer SparcStation 10/50 mit 32 MB in Min.:Sek.: 1:15 (C++)

7.1.7.2. Substituierte Pyrazole

Aus einer Hydrazinkomponente und einer 1,3-Dicarbonylverbindung können substituierte Pyrazole synthetisiert werden, wobei aus unsymmetrisch substituierten Verbindungen je zwei verschiedene Produkte entstehen [56]. Als Hydrazinkomponenten für die kombinatorische Synthese wurden die Verbindungen aus Abbildung 188 eingesetzt und mit den 1,3-Dicarbonylverbindungen aus Abbildung 189 zur Reaktion gebracht.

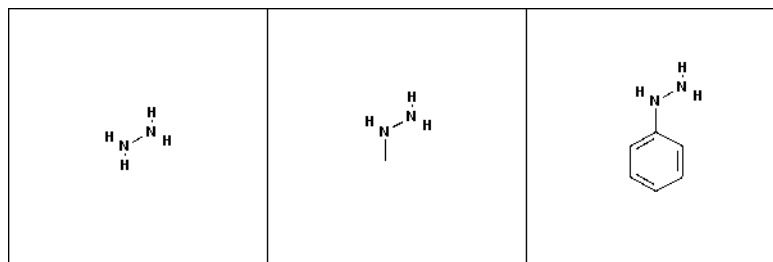


Abbildung 188: Hydrazine.

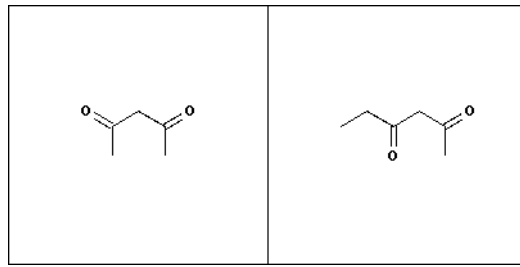


Abbildung 189: 1,3-Dicarbonylverbindungen.

EROS7 verwendete dazu einen Reaktor mit drei Phasen, wie sie in Abbildung 190 dargestellt sind.

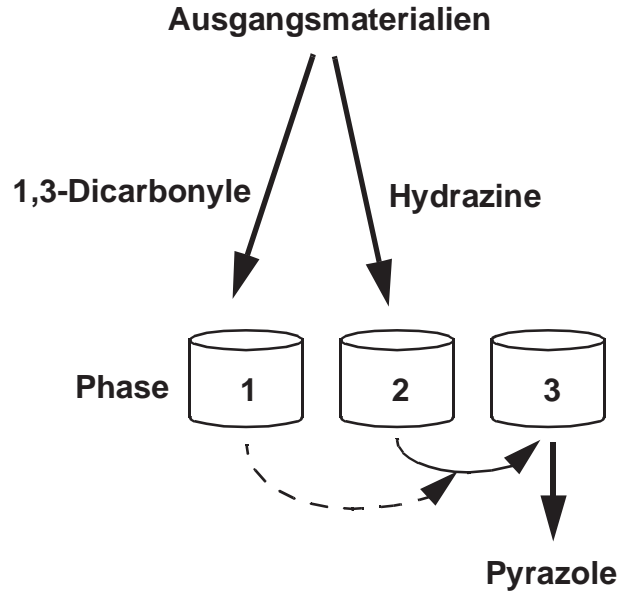


Abbildung 190: Phasen für die Generierung der Pyrazole. Phase zwei erzeugt Grenzflächenreaktionen und hat Kontakt zur Phase eins, die wie Phase drei inert ist.

Der einzige Reaktionstyp der Bildung der Pyrazole (siehe Abbildung 191) teilt seine Produkte der Phase drei zu.

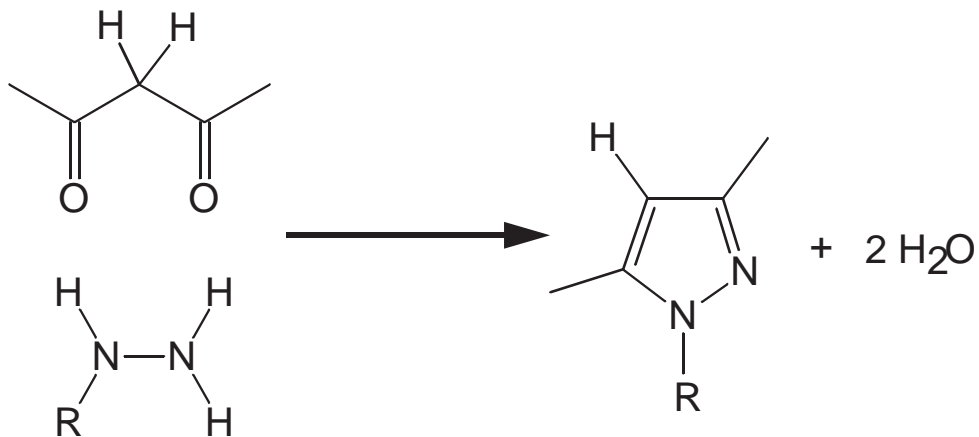


Abbildung 191: Bildung der Pyrazole.

Wird das bei der Reaktion gebildete Wasser entfernt, erhält man in der Phase drei ausschließlich die erzeugten Pyrazole, wie sie in Abbildung 192 gezeigt sind.

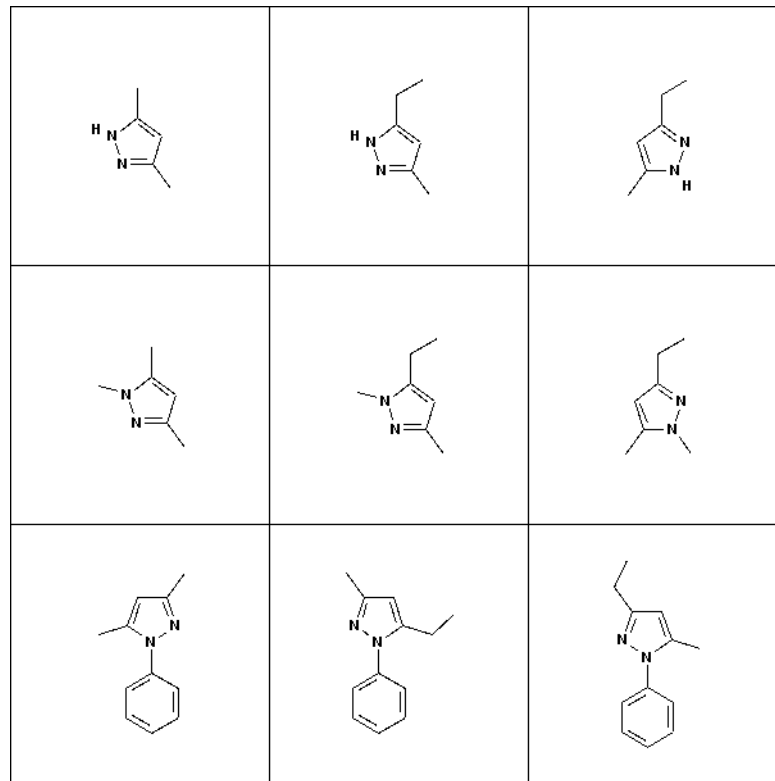


Abbildung 192: Erzeugte Pyrazole.

Berechnungszeit auf einer SparcStation 10/50 mit 32 MB in Min.:Sek.: 0:09 (C++)

7.1.7.3. Tripeptide aus drei möglichen Aminosäuren

Auch die Synthese aller möglichen Tripeptide aus drei Aminosäuren kann als kombinatorische Synthese aufgefaßt werden. Ein Satz von drei Aminosäuren reagiert mit einem zweiten Satz aus drei Aminosäurechloriden, wobei Dipeptide gebildet werden. Setzt man diese erneut mit einem dritten Satz von Aminosäurechloriden um, erhält man Tripeptide. Handelt es sich bei allen Sätzen um die gleichen drei Aminosäuren (Serin, Alanin und Phenylalanin), erhält man so den vollständigen Satz verschiedener Tripeptide aus drei Aminosäuren. Da die beiden Sätze an Aminosäurechloriden identisch sind, wurden sie zusammengefaßt und es ergibt sich ein Reaktions- und Phasenschema, wie es Abbildung 193 zeigt.

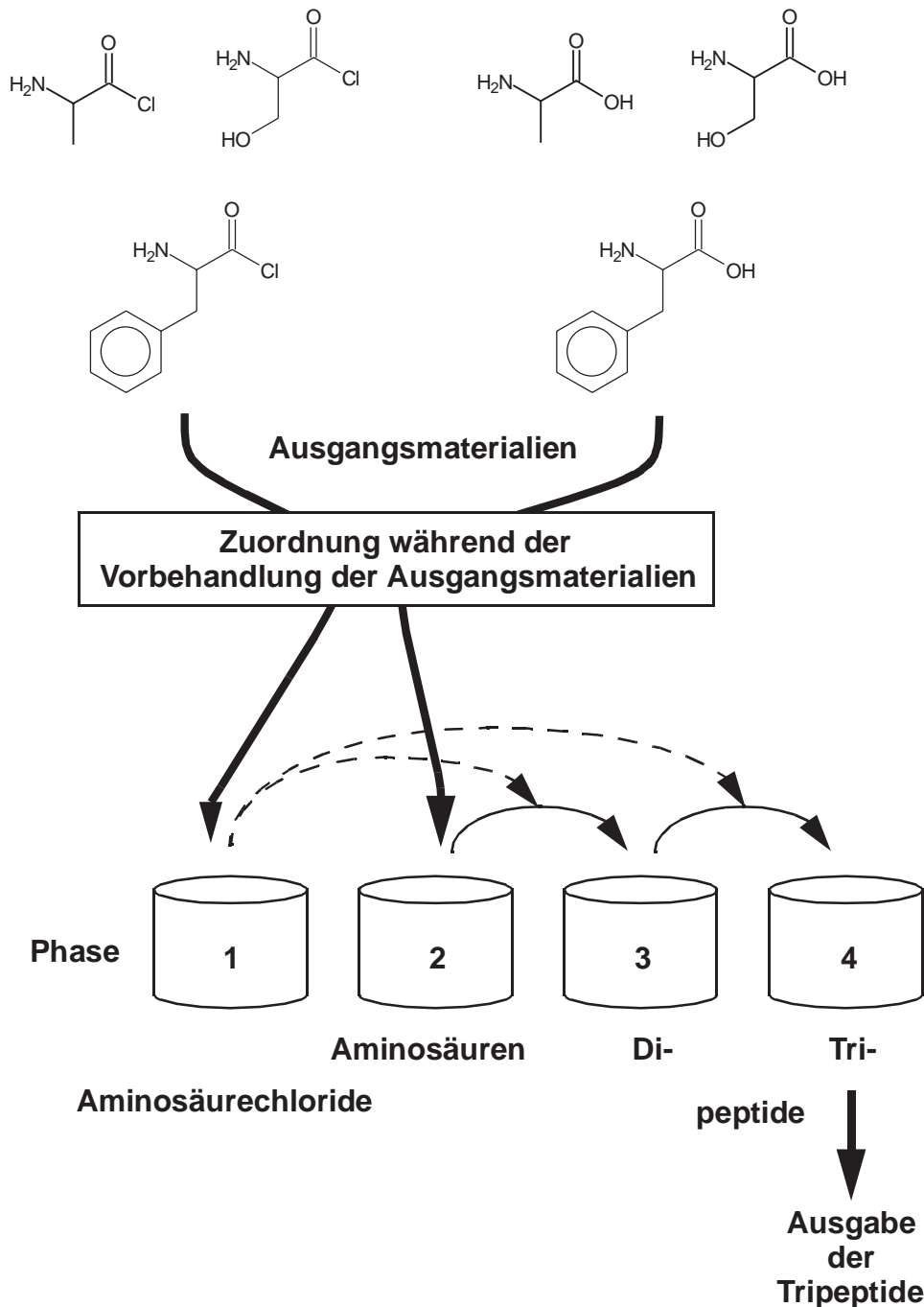


Abbildung 193: Erzeugung aller Tripeptide aus Serin, Alanin und Phenylalanin. Die Phasen 1 und 4 sind inert und die Phasen 2 und 3 haben einen Phasenkontakt zur Phase 1 und erzeugen Grenzflächenreaktionen. Die Produkte der Reaktionen, die die Phase 2 erzeugt (ein Edukt aus der Phase 2 und eines aus der Phase 1), kommen in die Phase 3 und die von Phase 3 in Phase 4.

Als Reaktionstyp wurde die selektive Amidbildung aus primären Aminen und Säurechloriden (siehe Abbildung 194) eingesetzt. Alle anderen mit Säurechloriden reagierenden Gruppen, wie Hydroxylgruppen gelten als geschützt. Es wird also so getan, als wären an ihnen Schutzgruppen, die direkt nach der Reaktion wieder entfernt werden.

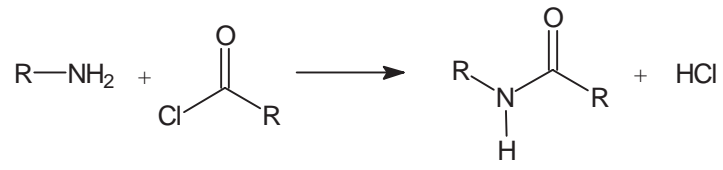


Abbildung 194: Selektive Bildung der Peptidbindung aus einem primären Amin und einem Säurechlorid.

Mit den in Abbildung 193 gezeigten Aminosäuren/Aminosäurechloriden (Serin, Alanin und Phenylalanin bzw. deren Säurechloride) erhält man 27 verschiedene Tripeptide, die in Abbildung 195 dargestellt sind.

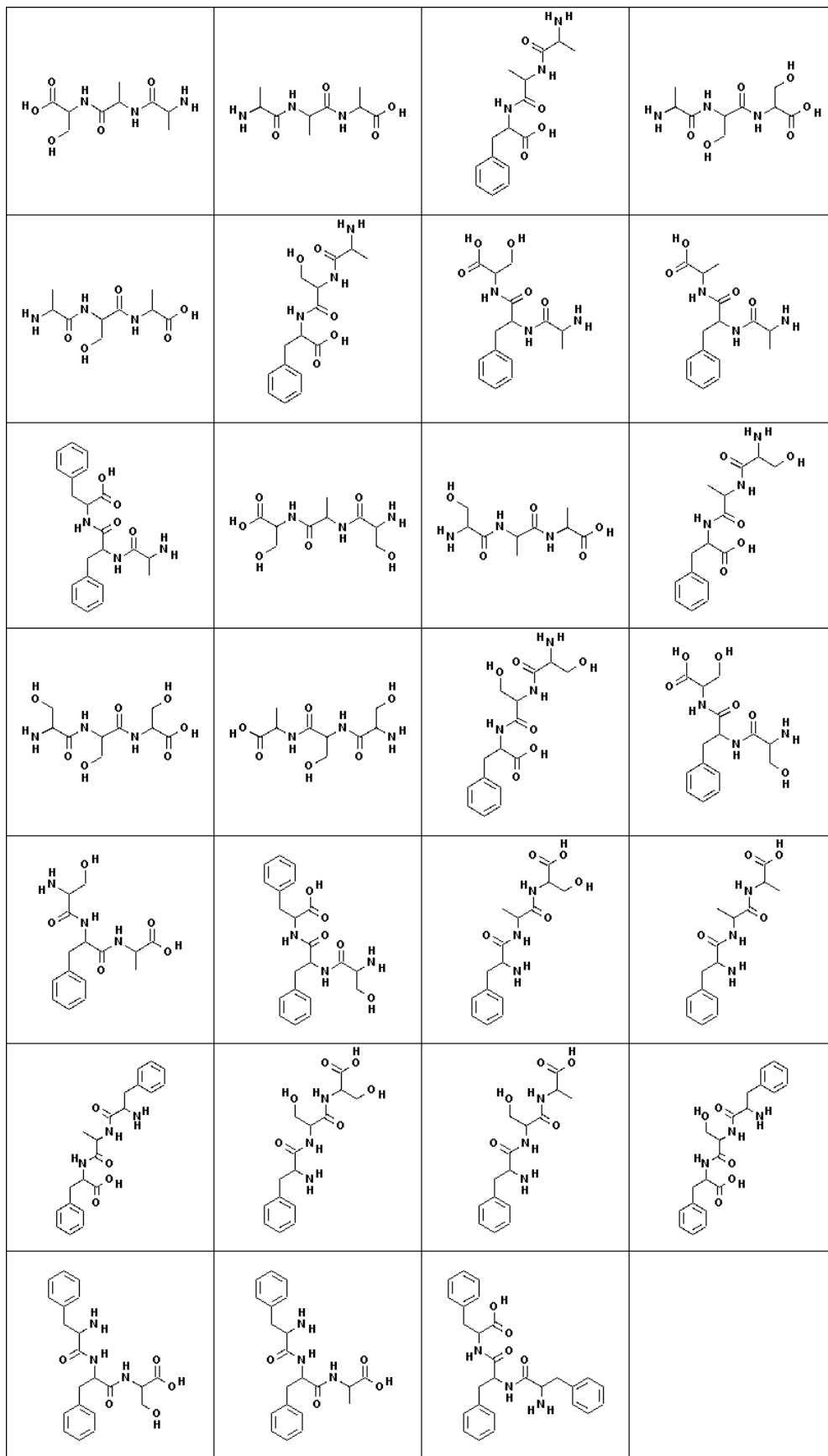


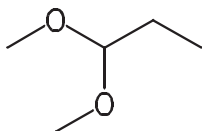
Abbildung 195: Tripeptide aus drei verschiedenen Aminosäuren (Serin, Alanin und Phenylalanin).

Berechnungszeit auf einer SparcStation 10/50 mit 32 MB in Min.:Sek.: 0:59 (C++)

7.2. Simulation von Massenspektren

Neben der Simulation von organischen Reaktionen können mit EROS7 auch die Prozesse in einem Massenspektrometer vorhergesagt werden. Mit der n - und π -Ionisation sowie den vier Fragmentierungstypen α -Spaltung, Oniumreaktion, Carbonyleliminierung, McLafferty-Spaltung und der Wasserstoffumlagerung, wie sie im Kapiteln 6.10.1 beschrieben sind, lassen sich die Elektronenstoßmassenspektren von Verbindungen simulieren, bei denen keine anderen Fragmentierungstypen eine wichtige Rolle spielen. Hierzu wurden bereits im Kapiteln 6.10.2 Beispiele gezeigt. Für die Simulation der im folgenden gezeigten Massenspektren wurden die oben erwähnten Reaktionstypen zusammen mit einem Reaktor aus einer Phase im Modus „monomolekular“ verwendet (siehe Abbildung 196), die die Reaktionen erster Ordnung im Massenspektrometer generiert. Die Ionisation ist eine Reaktion pseudoerster Ordnung, da hier ein Elektron und ein Molekül der Ausgangsverbindung zusammentreffen. Ausgangsmaterial ist die Verbindung, von der das Massenspektrum simuliert wird.

Verbindung, von der
das Massenspektrum
berechnet wird, z.B.:



Ausgangsmaterialien

eine Phase im Modus
„monomolekular“

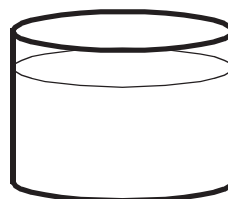


Abbildung 196: Phasen zur Simulation von Massenspektren.

Eine Verbindungsklasse, für die die Massenspektren gut simuliert werden können, sind die Acetale. Abbildung 197 zeigt hierfür ein Beispiel.

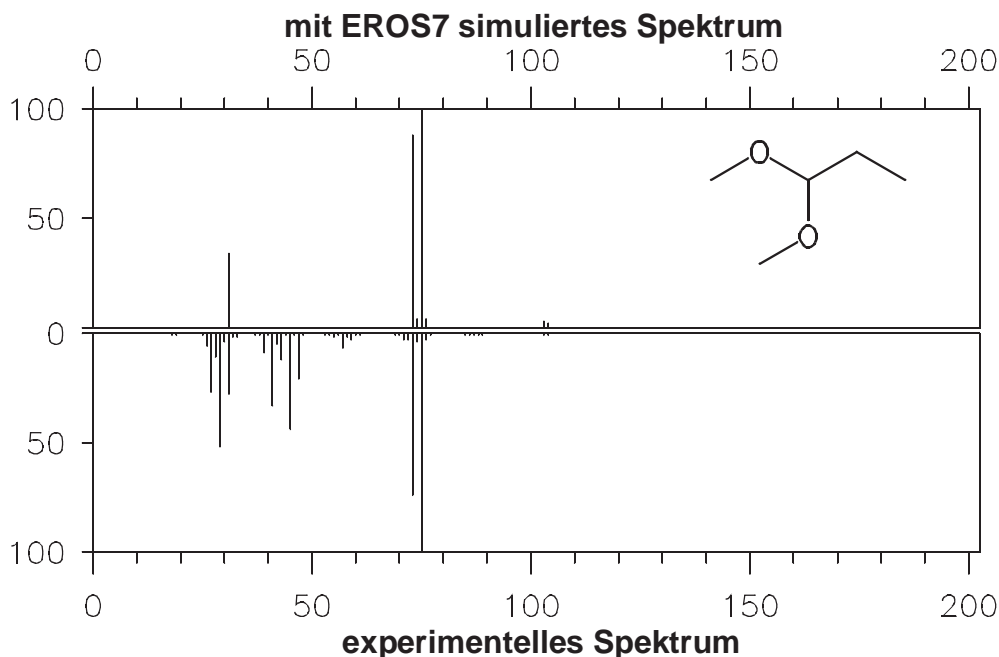


Abbildung 197: Vergleich des mit EROS7 simulierten Massenspektrums (ohne Wasserstoffumlagerung mit dem experimentellen aus [22] für

1,1-Dimethoxypropan.

Berechnungszeit auf einer SparcStation 10/50 mit 32 MB in Min.:Sek.: 0:05 (C++)

Aber auch zur Vorhersage der Spektren von Aminen eignen sich die eingesetzten, bewerteten Fragmentierungsregeln (siehe Abbildung 198 und Abbildung 199).

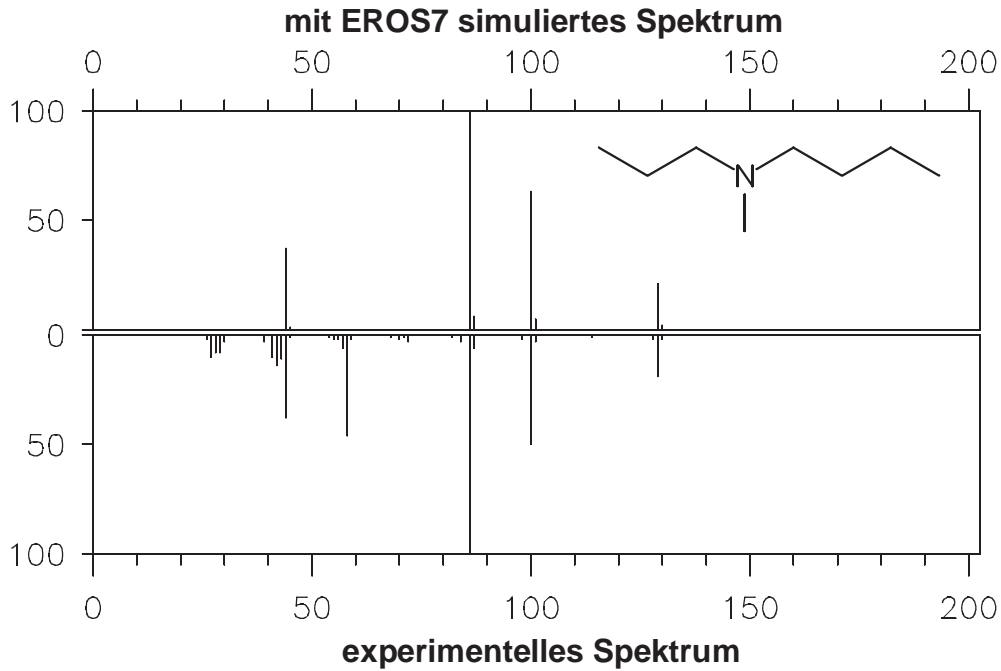


Abbildung 198: Vergleich des mit EROS7 simulierten Massenspektrums (ohne Wasserstoffumlagerung mit dem experimentellen aus [22] für Butylmethylpropylamin.

Berechnungszeit auf einer SparcStation 10/50 mit 32 MB in Min.:Sek.: 0:07 (C++)

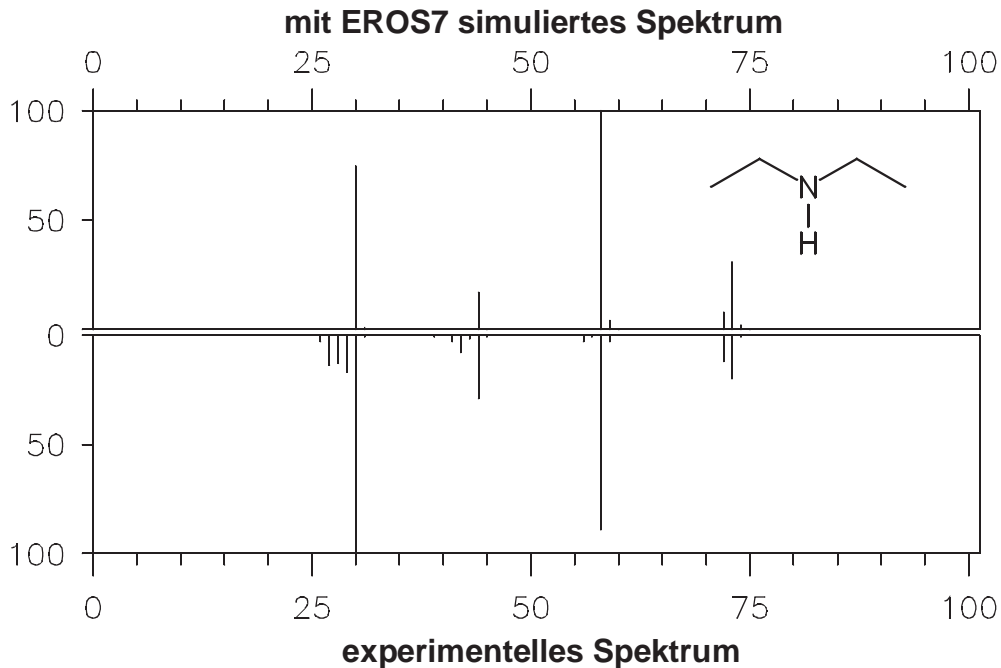


Abbildung 199: Vergleich des mit EROS7 simulierten Massenspektrums (ohne Was-

serstoffumlagerung mit dem experimentellen aus [22] für Diethylamin.

Berechnungszeit auf einer SparcStation 10/50 mit 32 MB in Min.:Sek.: 0:03 (C++)

Unter Einsatz der Wasserstoffumlagerung lassen sich die Massenspektren von Estern gesättigter Fettsäuren simulieren, wie das schon in Kapitel 6.10.2 dargestellte Spektrum von Methylstearat (siehe Abbildung 200).

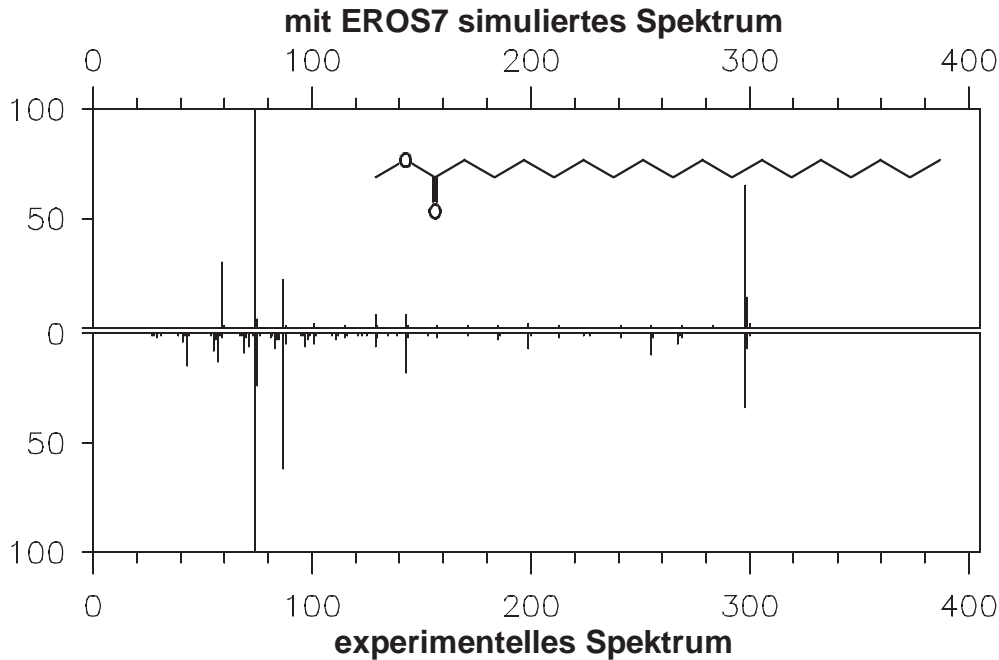


Abbildung 200: Vergleich des mit EROS7 simulierten Massenspektrums (mit Wasserstoffumlagerung) mit dem experimentellen aus [22] für Methylstearat.

Berechnungszeit auf einer SparcStation 10/50 mit 32 MB in Min.:Sek.: 6:49 (C++)

Durch die Vielzahl der erzeugten Wasserstoffumlagerungen erhöht sich die Berechnungszeit ganz erheblich und steht damit in keinem Verhältnis zu den hinzugekommenen Signalen im Spektrum. Dies legt in Zukunft den Einsatz der Wasserstoffumlagerung in der Weise nahe, bei dem jedes Produkt im gesamten Umlagerungssystem direkt aus dem Molekülion gebildet wird. Auf diese Weise würden viele Umlagerungsreaktionen zusammengefaßt, was einen enormen Geschwindigkeitsvorteil haben sollte. (Siehe auch letzter Absatz des Kapitels 6.10.2.)

7.3. Anwendungen mit mehreren Reaktoren

In den Kapiteln 7.1 und 7.2 wurden Anwendungen vorgestellt, die alle mit einem Reaktor auskommen. Anwendungen mit mehreren Reaktoren zeichnen sich dadurch aus, daß in den Reaktoren völlig verschiedene Dinge simuliert werden. Dies kann beispielsweise eine Reaktionsvorhersage in einem Rührkessel oder eine kombinatorische Synthese sein, der die Simulation der Massenspektren in einem zweiten Reaktor folgt.

7.3.1. Reaktionsvorhersage und Simulation der Massenspektren der Produkte

Für die Simulation der Reaktionen von Propanal mit Ethanol und in Folgereaktionen auch mit ihren Reaktionsprodukten wird Reaktor eins mit einer Phase im Modus „Rührkessel“ verwendet, wobei beide Edukte dem Reaktor eins gemeinsam zugeführt werden. Die so gebildeten Ver-

bindungen, die Ausgangsverbindungen und deren Produkte, werden nun einzeln, zusammen mit ihren Konzentrationen an den zweiten Reaktor übergeben. (Für die gemeinsame Übergabe siehe 7.3.2.) Die einzelne Übergabe bedeutet, daß die Berechnungen im zweiten Reaktor für jede Substanz getrennt voneinander durchgeführt wird. Im zweiten Reaktor mit einer Phase im Modus „monomolekular“ wird für jede Verbindung das Massenspektrum vorhergesagt. (Siehe Abbildung 201.)

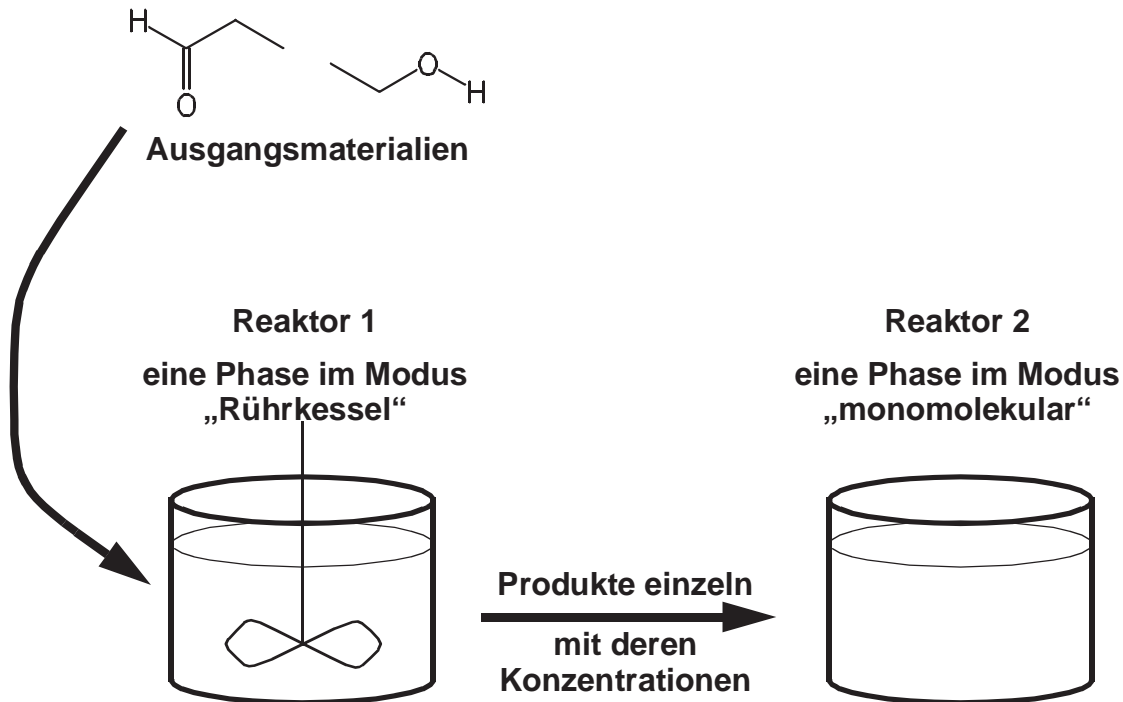


Abbildung 201: Aufbau der Reaktoren und Phasen.

Jeder der beiden Reaktoren hat dabei seine eigene Kinetik. So wurde für den Reaktor eins der Gear-Algorithmus verwendet, während zur Simulation im Reaktor zwei die Wahrscheinlichkeitskinetik eingesetzt wurde. Die Reaktoren und Phasen werden folgendermaßen festgelegt:

```

proc Tcl_rule {} {
  global irule op sub_op rule_trace
  global_name
  global rname rdate kin_mode convlim rxtime minconc k radic_els rctr level
  global phases_p_r phase_contacts phprop comp_edus in_phase ophase
  global attrib_1 center_1 rule_1
  ...
  switch $irule {
  GLOBAL {
    switch $op {
    INIT_RULES {
      set rname „mix-ms.tcl“
      putco name rname
      set rdate „30.09.1997“
      putco date rdate
      # Reaktoren und Phasen
      # 2 Reaktoren mit je einer Phase
      set phases_p_r {1,1,0}
      # Phasen mit den Ausgangsmaterialien fuer die Reaktoren
      # Reaktor 1: aus Datei, Reaktor 2: Phase 1
      set in_phase {0,1}
      # in den Reaktor 1 kommen alle Edukte zusammen,
      # in den Reaktor 2 getrennt nacheinander
      set comp_edus {1,0}
      # keine Phasenkontakte
    }
  }
}

```

```

set phase_contacts(0) 0
# Phasenmodi
set phprop(0) „$MIX“
set phprop(1) „$MONOMOLEC“
# Definieren der Ausgabephase
set ophase 2
# Kinetikmodi der Reaktoren
set kin_mode(0) gear
set kin_mode(1) prob_kin
# Initialisieren der Reaktivitätsvariablen
set k 0.0
# min. Konzentration, Reaktionszeit und geforderter Umsatz
set minconc 1.0e-5
set rxtime 3600.
set convlim 0.95
# Uebergabe der Variablen an EROS7
putco phases_per_reactor phases_p_r
putco phase_contacts phase_contacts
putco phase_property phprop
putco use_all_educts_together comp_edus
putco input_phase_for_reactors in_phase
putco output_phase ophase
putco kinetic_model kin_mode
put reactivity k
putco minimal_concentration minconc
putco reaction_time rxtime
putco conversion_limit convlim
# Holen der Variablen reactor und level von EROS7
get reactor rctr
get level level
return OK
}

```

In den beiden Reaktoren laufen überdies nicht nur Reaktionen mit verschiedenem Typ ab, sondern liefern auch einmal eine Geschwindigkeitskonstante und im anderen Fall eine Reaktionswahrscheinlichkeit. Die Regeln enthalten alle möglichen Reaktionstypen und überprüfen mit den Bedingungen des gesamten Eduktensembles (CONSTR, sub_op=0) auch, ob dieser Reaktionstyp im aktuellen Reaktor stattfindet.

```

...
CONSTR {
  switch $sub_op {
    0 {
      if {$rctr != 1} {return BAD}
      ...
      return OK
    }
  }
}

```

Die Ionisation ist beispielweise nur im Reaktor zwei zugelassen. Um dies entscheiden zu können, werden von EROS7 während der Initialisierung der Regeln Variablen übergeben, die immer den aktuellen Reaktor, die Phase und die Reaktionsebene enthalten. Hat man sich bei der Regelinitialisierung den Zeiger auf die Variable mit der Reaktornummer abgeholt, kann mit den Bedingungen des Eduktensembles auch festgelegt werden, welche Reaktionstypen im aktuellen Reaktor stattfinden.

Für den Reaktor eins waren Reaktionstypen zugelassen, die aus Propanal und Ethanol säurekatalysiert das Halbacetal, Acetal und den Diethylether, sowie zusammen mit dem dabei entstehenden Wasser auch das Hydrat des Propanals gebildet haben. Zur Simulation der Massenspektren im Reaktor zwei wurden die Ionisation und Fragmentierungen eingesetzt, die auch schon zur ausschließlichen Massenspektrenvorhersage (siehe Kapitel 6.10.2 und 7.2) verwendet wurden. Auf die Wasserstoffumlagerung wurde dabei verzichtet.

Da die Signale des Massenspektrums vor der Ausgabe von EROS7 so skaliert werden, daß der Basepeak 100% erhält, hat die vom Reaktor eins bestimmte Endkonzentration der Substanz keinen Einfluß auf die Spektren, weshalb hier nicht darauf eingegangen wird (im Gegensatz zum nächsten Kapitel 7.3.2).

Die simulierten Massenspektren sind zusammen mit einem experimentellen Spektrum zum Vergleich in den Abbildungen 202 bis 208 dargestellt. Für drei der Verbindungen stand allerdings kein experimentelles Massenspektrum zur Verfügung: Halbacetal, Wasser und Propanalhydrat. Vom Propanalhydrat und Halbacetal dürfte die Stabilität der Verbindungen nicht ausreichen, daß von ihnen selbst ein Massenspektrum aufgenommen werden kann. Sie werden schon vor der Ionisation Wasser bzw. Ethanol abspalten. Ein solches Verhalten wird jedoch nicht von den Fragmentierungsregeln berücksichtigt. Statt der angegebenen, simulierten Spektren müßte das Gemischspektrum der Spaltungsprodukte angegeben werden.

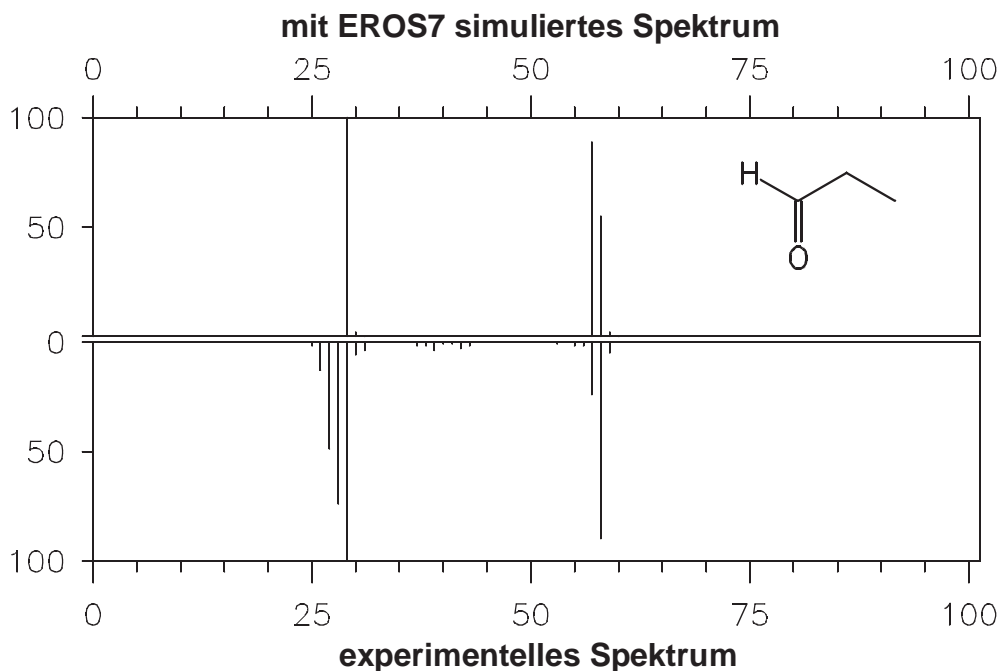


Abbildung 202: Simuliertes Spektrum von Propanal. Experimentelles Spektrum [22].

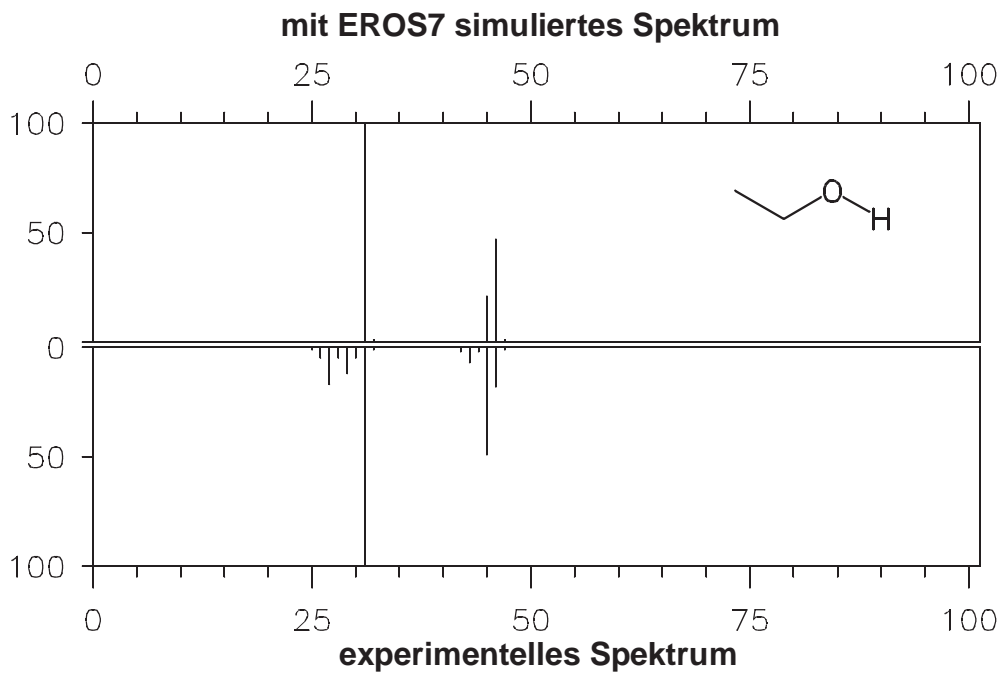


Abbildung 203: Simuliertes Spektrum von Ethanol. Experimentelles Spektrum [22].

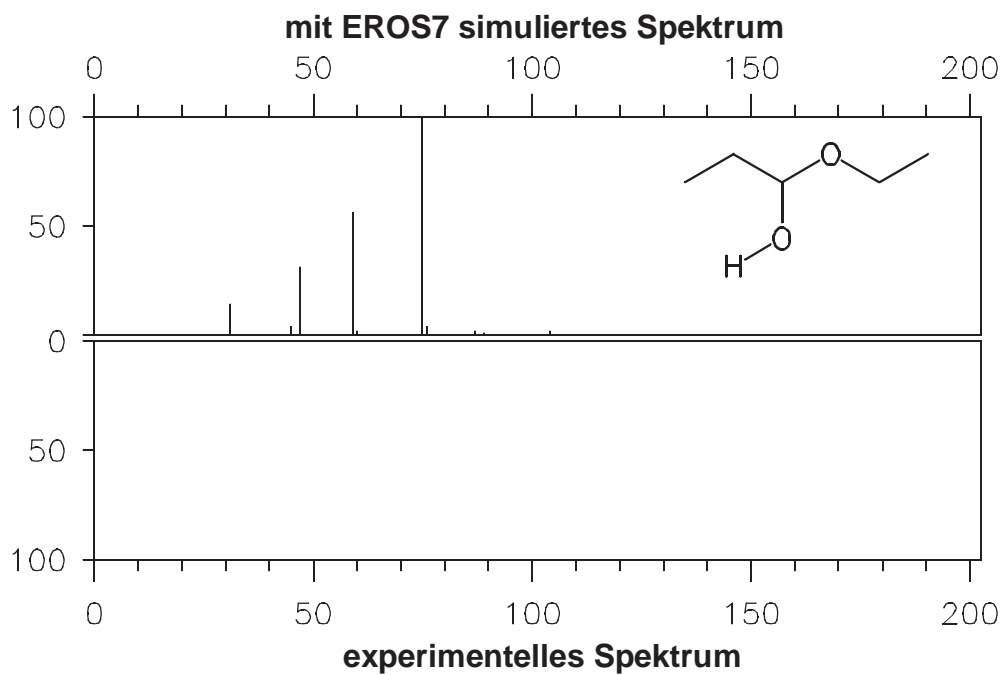


Abbildung 204: Simuliertes Spektrum des Halbacetals. (siehe oben)

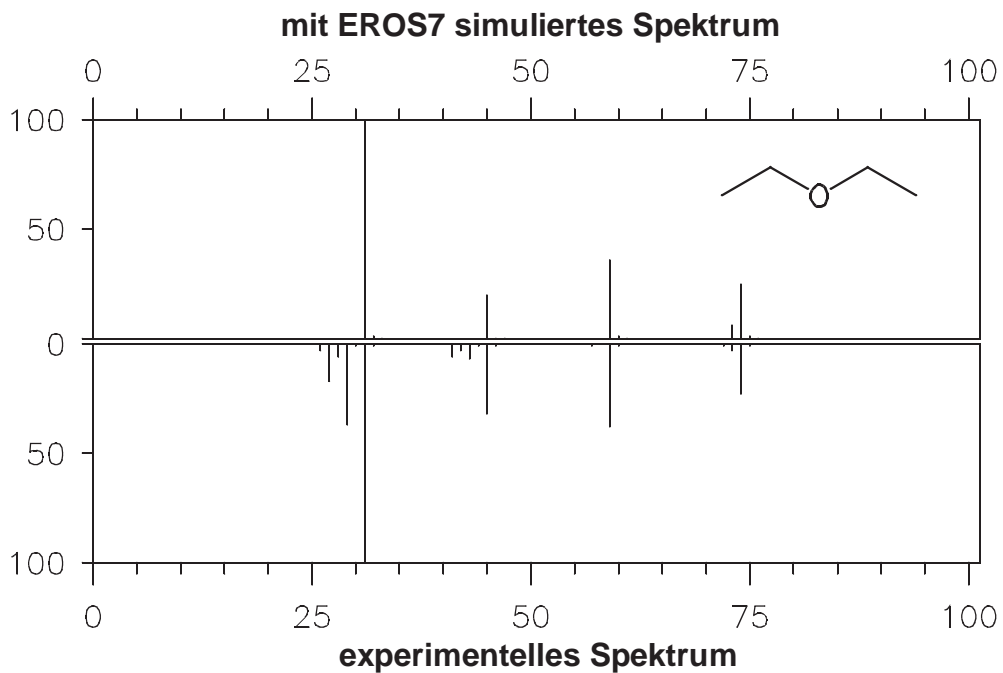


Abbildung 205: Simuliertes Spektrum von Diethylether. Experimentelles Spektrum [22].

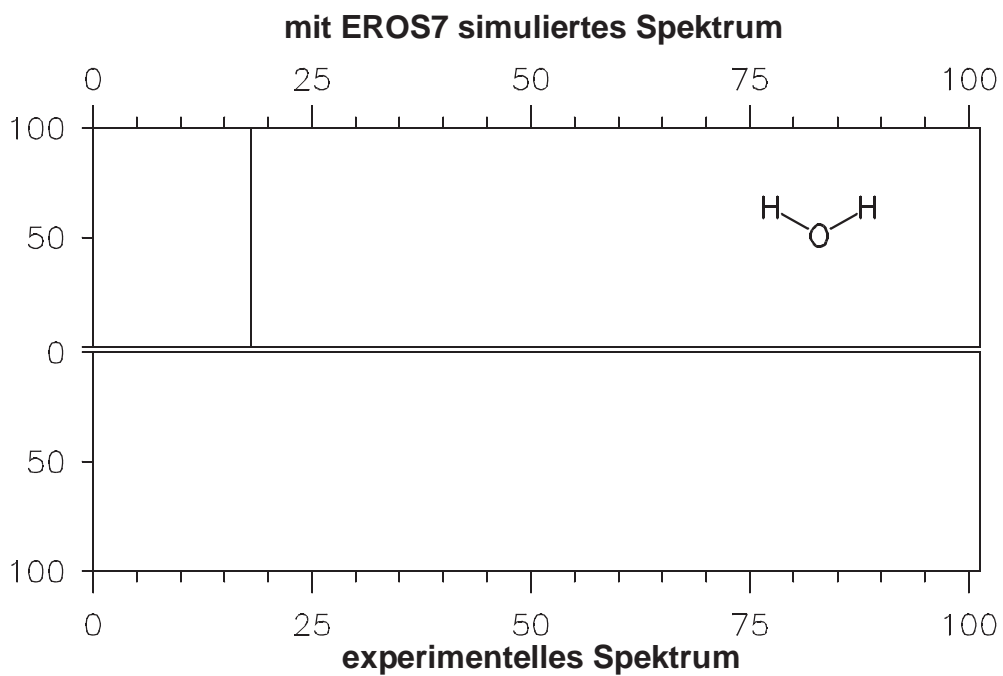


Abbildung 206: Simuliertes Spektrum von Wasser. Es stand kein experimentelles Spektrum zur Verfügung.

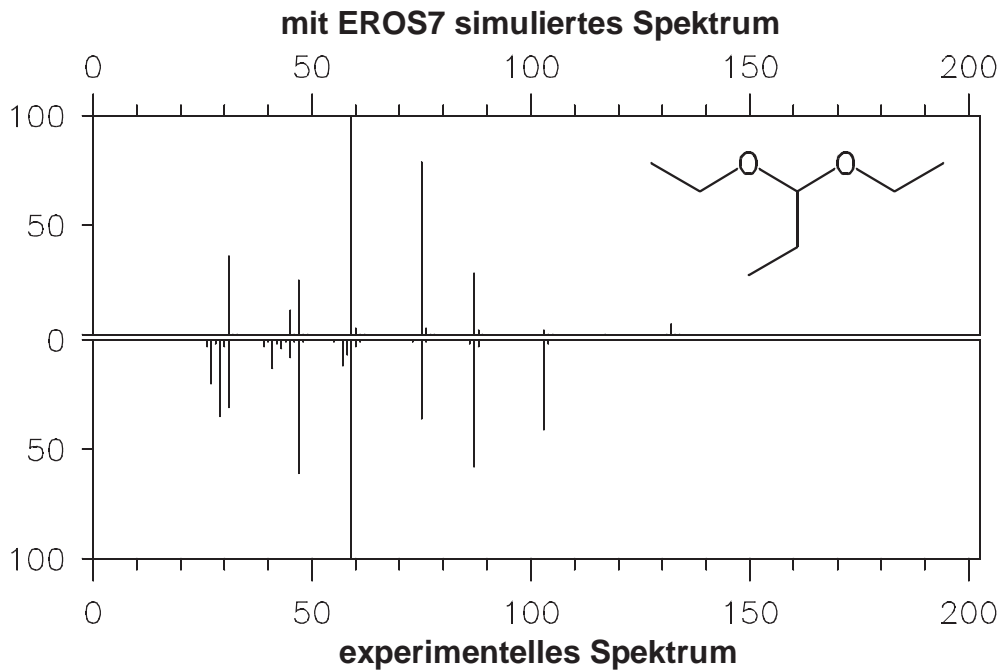


Abbildung 207: Simuliertes Spektrum des Acetals. Experimentelles Spektrum [22].

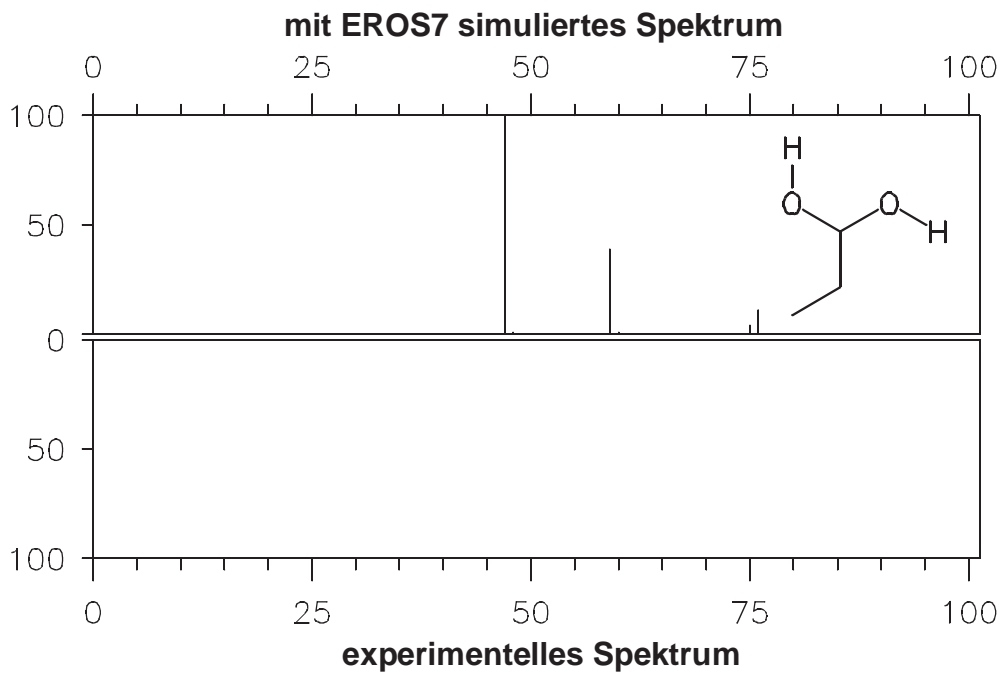


Abbildung 208: Simuliertes Spektrum von Propanalhydrat. (siehe oben)

Berechnungszeit auf einer SparcStation 10/50 mit 32 MB in Min.:Sek.: 0:50 (C++)
 (für alle Spektren, 50% Umsatz von Ethanol)

7.3.2. Simulation des Spektrums einer Reaktionsmischung

Ändert man die zweistufige Simulation aus Kapitel 7.3.1 ab und übergibt die Produkte des Reaktors eins zusammen an den Reaktor zwei, werden im Reaktor zwei nicht die Spektren der Produkte vorhergesagt, sondern das Massenspektrum, das sich ergibt, wenn das Spektrum vom

Gemisch aufgenommen wird. Hierbei geben die Konzentrationen der im Reaktor eins gebildeten Produkte an, mit welcher relativen Intensität die Signal dieser Verbindung im Spektrum zu sehen sind. Abbildung 209 zeigt den angenommenen Konzentrationsverlauf für die Reaktionsvorhersage im Reaktor eins, wobei die Endkonzentrationen an den Reaktor zwei zur Spektrenvorhersage übergeben werden. Da die Geschwindigkeitskonstanten für die einzelnen Prozesse nur sehr grob abgeschätzt wurden, können die Konzentrationsverläufe vom tatsächlichen abweichen. Aus diesem Grund soll auf sie nicht näher eingegangen werden. Dies stellt allerdings kein Problem dar, da hier nur gezeigt werden soll, daß EROS7 in der Lage ist, mehrere Simulation miteinander zu kombinieren.

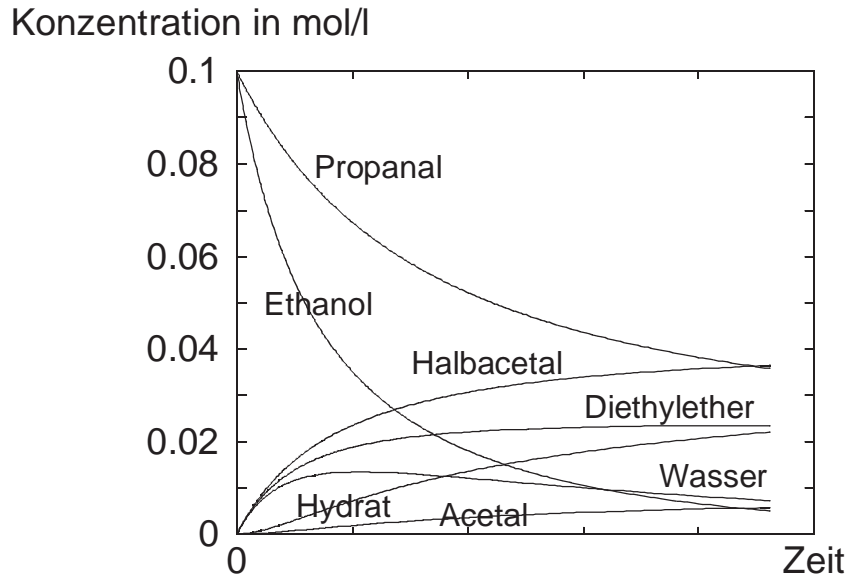


Abbildung 209: Angenommener Konzentrationsverlauf der Verbindungen im Reaktor eins.

Das Gemischspektrum (siehe Abbildung 210) ergibt sich aus der mit den Endkonzentrationen (siehe Abbildung 209) gewichteten, simulierten Massenspektren der Abbildungen 202 bis 208. Die Gewichtung erfolgt dabei für den Gesamtionenstrom der Spektren, wie es einem Gemischspektrum entspricht.

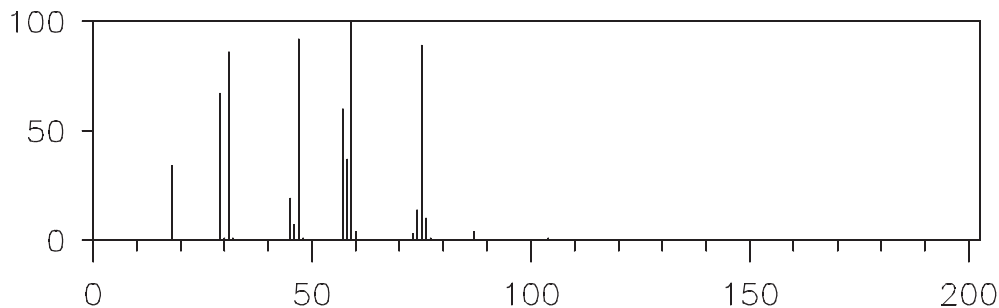


Abbildung 210: Simuliertes Gemischspektrum.

Berechnungszeit auf einer SparcStation 10/50 mit 32 MB in Min.:Sek.: 1:14 (C++)
(95% Umsatz von Ethanol)

8. Zusammenfassung und Ausblick

8.1. Massenspektrensimulation

Im Rahmen dieser Arbeit wurde anfänglich die automatische Ableitung von Reaktivitäten für die im Massenspektrometer ablaufenden Fragmentierungen und Umlagerungen aus Elektronenstoßmassenspektren unter Verwendung der dazugehörigen Struktur untersucht. Für die verschiedenen Reaktionstypen wurde dann der Zusammenhang zwischen Struktur und Reaktivität in Form von Funktionen und neuronalen Netzen ermittelt. Das auf diese Weise gewonnene Wissen über die Prozesse im Spektrometer kann schließlich zur Simulation der Massenspektren neuer Verbindungen genutzt werden.

Dabei stellte sich heraus, daß die eingesetzte Repräsentation der chemischen Struktur, Bindungslisten, für die Beschreibung der Vorgänge in einem Massenspektrometer nicht ausreichen. Bindungslisten kodieren zum Beispiel die Bindungen in einem Benzolmolekül nach Lewis als alternierende Einfach- und Doppelbindungen. Diese Darstellung bringt zunächst die Schwierigkeit, daß vor jeder Reaktion alle mesomeren Grenzstrukturen erzeugt werden müssen, um zu allen möglichen Reaktionen zu gelangen. Speziell bei der Generierung dieser Grenzstrukturen hat sich ein weiteres Defizit der Bindungslisten ausgewirkt. Die Bindungsliste macht keine Angaben darüber, welche Orientierungen die einzelnen Orbitale besitzen. Betrachten wir dazu einen ionisierten Enolether in Abbildung 211.

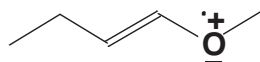


Abbildung 211: Ionisierter Enolether in der Darstellung einer Bindungsliste.

In dieser Darstellung kann nicht festgestellt werden, ob nun das Radikal oder das freie Elektronenpaar mit dem π -System der Doppelbindung konjugiert ist. Um zu allen richtigen mesomeren Strukturen einer Verbindung zu kommen, muß man alle möglichen Orientierungen zulassen. Gibt es mehrere mögliche Ausrichtungen der Atomorbitale und lokale Elektronenverschiebungen in der Darstellung der Struktur als Bindungsliste, was bei vielen Radikalkationen der Fall ist, öffnet man so aber auch den Weg, „Grenzstrukturen“ zu erzeugen, die nicht mesomer zueinander sind.

Trotz dieser Schwierigkeiten konnte gezeigt werden, daß es auf diesem Wege der automatischen Ableitung von Reaktivitäten für die Fragmentierungs- und Umlagerungsprozesse möglich ist, bei einer beschränkten Zahl von Verbindungsklassen Teile der Massenspektren inklusive der Intensitäten der Signale richtig vorherzusagen. Für wichtige Fragmentierungsprozesse, wie die induktive Spaltung, konnte jedoch keine Struktur-Reaktivitäts-Beziehung gefunden werden. Dies liegt an der für Radikalkationen unzureichenden Darstellung der chemischen Struktur als Bindungsliste, beschränkter Definitionsmöglichkeiten der Reaktionstypen (keine verzweigten Substrukturen) und daran, daß die Aktivierungsenergie der Ionen, die von der Entstehungsgeschichte des Kations abhängt, berücksichtigt werden konnte.

8.2. Neue Datenstruktur und Reaktionsvorhersage

Aus den in 8.1 genannten Gründen wurde begonnen, zusammen mit Frau S. Bauerschmidt (neue MO-orientierte, chemische Datenstruktur RICOS) und Frau L. Steinhauer (Lösung der Differentialgleichungen, neuronale Netze und Anschluß der Tcl-Regeln), ein neues Reaktionsvorhersageprogramm zu entwickeln, das diese Beschränkungen nicht mehr aufweist. Bei der Entwicklung von EROS7 wurden einige Module ergänzt, damit es neben der Massenspektrensimulation auch in einer Reihe anderer Bereiche eingesetzt werden kann, wie der Reaktionsvor-

hersage, von der auch sein Name abgeleitet ist: Erzeugung von Reaktionen für die Organische Synthese.

Mit EROS7 konnte ein System geschaffen werden, das nicht nur organische und metallorganische Verbindungen sowie Radikalkationen behandeln, sondern diese auch unter verschiedenen Reaktionsführungen umsetzen kann. Hierzu wurde das Konzept der Reaktoren und Phasen eingeführt.

EROS7 ist ein Expertensystem, das die verschiedenartigsten Reaktionsvorhersagen ausführen kann. Welche dies sind und wie sie im einzelnen geschehen, wird in der extern zum System liegenden Regeldatei festgelegt. Dazu gehören die Beschreibungen der Reaktionstypen mit ihren Bewertungsfunktionen zur Ermittlung der Reaktivität der erzeugten Reaktionen sowie die Zahl und die Arten der Phasen und Reaktoren. Mit den verschiedenen Einstellung lassen sich so sowohl großtechnisch durchgeführte (kontinuierliche Zuführung der Reaktanden) als auch Synthesen im Labormaßstab simulieren.

8.3. Arten der Reaktionsführung

Zunächst sollen die Möglichkeiten der verschiedenen Reaktionsführungen für Prozesse im Labor und für industrielle Chargenprozesse näher betrachtet werden. Chargenprozesse werden zur Herstellung von Substanzen in kleinen Mengen verwendet, wie es für die meisten Medikamente zutrifft. Für all diese Synthesen gilt, daß eine bestimmte Menge der Ausgangssubstanzen eingesetzt wird, aber während der Reaktion weder weitere Ausgangsmaterialien hinzugefügt, noch Produkte entnommen werden. Die Reaktion findet dabei in einem Rührkessel statt, der den Dreihalskolben, das Schlenkrohr oder den industriellen Reaktionskessel symbolisiert. In ihm werden die Edukte ständig vermischt, wobei jede der sich im Kessel befindlichen Verbindungen mit einer anderen reagieren kann. Alternativ dazu ist es auch möglich, die Reaktionsgenerierung auf Reaktionen erster und pseudoerster Ordnung zu beschränken. Diese Art der Reaktionsführung wird eingesetzt um den Abbau von Chemikalien in der Umwelt oder die Metabolisierung von Arzneimitteln vorherzusagen.

Neben der Simulation der entstehenden Strukturen kann EROS7 aber auch die Konzentrationsverläufe der einzelnen Komponenten bestimmen. Hierzu werden die Geschwindigkeitskonstanten für die erzeugten Reaktionen benötigt. Sie können, abhängig von den physikochemischen Eigenschaften der Reaktanden, in den Regeln bestimmt werden. Die Zuordnung der Eigenschaften der Edukte und Produkte zur Reaktivität der Reaktion kann dabei aus einer Konstanten, einer mathematischen Funktion oder dem Aufruf eines neuronalen Netzes bestehen. Als Netztypen stehen das Backpropagationnetz, das Kohonennetz und das Counterpropagationnetz zur Verfügung. Es können Reaktionen erster, pseudoerster und zweiter Ordnung behandelt werden. Für die Vorhersage enzymkatalysierter Reaktionen besteht überdies die Möglichkeit, ein kinetisches Verhalten der Reaktion nach Michaelis-Menten oder nullter Ordnung anzunehmen. All dies kann für einen, aber auch für mehrere Phasen simuliert werden. Um den Austausch zwischen den Phasen berücksichtigen zu können, werden zu den Geschwindigkeitskonstanten der Reaktionen auch Werte für die Menge des Stoffübergangs von einer in eine andere Phase benötigt. So können beispielsweise auch die verschiedenen Kompartimente einer Pharmakokinetik formuliert werden. Es kann dadurch die Absorption des Medikaments ins Blut, die Speicherung im Gewebe, die Ausscheidung und die Verfügbarkeit am Wirkort simuliert werden. Arzneimittel werden meist nicht als Infusion verabreicht, sondern in Form von Tablette, Dragees und Spritzen. Auch die Konzentrationsverläufe bei einer periodischen und frei wählbaren Dosierungsweise, inklusive der Mehrfachdosierung zu einem Zeitpunkt, können mit EROS7 vorhergesagt werden.

Bei der Infusion wird die Arznei im Gegensatz zur meistgenutzten Darreichungsform der Tablette oder des Dragees dem Körper des Patienten kontinuierlich zugeführt und die Abbaupro-

dukte und möglicherweise auch Anteile des Medikaments verlassen auch wieder den Organismus. Auch großtechnisch werden Synthesen zur Produktion von Verbindungen im Maßstab vieler Tonnen pro Jahr kontinuierlich durchgeführt. Man erspart sich so die Totzeiten der Reaktionsbehälter, in denen sie entleert und neu befüllt werden. Zum Einsatz kommen hierfür Rührkessel, denen ständig neue Ausgangsmaterialien zugeführt und ein Teil der Reaktionsmischung entnommen wird. Für ein anderes Verweilzeitverhalten der Edukte im Reaktionsgefäß kommen auch das Reaktionsrohr und die Rührkesselkaskade zum Einsatz. All diese Arten der Reaktionsführung können mit EROS7 simuliert werden.

Ein weiteres Gebiet, in dem EROS7 eingesetzt werden kann, ist die kombinatorische Chemie. Hierfür spielen die kinetischen Betrachtungen üblicherweise keine Rolle. Ohne Hilfe des Computer ist es mühsam, alle Verbindungen vollständig zu bestimmen, die entstehen, wenn je eine der Verbindungen aus mehreren Sätzen miteinander reagieren. Hierbei können schnell Tausende von neuen Substanzen gebildet werden. Die Arbeit, ihre Strukturen anzugeben, kann von EROS7 abgenommen werden.

Schließlich ist es auch möglich mit EROS7 Massenspektren zu simulieren. Die Regeln, die für die in dieser Arbeit gezeigten, simulierten Massenspektren verwendet wurden, haben jedoch einen sehr begrenzten Einsatzbereich. Für die Nutzung von EROS7 zur Vorhersage von Massenspektren müßten diese Regeln stark verbessert werden.

Jede der erwähnten Vorhersagen kann alleine eingesetzt oder auch mit einer anderen kombiniert werden. So ist es möglich die Massenspektren aller bei einer normalen oder auch kombinatorischen Synthese entstehenden Verbindung zu simulieren.

Im Kapitel 7 dieser Arbeit konnten hierfür einige Beispiele gezeigt werden. In Zukunft werden diesen noch viele hinzugefügt werden, die die breiten Einsatzmöglichkeiten von EROS7 zeigen werden. Auch die Umkehrung der Reaktionsvorhersage - die Syntheseplanung - ist hierbei nicht ausgeschlossen.

9. Für die Darstellungen verwendete Programme

Als Programme zur Darstellung von Strukturformeln wurden Chemwin [75], xmed++ [76] und in ganz einfachen Fällen FrameMaker [77] eingesetzt, mit dem auch die Arbeit geschrieben wurde. Waren die Strukturen das Ergebnis eines Programmlaufs, wurden sie mit csbr, dem Browser aus den CACTVS-System [78] gezeichnet. Die Spektren inklusive der dazugehörigen Strukturen wurden mit VEGAS [79] gezeichnet und mit hpgl2mif [80] vom HPGL-Format (Format für Plotter von Hewlett Packard) in das MIF-Format (Maker Interchange Format, eines der Formate von FrameMaker [77]) umgewandelt. Die Diagramme, bei denen die berechneten gegen die vorgegebenen Reaktionswahrscheinlichkeiten aufgetragen sind, wurden mit yplot [81] dargestellt. Konzentrationsverläufe sind mit gnuplot [82] gezeichnet, die Darstellungen von Reaktionsnetzwerken mit treetool [83]. Für einige Nachbearbeitungen von Grafiken wurden Corel Photopaint [84] sowie xpaint [85] zur Formatkonvertierung eingesetzt.

10. Literatur

- [1] F. W. McLafferty
Interpretation of Mass Spectra
University Science Books, Mill Valley, Calif., 1980
- [2] K. Dialer, A. Löwe
Chemische Reaktionstechnik
Hanser-Verlag, München, 1975
- [3] W. Vauck, H. Müller
Grundoperationen chemischer Verfahrenstechnik
Verlag Chemie, Weinheim, 8. Auflage, 1990
- [4] H. Lohninger, K. Varmuza
Selective Detection of Classes of Chemical Compounds by Gas Chromatography / Mass Spectroscopy / Pattern Recognition, Polycyclic Aromatic Hydrocarbon and Alkanes
Anal. Chem., **57**, 236-244, 1987
- [5] K. Funatsu, A. C. Del Carpio, S. Sasaki
Automatic Structure Elucidation System CHEMICS
Fresenius Z. Anal. Chem., **324**, 750-759, 1986
- [6] W. Bremser
Angew. Chem., **100**, 252, 1988; Angew. Chem., Int. Ed. Engl., **27**, 247, 1988
- [7] SpecInfo 2.0 von der Chemical Concepts GmbH, Weinheim
- [8] D. Henneberg
MPI für Kohlenforschung Mülheim a. d. Ruhr, persönliche Mitteilungen
- [9] J. Gasteiger, C. Jochum, M. Marsili, J. Thoma
The synthesis planning program EROS
MATCH **6**, 177-199, 1979
- [10] P. Röse
Ein Computerprogramm zur Vorhersage organisch-chemischer Reaktionen,
Entwicklung und Anwendung
Dissertation, Technische Universität München 1990
- [11] P. Röse, J. Gasteiger
Automated Derivation of Reaction Rules for the EROS 6.0 System for Reaction Prediction
Anal. Chim. Acta, **235**, 163-168, 1990
- [12] J. Gasteiger, U. Hondelmann, P. Röse, W. Witzendichler
Computer Assisted Prediction of the Degradation of Chemicals: Hydrolysis of Amides and Benzoylphenylureas
J. Chem. Soc., Perkin Trans. **2**, 193, 1995
- [13] J. Bauer, E. Fontain, I. Ugi
Computer-Assisted Bilateral Solution of Chemical Problems and Generation of Reaction Networks
Anal. Chim. Acta., **210**, 123-134, 1988
- [14] K. Funatsu
Toyohashi University of Technology, persönliche Mitteilungen
- [15] R. Höllering, J. Gasteiger, W. Hanebeck
Simulation of Mass Spectra of Organic Compounds by Massimo
in AIP conference proceedings **330**: E.C.C.C. 1
F. Bernardi, J. L. Rivail (Hrsg.)
American Institute of Physics, Woodbury, NY, 1995, 782

- [16] J. Gasteiger, W. D. Ihlenfeldt, R. Fick, J. R. Rose
Similarity Concepts for the Planning of Organic Reactions and Syntheses
J. Chem. Inf. Comput. Sci., **32**, 700-712, 1992
- [17] E. J. Corey, W. T. Wipke
Science, **166**, 178, 1969
- [18] H. Budzikiewicz, C. Djerassi and D. H. Williams
Mass Spectrometry of Organic Compounds
Holden Day, San Francisco, Calif., 1967
- [19] F. W. McLafferty and D. B. Stauffer
The Wiley/NBS Registry of mass spectral data
1-7, Wiley, New York, 1989
- [20] S. R. Heller and G. W. A. Milne
EPA/NIH Mass spectral data base
1-5, U. S. Government printing office, 1978
- [21] D. Henneberg und B. Weimann
Spektrenbibliotheken in der massenspektrometrischen Analytik
Softwareentwicklung in der Chemie **2**, J. Gasteiger (Ed.), 153-157,
Springer Verlag, Heidelberg, 1988
- [22] MassLib vom MPI für Kohlenforschung, Mülheim a. d. Ruhr
- [23] J. Gasteiger, W. Hanebeck, K.-P. Schulz, S. Bauerschmidt, and R. Höllering
Automatic Analysis and Simulation of Mass Spectra
Computer-Enhanced Analytical Spectroscopy **4**
L. Charles (Ed.), Plenum Press New York, 97-133, 1993
- [24] K.-P. Schulz, S. Bauerschmidt, R. Höllering, J. Gasteiger
Automatic Elucidation of Reactions in the Mass Spectrometer
in AIP conference proceedings **330**: E.C.C.C. 1
F. Bernardi, J. L. Rivail (Hrsg.)
American Institute of Physics, Woodbury, NY, 1995, 726-733
- [25] K.-P. Schulz
Computergestützte Untersuchungen über Zusammenhänge zwischen Struktur und
Massenspektrum
Dissertation, Technische Universität München 1992
- [26] W. Hanebeck
Simulation und Rekonstruktion von Reaktionen im Massenspektrometer
Dissertation, Technische Universität München 1991
- [27] G. Spitteller und M. Spitteller-Friedmann
Zur theoretischen Deutung der Massenspektren organischer Verbindungen
Liebig's Ann. Chem., **690**, 1, 1965
- [28] J. Gasteiger and M. G. Hutchings
*Quantification of effective polarisability. Applications to studies of X-ray photoelectron
spectroscopy and alkylamine protonation*
J. Chem. Soc. Perkin **2**, 559-564, 1984
- [29] J. Gasteiger, M. Marsili
*Iterative Partial Equalisation of Orbital Electronegativity - A Rapid Access to Atomic
Charges*
Tetrahedron **36**, 3219-3228, 1980
- [30] M. D. Guillen, J. Gasteiger
*Extension of the Method of Iterative Partial Equalization of Orbital Electronegativity to
Small Ring Systems*
Tetrahedron **39**, 1331-1335, 1983

- [31] J. Gasteiger, H. Saller
Berechnungen der Ladungsverteilung in konjugierten Systemen durch eine Quantifizierung des Mesomeriekonzepts
Angew. Chem. **97**, 699-701, 1985
- [32] H. Saller
Quantitative empirische Modelle für elektronische Effekte in π -Systemen und für die chemische Reaktivität
Dissertation, Technische Universität München 1985
- [33] A. Fröhlich
Modelle zur Beschreibung elektronischer Effekte in π -Systemen und ihr Einsatz in der Modellierung der elektrophilen aromatischen Substitution
Dissertation, Technische Universität München 1993
- [34] J. Gasteiger, P. Jakob, U. Strauß
Critical Evaluation of Additivity Schemes for Estimating Heats of Atomization
Tetrahedron **35**, 139-146, 1979
- [35] J. Gasteiger, O. Dammer
Automatic Estimation of Ring Strain Energies
Tetrahedron **34**, 2939-2945, 1978
- [36] K. Rafeiner
Quantifizierung struktureller Wechselwirkungen für die empirische Berechnung der Bildungsenthalpie
Dissertation, Technische Universität München 1993
- [37] K. Rafeiner, J. Gasteiger
Calculation of Heats of Formation for Organic Radicals and Cations
Softwareentwicklung in der Chemie **5**, J. Gmehling (Hrsg.)
Springer Verlag, Heidelberg, 1991, 69-75
- [38] Ch. Hiller
Ein direkter Weg von der Konstitution zum dreidimensionalen Molekülmodell
Dissertation, Technische Universität München 1989
- [39] J. Gasteiger, C. Jochum
An Algorithm for the Preception of Synthetically Important Rings
J. Chem. Inf. Comput. Sci., **19**, 43-48, 1979
- [40] D. G. Kleinbaum
Logistic Regression
Springer-Verlag New York 1994
- [41] V. Simon
Bewertungsmechanismen für Reaktionen und Anwendung neuronaler Netze zur Reaktionsvorhersage
Dissertation, Technische Universität München 1993
- [42] V. Simon, J. Gasteiger, and J. Zupan
A Combined Application of Two Different Neural Network Types for the Prediction of Chemical Reactivity
J. Am. Chem. Soc. **115**, 9148, 1993
- [43] T. Aoyama, H. Ichikawa
Reconstruction Learning of Weight Matrices in Neural Networks - A Method of Correlating Outputs with Inputs
Chem. Pharm. Bull. **39**(5), 1222, 1991
- [44] J. Gasteiger, J. Zupan
Neural Networks for Chemists: An Introduction
VCH-Verlag, Weinheim, 1993

- [45] T. Kohonen
Self-Organisation and Assoziative Memory
Springer-Verlag, New York, 3. Auflage, 1989
- [46] W. Hanebeck, J. Gasteiger
Rapid Empirical Calculation of the First (n and π) Ionisation Potential of Organic Molecules
Journal of Computational Chemistry, **14**(2), 138, 1993
- [47] W. Vetter
F. Hoffmann LaRoche AG Basel, persönliche Mitteilungen
- [48] T. Körtvélysi, L. Seres
Theoretical Studies on the 1,N ($N = 2, 3, 4, 5$) Intramolecular Radical H-Atom Shifts
in AIP conference proceedings **330**: E.C.C.C. 1
F. Bernardi, J. L. Rivail (Hrsg.)
American Institute of Physics, Woodbury, NY, 1995, 243
und dazugehöriges Poster
- [49] T. Clark
A Handbook of Computational Chemistry
John Wiley & Sons, 1985
- [50] G. Rauhut, A. Alex, J. Chandrasekhar, T. Steinke, W. Sauer, B. Beck,
M. Hutter, and T. Clark
VAMP5
Universität Erlangen 1995
- [51] S. Bauerschmidt, J. Gasteiger
Overcoming the Limitations of a Connection Table Description: A Universal Representation of Chemical Species
J. Chem. Inf. Comput. Sci., **37**, 705-714, 1997
- [52] S. Bauerschmidt
Repräsentation von Molekülstrukturen zur computergestützten Behandlung chemischer Reaktionen
Dissertation, Friedrich-Alexander-Universität Erlangen-Nürnberg 1997
- [53] P. W. Atkins
Physikalische Chemie
VCH-Verlag, Weinheim, 1. Auflage, 1987
- [54] N. N. Greenwood, A. Earnshaw
Chemie der Elemente
VCH, Weinheim, 1990
- [55] W. D. Ihlenfeldt, J. Gasteiger
Hash Codes for the Identification and Classification of Molecular Structure Elements
J. Comput. Chem., **15**, 793-813, 1994
- [56] A. Streitwieser Jr., C. H. Heathcock
Introduction to Organic Chemistry
Macmillan Publishing Company, New York, 3. Auflage, 1985
- [57] F. Patat, K. Kirchner
Praktikum der Technischen Chemie
Walter de Gruyter, Berlin, 1986
- [58] C. W. Gear
Numerical Initial Value Problems in Ordinary Differential Equations
Prentice Hall, Englewood Cliffs, N.J. 1971

- [59] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling
Numerical Recipes, The Art of Scientific Computing,
Cambridge University Press, 1989
- [60] N. Köckler
Numerische Algorithmen in Softwaresystemen
Teubner, Stuttgart, 1990
- [61] L. Steinhauer, R. Höllering, S. Bauerschmidt, T. Kostka, J. Gasteiger
Kinetic Calculations by the Simulation of Organic Reactions
ZAMM (Zeitschrift für angewandte Mathematik und Mechanik), 1997, eingereicht
http://www.uni-regensburg.de/Fakultaeten/nat_Fak_I/Mennicken/gamm97/s20/steinhauer/steinhauer.html
- [62] T. Kostka
Universität Erlangen, persönliche Mitteilungen
- [63] J. K. Ousterhout
Tcl and the Tk Toolkit
Adison-Wesley, 1984
- [64] SUNWsprow / SC4.0 aus Solaris 2.4, Sun Microsystems Grasbrunn bei München
- [65] H. P. Christen
Grundlagen der organischen Chemie
Verlag Sauerländer, Aarau, Verlag Diesterweg-Salle, Frankfurt a. M., 2. Auflage 1972
- [66] E. Glaser
Pharmakokinetik, Grundlagen, lineare Methoden, Rechenverfahren, Auswertemethoden
pmi-Verlag, Frankfurt a. M., 1985
- [67] J. C. Fröhlich (Ed.)
Plasmaspiegel - Wirkungsbeziehungen von Pharmaka, Kapitel 3
Gustav Fischer Verlag, Stuttgart, 1986
- [68] H. Lüllmann, K. Mohr, A. Ziegler
Taschenatlas der Pharmakologie
Georg Thieme Verlag Stuttgart, 3. Auflage, 1996
- [69] R. Fick
Konzepte zur Syntheseplanung: Strukturelle Ähnlichkeit und strategische Bindungen
Dissertation, Friedrich-Alexander-Universität Erlangen-Nürnberg 1996
- [70] Joshel, Butz
J. Amer. Chem. Soc., **63**, 3350, 1941
- [71] Soncony-Vacuum Oil Co.
Patent US 2430137, 1946
- [72] Joshel
Patent US 2349232, 1941
- [73] Socony-Vacuum Oil Co.
Patent US 2430137, 1946
und Joshel
Patent US 2349232, 1941
- [74] B. A. Bunin, M. J. Plunkett, J. A. Ellman
Synthesis and Evaluation of Three 1,4-Benzodiazepine Libraries
in Combinatorial Peptide and Nonpeptide Libraries, A Handbook, G. Jung (Ed.)
VCH Verlagsgesellschaft, Weinheim, 1996
- [75] Chemwin, Soft Shell International, Grand Junction, CO, USA
- [76] W. Witznibichler
xmed++
Technische Universität München

- [77] FrameMaker, Frame Technology
- [78] W. D. Ihlenfeldt, Y. Takahashi, H. Abe, J. Gasteiger
Rapid and Portable Development of Visualisation Tools for Computational Chemistry Applications
J. Mol. Graphics
W. D. Ihlenfeldt, Das CACTVS-System
<http://schiele.organik.uni-erlangen.de/cactvs/>
- [79] W. Hanebeck
VEGAS
Technische Universität München
- [80] W. Witzenbichler
hpgl2mif
Technische Universität München
- [81] U. Hondelmann
YPLOT
Technische Universität München
- [82] The GNU Projekt for Free Software Foundation
<http://www.gnu.ai.mit.edu>
- [83] M. Wagener
treetool
Technische Universität München
- [84] Corel Corporation
<http://www.corel.com>
- [85] D. Koblas
xpaint
Extra Mile Consulting, Mountain View, Californien USA
- [86] X. Li, M. Wagener, J. Gasteiger
KMAP
Universität Erlangen-Nürnberg

Anhänge:

Anhang A. User-Manual zum EROS7-System	211
Anhang B. Manual zum Abfassen neuer Regeln für das EROS7-System	215
Anhang C. 103 Moleküle für die MS-Bewertung	251
Anhang D. Von FRANZ bewertete Reaktionen.....	253
Anhang E. Chlorierte Dioxine	267
Anhang F. Verwendete Regeldateien.....	271
Anhang G. Publikationen	297
Anhang H. Lebenslauf	299

A. User-Manual zum EROS7-System

A.1. Aufbau von EROS7

Das wissensbasierte Reaktionsvorhersagesystem EROS7 ist aus mehreren Teilen aufgebaut: der im Kern liegenden Repräsentation chemischer Strukturen, der Berechnung physikochemischer Parameter sowie der Reaktionsgenerierung, des Aufbaus des Reaktionsnetzwerks und der Wissensbasis, den Regeln. In den Regeln wird festgelegt, wie, wann und wo Reaktionen an den Ausgangsmaterialien bzw. den Reaktionsprodukten durchgeführt werden.

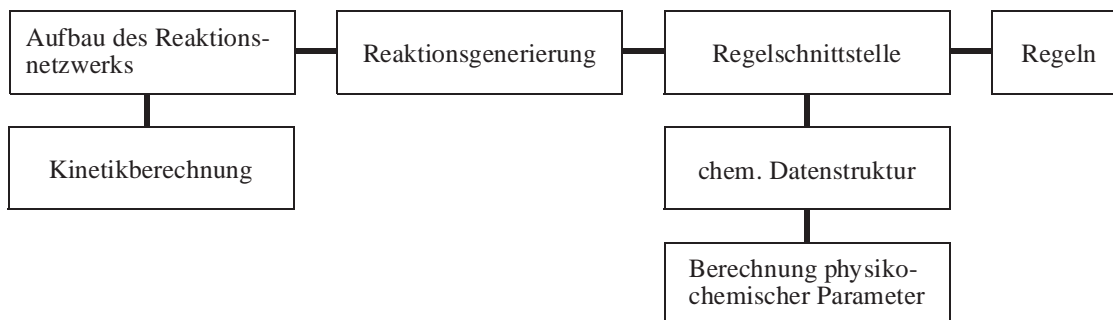


Abbildung 212: Aufbauschema von EROS7.

Das gesamte Reaktionsnetzwerk gliedert sich dabei in mehrere Bereiche, die Reaktoren. Für einen Reaktor werden zunächst alle Reaktionen erzeugt, sowie die kinetischen Daten berechnet. Das Ergebnis eines Reaktors kann dann als Ausgangsmaterial (inklusive der Konzentrationen und Stoffströme) für einen weiteren Reaktor verwendet werden. So ist es möglich, die Reaktionen in einem Rührkessel mit zwei Phasen, zum Beispiel Octanol und Wasser, zu simulieren und die anschließende Aufarbeitung der Produkte der Octanolphase in einem zweiten Reaktor vorherzusagen. Ergebnis der gesamten Vorhersage sind dann die aufgearbeiteten Produkte und deren Konzentrationen / Stoffströme.

Wie schon im gerade beschriebenen Beispiel erwähnt, gliedert sich ein Reaktor wiederum in eine oder mehrere Phasen. Die Reaktionen eines Reaktors werden ebenenweise durchgeführt, wobei für jede Reaktionsebene eines Reaktors je eine Ebene aller Phasen des Reaktors vorgenommen wird. Bei diesen Reaktionen kann es sich sowohl um Reaktionen innerhalb einer Phase, wie auch um Phasenübergänge und Phasentransferreaktionen handeln. Für die Konzentrationsberechnung der einzelnen Substanzen können verschiedene kinetische Modelle verwendet werden. Dieses Modell bezieht sich auf alle Reaktionen und Phasen eines Reaktors. Für die einzelnen Reaktionen selbst wird zunächst die Anwendbarkeit überprüft und nach der Reaktion abhängig von physikochemischen Parametern der Edukte und/oder der Produkte sowie Umgebungseinflüssen und der Entstehungsgeschichte der Produkte ein Reaktivitätswert berechnet. Diese Reaktivität wird dann in der kinetischen Evaluation verwendet.

A.2. Benutzerschnittstelle

Zur Benutzerschnittstelle zählen die Regeln (Wissensbasis) und die Aufrufparameter des Programms EROS7. Die meisten Einstellungen für EROS7 werden in den Regeln vorgenommen (siehe Anhang B). Dazu gehören die Reaktionstypen, die Phasen und Reaktoren sowie die Definition der Kinetik.

A.2.1. Die Regeln

Der größte Teil von EROS7 ist in C++ programmiert. Auch die Regeln können in C++ geschrie-

ben werden. Dabei ist der Aufbau der Regeln nicht, wie es für C++ üblich wäre, objektorientiert, sondern wie in der algorithmischen Programmiersprache C gegliedert. Dies hat den Vorteil, daß zum Abfassen einer Regel nur geringe C++-Kenntnisse benötigt werden und, was viel wichtiger ist, daß es möglich ist, für die Regelschnittstelle ein Interface zu einer anderen, auch nicht objektorientierten, Sprache zu schreiben. So besitzt EROS7 ein Tcl-Interface, was weitere Vorteile bietet. So besteht die Möglichkeit, die Regeln in der Entwicklungsphase über den dazugebundenen Tcl-Interpreter ausführen zu lassen. Man kann die Regeln schnell, ohne sie neu übersetzen und EROS7 neu linken zu müssen, abändern und sofort testen. Da es auch Tcl-Compiler gibt, die die Tcl-Regeln in C übersetzen, kann man die ausgereiften Regeln übersetzen und in EROS7 einbinden. Dies sollte einen Geschwindigkeitsvorteil bringen und kann praktisch sein, wenn man das Programm an Dritte weitergibt. Diese Lösung ist dennoch flexibel, da der Tcl-Interpreter auch dann Teil von EROS7 ist und jederzeit über eine Kommandozeilenoption auf die externen Tcl-Regeln umgeschaltet werden kann. So hat man ein System, das die Vorteile der einzelnen Methoden der Anbindung der Wissensbasis in sich vereinigt.

Für die Tcl-Regeln gibt es darüber hinaus noch zwei weitere Möglichkeiten. So können in einem Tcl-File mehrere Regelfunktionen verschiedenen Namens enthalten sein, wobei eine selektiert werden kann. Möchte man später auf eine andere Sprache für die Regeln - aus welchen Gründen auch immer - umsteigen, muß nur ein neues Interface zur C++-Regelschnittstelle geschrieben werden.

A.2.2. Die Aufrufparameter

Aber auch durch die Aufrufparameter können die Berechnungen von EROS7 beeinflußt werden. Alle Optionen von EROS7 haben die Form `-Option=Wert`. Besteht die Option nur aus einem einzigen Buchstaben, kann das `=`-Zeichen entfallen, wobei der Wert direkt an die Option angehängt wird. EROS7 kennt keine zwingende Option, meist werden jedoch die Optionen für die Selektion der Datei mit den Ausgangsverbindungen (`-i`), die zu verwendenden Regeln (`-r`), die Zahl der durchzuführenden Reaktionsebenen (`-levels`) sowie die Ausgabedateien für die Reaktionen (`-o`) und Strukturen (`-s`) angegeben. Die Option `-i` entfällt nur in dem seltenen Fall, daß EROS7 als Schnittstelle zur Datenstruktur RICOS für andere Programme verwendet wird. (Teile der Aufrufparameter in [] sind optional.)

A.2.2.1. Eingabedatei

Die Moleküle, mit denen EROS7 seine Reaktionen durchführt, werden aus einer Datei gelesen. Diese wird mit der Option `-i` angegeben. Soll zusätzlich das Format, in dem EROS7 die Eingabedatei erwartet, vom Standardwert (ctx-Datei, siehe unten) auf MO-orientierte Kodierung umgestellt werden, kann dies mit `-fr=mo` geschehen.

Option	Default-Wert	Werte
<code>-iinput_file</code>	keiner	Dateiname (<code>in</code>)
<code>-fr=format</code>	ctx	ctx bzw. mo

A.2.2.2. Regeln

Eine der wichtigsten Angaben ist die, welche Regeln verwendet werden sollen. Wird die Option `-r` nicht angegeben, werden die dazugebundenen, kompilierten C++-Regeln aufgerufen. Ist der Tcl-Interpreter zu EROS7 gebunden worden, können auch die kompilierten Tcl-Regeln (`compiled_tcl`) und Tcl-Regeln, die in einer Datei stehen und dann interpretiert werden, selektiert werden. Um die Regeln aus einer Tcl-Regeldatei zu laden, wird für die Regeln der entsprechende Dateiname angegeben. Optional kann bei den Tcl-Regeln nach dem Namen der Regeldatei bzw. `compiled_tcl` auch noch durch eine Doppelpunkt abgetrennt der Name der Regelprozedur angegeben werden (z.B. `f`, siehe unten). Standardmäßig heißt sie `Tcl_rule`.

Option	Default-Wert	Werte
<code>-r[debug:]rules[:f]</code>	<code>compiled_c++</code>	Regeldatei

Durch die Angabe `-rshow` gibt EROS7 die Namen aller zusätzlich eingebauten Regeln aus und beendet den Programmablauf ohne Reaktionen zu generieren. Wird bei Tcl-Regeln `debug:` vor den Namen der Regeln gestellt, wartet der Tcl-Interpreter nach jedem der unzähligen Aufrufe der Regelfunktion auf eine Benutzereingabe. Diese Möglichkeit ist zur Fehlersuche gedacht.

A.2.2.3. Steuerung der Reaktionsgenerierung

Die meist gebrauchte Option aus dem Bereich der Steuerung der Reaktionsgenerierung ist die Festlegung der maximal erzeugten Reaktionsebenen mit `-levels`.

Option	Default-Wert	Werte
<code>-levels=1</code>	1	1, 2, 3, ...
<code>-e2</code>	32767	1, 2, 3,
<code>-vpi:3.14,num#7</code>	keiner	siehe Text

Die beiden anderen hier aufgeführten Aufrufparameter sind für den versierten Benutzer, der zugleich sehr gute Kenntnisse über die verwendeten Regeln besitzt. Die Phasen der Regeln sind von 1 bis n durchnummeriert. Mit der Option `-e` kann die Nummer der Phase angegeben werden, die als letzte Reaktionen generiert. Angenommen in den Regeln seien fünf Phasen definiert, die alle Reaktionen erzeugen, könnten durch die Option `-e3` die Phasen vier und fünf von der Reaktionsgenerierung ausgenommen werden.

Mit der Option `-v` können, jeweils durch ein Komma getrennt, mehrere Variablen erzeugt werden, die an die Regeln übergeben werden. Diese können darauf reagieren, müssen aber nicht. Als erstes wird immer der Name, unter dem diese Variable an die Regeln übergeben werden soll, angegeben, der ein Separator folgt. Dieser trennt nicht nur den Wert vom Namen, sondern gibt auch an, um welchen Datentyp es sich handelt. `#` steht für *int*, `:` für *double* und `$` für *const char**.

A.2.2.4. Traceoutput

Normalerweise gibt EROS7 nur Warnungen und Fehlermeldungen aus. Leider halten sich nicht alle eingebundenen Modulpakete daran. Indem man schrittweise das Tracelevel (`-trace`) erhöht, erhält man zusätzliche Informationen über alle Bereiche von EROS7, inklusive der Regeln. Da so eine große Menge an Daten anfallen kann, ist es möglich diese in eine Datei umzulenken (`-tout`).

Option	Default-Wert	Werte
<code>-trace=1</code>	0	0, 1, 2, ...
<code>-tout=trace_file</code>	<code>stdout</code>	Dateiname (out)

A.2.2.5. Ausgabedateien

Eine Berechnung mit EROS7 ist wenig sinnvoll, wenn die Ergebnisse nicht ausgegeben werden. Da keine der Ausgabedateien standardmäßig geschrieben wird, ist darauf zu achten, daß eine der drei Optionen angegeben ist: `-o`, `-s` oder `-ms`. Da bei der Verwendung von EROS7 als Schnittstelle keine der Dateien geschrieben werden soll, ein vorgegebener Wert durch die Angabe der Option mit einem Wert zwar geändert, aber nicht gelöscht werden kann, wurde dies so gemacht.

Option	Default-Wert	Werte
-oreaction_file	keiner	Dateiname (out)
-sstruktur_file	keiner	Dateiname (out)
-sort=modus	p	p, c oder n
-p1	siehe Text	1, 2, 3, ...
-ms=spectra_file	keiner	Dateiname (out)
-fw=format	ctx	ctx bzw. mo

Die Option `-o` gibt den Namen der Datei an, in die die Reaktionen geschrieben werden. Sollen auch die Strukturen einer bestimmten Phase ausgegeben werden, kann dies mit `-s` aktiviert werden. Standardmäßig ist dies die in den Regeln angegebene `output_phase`. Möchte man jedoch die Aggregate einer anderen Phase, kann man dies durch `-p` selektieren. Die Reihenfolge, in der die Strukturen in die Datei geschrieben werden, richtet sich normalerweise nach der Persistenz, wobei mit dem größten Wert begonnen wird. Durch `-sort` kann dies auf fallende Konzentration (c) oder numerisch aufsteigend (n) geändert werden.

Die Option `-ms` initiiert, daß aus den Endkonzentrationen, Ladungen und Strukturen der Phase ein Massenspektrum berechnet wird, das zusammen mit den Strukturen der Ausgangsverbindungen in die angegebene Datei geschrieben wird.

Schließlich kann noch das Format, in dem die Strukturen in die verschiedenen Dateien geschrieben werden, mit `-fr` bestimmt werden.

A.3. Ein Beispielsaufruf von EROS7

Im folgenden Beispiel liest EROS7 die Ausgangsverbindungen aus der Datei `mols.ctx`, erzeugt maximal fünf Reaktionsebenen mit den Tcl-Regeln in der Datei `regeln.tcl` und schreibt die Reaktionen in die Datei `rxs.ctx`. Die Strukturen der in den Regeln angegebenen Ausgabephase werden in die Datei `str.ctx` ausgegeben. Alle Dateien haben das Format CTX.

```
eros7 -imols.ctx -rregeln.tcl -levels=5 -orxs.ctx -sstr.ctx
```

<code>mols.ctx</code>	Datei mit den Ausgangsverbindungen
<code>regeln.tcl</code>	Tcl-Regeldatei
<code>rxs.ctx</code>	Datei, in die die Reaktionen geschrieben werden
<code>str.ctx</code>	Datei, in die die Strukturen der Ausgabephase geschrieben werden

Dieses Beispiel stellt den Standardfall für den Aufruf von EROS7 dar.

B. Manual zum Abfassen neuer Regeln für das EROS7-System

B.1. Gliederung der Regeln

Die Regeln gliedern sich in mehrere Teile: einen globalen Teil und je einen Teil je Reaktionsregel. Die Nummern der Reaktionsregeln können zwischen 1 und 1000 beliebige Werte annehmen. Die Regelnummern müssen nicht zusammenhängend sein, müssen jedoch eindeutig sein. Der Rückgabewert ist vom Typ RULEOK und kann die beiden Werte OK und BAD annehmen. OK bedeutet, daß der durch `irule`, `op` und `sub_op` selektierte Regelteil erfolgreich durchgeführt wurde, wogegen BAD einen Fehler anzeigt. Ein Fehler in diesem Sinne kann sein, daß die Bedingungen für diese Reaktion nicht erfüllt sind oder tatsächlich ein Fehler aufgetreten ist. Der Regelfunktionsaufruf wird dann verworfen und es wird versucht, den Aufbau des Reaktionsnetzwerks fortzusetzen. Der Grundaufbau einer Regeldatei sieht folgendermaßen aus:

C++-Code:

```
#include "rx_eifc.h"
#include "rx_datex.h"
#include <iostream.h>
#include "rules.h"

RULEOK rules_cc (ostream &print, rx_eifc *eifc, rx_datex
*val, int irule, OP op, int sub_op, int test, int trace) {
// Deklaration der Variablen
    switch (irule) {
        case GLOBAL:
            // globale Regelinformationen
        case 1:
            // Regel 1
        case 7:
            // Regel 7
        default:
            return BAD;
    }
    return OK;
}
```

In Tcl sieht der entsprechende Code folgendermaßen aus:

Tcl-Code:

```
proc Tcl_rule {} {
    global_name
    # Deklaration der Variablen irule op sub_op test trace
    switch $irule {
        GLOBAL {
            # globale Regelinformationen
        }
        RULE_1 {
            # Regel 1
        }
    }
}
```

```

    RULE_7 {
        # Regel 7
    }
    default {
        return BAD
    }
}

```

Dabei ist zu beachten, daß in Tcl die Aufrufparameter `irule`, `op`, `sub_op`, `test` und `trace` mit der Prozedur `global_name` auf die entsprechenden globalen Tcl-Variablen gelinkt werden. Die Pointer auf die Objekte der Klassen `rx_datex` und `rx_eifc` sind in Tcl nicht sichtbar, werden aber von den definierten Funktionen intern verwendet. Für `rx_datex` sind dies `get`, `put`, `getco` und `putco`. Die Funktionen, die den `rx_eifc`-Pointer intern verwenden, sind hier wegen ihrer größeren Anzahl nicht aufgeführt. Für die `ostream`-Referenz `print` gibt es in Tcl ein zu `puts stdout` äquivalentes Kommando `print`, das auf den vom Kernsystem zur Verfügung gestellten Stream schreibt.

Eine mögliche Fehlerquelle in Tcl sind die Kommentare (`# ...`). Die Kommentare erfahren in Tcl keine Sonderbehandlung, sondern das `#`-Zeichen ist ein Befehl, der alle seine Argumente verwirft. Aus diesem Grunde dürfen Kommentare nicht an allen Stellen der Tcl-Prozedur stehen, sondern nur an Stellen, an denen auch andere Kommandos enthalten sein dürfen. Der nachfolgende Ausschnitt verdeutlicht dies.

```

proc Tcl_rule {} {
    global_name
    # ok
    switch $irule {
        GLOBAL {
            # ok
        }
        # Fehler
        RULE_1 {
            # ok
            set n 1 # Fehler
            # ok
        }
        default {
            return BAD
        }
    }
}

```

B.2. Globale Regelinformationen

In den globalen Regelinformationen werden folgende Dinge festgelegt:

- Name und Datum der Regeln
- der Variablen austausch zwischen dem Kernsystem und den Regeln
- die Zahl, der Zusammenhang und die Eigenschaften der Phasen und Reaktoren
- die Kinetik
- die Vorbehandlungsfunktionen für die Ausgangsmaterialien und die Edukte
- die Nachbehandlungs(Verteilungs)funktion für alle Produkte eines Eduktensembles
- die Modellfunktionen, die nicht zu den Regeln selbst gehören, aber vom Kernsystem aufge-

rufen werden können

- die Cleanup-Funktion, die vor Beendigung der Programms aufgerufen wird

In den Regeln sieht die Gliederung der globalen Regelinformationen wie folgt aus:

C++-Code:

```
switch (op) {
case INIT_RULES:
    // Name, Datum, Variablen, Phasen, Reaktoren, Kinetik
case PREP_ROOT:
    // Vorbehandlungsfunktion für die Ausgangsmaterialien
case PREP_EDUCT:
    // Vorbehandlungsfunktion für die Edukte
case DISTRIBFUNC:
    // Nachbehandlungs(Verteilungs)funktion für alle Produkte
case MODFUNC:
    // Modellfunktionen
case FINISH:
    // Cleanup
default:
    return BAD;
}
return OK;
```

Tcl-Code:

```
switch $op {
INIT_RULES {
    # Name, Datum, Variablen, Phasen, Reaktoren, Kinetik
}
PREP_ROOT {
    # Vorbehandlungsfunktion für die Ausgangsmaterialien
}
PREP_EDUCT {
    # Vorbehandlungsfunktion für die Edukte
}
DISTRIBFUNC {
    # Nachbehandlungs(Verteilungs)funktion für alle Produkte
}
MODFUNC {
    # Modellfunktionen
}
FINISH {
    # Cleanup
}
default {
    return BAD
}
}
```

B.2.1. Variablen

Der Aufrufparameter val der C++-Regelfunktion ist ein Zeiger auf ein Objekt der Klasse

`rx_datex`, über das der Austausch der Zeiger auf die Variablen erfolgt. Für die verschiedenen Variablentypen gibt es die Elementfunktionen `put` und `get` für Variablen sowie `putco` und `getco` für konstante Werte.

Die Deklaration der Variablen, die vom Kernsystem geholt werden bzw. an das übergeben werden, müssen in der ganzen Regelfunktion zugreifbar sein und dürfen nach dem Verlassen ihren Wert nicht verlieren. Variablen, die an das Kernsystem übergeben werden, wie zum Beispiel die Reaktivität (`double`), werden `static` deklariert und mit einem Anfangswert initialisiert. Für Variablen des Kernsystems, die gelesen werden, wie das Reaktionslevel (`int`), wird ein `static const`-Pointer deklariert.

In Tcl greifen die Funktionen `put`, `get` und `putco` intern auf den in Tcl nicht sichtbaren Pointer `val` zu.

C++-Code:

```
static const int *var_level;
static double var_reactivity = 0.;
```

Tcl-Code:

```
# Für Variablen vom Kernsystem ist in Tcl
# keine Deklaration nötig. Sie werden intern
# automatisch als global deklariert.
global var_reactivity
set var_reactivity 0.0
```

Die Übergabe der Adressen der Variablen erfolgt dann unter `INIT_RULES` mit:

C++-Code:

```
val->get("level", &var_level);
val->put("reactivity", &var_reactivity);
```

Tcl-Code:

```
get level var_level
put reactivity var_reactivity
```

Kann die Adresse der Variable "level" nicht geholt werden, wird in C++ `var_level` auf 0 gesetzt. Um in Tcl zu überprüfen, ob die Variable "level" geholt werden konnte, kann der Rückgabewert des Kommandos `get` überprüft werden:

Tcl-Code:

```
set flag [get level var_level]
```

Hat die Variable `flag` den Wert 1, konnte die Variable `var_level` abgeholt werden. Im anderen Fall hat `flag` den Wert 0. In Tcl besteht außerdem die Möglichkeit, bei `put` und `putco` den Namen der Variablen wegzulassen. Er ist dann identisch mit dem Namen, unter dem die Variable übergeben wird.

Die Variablen können dann in beliebigen Regelteilen folgendermaßen gelesen bzw. gesetzt werden:

C++-Code:

```
int i = *var_level;
var_reactivity = 1.4e-8;
```

Tcl-Code:

```
set i var_level
set var_reactivity 1.4e-8
```

Immer, wenn man `*var_level` verwendet, erhält man das aktuell gültige Level. Für die Re-

aktivität holt sich das Kernsystem nach jeder erfolgreichen Reaktion den aktuellen Wert. Dies gilt im übrigen auch für alle anderen an das Kernsystem übergebenen Variablen, die vom Typ `int` oder `double` sind. Eine Reaktion ist beendet, wenn der entsprechende Regelteil verlassen wird oder wenn eine der beiden Funktionen `save_active_ens_as_product` bzw. `save_ens_as_product` aufgerufen wird. Die Werte der Variablen werden im Reaktionsfile unter `/REACLOG` mit ausgegeben. Wenn die Kinetik angeschaltet ist, wird die Variable `re-activity` vom Typ `double` benötigt. Sie gibt die Geschwindigkeitskonstante der Reaktion an bzw. die Wahrscheinlichkeit beim Modus `prob_kin` der Kinetik.

Als Variablen vom Kernsystem stehen zur Verfügung:

`level`, `phase` und `reactor` (alle vom Typ `int`), sowie zu Testzwecken `temperature`, `pressure` (beide `double`), `hihi` (`int`), `program_name` (Konstante `const char*`), `string_array` (Konstante `const char *const *`), `variable_string` (`const char*`) und `array_probe` (`vector<int>`).

B.2.2. Name und Datum

Wie unter dem Punkt Variablen schon angesprochen, können mit der Klasse `rx_datex` auch konstante Werte übergeben werden. Dies wird beispielsweise für den Namen und das Datum der Regeln verwendet.

C++-Code:

```
val->putco("name", "Regelname");
val->putco("date", "13.11.1996");
```

Tcl-Code:

```
set name_var "Regelname"
putco name name_var
set date_var 13.11.1996
putco date date_var
```

Vom Kernsystem wird der Regelname dann folgendermaßen abgeholt:

```
const char *name;
var->getco("name", &name);
cout << "The name of the rules is " << name << endl;
```

Vergleicht man den C++-Code für die Übergabe von Konstanten mit dem der Variablen, so erkennt man, daß bei den Konstanten im Aufruf von `putco` die Variable selbst steht und bei `getco` der Zeiger auf die Variable und nicht, wie bei der Übergabe der Variablen in den Funktionen `put` und `get`, der Zeiger auf die Variable bzw. der Zeiger auf den Zeiger. Die Referenzierung der Konstanten ist gegenüber den Variablen bei der Übergabe also um eins erniedrigt.

B.2.3. Reaktoren und Phasen

Auch die Einstellungen der Reaktoren und Phasen werden im Teil `INIT_RULES` der Regeln festgelegt. Zunächst müssen die Anzahl der Reaktoren und Phasen sowie die Zugehörigkeit der Phasen zu den Reaktoren angegeben werden. Dies geschieht mit der Konstante `"phases_per_reactor"`, die von Typ `const int[]` ist. Jede Zahl in diesem `int`-Array steht dabei für die Anzahl der Phasen im Reaktor. Das `int`-Array wird mit 0 abgeschlossen. Im folgenden Beispiel werden 2 Reaktoren mit zwei Phasen bzw. einer Phase definiert. Der Reaktor 1 enthält die Phasen 1 und 2 und der Reaktor 2 die Phase 3.

C++-Code:

```
{
static const int ppr[] = { 2, 1, 0 };
```

```

val->putco("phases_per_reactor", ppr);
}

```

Tcl-Code:

```

global ppr
set ppr {2,1,0}
putco phases_per_reactor ppr

```

Als nächstes muß festgelegt werden, welche Verbindungen als Ausgangsmaterialien für die einzelnen Reaktoren dienen. Dabei ist zu beachten, daß die Startmaterialien für jeden Reaktor in die erste Phase dieses Reaktors gegeben werden. Im obigen Beispiel also in Phase 1 für den Reaktor 1 und in Phase 3 für den Reaktor 2. Für alle Reaktoren wird angegeben, wo die Startmaterialien herkommen. Dies sind grundsätzlich alle Verbindungen einer bestimmten Phase, die angegeben wird. Dementsprechend wird für jeden Reaktor die Nummer der Phase mit den Ausgangsmaterialien angegeben, wobei für den ersten Reaktor noch keine Strukturen aus einer anderen Phase zur Verfügung stehen. Für ihn wird die Phase 0 angegeben. Sollen nun im obigen Beispiel die Produkte aus Phase 2 in den Reaktor 2 übernommen werden, ist dies folgendermaßen zu kodieren:

C++-Code:

```

{
static const int inp_phase[] = { 0, 2 };
val->putco("input_phase_for_reactors", inp_phase);
}

```

Tcl-Code:

```

global inp_phase
set inp_phase {0,2}
putco input_phase_for_reactors inp_phase

```

Normalerweise werden alle vom File eingelesenen Ausgangsmaterialien der Phase 1 zugeteilt, es besteht aber auch die Möglichkeit, sie in beliebige andere Phasen zu geben. Dazu kann im CTX-File für jedes Molekül die Eigenschaft M_PHASE angegeben werden, die dann die Nummer der Phase enthält. Beim Einfügen der Aggregate wird diese in die Eigenschaft AG_PHASE kopiert, die dann ausgewertet wird. Enthält das Eingabefile keine Phasenangabe, wird noch nachgesehen, ob in INIT_RULES eine Variable des Typs `vector<int>` unter dem Namen `start_phase` übergeben wurde. Ist dies der Fall, gibt `sphs[0]` die Phase für das erste Aggregat an. Kann keine Angabe über die Anfangsphase gefunden werden, kommt das Aggregat in die Phase 1. Im folgenden Beispiel wird angegeben, daß das Aggregat 1 in die Phase 3 kommt:

C++-Code:

```

static vector<int> sphs(1);
sphs[0] = 3;
val->put("start_phase", &sphs);

```

Tcl-Code:

```

global sphs
set sphs {3}
put start_phase sphs

```

Dabei ist es nur wichtig, daß in INIT_RULES die Variable für die Einträge der Phasen für die Ausgangsmaterialien übergeben wird. Die Einträge können auch erst während der Vorbehandlung der Ausgangsmaterialien gesetzt werden. Die Vorbehandlung kann abhängig von der Zahl

der Eduktensembles mehrfach aufgerufen werden, wobei nach jedem Aufruf die Phasen für alle gerade vorbehandelten Aggregate gesetzt sein müssen, da sie direkt danach von EROS7 gelesen werden. Alle Aggregate werden von null ab mit jedem Aufruf von PREP_ROOT durchnummeriert, wobei für jedes Aggregat im Ensemble hochgezählt wird. Ist ein Aggregat ein zweites Mal enthalten, wird nicht hochgezählt.

Desweiteren kann für jeden Reaktor noch festgelegt werden, ob die Edukte ein Ausgangsgemisch darstellen oder ob sie getrennt voneinander behandelt werden sollen. Dafür gibt es für jeden Reaktor ein Flag (1=Gemisch, 0=jede Verbindung einzeln). Sollen zunächst im ersten Reaktor alle Ausgangsmaterialien miteinander reagieren, werden aber danach chromatografisch getrennt und getrennt voneinander, aber gleich, aufgearbeitet, ergibt sich folgender Code:

C++-Code:

```
{
static const int all_educts[] = { 1, 0 };
val->putco("use_all_educts_together", all_educts);
}
```

Tcl-Code:

```
global all_educts
set all_educts {1,0}
putco use_all_educts_together all_educts
```

Jetzt fehlt nur noch, daß festgelegt wird, welche Verbindungen im Strukturfile ausgegeben werden. Um die Verbindungen der Phase 3 auszugeben, schreibt man:

C++-Code:

```
val->putco("output_phase", 3);
```

Tcl-Code:

```
global ophase
set ophase 3
putco output_phase ophase
```

Die Angabe der Ausgabephase kann jedoch beim Aufruf von EROS7 mit der Option -p überschrieben werden.

Bei der Kodierung von "input_phase_for_reactors" und "use_all_educts_together" ist darauf zu achten, daß für alle Reaktoren ein entsprechender Wert angegeben ist, da intern die Größe des allocierten int-Arrays nicht abgeprüft werden kann.

B.2.4. Phaseneigenschaften

Für die definierten Phasen wird noch der Modus festgelegt, der bestimmt, wie die Phase ihre Edukte für Reaktionen zusammenstellt. Folgende Modi stehen zur Verfügung:

- INERT
- MONOMOLEC
- MIX
- MIX_NO_A_A
- SURFACE
- TUBE

Dies sind const ints und die Modi der Phasen 1, 2, und 3 werden zum Beispiel folgendermaßen für obige Reaktoren gesetzt:

C++-Code:

```
{
static const int phprop[]={MIX, MIX, MONOMOLEC};
val->putco("phase_property", phprop);
}
```

In Tcl sind die oben angegebenen Modi als interne Variablen definiert, die nur gelesen werden können.

Tcl-Code:

```
global phprop
set phprop "$MIX,$MIX,$MONOMOLEC"
putco phase_property phprop
```

B.2.4.1. INERT

Eine Phase, die mit INERT gekennzeichnet ist, generiert selbst keine Reaktionen, dient allerdings als Aggregatspeicher für andere Phasen, die den Modus SURFACE oder auch MIX oder MIX_NO_A_A hat. Funktionsweise siehe Modus SURFACE.

B.2.4.2. MONOMOLEC

Der Begriff MONOMOLEC ist von monomolekular abgeleitet. In diesem Modus erzeugt die Phase Reaktionen, bei denen ein einziges Aggregat als Edukt verwendet wird. Die Phase erzeugt so Reaktionen erster und pseudoerster Ordnung.

B.2.4.3. MIX (Modus von EROS6) und MIX_NO_A_A

Ist die Phase im Modus MIX, produziert sie alle Reaktionen ausgehend von einem Aggregat und alle Reaktionen, die von allen möglichen Paaren aller Aggregate dieser Phase ausgehen. Im Modus MIX werden dabei auch die Reaktionen erzeugt, die zweimal das Aggregat X als Edukte haben. Im Modus MIX_NO_A_A werden diese Reaktionen nicht erzeugt. Zusätzlich werden noch alle Reaktionen wie im Modus SURFACE durchgeführt.

B.2.4.4. SURFACE

Im Modus SURFACE erzeugt die Phase alle Reaktionen, die von zwei Aggregaten ausgehen, wobei eines der Aggregate aus der Phase selbst stammt, wohingegen das andere aus einer angrenzenden Phase kommt. Dazu muß festgelegt werden, welche Phasen benachbart sind (oder in Kontakt stehen). Die Nachbarschaften werden als 0-terminiertes Array aus Zahlenpaaren angegeben. Zum Beispiel: 1, 2, 2, 3, 0. Dies bedeutet, daß Phase 1 mit Phase 2 in Kontakt steht und Phase 2 mit Phase 3. Die andere Phase ist häufig im Modus INERT und speichert nur eine bestimmte Gruppe von Verbindungen, wie für dem Einsatz zur kombinatorischen Chemie.

C++-Code:

```
{
static const int phct[]={ 1, 2, 2, 3, 0 };
val->putco("phase_contacts",phct);
}
```

Tcl-Code:

```
global phct
set phct {1, 2, 2, 3, 0}
putco phase_contacts phct
```

Um Reaktionen nicht doppelt zu erzeugen, ist bei der Angabe der Phasenkontakte die Reihen-

folge von Bedeutung. 1, 2 ist also nicht gleich 2, 1. Im ersten Fall werden die Reaktionen von Phase 1 und im zweiten von Phase 2 erzeugt. Dies macht keinen Unterschied, solange weder in Phase 1 noch in Phase 2 während der Reaktionsgenerierung neue Aggregate in einer der Phasen gebildet werden. Dies ist allerdings oft nicht der Fall, weshalb die Reihenfolge von Bedeutung ist.

B.2.4.5. TUBE (Modus von EROS5)

In diesem Modus werden die Aggregaten des Ausgangsensemles zusammengehalten und auch die Folgereaktionen nur mit den nach den vorher abgelaufenen Reaktionen vorliegenden Substanzen durchgeführt. Ist also im Ausgangsensemble ein Wassermolekül und wurde dies in einem ersten Hydrolyseschritt verbraucht, steht Wasser in den Folgereaktionen nicht mehr zur Verfügung, wohl aber die Reaktionsprodukte.



Im Modus Tube ist keine Kinetik möglich.

B.2.5. Kinetik

Abschließend zu den Einstellungen der Phasen und Reaktoren muß noch das Kinetikmodell für jeden Reaktor gesetzt werden. Dabei stehen folgende Methoden zur Verfügung:

- none
- gear
- runge_kutta
- runge_kutta_merson
- prob_kin

Mit `none` wird die Kinetik abgeschaltet. `gear`, `runge_kutta` und `runge_kutta_merson` sind Methoden zur Lösung der Differentialgleichungen der Reaktionsgeschwindigkeiten. `prob_kin` berechnet die Substanzkonzentrationen aufgrund von Reaktionswahrscheinlichkeiten.

Die Methoden `runge_kutta` und `runge_kutta_merson` sind Methoden vom Typ Runge-Kutta vierter Ordnung mit automatisch bestimmten Integrationsschritten und zählen zu den klassischen numerischen Methoden zur Lösung von Differentialgleichungen. Sie besitzen eine höhere Lösungsgenauigkeit als z.B. die Methoden von Euler, auf denen grundsätzlich der Gear-Algorithmus basiert. Der Gear-Algorithmus wurde speziell für die Lösung von Differentialgleichungen physikalischer Prozesse geschrieben und enthält sogenannte „multiple time scales“, einen Mechanismus zur Auswahl des Integrationsschrittes durch eine komplizierte Analyse des Verhaltens der Differentialgleichungen. Wie alle „Implicit Integration Schemes“, zu denen der Gear-Algorithmus zählt, kann er „numerical damping“- und Resonanzprobleme haben, die besonders bei einem großen System aus Differentialgleichungen zu Tage treten. Die Methoden `runge_kutta` und `runge_kutta_merson` sind überdies nicht, wie `gear`, auf Differentialgleichungen erster Ordnung beschränkt. Es treten allerdings zur Zeit in EROS7 nur Differentialgleichungen erster Ordnung auf.

Für die Wahrscheinlichkeitskinetik ist zu beachten, daß sie nur für monomolekulare Reaktionen geeignet ist. Hat eine der Reaktionen im Kinetikmodus `prob_kin` das Attribut `rearrangement`, gibt die Variable `reactivity` den Prozentsatz an, der reagiert. Mehr dazu steht in Kapitel 6.9.2.

Die Werte der double-Variablen `reactivity` sind die Geschwindigkeitskonstanten bzw. die Wahrscheinlichkeit (`prob_kin`) der Reaktionen. Die Konzentrationen der Ausgangsmaterialien werden bevorzugt aus dem Struktureingabefile gelesen (`AG_CONCENTRATION` bzw. `M_CONCENTRATION`). Sind dort keine Konzentrationen angegeben, kann man sie, wie unten für einen einzigen Ausgangsstoff gezeigt, auch in der Regel setzen. Die tatsächliche Zuweisung

der Konzentration kann auch erst während der Vorbehandlung der Ausgangsmaterialien geschehen (siehe B.2.3). Werden auch hier keine Konzentrationen angegeben (kein `val->put("start_conc", ...)`), werden die Konzentrationen der Ausgangsverbindungen auf 0.1 mol/l gesetzt. Gleiches gilt für die kontinuierlichen Zuflüsse der Ausgangsmaterialien. Hier sind es `AG_FLOW` bzw. `M_FLOW` oder gegebenenfalls der `vector<double>` unter dem Namen `start_flow`. Wird so kein Zufluß der Ausgangsmaterialien gefunden, wird er auf 0.0 mol/s (keine Zufluß) gesetzt.

Mit `minimal_concentration` wird diejenige Konzentration angegeben, die ein Reaktionsprodukt mindestens haben muß, damit ausgehend von ihm noch weitere Reaktionen erzeugt werden. Da zu diesem Zeitpunkt noch keine Reaktionen ausgehend von diesem Produkt erzeugt wurden, ist die Konzentration, die zum Vergleich herangezogen wird, diejenige ohne Berücksichtigung etwaiger Abbaureaktionen. Die beiden Werte `conversion_limit` und `reaction_time` bestimmen die Reaktionszeit. `reaction_time` ist die maximale Reaktionszeit in Sekunden, die allerdings nicht erreicht wird, wenn vorher mindestens 50% (`conversion_limit = 0.5`) der Ausgangssubstanzen umgesetzt sind.

Arbeitet man mit mehreren Phasen, ist es unter Umständen noch interessant, die Volumina der einzelnen Phasen zu setzen. Werden keine Volumina gesetzt, werden sie intern all mit 1.0 l angenommen. Außerdem kann man noch die Flüsse der Phasen und wohin sie fließen angeben. Der `vector<double>` `phase_flow` gibt die Flüsse der Phasen in l/s (default 0.0) und der `vector<int>` `phase_flow_to` die Phase im gleichen Reaktor, in die die Flüsse gehen, an (0 bedeutet: der Fluß wird nicht von einer anderen Phase aufgenommen). Dabei ist es unerheblich, ob beide Phasen den gleichen Fluß aufweisen, da der Stoffstrom, der aus der einen Phase herauskommt, in die neue Phase hineingeht. Es wird angenommen, daß genau soviel Lösungsmittel zugesetzt bzw. entzogen wird, daß der Volumenstrom der einen Phase in den Volumenstrom der anderen Phase übergeht.

C++-Code:

```
{
static vector<double> sconc(1), vols(3), flow(3);
static vector<int> to_phase(1), flow_to(3);
sconc[0] = 0.10;
to_pahse[0] = 1; // Aggregat 1 kommt in Phase 1 (default)
vols[0] = 1.0; // Volumen für Phase 1
vols[1] = 1.0; // Volumen für Phase 2
vols[2] = 1.0; // Volumen für Phase 3
flow[0] = 0.;
flow[1] = 0.;
flow[2] = 0.;
flow_to[0] = 0;
flow_to[1] = 0;
flow_to[2] = 0;
static const char *const kin_mode[]={ "gear", "gear" };
val->putco("kinetic_model", kin_mode);
val->putco("conversion_limit", 0.5);
val->putco("reaction_time", 5.e7);
val->putco("minimal_concentration", 1.e-10);
val->put("start_conc", &sconc);
val->put("start_phase", &to_pahse);
val->put("phase_volume", &vols);
val->put("phase_flow", &flow);
```

```

val->put("pahse_flow_to", &flow_to):
}

```

Tcl-Code:

```

global kin_mode min_conc conv_lim reac_time sconc vols
global to_phase
set sconc(0) 0.1
set to_phase(0) 1 # Aggregat 1 kommt in Phase 1 (default)
set vols {1.0,1.0,1.0}
set flow {0.,0.,0.}
set flow_to {0,0,0}
set kin_mode {"gear","gear"}
set min_conc 1.e-10
set conv_lim 0.5
set reac_time 5.e7
putco kinetic_model kin_mode
putco conversion_limit conv_lim
putco reaction_time reac_time
putco minimal_concentration min_conc
put start_conc sconc
put start_phase to_phase
put phase_volume vols
put phase_flow flow
put phase_flow_to flow_to

```

Hat man einen der Kinetikmodi gear, runge_kutta oder runge_kutta_merson gewählt, ist es auch möglich, für einzelne Reaktionstypen eine Kinetik nach Michaelis-Menten oder nullter Ordnung festzulegen. Diese Arten der Kinetik treten bei enzymatischen Reaktionen auf. Bei ihnen ist eine Gleichgewichtsreaktion zur Bildung des Enzym-Substrat-Komplexes der enzymkatalysierten Reaktion vorgelagert (siehe Abbildung 213).

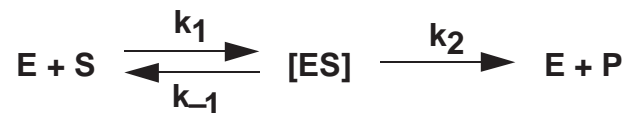


Abbildung 213: Enzymreaktion.

Diese Einzelreaktionen können von EROS7 so erzeugt werden, es setzt allerdings voraus, daß das Enzym (bzw. eine vereinfachte Form) als Reaktand eingesetzt wird. Da dies oft nicht gewünscht wird, wurde für EROS7 auch die Kinetik nach Michaelis-Menten eingeführt (siehe Formel 17). Es ergibt sich eine vereinfachte Reaktion, wie sie in Abbildung 214 gezeigt ist.



Abbildung 214: Vereinfachte Enzymreaktion.

$$\frac{dP}{dt} = \frac{k_2 \cdot [E]_0 \cdot [S]}{K_m + [S]} \quad \text{mit} \quad K_m = \frac{k_2 + k_{-1}}{k_1} \quad (17)$$

Sind solche Reaktionstypen in den Regeln enthalten, übergibt man zusätzlich zur reactivity-Variablen auch eine double-Variable unter dem Namen Km, in der für diese Reaktionstypen der Wert der Michaelis-Konstante übergeben wird. Der reactivity-Varia-

ble wird bei diesen Reaktionstypen das Produkt aus k_2 und der Enzymkonzentration $[E]_0$ zugewiesen. Der Reaktionstyp erhält das Attribut `michaelis_menten_kinetics` (siehe B.3.1). Befindet sich die Enzymreaktion ständig im Sättigungsbereich, kann man durch die Angabe des Attributs `force_zero_order` für den Reaktionstyp eine Kinetik nullter Ordnung erzwingen.

Ist der Gear-Algorithmus als Kinetik gewählt, kann mit EROS7 auch eine Mehrfachdosierung simuliert werden. Hierzu übergibt man eine `const double` unter dem Namen `multi_dose` mit der Zeit zwischen den Dosierungen. EROS7 erhöht dann die Konzentration der Ausgangsverbindungen in gleichen Zeitintervallen um die Anfangskonzentrationen. Soll mehrfach dosiert werden, aber nicht in gleichen Zeitabständen, kann unter dem Namen `multi_dose` auch ein `vector<double>` mit den Zeiten der Dosierung übergeben werden. Auch in diesem Fall wird mit der Anfangskonzentration der Ausgangsverbindungen begonnen. Wird als erste Zeit 0.0 Sekunden angegeben, kann jedoch auch mit der doppelten Konzentration angefangen werden.

B.2.6. Weitere globale Einstellungen

Zu den weiteren globalen Einstellungen gehören die Festlegung, was im Falle eines internen Fehlers zu geschehen hat. Fast alle Zugriffsroutinen auf das Ensemble geben ein Fehlerflag zurück. Da aber in aller Regel diese nicht überprüft werden, ist die Voreinstellung so, daß alle Reaktionen, bei denen ein Fehler aufgetreten ist, verworfen werden. Dies geschieht auch dann, wenn der entsprechende Regelteil `FUNC` mit `OK` zurückkehrt. Schreibt man Regeln, bei denen alle Fehlerflags überprüft werden und der Regelteil davon abhängig weitergeführt wird, kann man die interne Löschung von Reaktionen mit Fehlern folgendermaßen im Regelteil `INIT_RULES` abschalten:

C++-Code:

```
val->putco("no_discard_on_error", 1);
```

Tcl-Code:

```
global nodisonerr
set nodisonerr 1
putco no_discard_on_error nodissonerr
```

Analog der internen Fehlerüberprüfung findet auch eine Überprüfung der Korrektheit der Produkte statt. Der Modus für diese Valenzüberprüfung kann gesetzt werden mit:

C++-Code:

```
val->putco("valence_mode", "organic");
```

Tcl-Code:

```
global vmode
set vmode organic
putco valence_mode vmode
```

Es stehen die Möglichkeiten `organic`, `inorganic`, `ms` und `ms-inorganic` zur Verfügung. Hat der Modus den Wert `none`, ist die Valenzüberprüfung abgeschaltet.

B.2.7. Vorbehandlungsfunktionen für die Ausgangsmaterialien (PREP_ROOT)

Diese Funktion wird für jedes eingelesene Eduktensemble einmal aufgerufen. Hier können Vorbehandlungen der Ausgangsmaterialien, wie das Markieren saurer Wasserstoffatome oder funktioneller Gruppen vorgenommen werden. Es ist aber auch möglich, hier die Phasen,

Konzentrationen und Zuflüsse für die Ausgangsverbindungen festzulegen (siehe B.2.3). In Bezug auf Modifikationen am Ensemble entspricht der Umfang der Möglichkeiten denen in FUNC der einzelnen Regeln (siehe unten). Im Unterschied von FUNC findet keinerlei Variablenaus-tausch statt und das vorbehandelte Ausgangsensemble ist dasjenige, das nach ein oder mehreren Reaktionen herausgekommen ist. In aller Regel ist allerdings keine Vorbehandlung nötig und es genügt folgende Zeile in dieser Funktion:

C++-Code:

```
return OK;
```

Tcl-Code:

```
return OK
```

B.2.8. Vorbehandlungsfunktionen für die Edukte (PREP_EDUCT)

Wie für die Ausgangsmaterialien kann auch für jedes Eduktensemble einer Reaktion eine Vorbehandlungsfunktion geschrieben werden. Auch sie hat den gleichen Funktionsumfang und das letzte Eduktensemble in der Reaktionsfolge ist das vorbehandelte. Eine Rolle spielt diese Funktion vorwiegend im Zusammenhang mit der Verteilungsfunktion oder zum Ausschluß bestimmter Verbindungen von der Weiterreaktion. In aller Regel kann aber auch diese entfallen:

C++-Code:

```
return OK;
```

Tcl-Code:

```
return OK
```

B.2.9. Die Verteilungsfunktion (DISTRIBFUNC)

Diese Funktion wird immer dann aufgerufen, wenn ausgehend von einem Eduktensemble alle Reaktionen durchgeführt wurden. Es besteht hier die Möglichkeit, zu entscheiden, ob einzelne Reaktionen verworfen werden sollen. Bei symmetrischen Reaktionen an unsymmetrischen Molekülen kann es nicht ganz ausgeschlossen werden, daß eine Reaktion mehrfach erzeugt wird. Die weiteren Reaktionen können verworfen werden, aber auch mit der ersten zusammengefaßt werden, indem man den Symmetriefaktor der ersten Reaktion auf die Summe aller Symmetriefaktoren setzt. Desweiteren können hier noch weitere Werte einer Reaktion berechnet werden. Ein gewichteter Reaktivitätswert bezüglich der Konkurrenzreaktionen wäre hier denkbar. Der folgende Code zeigt, was intern ausgeführt wird, wenn die Verteilungsfunktion lediglich aus dem Befehl `return OK;` besteht:

C++-Code:

```
{
print << "DISTRIBFUNC called.\n";
int    n_rxs=-1, irx, cur_rx;
val->getco("#rxs", &n_rxs);
if (n_rxs < 0) return BAD;
static vector<int> ok_var;
ok_var.resize(n_rxs);
const vector<double> *vprob;
val->get("reactivity", &vprob);
const vector<int> *eqv_rxn, *ophs;
```

```

val->get("eqv_rxs", &eqv_rxn);
val->get("rxn_phase", &ophs);
int cur_rx = 2, iphs = (*ophs)[0], cphs, irx2, cphs2;
for (irx=1; irx<=n_rxs; irx++) {
    if ((*eqv_rxn)[irx] == cur_rx) {
        cur_rx++;
        ok_var[irx-1] = 1;
    } else {
        ok_var[irx-1] = 0;
    }
    cphs = (*ophs)[irx];
    if (((*eqv_rxn)[irx]==1)&&(iphs!=cphs)) {
        bool take = true;
        for (irx2=1; irx2<irx; irx2++) {
            cphs2 = (*ophs)[irx2];
            if (((*eqv_rxn)[irx2]==1)&&(cphs==cphs2)) {
                take = false;
                break;
            }
        }
        if (take) ok_var[irx-1] = 1;
    }
    print << "eqv product " << (*eqv_rxn)[irx] << " ";
    if (ok_var[irx-1] == 1) {
        print << "accepting reaction ";
    } else {
        print << "rejecting reaction ";
    }
    print << irx << " with a probability of ";
    print << (*vprob)[irx-1] << endl;
}
val->put("rx_ok", &ok_var);
return OK;
}

```

Tcl-Code:

```

print "DISTRIBFUNC called.\n"
get "#rxs" n_rxns
if {$n_rxns < 0} {return BAD}
global ok_var
resize ok_var $n_rxns
get reactivity vprob
get eqv_rxs eqv_rxn
get rxn_phase ophs
set cur_rx 2
set iphs $ophs(0)
for {set irx 1} {$irx<=$n_rxns} {incr irx} {
    set idx [expr $irx-1]
    if {$eqv_rxn($irx) == $cur_rx} {
        incr cur_rx
    }
}

```

```

        set ok_var($idx) 1
    } else {
        set ok_var($idx) 0
    }
    set cphs $ophs($irx)
    if {{ $eqv_rxn($irx)==1 }&&{ $iphs!=$cphs }} {
        set take 1
        for {set irx2 1} {$irx2<$irx} {incr irx2} {
            set cphs2 $ophs($irx2)
            if {{ $eqv_rxn($irx2)==1 }&&{ $cphs==$cphs2 }} {
                set take 0
                break
            }
        }
        if {$take} {set ok_var($idx) 1}
    }
    print "eqv product $eqv_rxn($irx)    "
    if {$ok_var($idx) == 1} {
        print "accepting reaction"
    } else {
        print "rejecting reaction"
    }
    print " $irx with a probability of $vprob($idx)\n"
}
put rx_ok ok_var
return OK

```

Die Variable #rxs gibt die Zahl der erfolgreich durchgeführten Reaktionen an und gibt damit die Dimension der Vektoren der anderen Variablen an. eqv_rxs ist die Äquivalenzklasse des Produkts, wobei 1 das Edukt selbst ist. Die Äquivalenzklasse des Produkts der ersten Reaktion bzw. die Phase für die Produkte (rxn_phase) erhält man mit dem Index 1; der Index 0 ist für das Edukt. Das Ergebnis der Verteilungsfunktion ist die Variable rx_ok, die mit 1 angibt, daß die Reaktion behalten wird bzw. mit 0, daß die Reaktion verworfen wird. Neben den Variablen #rxs, eqv_rxn, rxn_phase, rx_symmetry, rx_rule_nr und reactivity stehen in dieser Funktion auch alle Variablen der Reaktion zur Verfügung, die wie der Reaktivitätswert EROS7 bekanntgegeben wurden und vom Typ int oder double sind. Sie kommen dann hier als vector<int> bzw. vector<double> für alle Reaktionen eines Edukts an. Für die Reaktivität und alle weiteren während der Reaktion berechneten Variablen gilt Index 0 für die erste Reaktion. Nur für eqv_rxn, rxn_phase und rx_ok hat der Eintrag für die Reaktion 1 den Index 1.

Liefert die DISTRIBFUNC keinen vector<int> unter dem Namen rx_ok, wird der oben angegebene in EROS7 eingebaute Algorithmus verwendet, der alle Reaktionen verwirft, die identisch sind und deren Produkte in die gleiche Phase gehen (siehe Beispiel oben). Behalten wird nur die erste Reaktion. Daneben werden auch noch die invarianten Reaktionen verworfen, bei denen Edukt und Produkt identisch sind und der gleichen Phase angehören. Soll dies genutzt werden, sieht der Regelteil folgendermaßen aus:

C++-Code:

```
return OK;
```

Tcl-Code:

```
return OK
```

Kehrt die Funktion mit BAD zurück, werden alle Reaktionen dieses Edukts verworfen. Will man hier die Reaktivität und / oder die Symmetriezahl einer Reaktion anpassen, so kann man dies tun, indem man die einen `vector<double>` als `reactivity` und einen `vector<int>` als `symmetry` übergibt. Die Reaktivität muß in diesem Fall von den Reaktionen in eine andere Variable geschrieben werden. Der von EROS7 berechnete Symmetriefaktor wird als `vector<int>` unter dem Namen `rx_symmetry` bereitgestellt. Es können hier aber auch weitere Werte für die Reaktionen berechnet werden. Werden sie an EROS7 zurückgegeben, werden diese auch bei den Reaktionen abgespeichert und im Reaktionsfile ausgegeben.

B.2.10. Die Modellfunktionen (MODFUNC)

In diesem Regelteil ist es möglich, dem Programm vom Benutzer definierte Berechnungsfunktionen zur Verfügung zu stellen. Welche Funktionen verwendet werden können und Daten vom Programm geliefert bzw. erwartet werden, hängt ganz vom Programm selbst ab. EROS7 verwendet derzeit keine dieser Funktionen. Der Code, der besagt, daß keine Modellfunktion vorhanden ist, sieht folgendermaßen aus:

C++-Code:

```
return BAD;
```

Tcl-Code:

```
return BAD
```

Wenn sie implementiert werden soll, sieht sie etwa folgendermaßen aus:

C++-Code:

```
if (test) {
    val->putco("name", "meine Modellfunktion");
    return OK;
} else {
    static vector<double> *in, out;
    val->get("var_of_eros7", &in);
    if (!in) return BAD;
    // mache etwas
    out[0] = 17.34;
    val->put("result_of_my_func", &out);
    return OK;
}
```

Tcl-Code:

```
if {$test} {
    set tvar "meine Modellfunktion"
    putco name tvar
    return OK
} else {
    set flag [get var_of_eros7 in]
    if {$flag == 0} return BAD
    # mache etwas
    set out(0) 17.34
    put result_of_my_func out
    return OK
}
```

B.2.11. Die Cleanup-Funktion (FINISH)

Diese Funktion wird aufgerufen, nachdem das gesamte Reaktionsnetzwerk aller Reaktoren aufgebaut wurde. Sie wird allerdings nur in Spezialfällen, wie der Nutzung von EROS7 als Schnittstelle zur chemischen Datenstruktur RICOS für andere Anwendungen (siehe 6.13), gebraucht, weshalb normalerweise folgender Code ausreicht:

C++-Code:

```
return OK;
```

Tcl-Code:

```
return OK
```

B.3. Die einzelnen Reaktionsregeln

Auch die einzelnen Regeln sind wieder in mehrere Teile gegliedert und zwar in folgende drei:

- RULE_INFO
- CONSTR
- FUNC

In RULE_INFO werden der Regelname, die Regelattribute und die Reaktionssubstruktur angegeben. Die Überprüfung der Bedingungen an der Reaktionssubstruktur findet im Teil CONSTR statt. Durchgeführt wird die Reaktion schließlich in FUNC inklusive der Berechnung der Reaktivitätsfunktion aus Edukt- und Produktparametern. Der Aufbau einer Regel sieht dann wie folgt aus:

C++-Code:

```
// Regel Nummer 1
case 1:
    switch (op) {
    case RULE_INFO:
        {
            // Reaktionssubstruktur, Name, Attribute
            return OK;
        }
    case CONSTR:
        // Bedingungen an der Reaktionssubstruktur
        return OK;
    case FUNC:
        // Reaktionsfunktion
        return OK;
    default:
        return BAD;
    }
return OK;
```

Tcl-Code:

```
RULE_1 {
    switch $op {
    RULE_INFO {
        # Reaktionssubstruktur, Name, Attribute
    }
    }
}
```

```

CONSTR {
    # Bedingungen an der Reaktionssubstruktur
}
FUNC {
    # Reaktionsfunktion
}
default {
    return BAD
}
}
return OK
}

```

B.3.1. Regelname, Attribute

Jede Regel hat einen Namen sowie ein NULL-terminiertes `const char`-Array mit den Attributen der Regel. Die Attribute der Regel werden dazu verwendet, einen Reaktionstyp für die Wahrscheinlichkeitskinetik als Umlagerungsreaktionstyp (`rearrangement`) zu kennzeichnen oder für sie eine Michaelis-Menten-Kinetik (`michaelis_menten_kinetics`) bzw. eine Kinetik nullter Ordnung (`force_zero_order`) festzulegen. Ist ein entsprechendes trace-Level gewählt, werden die Attribute und der Name für die einzelnen Regeln ausgegeben.

C++-Code:

```

val->putco("rule_name", "Enzymreaktion");
static const char *const attrib_al[]={ "michaelis_menten_kinetics", NULL };
val->putco("attributes", attrib_al);

```

Tcl-Code:

```

global attrib_1 rule_1
set rule_1 "Enzymreaktion"
putco rule_name rule_1
set attrib_1(0) michaelis_menten_kinetics
set attrib_1(1) NULL
putco attributes attrib_1

```

B.3.2. Die Reaktionssubstruktur

Der wichtigste Teil von `RULE_INFO` ist die Kodierung der Reaktionssubstruktur (RSS). Die Reaktionssubstruktur kann aus 0 bis beliebig vielen Teilzentren bestehen, die wiederum beliebig verzweigt sein können. Auch Ringe sind in der Reaktionssubstruktur zugelassen. Da die Reaktionssubstruktur so großzügig gestaltet ist, wird sie als Satz von Nachbarschaftsbeziehungen beschrieben. Die Atome in der Reaktionssubstruktur sind von 1 bis N durchnummeriert. Sie werden im übrigen auch in dieser Reihenfolge gesucht, was für die Numerierung in der RSS von Bedeutung ist. Aus Effizienzgründen empfiehlt es sich, den spezifischsten Atomen in der RSS die niedrigsten Nummern zu geben.

Die Kodierung der RSS ist ein 0-terminiertes Array aus Zahlentriplets:

```
a1, a2, d
```

`a1` und `a2` sind die Nummern der Atome in der RSS, wobei die Reihenfolge ohne Bedeutung ist. `d` ist für Erweiterungen reserviert und kann im Moment nur die Werte 0 oder 1 annehmen, die beide bedeuten, daß die Atome `a1` und `a2` direkt benachbart sind. Ursprünglich war gedacht, daß man mit `d` auch einen größeren Abstand zwischen den Atomen angeben kann. Es ist aber flexibler, dies über Eigenschaften zu lösen und die Überprüfung im Teil `CONSTR` vorzunehmen. Für eine 3-atomige RSS 1-2-3 sieht der Code folgendermaßen aus:

C++-Code:

```
static const int center_al[]={ 1, 2, 0, 2, 3, 0, 0 };
val->putco("center_connectivity", center_al);
```

Tcl-Code:

```
global center_1
set center_1 {1,2,0,2,3,0,0}
putco center_connectivity center_1
```

Die Atome eines Teilzentrums sollten in einer Reihe durchnummeriert sein, wobei die Teilzentren automatisch erkannt werden, da keine Atome aus verschiedenen Teilzentren Nachbarschaftsbeziehungen besitzen. Will man explizit ein einzelnes Atom als Teilzentrum spezifizieren, kann man das mit

```
a3, a3, 0
```

tun. Dies ist nur nötig, wenn es das letzte Atom in der RSS ist.

B.3.3. Bedingungen

Im Teil CONSTR einer Regel werden die Bedingungen, die an eine RSS über die reine Konnektivität hinaus gestellt werden, überprüft. Dies geschieht ebenso schrittweise, wie das Reaktionszentrum gesucht wird. Als Indikator, wieviele Atome im aktuellen Reaktionszentrum schon gefunden wurden, dient dabei der Übergabeparameter `sub_op`, der in Tcl implizit definiert ist. Für eine Reaktion wird die Funktion CONSTR nacheinander mit `sub_op = 0, 1, 2, ...` aufgerufen, bis alle Atome in der RSS überprüft wurden. Ist `sub_op = 0`, können die Eigenschaften des Ensembles, der Aggregate und Moleküle getestet werden. Für die Werte 1, 2, ... sind entsprechend viele Atome der RSS gefunden. Sind zwei Atome gefunden, können alle Eigenschaften der Atome 1 und 2 überprüft werden. Es empfiehlt sich allerdings, die Überprüfungen, die nicht vom Atom 2 abhängen, bereits für `sub_op = 1` durchzuführen. Benötigt man Oderverknüpfungen zwischen Eigenschaften der Atome 1 und 2, können diese erst überprüft werden, wenn auch das zweite Atom gefunden wurde.

C++-Code:

```
switch (sub_op) {
case 0: {
    // Überprüfung des Eduktensembles
    break;}
case 1:
    // Überprüfung von Atom 1 in der RSS
    break;}
case 2:
    // Überprüfung von Atom 2
    // sowie Oderbedingungen von Atom 1 und 2
    break;}
case 3:
    // Überprüfung von Atom 3
    // sowie Oderbedingungen der Atome 3 und 1 sowie 3 und 2
    break;}
default:
    return BAD;
}
return OK;
```

Tcl-Code:

```
switch $sub_op {
0 {
  # Überprüfung des Eduktensembles
}
1 {
  # Überprüfung von Atom 1 in der RSS
}
2 {
  # Überprüfung von Atom 2
  # sowie Oderbedingungen von Atom 1 und 2
}
3 {
  # Überprüfung von Atom 3
  # sowie Oderbedingungen der Atome 3 und 1 sowie 3 und 2
}
default {
  return BAD
}
}
```

Die einzelnen Teile von CONSTR für die unterschiedliche Anzahl gefundener Atome der verschiedenen Regeln werden im übrigen für die Regeln und die Reaktionen durcheinander aufgerufen. Es ist also fast ausgeschlossen, sich Werte zu merken, wenn ein Atom gefunden wurde, und diese zu verwenden, wenn auch ein weiteres Atom gefunden wurde.

B.3.3.1. Funktionalität

Um festzustellen, ob eine gefundene (Teil-)Substruktur den Anforderungen genügt oder nicht, können in CONSTR die Eigenschaften der Edukte überprüft werden. Dazu müssen zunächst die entsprechenden Eigenschaften abgefragt werden. Zu den Eigenschaften gehören auch Angaben über die Beschaffenheit der chemischen Struktur. Die Abfrage geschieht mit den Funktionen:

C++-Code:

```
bool flag;
var_type variable;
flag = eifc->prop(variable, prop_name, index_1, index_2);
```

Tcl-Code:

```
set flag [prop prop_name index_1 index_2 variable]
```

Abhängig von der Eigenschaft `prop_name` entfallen die Indizes teilweise. Die überwiegende Zahl aller Eigenschaften außer derer des ganzen Ensembles und derer von Nachbaratomen haben nur einen Index (Aggregat, Molekül, Gruppe, Elektronensystem und Atom). `index_2` entfällt in diesem Fall. Ensembleeigenschaften haben gar keinen Index und Nachbarschaftseigenschaften beide Indizes. Der Rückgabewert der Funktionen `prop` ist ein Flag, das angibt, ob die Eigenschaft erfolgreich abgeholt werden konnte (`true` = erfolgreich, Tcl: 1).

Die Indizes sind dabei diejenigen des Ensembles. Die Angabe des Index 1 für eine Atomeigenschaft bedeutet, daß diese Eigenschaft für das Atom 1 im Ensemble angefordert wird. Will man die Eigenschaft des ersten Atoms in der RSS, so muß man sich zunächst den Index dieses Atoms holen. Diese etwas umständlichere Methode ist durchaus beabsichtigt, da man so das ganze Ensemble im Zugriff hat und, wenn gewünscht, auch die Atome usw. in der Nachbarschaft der

RSS überprüfen kann.

Das folgende Beispiel zeigt, wie man sich die Ordnungszahl des ersten Atoms in der RSS abholen kann.

C++-Code:

```
bool flag;
int atom_index, elem;
atom_index = eifc->center(1);
flag = eifc->prop(elem, "A_ELEMENT", atom_index);
if (flag) {
    print << "Das Atom 1 in der RSS ist " << atom_index;
    print << " und hat die Ordnungszahl " << elem << endl;
}
```

Tcl-Code:

```
set atom_index [center 1]
set flag [prop A_ELEMENT $atom_index elem]
if {$flag} {
    print "Das Atom 1 in der RSS ist $atom_index"
    print " und hat die Ordnungszahl $elem\n"
}
```

In C++ kann der Name der Eigenschaft wahlweise als `const char* ("A_ELEMENT")` angegeben werden oder man macht sich allgemein ein Objekt der Klasse `rx_prent`, das dann immer wieder verwendet werden kann. Dieses Objekt wird dann statt des Eigenschaftsnamens übergeben und mit dem Konstruktor erzeugt, der den Namen als `const char*` erwartet. Da intern nicht bei jeder Abfrage die Eigenschaft über einen Stringvergleich in mehreren Listen gesucht werden muß, ist die etwas schnellere Methode in C++ eine Eigenschaft abzufragen diejenige über die Erzeugung eines Objekts der Klasse `rx_prent`.

C++-Code:

```
static rx_prent ordnz("A_ELEMENT");
bool flag;
int atom_index, elem;
atom_index = eifc->center(1);
flag = eifc->prop(elem, ordnz, atom_index);
if (flag) {
    print << "Das Atom 1 in der RSS ist " << atom_index;
    print << " und hat die Ordnungszahl " << elem << endl;
}
```

In C++ muß die Variable vom richtigen Typ sein, wobei es Eigenschaften mit folgenden Typen gibt:

- int
- double
- vector<int>
- vector<double>
- e_bit64
- const char*

C++-Code:

```
bool prop(          int &var, const char* prop_name, int idx1=0, int idx2=0);
```

```

bool prop(          double &var, const char* prop_name, int idx1=0, int idx2=0);
bool prop(   vector<int> &var, const char* prop_name, int idx1=0, int idx2=0);
bool prop(vector<double> &var, const char* prop_name, int idx1=0, int idx2=0);
bool prop(          e_bit64 &var, const char* prop_name, int idx1=0, int idx2=0);
bool prop( const char * &var, const char* prop_name, int idx1=0, int idx2=0);

bool prop(          int &var, const rx_prent &p, int idx1=0, int idx2=0);
bool prop(          double &var, const rx_prent &p, int idx1=0, int idx2=0);
bool prop(   vector<int> &var, const rx_prent &p, int idx1=0, int idx2=0);
bool prop(vector<double> &var, const rx_prent &p, int idx1=0, int idx2=0);
bool prop(          e_bit64 &var, const rx_prent &p, int idx1=0, int idx2=0);
bool prop( const char * &var, const rx_prent &p, int idx1=0, int idx2=0);

rx_prent(const char *name);

```

Darüber hinaus gibt es noch ein paar weitere nützliche Funktionen, mit deren Hilfe man feststellen kann, welche Eigenschaften verfügbar sind und welche Typen sie besitzen:

```

const char* prop_name(int idx);
const char* prop_type(const char *prop_name);
int         prop_idxs(const char *prop_name);

rx_prent(int idx);

```

In Tcl gibt es keine Unterscheidung bezüglich der Variablentypen und damit auch nur eine Funktion.

Tcl-Code:

```

prop property_name $idx1 $idx2 variable_name
set property_name prop_name($idx)
set ptype prop_type($property_name)
set idxs prop_idxs($property_name)

```

Der Name der Variablen kann in Tcl entfallen. Er ist dann identisch mit dem Namen der Eigenschaft.

Die so erhaltenen Eigenschaften können überprüft und verrechnet werden, wodurch entschieden wird, ob die (Teil-)Substruktur für die angestrebte Reaktion geeignet ist oder nicht.

B.3.3.2. Der Rückgabewert

Hat man festgestellt, ob die gefundene (Teil-)Substruktur für die geplante Reaktion geeignet ist oder nicht, muß man dies auch EROS7 mitteilen. Dies geschieht durch die unterschiedlichen Rückgabewerte der Regelfunktion. OK bedeutet, daß die Substruktur in Ordnung ist, BAD daß auf dieser Substruktur die Reaktion nicht durchgeführt werden kann.

Im folgenden Beispiel muß es sich bei der Überprüfung von Atom 1 in der RSS um ein Kohlenstoffatom handeln.

C++-Code:

```

bool flag;
int atom_index, element;
atom_index = eifc->center(1);
flag = eifc->prop(elem, "A_ELEMENT", atom_index);
if (elem != 6) return BAD;
return OK;

```

Tcl-Code:

```

set atom_index [center 1]
set flag [prop A_ELEMENT $atom_index elem]
if {$elem != 6} {return BAD}
return OK

```

B.3.4. Die Reaktionsfunktion

Die Reaktionsfunktion besteht aus mehreren Teilen. Im einfachsten Fall aus einer Eduktfunktion, der Reaktionsgenerierung und einer Produktfunktion. Die Edukt- und Produktfunktionen dienen der Berechnung der Reaktivität. Die einzelnen Teile sind nicht klar voneinander getrennt, sondern sie werden automatisch dadurch erkannt, daß entweder Eigenschaften abgefragt werden (Funktionen) oder Veränderungen am Ensemble vorgenommen werden (Reaktionsgenerator). Ausgehend vom Produkt der Reaktion kann sich eine weitere Teilreaktion mit entsprechender Produktfunktion anschließen. So ist es möglich, auch Eigenschaften von Zwischenprodukten oder Übergangszuständen bei der Berechnung der Reaktivität zu berücksichtigen. Die ausgegebene Reaktion ist diejenige vom Edukt zum letzten erzeugten Produkt.

B.3.4.1. Eduktfunktion

Hier können Eigenschaften abgefragt und auch schon verrechnet werden. Der Funktionsumfang ist identisch mit dem, der für die Überprüfung der Bedingungen verwendet werden kann (siehe B.3.3).

B.3.4.2. Reaktionsgenerierung

In diesem Teil werden Veränderungen am Ensemble vorgenommen, also die Reaktion selbst durchgeführt. Auch die Reaktionsdurchführung gliedert sich wieder in mehrer, mögliche Bereiche:

- Umordnung der Elektronensysteme mit der Valence-Bond-Emulation
- Umordnung der Elektronensysteme nach der MO-orientierten Methode
- Erzeugung von Atomen und Löschen von Aggregaten
- Erzeugung und Vernichtung von Gruppen
- Erzeugen neuer Eigenschaften
- Setzen von Eigenschaften
- Steuerung des aktiven Ensembles
- spezieller In- und Output
- Phase für die Produkte
- Symmetriefaktor
- Fehlerflags

Die Funktionen dieser Bereiche sollen im folgenden näher erläutert werden.

Umordnung der Elektronensysteme mit der Valence-Bond-Emulation

In EROS7 werden die chemischen Strukturen grundsätzlich mit der MO-orientierten Datenstruktur RICOS repräsentiert. Zur einfacheren Handhabung gibt es neben den zur MO-orientierten Datenstruktur passenden Funktionen zur Reaktionsgenerierung auch Emulationsroutinen, die die MO-orientierte Datenstruktur so behandeln, als würden sie auf einer Valence-Bond-Datenstruktur arbeiten. Dies sind die Funktionen `change_bond_order` und `change_free_els`. Bei der Änderung der Bindungsordnung sind `ia1` und `ia2` (siehe unten) die Indizes der Atome, zwischen denen die Bindungsordnung geändert werden soll und `delta` die Änderung. Ist `delta` größer als null, werden entsprechend viele Bindungen erzeugt, für `delta < 0` gebrochen. Das optionale Flag `single_aro_split` gibt schließlich noch an, ob der einfache Bruch eines cyclischen π -Systems als Bindungsbruch gewertet werden soll (=1) oder nicht (=0). Ist `single_aro_split = 0` und spaltet man in Benzol eine C-C-Bindung, wird die bevorzugt durchgeführt Spaltung des π -Systems nicht als Bindungsbruch interpretiert. Dies hat zur Folge, daß ein weiteres Elektronensystem gespalten wird, jetzt das σ -System, und der Benzolring ist somit an dieser Stelle komplett aufgebrochen worden. In der Regel will man aber das π -System des Aromaten an zwei Stellen aufbrechen und die C-C- σ -Bindungen intakt lassen. Deshalb ist der default-Wert für `single_aro_split = 1`. So

wird im ersten Schritt das cyclische π -System zwar nicht gebrochen, aber zum Bruch markiert. Im zweiten Schritt bricht das cyclische π -System dann unabhängig von `single_aro_split` in zwei Teile. `change_bond_order` erkennt auch selbstständig Konjugationen und hebt diese gegebenenfalls auf, bevor der eigentliche Bindungsbruch erfolgt. Dabei wird folgendermaßen vorgegangen: es wird zunächst ein π -System (falls vorhanden) gespaltet und überprüft, ob es sich dabei um einen homolytischen Bindungsbruch gehandelt hat. Dies ist der Fall, wenn beide resultierenden Elektronensysteme eine ungerade Zahl an Elektronen haben, wobei die Elektronen so verteilt werden, daß nach Möglichkeit so viele Elektronen in beide Elektronensysteme gegeben werden, wie die nun am Elektronensystem beteiligten Atome zum ursprünglichen Elektronensystem beigetragen haben. Da auf diese Weise auch Konjugationen aufgehoben werden können, aber entstehende Lonepairs und π -Systeme nicht automatisch konjugiert werden, gibt es die Funktion `conjugate_elsys`. Sie konjugiert die entstandenen Elektronensysteme nach Möglichkeit, was während der einzelnen Reaktionsschritte hinderlich wäre, da für die nächsten Reaktionsschritte die Konjugation höchstwahrscheinlich wieder aufgehoben werden müßte. Außerdem gibt es uneindeutige Fälle, in denen aufgrund von Heuristiken eine der Möglichkeiten gewählt wird, die aber im Einzelfall nicht der gewünschten entsprechen muß. So besteht die Möglichkeit, die Elektronensysteme selbst zu konjugieren oder dies von der Funktionen `conjugate_elsys` automatisch erledigen zu lassen. Funktionen, die ein Flag zurückgeben, kommen beim Fehlschlagen mit `false` bzw. 0 in Tcl zurück. Wird die Funktion `conjugate_elsys` nach Abschluß der Reaktion mit einer Reaktionsgenerierung nach der Valence-Bond-Emulation nicht aufgerufen, kann es zu unbeabsichtigten Effekten kommen. Da der für die Erkennung der Identität von Produkten verwendete Hashcode davon abhängt, ob Elektronensysteme konjugiert sind oder nicht, wird ein Produkt unter Umständen mehrfach (konjugiert und nicht konjugiert) erzeugt und als unterschiedliche Moleküle gespeichert. Dies liegt daran, daß der Hashcode für die MO-orientierte Kodierung von Verbindungen gedacht ist, bei der eine Konjugation auch gleichzeitig eine Planarität bedeutet. Entsteht ein Molekül also einmal konjugiert und ein anderes Mal nicht konjugiert, hat man tatsächlich zwei verschiedene Strukturen vor sich, eine planare Struktur und eine, bei der der Substituent so verdreht ist, daß die Konjugation aufgehoben ist. Bei einer Mischung der Reaktionsgenerierung mit der Emulation und der MO-orientierten Methode wird die Funktion `conjugate_elsys` nahezu unerläßlich. Für die Änderung der Zahl der freien Elektronen mit der Funktion `change_free_els` wird lediglich das Atom `ia` und die Änderung `i` der Elektronen (z.B.: +1 oder -2) angegeben.

Mit der Funktion `valence` kann überprüft werden, ob ein erzeugtes Produkt eine korrekte Struktur besitzt. `mode` gibt dabei den Modus an, mit dem die Prüfung durchgeführt wird. Es stehen zur Auswahl: `organic`, `inorganic`, `ms`, `ms-inorganic`, `none`, ...

C++-Code:

```
// emulation
void change_bond_order(int ial, int ia2, int delta, int single_aro_split=1);
void change_free_els(int ia, int i);
void conjugate_elsys();
bool valence(int mode=0);
```

Tcl-Code:

```
# emulation
change_bond_order $ial $ia2 $delta $single_aro_split
change_free_els $ia $i
conjugate_elsys
set flag [valence $mode]
```

Die Funktion `change_bond_order` verwendet für das Brechen von Bindungen intern einige Heuristiken zur automatischen Verteilung der Elektronen auf die beiden neuen Elektronensy-

steme. Eine Bindung gilt dann als gebrochen, wenn beide Elektronensysteme nach dem Bruch radikalisch sind. Dies bedeutet, daß beim Bruch von nicht cyclischen π -Systemen die mesomere Grenzstruktur genommen wird, bei der an der zu brechenden Stelle nach Möglichkeit keine Doppel- oder Dreifachbindung ist. „Die Mesomerie weicht also einem Bindungsbruch aus.“ Gebrochene cyclische π -Systeme zählen dann als Bindungsbruch, wenn `single_aro_split` gleich 1 bzw. nicht angegeben ist. Welche der π -Bindungen einer Dreifachbindung gespalten wird, ist zufällig. Die eingebauten Heuristiken haben nur eine begrenzte Gültigkeit, für die die Emulationsfunktionen angewendet werden können. Mit der Beschränkung auf neutrale organische Verbindungen und Oniumionen aus Hauptgruppenelementen, bei denen keine Oktetaufweitung stattgefunden hat, läßt sich dennoch fast die ganze organische Chemie formulieren. Eine Schwierigkeit ergibt sich bei Strukturen, die ein Stickstoffatom mit zwei σ - und zwei π -Systemen enthalten. Hier wird angenommen, daß dieses Atom zwei Elektronen zum betrachteten π -System beiträgt, wobei es auch eines sein kann, was aber nicht eindeutig entschieden werden kann. Es könnte sich um ein konjugiertes freies Elektronenpaar, eine Einfachbindung sowie eine Doppelbindung handeln oder um eine Kombination aus einer Dreifach- und einer Einfachbindung bzw. zweier Doppelbindungen, wobei das Stickstoffatom eine positive Formalladung tragen würde. Formalladungen sind nicht Bestandteil der MO-orientierten Molekülstruktur, da sie mit der Delokalisation der Elektronen auch delokalisiert werden können. Die Besetzungszahl des π -Systems kann nicht verwendet werden, da man ja gerade dabei ist, die Zahl der Elektronen zu bestimmen, die eigentlich in diesem Elektronensystem enthalten sein sollten, und zudem könnten ja noch weitere solcher Atome an dem π -System beteiligt sein. Spätestens dann ist eine Unterscheidung nicht mehr möglich. Sind in Molekülen σ -Bindungen mit mehr als zwei Zentren oder Komplexbindungen in den Molekülen enthalten, kann die Valence-Bond-Emulation nicht verwendet werden. In solchen Fällen muß auf die MO-orientierte Reaktionsgenerierung zurückgegriffen werden (siehe unten).

Umordnung der Elektronensysteme nach der MO-orientierten Methode

Bei der Reaktionsgenerierung nach der MO-orientierten Methode werden Elektronensysteme gespalten und zusammengefügt. Was ist ein Elektronensystem? Ein Elektronensystem ist die Summe aller Molekülorbitale mit gleicher Ausdehnung. So sind die drei bindenden und drei antibindenden MOs des π -Systems im Benzol ein Elektronensystem mit sechs Elektronen. Jede σ -Bindung und jedes Lonpair sind auch jeweils ein Elektronensystem mit zwei Elektronen. Neben den einfach- (Radikale), zweifach- und mehrfachbesetzten Elektronensystemen gibt es auch unbesetzte, wie im Fall eines nicht konjugierten Carbokations.

Drei Funktionen sind für die Reaktionsgenerierung zuständig:

- Elektronensysteme brechen (`split_elsy`)
- Elektronensysteme verbinden (`combine_elsys`)
- Die Zahl der Elektronen eines Elektronensystems ändern (`change_electrons`)

Um zwei Elektronensysteme zu verbinden übergibt man der Funktion `combine_elsys` die beiden Indizes der Elektronensysteme und den Typ des Elektronensystems, das gebildet werden soll. Als Typen gibt es folgende Möglichkeiten:

C++-Code:

- `es_sigma`
- `es_pi`
- `es_complex`
- `es_default`

Tcl-Code:

- `es_sigma`

- `es_pi`
- `es_complex`
- `es_default`

Bei den ersten drei Möglichkeiten wird ein entsprechendes Elektronensystem gebildet, wobei auch Lonepairs (besetzt oder unbesetzt) den Typ `es_pi` besitzen.

Für `es_sigma` sollte man zwei π -Systeme angeben, die nur ein einziges Atom enthalten. In diesem Fall ist klar, zwischen welchen Atomen das σ -System gebildet werden soll. Anderenfalls wird intern versucht herauszufinden, wo die Bindung erzeugt werden soll, es kann aber nicht garantiert werden, daß diese Stelle auch der entspricht, die beabsichtigt war. Mehrzentren- σ -Systeme können schrittweise aus besetzten bzw. unbesetzten Lonepairs (ein Atom- π -Systemen) aufgebaut werden. Zunächst entsteht ein normales 2-Elektronen-2-Zentren- σ -System, das dann nach und nach erweitert werden kann. Dazu verknüpft man das σ -System mit einem unbesetzten Lonepair und erhält ein Dreizentren- σ -System. Hierbei gibt es wiederum zwei Formen: σ -Systeme ohne Atom im Zentrum, wie die BBB-Bindung in Boranen, und das σ -System mit Atom im Zentrum, wie die BHB-Bindung in Diboran, bei der das Wasserstoffatom das Zentrum bildet. Primär entsteht beim schrittweisen Aufbau eines Mehrzentren- σ -Systems immer ein σ -System ohne Atom im Zentrum. Soll ein Atom als Zentrum gesetzt werden, kann dies anschließend mit der setzbaren Eigenschaft `EL_CENTER` geschehen, die den Index des Atoms angibt, das zum Zentrum werden soll.

`es_complex` erzeugt eine Komplexbindung, die analog der Bildung von Mehrzentren- σ -Systemen schrittweise aus einem besetzten π -System und mehreren unbesetzten Orbitalen (π -Systemen an einem Atom ohne Elektronen) des Metallatoms aufgebaut und gegebenenfalls auch wieder in diese gespalten werden.

`es_default` erzeugt ein σ -System, wenn die zwei Atome kein Elektronensystem verbindet - ansonsten ein π -System. Der Rückgabewert der Funktion ist der Index des gebildeten Elektronensystems.

Um die Indizes der Elektronensysteme zu erhalten, fragt man die Atomeigenschaft `A_ELECSYSS` bzw. die Nachbarschaftseigenschaft `N_ELECSYSS` ab und erhält die Indizes der Elektronensysteme an einem Atom bzw. die Elektronensysteme, an denen die zwei angegebenen Atome beteiligt sind.

C++-Code:

```
// Reaktionsgenerator
int combine_elsys(int iel1, int iel2, RX_ELSY_TYPE typ=es_default);
bool split_elsy(int ie, int ia1, int ia2, int &np1, int &np2, int iel1=-1);
bool change_electrons(int ie, int delta);
```

Tcl-Code:

```
# Reaktionsgenerator
set new_els [combine_elsys $iel1 $iel2 $created_els_type]
set flag [split_elsy $ie $ia1 $ia2 np1 np2 $iel1]
set flag [change_electrons $ie $delta]
```

`split_elsy` spaltet das Elektronensystem mit dem Index `ie` zwischen den Atomen `ia1` und `ia2`. `iel1` gibt die Zahl der Elektronen an, die das Elektronensystem erhalten soll, zu dem das Atom `ia1` gehört. Ist die Zahl negativ, werden die Elektronen nach Möglichkeit so verteilt, daß in den neuen Elektronensystemen so viele Elektronen enthalten sind, wie deren beteiligte Atome zum alten Elektronensystem beigesteuert haben, was allerdings dieselben Beschränkungen wie die automatische Verteilung der Elektronen bei der Spaltung von π -Bindungen nach der Valence-Bond-Methode (siehe oben) hat. In den Variablen `np1` und `np2` stehen nach dem Aufruf die Indizes der neu gebildeten Elektronensysteme, wobei einer der beiden auch dem Index des alten Elektronensystems entsprechen kann. Der Rückgabewert ist ein Flag, das angibt, ob das Elektronensystem gespalten werden konnte (C++: `true` / Tcl: 1 erfolgreich bzw. `false` / 0

nicht).

Handelt es sich bei dem zu spaltenden Elektronensystem um ein σ -System mit mehr als zwei Atomen, wird das Atom `ia1` aus dem σ -System herausgenommen und es entsteht am Atom `ia1` ein π -System mit einem Atom und `iel1` Elektronen. Ist `iel1` negativ oder nicht angegeben, erhält das π -System null Elektronen. Das Atom `ia2` hat in diesem Fall keine Bedeutung und muß auf 0 gesetzt werden. `np1` enthält nach dem Aufruf die Nummer des gebildeten π -Systems. Analoges gilt für Komplexbindungen.

Schließlich kann man mit `change_electrons` die Zahl der Elektronen des Elektronensystems `ie` bei positivem `delta` um `delta` erhöhen bzw. bei negativem um `delta` erniedrigen.

Erzeugung von Atomen und Löschen von Aggregaten

Bei Reaktionen, wie der Hydrolyse, hat man häufig Reaktionen pseudoerster Ordnung. Das in der Reaktion verbrauchte Wasser ist in großem Überschuß vorhanden, weshalb seine Konzentration als konstant angenommen werden kann. In diesen Fällen interessiert man sich häufig auch nicht dafür, das Wassermolekül bei den Ausgangsverbindungen zu haben. Deshalb besteht die Möglichkeit, auch innerhalb der Reaktionsgenerierung neue Atome einzufügen. Die Funktion `new_atom` erhält mit `ipernr` die Ordnungszahl des zu erzeugenden Atoms und gibt den Index dieses hinzugefügten Atoms zurück. Im Vektor `elss` stehen die Indizes der Elektronensysteme, die am neuen Atom hängen. In diesem Fall gibt man als Reaktivität die Geschwindigkeitskonstante pseudoerster Ordnung `k1` an (`k2`: Geschwindigkeitskonstante zweiter Ordnung):

$$k_1 = k_2 * [\text{H}_2\text{O}]$$

Bei komplexen Molekülen besteht auch die Möglichkeit das Molekül vom File einzulesen (siehe unten).

In manchen Situationen ist man auch an den Coprodukten nicht interessiert. Dies kann beispielsweise in der Massenspektroskopie für die abgespaltenen Neutralfragmente oder das entstehende Wasser bei einer Esterbildung in der kombinatorischen Chemie der Fall sein. Diese Aggregate können mit `delete_aggregate` gelöscht werden, wobei `agr_id` den Index des Aggregates angibt, das in den beiden erwähnten Beispielen aus einem einzigen Molekül besteht. Diesen Index erhält man unter anderem von der Atomeigenschaft `A_AGGREGATE`.

C++-Code:

```
int new_atom(int ipernr, vector<int> &elss);
bool delete_aggregate(int agr_id);
```

Tcl-Code:

```
set ia [new_atom $ipernr elss]
set flag [delete_aggregate $agr_id]
```

Erzeugung und Vernichtung von Gruppen

Eine weitere Möglichkeit der Reaktionsgenerierung ist die Bildung oder Spaltung von Aggregaten. Das bedeutet, daß Gruppen für die Wechselwirkung zwischen den Molekülen erzeugt bzw. gelöscht werden. Um welchen Typ einer Gruppe es sich handelt, wird durch ihren Namen festgelegt. Dies können beispielsweise sein: `h-bridge`, `ring`, `catenan`, etc.. Desweiteren kennt jede Gruppe noch ihre Zugehörigkeit zu einem Aggregat, Molekül, Atom oder Elektronensystem (`chemtyp`) und zwei Flags, `make_aggr` und `splittable`. `make_aggr = 1` gibt an, daß die Gruppe zwei oder mehrere Moleküle als Aggregat zusammenhält, wenn die Atome/Elektronensysteme aus verschiedenen Molekülen stammen. Ist `make_aggr = 0` wird die Gruppe gelöscht bzw. gespalten, wenn nach der Reaktion nicht alle Atome/Elektronensysteme zu einem Molekül gehören. Ob die Gruppe in diesem Fall gelöscht wird oder nur gespalten wird, liegt am Flag `splittable`. Ist dieses Flag 1, wird gespalten und die neuen Gruppen erhalten die Werte der gespaltenen Gruppe; ist es 0, wird die Gruppe gelöscht. Die übrigen Funk-

tionen erklären sich von selbst.

C++-Code:

```
// Gruppen Steuerung
int new_group(const char *name, const char* chemtyp, int idx,
              int make_aggr=1, int splittable=0);
int new_group(const char *name, ds_chem_type chemtyp, int idx,
              int make_aggr=1, int splittable=0);
bool delete_group(int gr_id);
bool add_atom_to_group(int gr_id, int at_id);
bool delete_atom_from_group(int gr_id, int at_id);
bool add_elsy_to_group(int gr_id, int el_id);
bool delete_elsy_from_group(int gr_id, int el_id);
void get_all_groups_of_type(const char *name, vector<int> &nums);
void get_groups_of(const char* chemtyp, int idx, vector<nt> &nums);
const char* get_type_of_group(int grp_idx);
const char* group_belongs_to(int grp_idx);
ds_chem_type group_belongs_to_ctyp(int grp_idx);
```

Tcl-Code:

```
// Gruppen Steuerung
set idx [new_group $name $chemtype $idx $make_aggr $splittable]
set flag [delete_group $gr_id]
set flag [add_atom_to_group $gr_id $at_id]
set flag [delete_atom_from_group $gr_id $at_id]
set flag [add_elsy_to_group $gr_id $el_id]
set flag [delete_elsy_from_group $gr_id $el_id]
get_all_groups_of_type $name grp_idx
get_groups_of $chemtype $dx
set name [get_type_of_group $grp_idx]
set tpye [group_belongs_to $grp_idx]
```

Erzeugen neuer Eigenschaften

Neben den berechneten Eigenschaften (PETRA-Parametern) gibt es auch solche Eigenschaften, die ihre Gültigkeit solange behalten, bis sie während der Durchführung einer Reaktion explizit umgesetzt werden. Dies sind die genealogischen Variablen. Man kann beliebig solche Eigenschaften neu erzeugen. Der Funktion `new_prop` werden vier Parameter übergeben, der Name der neuen Eigenschaft `prop_name`, der Datentyp der Eigenschaft (`int`, `double`, `vector<int>`, `vector<double>`, `e_bit64` oder `char*`), der Chemietyp (`atom`, `electronsyst`, `aggregate`, `molecule` oder `group`) sowie ein Flag `us4hash`, ob diese Eigenschaft für die Berechnung der Hashcodes verwendet werden soll (1 = ja, 0 = nein).

Diese so erzeugten Eigenschaften können dann während der Reaktion gesetzt werden (siehe unten).

C++-Code:

```
// create new property
bool new_prop(const char* prop_name, const char* dat_type, \
              const char* chm_type, int us4hash=0);
bool new_prop(const char* prop_name, const char* dat_type, \
              const char* chm_type, rx_prent& ent, int us4hash=0);
bool new_prop(const char* prop_name, proptype dat_type, \
              ds_chem_type chm_type, rx_prent& ent, int us4hash=0);
```

Tcl-Code:

```
set flag [new_prop $prop_name $dat_type $chm_type $us4hash]
```

Die Namen der Eigenschaften sollten der Konvention entsprechen, daß Eigenschaften für Atome mit `A_`, für Elektronensysteme mit `EL_`, für Gruppen mit `G_`, für Nachbarn mit `N_`, für Moleküle mit `M_` und für Aggregate mit `AG_` beginnen. Eigenschaften für das Ensemble (`E_`) können zwar erzeugt werden, werden allerdings nicht für Folgereaktionen gespeichert, da das

Produktensemble in EROS7 in die einzelnen Aggregate gespalten und als solche gespeichert wird. Dies dient der Reduktion des Speicherbedarfs. Für die Funktionen, in denen `rx_prent&ent` vorkommt, enthält `ent` nach dem Aufruf den richtigen Entry, der zum Abfragen der Eigenschaft verwendet werden kann.

Setzen von Eigenschaften

Die neu erzeugten Eigenschaften (genealogische Variablen) sowie setzbare Eigenschaften des Systems wie `EL_CENTER` können während der Reaktion folgendermaßen gesetzt werden:

C++-Code:

```
// set properties (ensemble vars)
bool set_prop(const int &var, const char* prop_name, int idx1=0, \
int idx2=0);
bool set_prop(const double &var, const char* prop_name, int idx1=0, \
int idx2=0);
bool set_prop(const vector<int> &var, const char* prop_name, int idx1=0, \
int idx2=0);
bool set_prop(const vector<double> &var, const char* prop_name, int idx1=0, \
int idx2=0);
bool set_prop(const e_bit64 &var, const char* prop_name, int idx1=0, \
int idx2=0);
bool set_prop(const char *const &var, const char* prop_name, int idx1=0, \
int idx2=0);
//
bool set_prop(const int &var, const rx_prent &p, int idx1=0, \
int idx2=0);
bool set_prop(const double &var, const rx_prent &p, int idx1=0, \
int idx2=0);
bool set_prop(const vector<int> &var, const rx_prent &p, int idx1=0, \
int idx2=0);
bool set_prop(const vector<double> &var, const rx_prent &p, int idx1=0, \
int idx2=0);
bool set_prop(const e_bit64 &var, const rx_prent &p, int idx1=0, \
int idx2=0);
bool set_prop(const char *const &var, const rx_prent &p, int idx1=0, \
int idx2=0);
```

Tcl-Code:

```
set flag [set_prop var $prop_name $idx1 $idx2]
```

In den genealogischen Variablen kann man Angaben wie die Anregungsenergie des Aggregates oder Umlagerungszähler speichern. Außerdem kann man mit ihrer Hilfe steuern, welche Reaktionstypen für Folgereaktionen zugelassen sind.

Steuerung des aktiven Ensembles

Bis jetzt haben wir nur von einer Reaktion mit Übergangszuständen und Zwischenprodukten gesprochen. Das Reaktionssystem von EROS7 ist aber auch dazu ausgelegt, ausgehend von einem gefundenen Reaktionszentrum mehrere Reaktionen zu erzeugen. Dies kann interessant sein, wenn zwei verschiedene Stereoisomere mit unterschiedlichen Reaktivitäten gebildet werden oder für Reaktionen, bei denen das Reaktionszentrum eigentlich keine Atome enthält, sondern nur Elektronensysteme, wie bei der Ionisationsregel in der Massenspektroskopie. Deshalb gibt es die folgenden Funktionen, um steuern zu können, auf welchem Ensemble weitergearbeitet wird und um schließlich festzulegen, daß es sich beim generierten Ensemble um das Produkt einer der Reaktionen handelt, die ausgehend vom gefundenen Reaktionszentrum erzeugt werden. Häufig dürften hierzu die Funktionen `save_active_ens_as_product` zum Speichern der Reaktion und `set_active_ens(1)`, um erneut mit dem Edukt zu beginnen, verwendet werden. Die Ensembles werden dabei in der Reihenfolge, in der sie erzeugt werden, durchnummeriert. Die Nummer des aktiven Ensembles kann mit der Funktion `get_active_ens` abgefragt werden. Wird keine der Funktionen

save_active_ens_as_product oder save_ens_as_product aufgerufen, ist automatisch das letzte erzeugte Ensemble das Produkt der Reaktion.

C++-Code:

```
// ensemblesteuerung
bool set_active_ens(int ie);
int get_active_ens();
void save_active_ens_as_product();
bool save_ens_as_product(int ie);
bool append_ens(int i);
void empty_ens();
```

Tcl-Code:

```
set flag [set_active_ens $i]
set i [get_active_ens]
save_active_ens_as_product
set flag [save_ens_as_product $i]
set flag [append_ens $i]
empty_ens
```

Spezieller In- und Output

Damit man nicht unbedingt Moleküle, die während der Reaktion hinzugefügt werden, aus einzelnen Atomen zusammenbauen muß, kann man auch Moleküle vom File einlesen. Dabei werden die Moleküle, unabhängig davon, wie oft die Funktion aufgerufen wird, tatsächlich nur beim ersten Mal eingelesen.

Zwei weitere Gründe für die speziellen IO-Routinen sind:

- die Möglichkeit, zu Tracing-Zwecken die Moleküle auszugeben und
- die Verwendung des Reaktionsgenerierungssystems als Kernsystem für andere Programme. Ersterer erklärt sich von selbst. Bei zweiterem ist daran gedacht, daß man ein EROS7 entwickelt, das „lediglich eine einzige Reaktion erzeugt“. Die Erzeugung dieser Reaktion wird dann von einem Subsystem übernommen, das an der Regelschnittstelle ansetzt und, wie oben erwähnt, aus der „einen Reaktion“ ein ganzes Reaktionsnetzwerk erzeugen kann. Die Einsatzgebiete der Programme, die man so implementieren kann, sind mannigfaltig.

Als Formate sind momentan CTX, SCREEN und CTX_MO implementiert.

Wenn mit der print_active_ens-Funktion gearbeitet wird, die const char* file_name als Übergabeparameter hat, oder man mit Tcl arbeitet, kann mit open_stream und close_stream das File zum Lesen und Schreiben öffnet bzw. geschlossen werden.

C++-Code:

```
// IO
void print_active_ens(ostream& fp, const char *format);
void print_active_ens(ostream& fp, e_format format=SCREEN);
void print_active_ens(const char* format, const char* file_name);
void open_stream(const char* name);
void close_stream(const char* name);
bool read_ens(const char* file_name);
bool read_ens(istream& fp);
```

Tcl-Code:

```
print_active_ens $format $file_name
open_stream $name
close_stream $name
read_ens $file_name
```

Phase für die Produkte

Wie oben erwähnt, kann EROS7 mehrere Phasen und Reaktoren aufbauen. Wird bei einer Reaktion nichts angegeben, bleiben die Produkte der Reaktion in der Phase, in der auch die Edukte waren. Um Verbindungen auch in andere Phasen zu bekommen, ist es notwendig, auch Phasen-

übergänge und Phasentransferreaktionen zuzulassen. Dies ist grundsätzlich mit jeder Reaktion möglich. Es muß nur während der Reaktionsdurchführung die Phase gesetzt werden, in die die Produkte kommen sollen. Dabei werden immer alle Produkte einer Reaktion in dieselbe Phase gegeben. Man kann sich dementsprechend zwei Typen von Phasentransferreaktionen vorstellen:

- reine Phasenübergänge, bei denen keine Modifikationen am Ensemble vorgenommen werden, sowie
- kombinierte Phasentransferreaktionen, die aus einer chemischen Reaktion und dem reinen Übergang bestehen.

Um die Phase der Produkte zu setzen, braucht man nur die Variable `result_phase` der Klasse `rx_eifc` auf die Nummer der Phase zu setzen, in die die Produkte gehen sollen. Dabei ist darauf zu achten, daß der Wert nicht außerhalb der existierenden Phasen liegt. Desweiteren sollte die Produktphase innerhalb des aktuellen Reaktors liegen. Soll ein Kinetikmodell verwendet werden, muß sie das auch, da die Produkte bei Phasenübergängen und Phasentransferreaktionen zwischen Reaktoren eine Konzentration von 0.0 erhalten. Die kinetischen Berechnungen wären also falsch. Dies liegt daran, daß jeder Reaktor seine eigene Kinetik besitzt.

Gesetzt wird die Produktphase wie folgt:

C++-Code:

```
eifc->result_phase = 2;
```

Tcl-Code:

```
set result_phase 2
```

Zu Beginn jeder Reaktion hat die Variable `result_phase` den Wert der Eduktphase. Die Phase des ersten Aggregates, die diese Reaktion gerade erzeugt, kann man auch aus der Variablen `phase`, die von EROS7 an die Regeln übergeben wird, erhalten.

Symmetriefaktor

Ist kein Atom im Reaktionszentrum, wie bei der Ionisationsreaktion für die Simulation von Massenspektren, oder handelt es sich um Reaktionen pseudoerster Ordnung, bei denen während der Reaktion Moleküle eingeschleust werden, kann EROS7 den Symmetriefaktor nicht korrekt berechnen. Für die Ionisationsreaktion wird der Symmetriefaktor auf die Symmetriezahl des ionisierten Elektronensystems gesetzt. Er wäre sonst immer 1. Im Fall der Reaktion pseudoerster Ordnung muß der Symmetriefaktor mit dem des eingeführten Moleküls multipliziert werden. Der Symmetriefaktor kann genauso gelesen und gesetzt werden wie die Phase für die Produkte.

C++-Code:

```
eifc->symmetry_factor *= 2;
```

Tcl-Code:

```
set symmetry_factor [expr $symmetry_factor*2]
```

Fehlerflags

Viele Funktionen geben ein Flag zurück, ob die angeforderte Aktion erfolgreich war. In den meisten Fällen werden diese Flags allerdings nicht überprüft werden. Deshalb wird bei jedem Fehler eine Warnung ausgegeben und diese Reaktion nachträglich verworfen. Dies wird so gemacht, da EROS7 nicht in der Lage ist festzustellen, ob das Flag überprüft wird oder nicht.

Überprüft man die gelieferten Flags, kann man diese Warnungen zusammen mit der Unterdrückung dieser Reaktionen im Regelteil `INIT_RULES` folgendermaßen abstellen:

C++-Code:

```
eifc->putco("no_discard_on_error", 1);
```

Tcl-Code:

```
global tvar
set tvar 1
putco no_dicard_on_error tvar
```

Hat man nicht in allen Fällen die Fehlerflags getestet, kann man vor dem Verlassen von FUNC mit `bool flag=eifc->get_err_ocur();` überprüfen, ob ein Fehler eingetreten ist (true) und gegebenenfalls den Regelfunktionsaufruf mit `return BAD;` beenden. Mit `eifc->set_warnings(true);` bzw. mit `false` wird das Fehlerflag zurückgesetzt.

B.3.4.3. Produktfunktion

Sind alle Reaktionsteilschritte durchgeführt, können die Eigenschaften der Produkte abgefragt werden. Die Werte werden schließlich zusammen mit denen der Edukte verrechnet und daraus der Reaktivitätswert berechnet. Dieser wird dann der Variablen zugewiesen, die dem Kernsystem unter dem Namen `reactivity` bekanntgegeben wurde.

Die Nummern der Atome, Elektronensysteme und Gruppen bleiben während der ganzen Reaktionsschritte gleich, damit man sein Reaktionszentrum auch in den Produkten wiederfindet. Die Nummern der Aggregate und Moleküle muß man sich dagegen nach jeder Modifikation des Ensembles wieder holen z.B. über die Atome. Da in den Reaktionen beispielsweise durch die Kombination zweier Elektronensysteme zu einem Elektronensysteme vernichtet werden können, aber die Indizes der anderen Elektronensysteme gleich bleiben, kann es zu Löchern in der Reihenfolge der Nummern der Elektronensysteme kommen. Anfänglich sind alle Elektronensysteme konsekutiv durchnummeriert, aber nach der Reaktion gibt es beispielsweise das Elektronensystem Nummer 5 nicht mehr. Um so etwas feststellen zu können, wenn man es nicht bei den Reaktionsfunktionen mitprotokolliert, gibt es spezielle Eigenschaften: `E_N...`, `E_LAST...` und `._EXIST`. Die Punkte stehen dabei für `AGGREGATES`, `MOLECULES`, `GROUPS`, `ELECSYSS` oder `ATOMS` bzw. der einzelne Punkt entsprechend für `AG`, `M`, `G`, `E` oder `A`. `E_N...` liefert die Anzahl der ... im aktuellen Ensemble, wohingegen `E_LAST...` die höchste verwendete Indexnummer für ... liefert. Der Unterschied liegt in den ungültigen Indexnummern, der durch die Löschung von Aggregaten auch bei Atomen usw. auftreten kann. Will man nun feststellen, ob ein bestimmter Index noch gültig ist, kann man sich die immer gültige Eigenschaft `._EXIST` für diesen Index holen. 1 bedeutet, daß der Index gültig ist, und 0, daß der Teil des Ensembles nicht (mehr) existiert.

Es gibt noch eine Reihe weiterer Eigenschaften, die nicht mit den Aggregaten/Molekülen gespeichert werden, sondern vom Regelinterface generiert werden. Alle Eigenschaften, die Indizes zurückliefern, gehören dazu:

<code>A_NEIGHBORS</code>	die Indizes der Nachbaratome
<code>N_ELECSYSS</code>	die Indizes der Elektronensysteme, die zwischen zwei Atomen sind
<code>A_ELECSYSS</code>	die Indizes der Elektronensysteme, an denen das Atome beteiligt ist
<code>A_AGGREGATE</code>	der Index des Aggregates, zu dem das Atom gehört
<code>A_MOLECULE</code>	der Index des Moleküls, zu dem das Atom gehört
<code>M_AGGREGATE</code>	der Index des Aggregates, zu dem das Molekül gehört
<code>A_GROUPS</code>	die Indizes der Gruppen, zu denen das Atom gehört
<code>EL_ATOMS</code>	die Indizes der Atome, die am Elektronensystem beteiligt sind

aber auch:

<code>B_BOORD</code>	Die VB-Bindungsordnung, die bei einer möglichen mesomeren Grenzstruktur zwischen zwei Atomen herauskommt. Sind die Atome nicht benachbart, ist die Bindungsordnung 0.
<code>EL_SI_SIGMA</code>	Flag, ob das Elektronensystem ein σ -System ist (1=ja, 0=nein)
<code>EL_CENTER</code>	Lesen und Setzen des Zentrums für σ -Systeme mit mehr als 2 Atomen

sowie Setzen des Typs des σ -Systems. Ist der gesetzte Wert (Atomindex) 0, ist / wird es ein σ -System ohne Zentrum.

A_ELEMENT die Ordnungszahl (historisch bedingt)
A_LABEL das Atomlabel (historisch bedingt)
A_HASH_INDEX der Atomhashindex im gesamten Ensemble
A_NEIG_INDEX der Nachbarschaftsindex zweier Atome im gesamten Ensemble
A_SYMMETRY die Zahl der äquivalenten Atome
A_IDHALBE eine Testeigenschaft (historisch bedingt)
A_INDEX der interne Index der Datenstruktur, der sich nach jeder Reaktion ändern kann (historisch bedingt)

EL_ELECTRONS die Zahl der Elektronen

Eine vollständige Liste aller dem System bekannten Eigenschaften erhält man mit folgendem kurzen Regelstück, das man am besten beim ersten Aufruf des Teils PREP_ROOT aufruft:

C++-Code:

```
int idx=1, idxs;
const char *pname, *ptype;
while ((pname = eifc->prop_name(idx))) {
    ptype = eifc->prop_type(pname);
    idxs = eifc->prop_idxxs(pname);
    print << "Eigenschaft " << pname << " hat den Typ ";
    print << ptype << " und braucht " << idxs;
    print << " Indizes" << endl;
    idx++;
}
```

Tcl-Code:

```
set idx 1
while {$idx} {
    set pname [prop_name $idx]
    if ($pname) {
        set ptype [prop_type $pname]
        set idxs [prop_idxxs $pname]
        print "Eigenschaft $pname hat den Typ "
        print "$ptype und braucht $idxs Indizes\n"
        incr idx
    } else {
        set idx 0
    }
}
```

B.3.4.4. Ein sehr einfaches Beispiel

Das folgende Beispiel zeigt den RG22

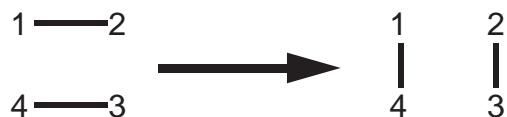


Abbildung 215: Reaktionsschema des RG22 (Reaktionsgenerator breche 2 Bindungen und mache 2 Bindungen).

mit der Valence-Bond-Emulation und der Funktion

$$k = (2.5 + q_{\sigma}(\text{Edukt, Atom 2}) - q_{\sigma}(\text{Produkt, Atom 3})) * 10^{-6}.$$

C++-Code:

```
int a1, a2, a3, a4;
a1 = eifc->center(1);
a2 = eifc->center(2);
a3 = eifc->center(3);
a4 = eifc->center(4);
double qsige2, qsigp3;
bool flag;

// Eduktfunktion
flag = eifc->prop(qsige2, "A_QSIG", a2);

// Reaktionsgenerator
eifc->change_bond_order(a1,a2,-1);
eifc->change_bond_order(a3,a4,-1);
eifc->change_bond_order(a2,a3,+1);
eifc->change_bond_order(a1,a4,+1);

// Produktfunktion
flag = eifc->prop(qsigp3, "A_QSIG", a3);

// Berechnung der Reaktivität
//     var_reactivity wurde unter INIT_RULES
//     mit put als reactivity übergeben
var_reactivity = (2.5 + qsige2 - qsigp3) * 1.e-6;

return OK;
```

Tcl-Code:

```
set a1 [center 1]
set a2 [center 2]
set a3 [center 3]
set a4 [center 4]

# Eduktfunktion
set flag [prop A_QSIG $a2 qsige2]

# Reaktionsgenerator
change_bond_order $a1 $a2 -1
change_bond_order $a3 $a4 -1
change_bond_order $a2 $a3 1
change_bond_order $a1 $a4 1

# Produktfunktion
set flag [prop A_QSIG $a3 qsigp3]
```


























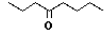


































```
# Berechnung der Reaktivität
#   var_reactivity wurde unter INIT_RULES
#   mit put als reactivity übergeben
set var_reactivity [expr (2.5 + $qsige2 - $qsigp3) * 1.e-6]

























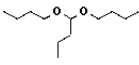



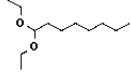
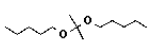
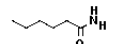
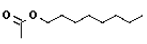

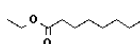

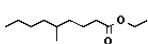
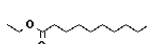
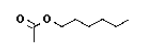
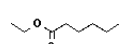
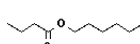
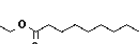
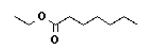
return OK
```

B.3.4.5. Neuronale Netze

Alternativ zur Berechnung der Reaktivität als Funktion kann sie auch mit neuronalen Netzen ermittelt werden. Zur Verwendung von Kohonen- und Counterpropagationnetzen siehe Anhang F.1 (Aufruf aus Tcl) bzw. für Backpropagationnetze siehe Anhang F.2 (Aufruf aus C++). Die verschiedenen Netzwerktypen können natürlich auch in den Regeln, die in der jeweils anderen Sprache geschrieben sind, eingesetzt werden.

C. 103 Moleküle für die MS-Bewertung

3066 4309  PROPIONSÄURE	4359 5860  ISOBUTTERSÄURE	3841 5212  BUTTERSÄURE, 3-METHYL-	9401 11880  BUTTERSÄURE, 2-METHYL-	9402 11881  PENTANSÄURE, 2-METHYL-	4037 5449  PENTANSÄURE (VALERIANISÄURE)
8490 10797  MEISSENSÄURE, METHYLESTER	7631 9773  ISOBUTTERSÄURE, METHYLESTER	1534 2414  BUTTERSÄURE, ETHYLESTER	1697 2597  PROPIONSÄURE, METHYLESTER	3971 5367  PENTANSÄURE, ISOPROPYLESTER	813 1482  PENTANSÄURE, TERT.BUTYLESTER
4583 6125  BUTTERSÄURE, METHYLESTER	4624 6176  ISOBUTTERSÄURE, TERT.BUTYLESTER	1732 2637  ACETALDEHYD	1350 2197  PROPIONALDEHYD	1349 2196  BUTYRALDEHYD	8968 11360  PENTANAL (VALERALDEHYD)
9618 12155  HEPTANAL	1287 2125  ISOBUTYRALDEHYD	1346 2193  HEXANAL, 2-ETHYL	846 1521  ACETON	429 919  2-BUTANONE	1318 2165  3-HEXANON
634 1246  3-PENTANONE	11815 14963  4-OCTANON	10011 12605  SULFID, DIETHYL	10339 12976  ETHYLCHLORID	11979 12977  ETHER, TERT.BUTYL METHYL	1261 2084  2-BUTANOL, 2-METHYL-
1307 2150  ETHER, METHYL PROPYL	1308 2151  ETHER, ISOPROPYL METHYL	197 554  TRIETHYLAMIN	264 658  ETHER, ETHYL METHYL	2695 3839  1-PENTANOL	297 707  ETHER, DIETHYL
3729 5082  2-PROPANOL	3730 5083  2-BUTANOL	3858 5229  1-PROPANAMIN	4270 5755  BUTAN, 2-CHLORO-	4349 5849  1-BUTANOL	4452 5965  1-HEPTANOL
4750 6332  METHAN, CHLORO-	4904 6506  DIISOPROPYLAMIN	5077 6707  DIPROPYLAMIN	5207 6867  PROPYLAMIN, N-ETHYL-N-METHYL	5208 6868  HEPTYLAMIN, N-ETHYL-N-METHYL	5646 7382  HEPTYLAMIN, N,N-DIMETHYL-
5791 7563  PROPYLAMIN, N-METHYL-N-PROP	5850 7641  ETHANAMIN, N-ETHYL-	591 1181  3-PENTANOL	7802 9985  4-OCTANOL	7818 10003  METHYLJODID	7819 10004  ETHYLJODID
7825 10012  PROPAN, 1-CHLORO-	7835 10022  PROPAN, 2-CHLORO-	7837 10024  PROPAN, 2-BROMO-2-METHYL	7838 10025  PROPAN, 1-BROMO-	7909 10101  1-PENTYLAMIN	7912 10104  DIBUTYLAMIN

7927 10119  1-HEXYLAMIN 9298 11750	802 1468  ETHANOL 957 1731	827 1501 Br  ETHAN, BROMO- 9657 12195	8406 10696  2-PROPANOL, 2-METHYL- 979 1753	9122 11542  2-PROPANAMIN, 2-METHYL- 9970 12555	9176 11601 HS  METHANTHIOL 833 1508
 2-BUTYLAMIN 835 1510	 ETHER, DIISOPROPYL 843 1518	 PROPAN, 2-CHLORO-2-METHYL 847 1522	 TRIBUTYLAMIN 848 1523	-Br METHAN, BROMO- 849 1524	 1-PENTEN, 2-METHYL- 904 1648
 1-HEXEN 3208 4472	 1-PENTEN 3170 4422	 1-BUTEN 3172 4426	 2-BUTEN 1486 2362	 ISOBUTEN 1975 2953	 2-HEXENE 3099 4345
 1-BUTEN, 2-METHYL- 3992 5393	 2-PENTEN, CIS-TRANS 7022 9054	 2-PENTEN, 2-METHYL- 7482 9584	 UTYRALDEHYD, DIMETHYLACETALDEHYD, DIISOPROPYLACE 8466 10771	 ETHAN, 1,1-DIETHOXY- 8957 11346	 ETHAN, 1,1-DIETHOXY- 9395 11874
 PROPAN, 1,1-ETHOXY-1-METHOXY 9905 12476	 BUTAN, 1,1-DIBUTOXY 229364	 PROPAN, 2,2-DIMETHOXY 2667 3807	 PROPAN, 1,1-DIETHOXY-2-METHOXY 2744 3898	 PROPAN, 1,1-DIETHOXY-2-METHOXY 2745 3899	 OCTAN, 1,1-DIETHOXY-2-METHOXY 3179 4435
 PROPAN, 2,2-BIS(PENTYLOXY) 3493 4810	 DECANSAEURE, ETHYLESTER 628-02-4	 ESSIGSAEURE, OCTYLESTER 4030 5442	 OCTANSAEURE, METHYLESTER 4031 5443	 OCTANSAEURE, ETHYLESTER 4034 5446	 NONANSAEURE, METHYLESTER 4106 5543
 ISAEURE, 5-METHYL-, ETHYLESTER 4450 5963	 DECANSAEURE, ETHYLESTER	 ESSIGSAEURE, HEXYLESTER	 HEXANSAEURE, ETHYLESTER	 BUTTERSAEURE, HEXYLESTER	 NONANSAEURE, ETHYLESTER
 HEPTANSAEURE, ETHYLESTER					

D. Von FRANZ bewertete Reaktionen

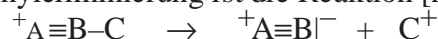
In den nachfolgenden Kapiteln sind von FRANZ, der mit der Darstellung der Strukturen als Bindungsliste arbeitet, bewertete Reaktionen aufgeführt, wobei jeweils nur das Edukt angegeben ist. Über der Struktur sind von links die Nummer der Reaktionsregel, also die Nummer des Reaktionstyps, angegeben, gefolgt von der Nummer der Reaktion und in eckigen Klammern den Nummern der Atome im Reaktionszentrum. Die Zahlen an den Atomen in der Struktur entsprechen diesen Atomnummern. Kennt man die Reaktion, ist daraus leicht die Reaktion zu ermitteln.

Für die Oniumreaktion ist die Reaktion [A-B-C-D]:



wobei zwischen B und D statt der Bindung auch eine längere Kette sein kann.

Für die Carbonyleliminierung ist die Reaktion [A-B-C]:



Für die McLafferty-Spaltung ist die Reaktion [A-B-C-D-E-F]:



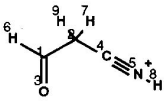
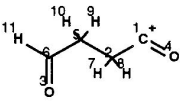
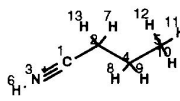
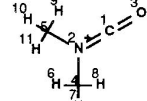
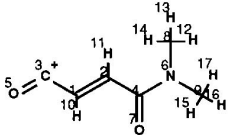
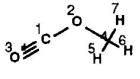
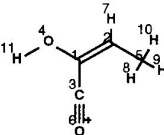
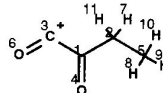
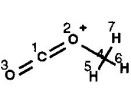
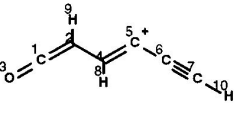
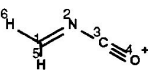
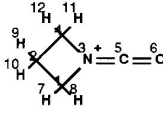
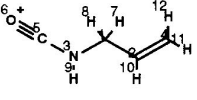
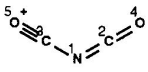
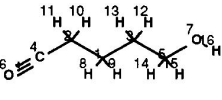
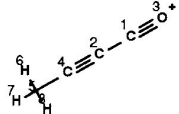
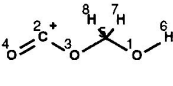
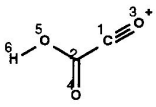
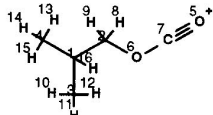
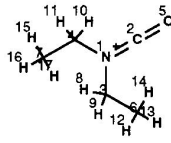
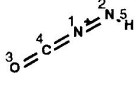
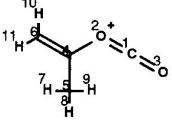
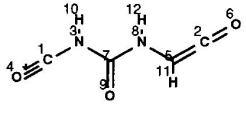
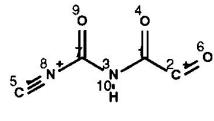
Unter den Strukturen sind die von FRANZ bestimmte Reaktionswahrscheinlichkeit sowie die Äquivalenzklassen der Reaktion angegeben. Die Datenbanken, in die FRANZ diese Reaktionen schreibt, sind wegen der Größe der Dateien auf eine bestimmte Anzahl von Reaktionen beschränkt. FRANZ versucht mit der reduzierten Anzahl von Reaktionen dennoch den gesamten Datenraum abzudecken. Dazu teilt FRANZ, abhängig von der topologischen Beschaffenheit des Reaktionszentrums und der ersten und zweiten Sphäre von Atomen, die Reaktionen in mehrere Äquivalenzklassen ein. Diese sind hinter EQC: angegeben. FRANZ sammelt nach Möglichkeit gleich viele Reaktionen in den verschiedenen Äquivalenzklassen. Ist ein gewisser geduldeter Überhang an Reaktionen in einer Klasse überschritten, werden in diese Klasse keine neuen Reaktionen mehr aufgenommen, bis der Überhang durch den Eintrag von Reaktionen in andere Klassen wieder unter den Schwellwert sinkt.

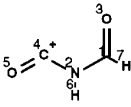
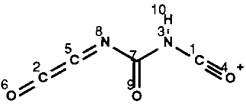
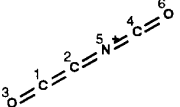
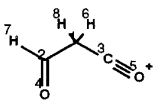
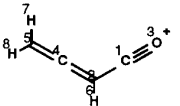
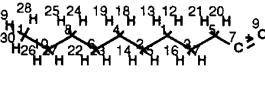
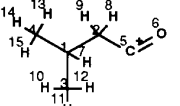
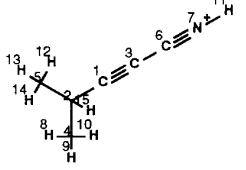
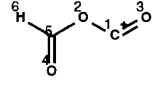
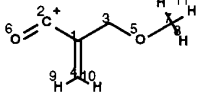
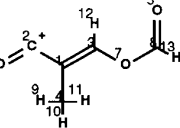
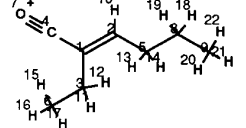
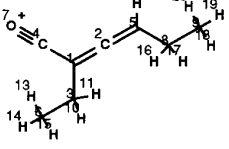
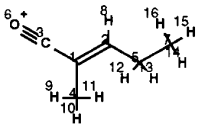
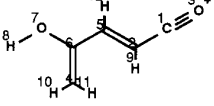
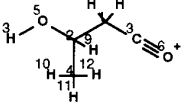
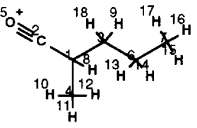
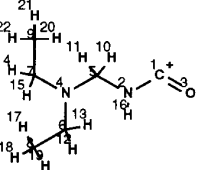
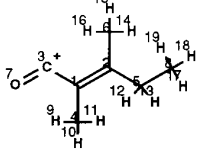
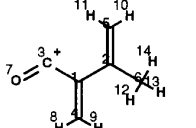
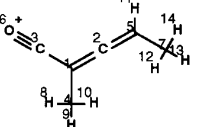
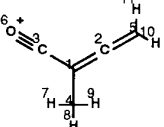
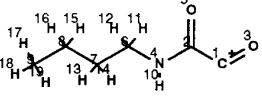
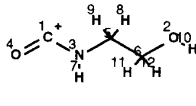
D.1. Bei Verwendung von Umlagerungsreaktionstypen

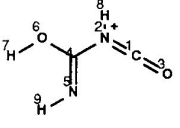
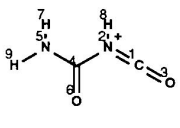
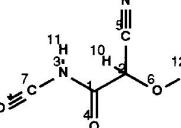
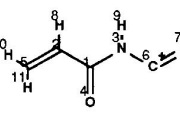
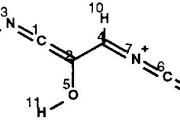
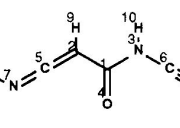
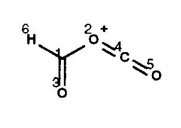
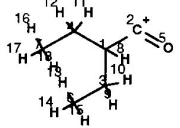
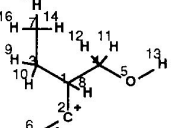
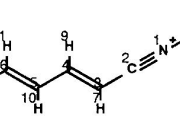
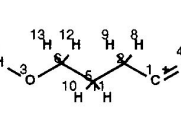
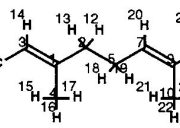
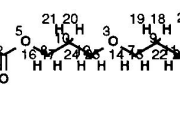
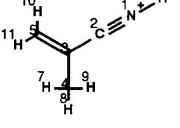
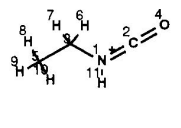
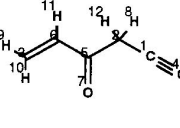
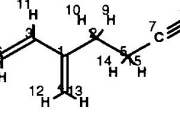
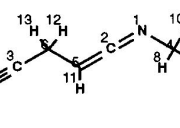
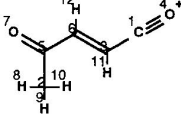
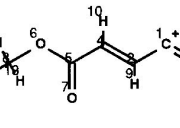
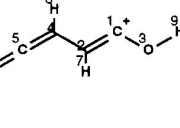
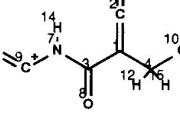
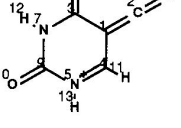
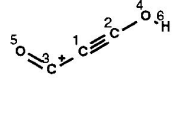
D.1.1. Carbonyleliminierung Klasse 1-Daten

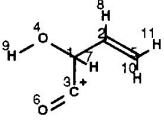
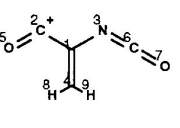
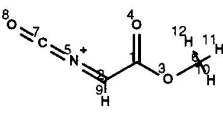
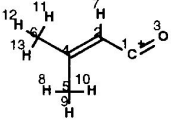
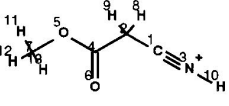
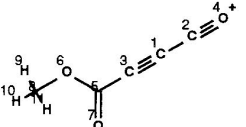
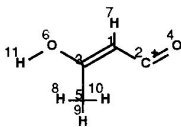
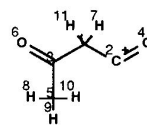
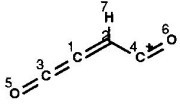
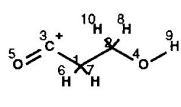
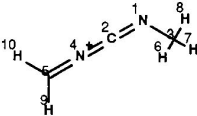
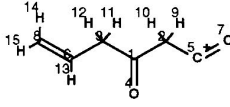
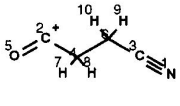
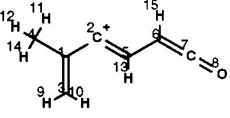
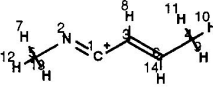
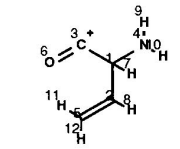
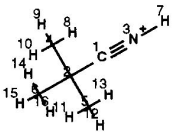
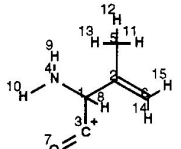
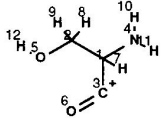
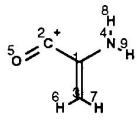
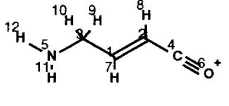
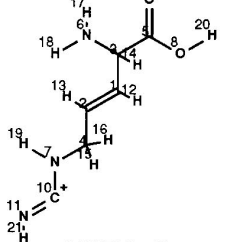
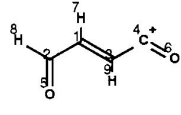
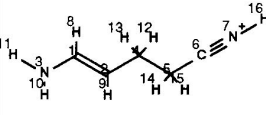
Beschriftung siehe Seite 253.

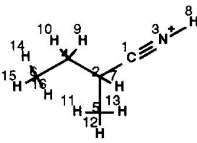
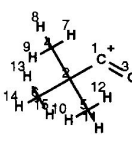
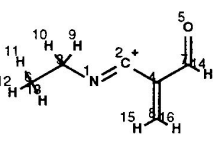
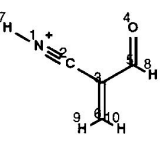
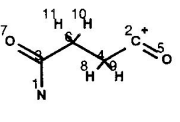
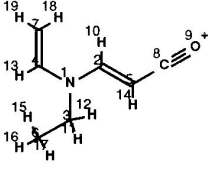
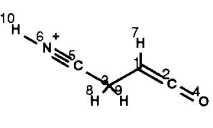
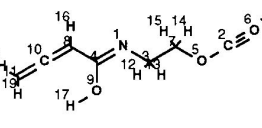
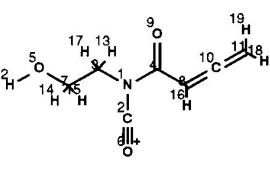
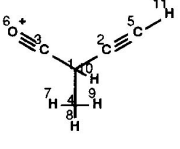
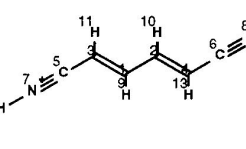
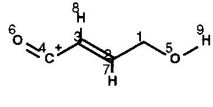
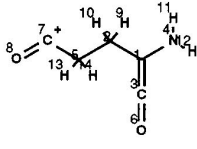
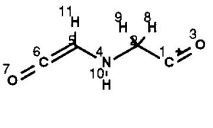
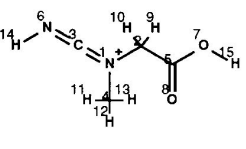
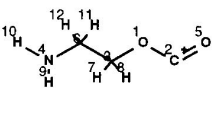
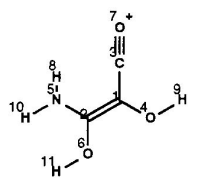
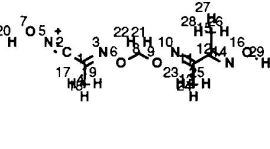
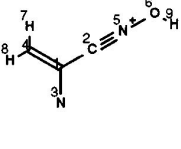
<p>9 4 [6-5-2]</p> <p>0.000 EQC: 1</p>	<p>9 9 [6-2-1]</p> <p>0.998 EQC: 2</p>	<p>9 11 [5-2-1]</p> <p>0.950 EQC: 2</p>	<p>9 12 [4-2-1]</p> <p>0.975 EQC: 3</p>
<p>9 15 [4-2-1]</p> <p>0.402 EQC: 4</p>	<p>9 16 [3-1-2]</p> <p>0.998 EQC: 2</p>	<p>9 19 [6-3-1]</p> <p>0.000 EQC: 5</p>	<p>9 20 [2-1-3]</p> <p>0.995 EQC: 6</p>
<p>9 21 [1-2-3]</p> <p>0.983 EQC: 1</p>	<p>9 27 [6-3-1]</p> <p>0.000 EQC: 7</p>	<p>9 28 [3-1-2]</p> <p>0.019 EQC: 1</p>	<p>9 36 [1-2-3]</p> <p>0.012 EQC: 4</p>
<p>9 38 [5-3-1]</p> <p>0.989 EQC: 1</p>	<p>9 39 [2-1-3]</p> <p>0.000 EQC: 8</p>	<p>9 41 [5-2-1]</p> <p>0.000 EQC: 9</p>	<p>9 48 [3-1-2]</p> <p>0.991 EQC: 10</p>
<p>9 49 [3-1-2]</p> <p>0.000 EQC: 11</p>	<p>9 50 [3-1-2]</p> <p>0.000 EQC: 12</p>	<p>9 73 [4-1-2]</p> <p>0.995 EQC: 13</p>	<p>9 78 [5-3-1]</p> <p>0.999 EQC: 1</p>
<p>9 79 [6-4-2]</p> <p>1.000 EQC: 1</p>	<p>9 80 [6-4-2]</p> <p>0.991 EQC: 1</p>	<p>9 83 [5-3-1]</p> <p>0.992 EQC: 1</p>	<p>9 84 [6-3-1]</p> <p>0.997 EQC: 1</p>

<p>9 87 [5-4-2]</p>  <p>0.957 EQC: 14</p>	<p>9 88 [4-1-2]</p>  <p>0.000 EQC: 12</p>	<p>9 100 [3-1-2]</p>  <p>0.953 EQC: 4</p>	<p>9 105 [3-1-2]</p>  <p>0.017 EQC: 15</p>
<p>9 110 [5-3-1]</p>  <p>0.986 EQC: 16</p>	<p>9 113 [3-1-2]</p>  <p>0.040 EQC: 5</p>	<p>9 116 [6-3-1]</p>  <p>0.000 EQC: 7</p>	<p>9 118 [6-3-1]</p>  <p>0.000 EQC: 17</p>
<p>9 119 [3-1-2]</p>  <p>0.027 EQC: 11</p>	<p>9 121 [3-1-2]</p>  <p>0.000 EQC: 10</p>	<p>9 246 [4-3-2]</p>  <p>0.000 EQC: 18</p>	<p>9 248 [6-5-3]</p>  <p>0.000 EQC: 15</p>
<p>9 250 [6-5-3]</p>  <p>0.000 EQC: 18</p>	<p>9 264 [5-3-1]</p>  <p>0.343 EQC: 19</p>	<p>9 268 [6-4-2]</p>  <p>0.000 EQC: 1</p>	<p>9 299 [3-1-2]</p>  <p>0.999 EQC: 1</p>
<p>9 310 [4-2-3]</p>  <p>0.529 EQC: 20</p>	<p>9 314 [3-1-2]</p>  <p>0.003 EQC: 9</p>	<p>9 318 [5-7-6]</p>  <p>1.000 EQC: 5</p>	<p>9 320 [5-2-1]</p>  <p>0.000 EQC: 15</p>
<p>9 329 [3-4-1]</p>  <p>0.000 EQC: 21</p>	<p>9 332 [3-1-2]</p>  <p>0.000 EQC: 11</p>	<p>9 369 [4-1-3]</p>  <p>0.000 EQC: 22</p>	<p>9 370 [6-2-1]</p>  <p>0.000 EQC: 23</p>

<p>9 374 [5-4-2]</p>  <p>0.000 EQC: 24</p>	<p>9 377 [4-1-3]</p>  <p>0.000 EQC: 22</p>	<p>9 379 [6-4-5]</p>  <p>0.872 EQC: 25</p>	<p>9 380 [5-3-1]</p>  <p>0.000 EQC: 8</p>
<p>9 399 [3-1-2]</p>  <p>0.540 EQC: 1</p>	<p>9 400 [9-7-5]</p>  <p>0.000 EQC: 3</p>	<p>9 407 [6-5-2]</p>  <p>0.981 EQC: 3</p>	<p>9 411 [7-6-3]</p>  <p>0.084 EQC: 4</p>
<p>9 428 [3-1-2]</p>  <p>0.000 EQC: 20</p>	<p>9 431 [6-2-1]</p>  <p>0.000 EQC: 17</p>	<p>9 432 [6-2-1]</p>  <p>0.000 EQC: 17</p>	<p>9 436 [7-4-1]</p>  <p>0.999 EQC: 6</p>
<p>9 440 [7-4-1]</p>  <p>0.998 EQC: 6</p>	<p>9 449 [6-3-1]</p>  <p>0.959 EQC: 6</p>	<p>9 465 [3-1-2]</p>  <p>0.985 EQC: 8</p>	<p>9 477 [6-3-1]</p>  <p>0.000 EQC: 8</p>
<p>9 481 [5-2-1]</p>  <p>0.999 EQC: 6</p>	<p>9 495 [3-1-2]</p>  <p>0.997 EQC: 26</p>	<p>9 499 [7-3-1]</p>  <p>0.983 EQC: 2</p>	<p>9 500 [7-3-1]</p>  <p>0.912 EQC: 2</p>
<p>9 502 [6-3-1]</p>  <p>0.967 EQC: 6</p>	<p>9 504 [6-3-1]</p>  <p>0.941 EQC: 6</p>	<p>9 517 [3-1-2]</p>  <p>0.000 EQC: 27</p>	<p>9 536 [4-1-3]</p>  <p>0.000 EQC: 24</p>

<p>9 562 [3-1-2]</p>  <p>0.000 EQC: 28 9 570 [8-6-7]</p>	<p>9 563 [3-1-2]</p>  <p>0.000 EQC: 28 9 575 [8-6-3]</p>	<p>9 564 [9-7-3]</p>  <p>0.000 EQC: 29 9 584 [5-4-2]</p>	<p>9 566 [7-6-3]</p>  <p>0.000 EQC: 24 9 586 [5-2-1]</p>
<p>0.000 EQC: 25 9 587 [6-2-1]</p> 	<p>0.000 EQC: 19 9 589 [1-2-3]</p> 	<p>0.000 EQC: 30 9 591 [4-1-2]</p> 	<p>0.994 EQC: 2 9 592 [8-6-3]</p> 
<p>0.971 EQC: 17 9 606 [4-1-2]</p> 	<p>0.969 EQC: 4 9 628 [1-2-3]</p> 	<p>0.000 EQC: 12 9 636 [4-2-1]</p> 	<p>0.999 EQC: 1 9 645 [4-1-2]</p> 
<p>0.000 EQC: 31 9 658 [8-7-5]</p> 	<p>0.961 EQC: 32 9 662 [7-3-6]</p> 	<p>0.000 EQC: 33 9 666 [4-1-3]</p> 	<p>0.936 EQC: 8 9 684 [3-1-2]</p> 
<p>0.996 EQC: 1 9 687 [6-5-4]</p> 	<p>0.998 EQC: 34 9 707 [5-9-7]</p> 	<p>0.795 EQC: 8 9 708 [6-2-1]</p> 	<p>0.967 EQC: 35 9 710 [5-3-1]</p> 
<p>0.938 EQC: 36</p> 	<p>0.000 EQC: 37</p> 	<p>0.000 EQC: 38</p> 	<p>0.000 EQC: 12</p> 

<p>9 726 [6-3-1]</p>  <p>0.981 EQC: 17</p>	<p>9 732 [5-2-1]</p>  <p>0.915 EQC: 27</p>	<p>9 737 [8-7-5]</p>  <p>0.000 EQC: 39</p>	<p>9 748 [3-1-2]</p>  <p>0.000 EQC: 3</p>
<p>9 757 [3-1-2]</p>  <p>0.000 EQC: 40</p>	<p>9 758 [4-2-1]</p>  <p>0.000 EQC: 41</p>	<p>9 760 [4-2-1]</p>  <p>0.000 EQC: 12</p>	<p>9 762 [4-2-1]</p>  <p>0.000 EQC: 12</p>
<p>9 788 [6-4-2]</p>  <p>0.979 EQC: 12</p>	<p>9 801 [5-3-1]</p>  <p>0.000 EQC: 12</p>	<p>9 872 [1-2-4]</p>  <p>0.000 EQC: 42</p>	<p>9 874 [7-5-2]</p>  <p>0.960 EQC: 12</p>
<p>9 887 [5-2-4]</p>  <p>0.982 EQC: 13</p>	<p>9 930 [8-7-6]</p>  <p>0.992 EQC: 10</p>	<p>9 931 [2-1-3]</p>  <p>0.972 EQC: 43</p>	<p>9 938 [6-3-1]</p>  <p>0.979 EQC: 44</p>
<p>9 939 [3-1-2]</p>  <p>0.000 EQC: 32</p>	<p>9 941 [7-3-1]</p>  <p>0.964 EQC: 44</p>	<p>9 947 [6-3-1]</p>  <p>0.019 EQC: 27</p>	<p>9 951 [5-2-1]</p>  <p>0.939 EQC: 44</p>
<p>9 954 [6-4-2]</p>  <p>0.989 EQC: 34</p>	<p>9 972 [11-8-0-7]</p>  <p>0.000 EQC: 45</p>	<p>9 976 [6-4-3]</p>  <p>0.000 EQC: 12</p>	<p>9 990 [7-6-5]</p>  <p>0.993 EQC: 4</p>

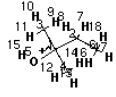
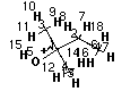
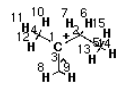
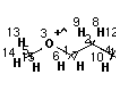
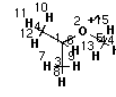
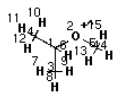
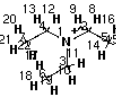
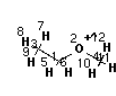
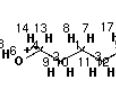
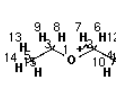
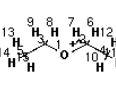
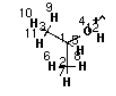
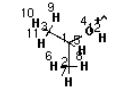
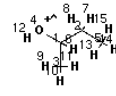
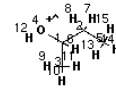
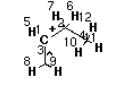
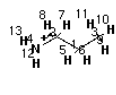
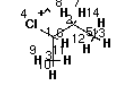
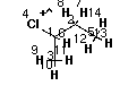
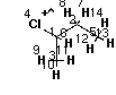
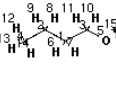
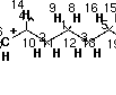
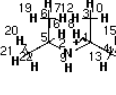
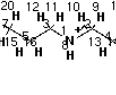
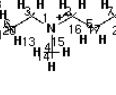
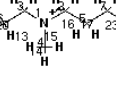
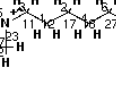
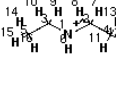
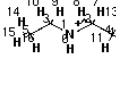
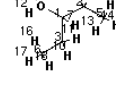
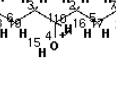
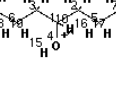
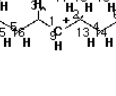
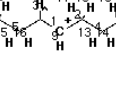
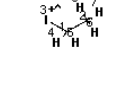
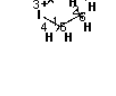
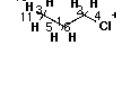
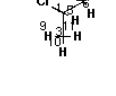
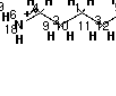
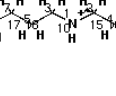
<p>9 996 [3-1-2]</p>  <p>0.994 EQC: 32</p>	<p>9 1008 [3-1-2]</p>  <p>0.941 EQC: 2</p>	<p>9 1020 [1-2-4]</p>  <p>0.000 EQC: 46</p>	<p>9 1022 [1-2-3]</p>  <p>0.000 EQC: 47</p>
<p>9 1023 [5-2-4]</p>  <p>0.000 EQC: 16</p>	<p>9 1064 [9-8-5]</p>  <p>0.999 EQC: 34</p>	<p>9 1070 [6-5-3]</p>  <p>0.000 EQC: 14</p>	<p>9 1090 [6-2-5]</p>  <p>0.000 EQC: 48</p>
<p>9 1092 [6-2-1]</p>  <p>0.000 EQC: 49</p>	<p>9 1101 [6-3-1]</p>  <p>0.998 EQC: 6</p>	<p>9 1141 [7-5-3]</p>  <p>0.969 EQC: 4</p>	<p>9 1209 [6-4-3]</p>  <p>0.986 EQC: 12</p>
<p>9 1227 [8-7-5]</p>  <p>0.999 EQC: 13</p>	<p>9 1246 [3-1-2]</p>  <p>0.957 EQC: 13</p>	<p>9 1247 [6-3-1]</p>  <p>0.000 EQC: 50</p>	<p>9 1257 [5-2-1]</p>  <p>0.994 EQC: 51</p>
<p>9 1285 [7-3-1]</p>  <p>0.989 EQC: 52</p>	<p>9 1292 [5-2-1]</p>  <p>0.000 EQC: 53</p>	<p>9 1294 [5-2-1]</p>  <p>0.000 EQC: 54</p>	

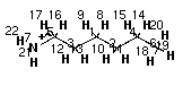
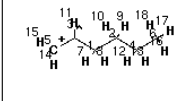
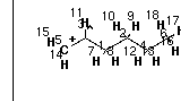
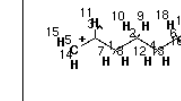
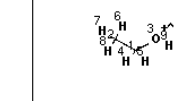
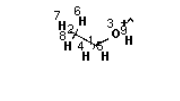
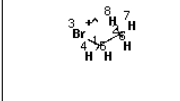
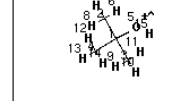
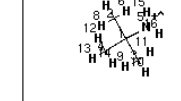
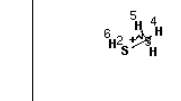
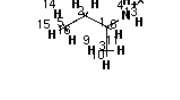
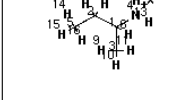
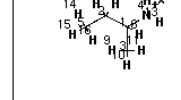
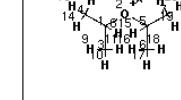
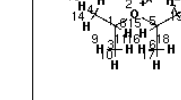
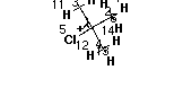
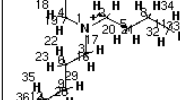
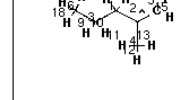
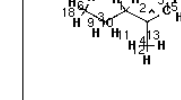
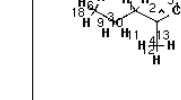
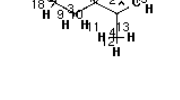
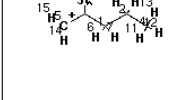
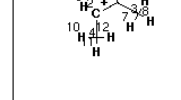
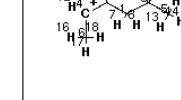
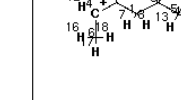
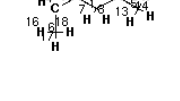
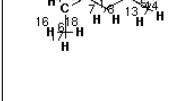
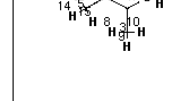
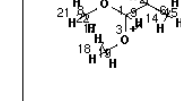
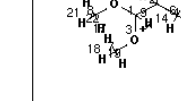
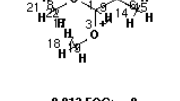
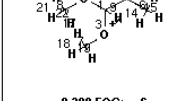
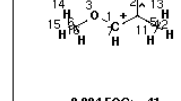
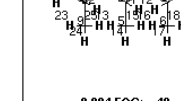
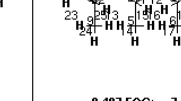
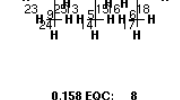
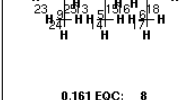
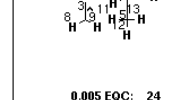
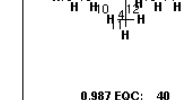
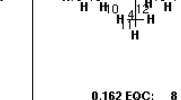
D.2. Ohne Umlagerungsreaktionstypen

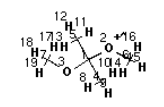
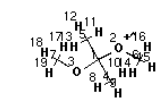
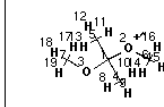
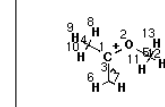
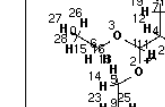
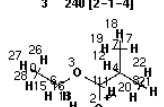
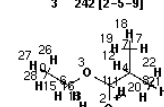
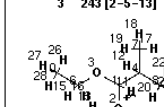
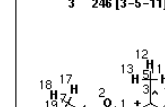
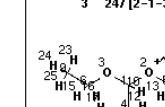
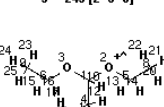
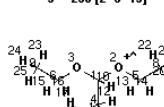
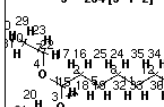
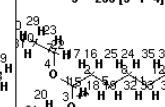
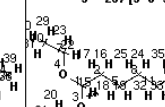
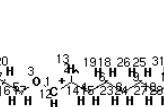
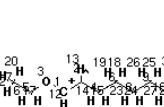
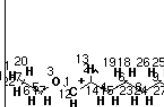
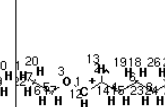
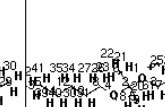
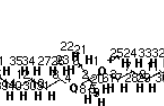
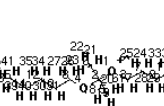
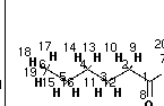
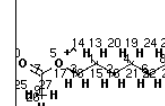
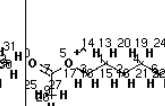
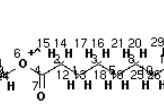
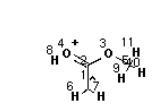
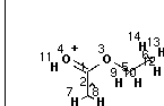
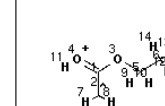
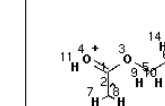
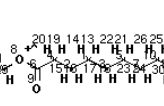
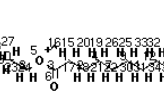
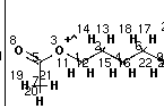
D.2.1. α -Spaltung

Beschriftung siehe Seite 253.

<p>3 2 [3-1-2]</p> <p>0.334 EQC: 1</p>	<p>3 3 [5-2-1]</p> <p>0.206 EQC: 2</p>	<p>3 4 [6-5-2]</p> <p>0.471 EQC: 1</p>	<p>3 5 [3-2-4]</p> <p>0.000 EQC: 3</p>	<p>3 6 [6-3-1]</p> <p>0.232 EQC: 2</p>
<p>3 8 [1-3-7]</p> <p>0.044 EQC: 4</p>	<p>3 9 [4-2-1]</p> <p>0.422 EQC: 1</p>	<p>3 10 [2-1-5]</p> <p>0.668 EQC: 5</p>	<p>3 12 [2-1-3]</p> <p>0.159 EQC: 1</p>	<p>3 14 [1-3-8]</p> <p>0.401 EQC: 5</p>
<p>3 18 [3-1-2]</p> <p>0.349 EQC: 1</p>	<p>3 20 [1-3-8]</p> <p>0.896 EQC: 5</p>	<p>3 21 [1-2-4]</p> <p>0.197 EQC: 7</p>	<p>3 27 [1-3-8]</p> <p>0.061 EQC: 5</p>	<p>3 28 [1-2-4]</p> <p>0.113 EQC: 10</p>
<p>3 32 [3-1-2]</p> <p>0.364 EQC: 2</p>	<p>3 36 [3-1-2]</p> <p>0.517 EQC: 12</p>	<p>3 37 [3-1-4]</p> <p>0.215 EQC: 13</p>	<p>3 38 [4-2-1]</p> <p>0.000 EQC: 12</p>	<p>3 39 [4-2-7]</p> <p>0.660 EQC: 13</p>
<p>3 40 [5-3-1]</p> <p>0.230 EQC: 12</p>	<p>3 46 [8-6-19]</p> <p>0.913 EQC: 13</p>	<p>3 47 [5-2-1]</p> <p>0.180 EQC: 14</p>	<p>3 48 [5-2-7]</p> <p>0.778 EQC: 13</p>	<p>3 49 [7-4-1]</p> <p>0.322 EQC: 14</p>
<p>3 50 [7-4-15]</p> <p>0.761 EQC: 13</p>	<p>3 51 [5-3-9]</p> <p>0.021 EQC: 13</p>	<p>3 54 [4-1-2]</p> <p>0.612 EQC: 12</p>	<p>3 55 [4-1-2]</p> <p>0.745 EQC: 12</p>	<p>3 56 [4-1-3]</p> <p>0.408 EQC: 12</p>
<p>3 57 [4-1-2]</p> <p>0.718 EQC: 12</p>	<p>3 58 [4-1-3]</p> <p>0.714 EQC: 12</p>	<p>3 59 [4-1-3]</p> <p>0.894 EQC: 12</p>	<p>3 60 [4-1-2]</p> <p>0.750 EQC: 12</p>	<p>3 61 [4-1-2]</p> <p>0.829 EQC: 12</p>
<p>3 62 [4-1-3]</p> <p>0.732 EQC: 12</p>	<p>3 63 [1-2-4]</p> <p>0.349 EQC: 16</p>	<p>3 64 [1-2-6]</p> <p>0.067 EQC: 17</p>	<p>3 66 [3-1-4]</p> <p>0.002 EQC: 19</p>	<p>3 67 [2-1-3]</p> <p>0.860 EQC: 20</p>

<p>3 69 [5-1-2]</p>  <p>0.970 EQC: 20</p>	<p>3 70 [5-1-3]</p>  <p>0.568 EQC: 20</p>	<p>3 73 [3-1-2]</p>  <p>0.240 EQC: 22</p>	<p>3 74 [3-1-2]</p>  <p>0.815 EQC: 23</p>	<p>3 77 [2-1-3]</p>  <p>0.620 EQC: 23</p>
<p>3 79 [2-5-13]</p>  <p>0.143 EQC: 21</p>	<p>3 80 [1-2-5]</p>  <p>0.487 EQC: 25</p>	<p>3 82 [2-1-3]</p>  <p>0.664 EQC: 23</p>	<p>3 85 [6-4-2]</p>  <p>0.578 EQC: 23</p>	<p>3 90 [1-2-4]</p>  <p>0.506 EQC: 23</p>
<p>3 91 [1-2-6]</p>  <p>0.085 EQC: 21</p>	<p>3 92 [4-1-2]</p>  <p>0.860 EQC: 23</p>	<p>3 93 [4-1-5]</p>  <p>0.112 EQC: 24</p>	<p>3 94 [4-1-2]</p>  <p>0.982 EQC: 23</p>	<p>3 95 [4-1-3]</p>  <p>0.519 EQC: 23</p>
<p>3 97 [3-1-2]</p>  <p>0.269 EQC: 22</p>	<p>3 98 [4-2-1]</p>  <p>0.769 EQC: 27</p>	<p>3 100 [4-1-2]</p>  <p>0.586 EQC: 18</p>	<p>3 101 [4-1-3]</p>  <p>0.082 EQC: 18</p>	<p>3 102 [4-1-6]</p>  <p>0.000 EQC: 29</p>
<p>3 103 [5-3-1]</p>  <p>0.830 EQC: 23</p>	<p>3 110 [6-4-14]</p>  <p>0.931 EQC: 9</p>	<p>3 113 [2-1-3]</p>  <p>0.520 EQC: 27</p>	<p>3 115 [1-2-4]</p>  <p>0.619 EQC: 27</p>	<p>3 119 [1-2-5]</p>  <p>0.857 EQC: 25</p>
<p>3 122 [1-4-13]</p>  <p>0.850 EQC: 26</p>	<p>3 123 [5-3-1]</p>  <p>0.962 EQC: 25</p>	<p>3 130 [1-2-4]</p>  <p>0.602 EQC: 27</p>	<p>3 131 [1-2-7]</p>  <p>0.098 EQC: 28</p>	<p>3 132 [4-1-2]</p>  <p>0.929 EQC: 23</p>
<p>3 139 [4-1-2]</p>  <p>0.716 EQC: 23</p>	<p>3 140 [4-1-3]</p>  <p>0.448 EQC: 23</p>	<p>3 143 [1-3-5]</p>  <p>0.019 EQC: 22</p>	<p>3 145 [3-1-2]</p>  <p>0.941 EQC: 22</p>	<p>3 148 [3-1-2]</p>  <p>0.010 EQC: 32</p>
<p>3 150 [3-1-4]</p>  <p>0.001 EQC: 31</p>	<p>3 151 [4-2-1]</p>  <p>0.111 EQC: 18</p>	<p>3 153 [4-1-2]</p>  <p>0.148 EQC: 18</p>	<p>3 158 [6-4-2]</p>  <p>0.938 EQC: 27</p>	<p>3 161 [1-2-4]</p>  <p>0.662 EQC: 27</p>

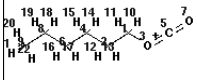
<p>3 163 [7-5-3]</p>  <p>0.981 EQC: 27</p>	<p>3 165 [5-3-1]</p>  <p>0.783 EQC: 22</p>	<p>3 167 [3-1-2]</p>  <p>0.910 EQC: 22</p>	<p>3 168 [3-1-7]</p>  <p>0.013 EQC: 9</p>	<p>3 170 [3-1-2]</p>  <p>0.736 EQC: 23</p>
<p>3 171 [3-1-4]</p>  <p>0.283 EQC: 21</p>	<p>3 173 [3-1-4]</p>  <p>0.002 EQC: 35</p>	<p>3 174 [5-1-2]</p>  <p>0.875 EQC: 20</p>	<p>3 175 [5-1-2]</p>  <p>0.892 EQC: 36</p>	<p>3 176 [2-1-3]</p>  <p>0.256 EQC: 17</p>
<p>3 177 [4-1-2]</p>  <p>0.985 EQC: 27</p>	<p>3 178 [4-1-3]</p>  <p>0.315 EQC: 27</p>	<p>3 179 [4-1-6]</p>  <p>0.066 EQC: 30</p>	<p>3 180 [2-1-3]</p>  <p>0.603 EQC: 23</p>	<p>3 181 [2-1-8]</p>  <p>0.153 EQC: 24</p>
<p>3 182 [5-1-2]</p>  <p>0.543 EQC: 37</p>	<p>3 183 [1-2-5]</p>  <p>0.545 EQC: 25</p>	<p>3 186 [5-2-4]</p>  <p>0.575 EQC: 22</p>	<p>3 187 [2-1-3]</p>  <p>0.319 EQC: 38</p>	<p>3 189 [2-4-11]</p>  <p>0.237 EQC: 11</p>
<p>3 190 [2-5-14]</p>  <p>0.001 EQC: 11</p>	<p>3 191 [3-1-2]</p>  <p>0.623 EQC: 22</p>	<p>3 192 [2-1-3]</p>  <p>0.707 EQC: 22</p>	<p>3 195 [4-2-1]</p>  <p>0.406 EQC: 22</p>	<p>3 197 [4-6-16]</p>  <p>0.059 EQC: 9</p>
<p>3 198 [2-1-3]</p>  <p>0.564 EQC: 22</p>	<p>3 200 [2-4-6]</p>  <p>0.191 EQC: 22</p>	<p>3 202 [1-3-8]</p>  <p>0.139 EQC: 11</p>	<p>3 203 [3-1-2]</p>  <p>0.995 EQC: 39</p>	<p>3 204 [3-1-4]</p>  <p>0.805 EQC: 40</p>
<p>3 205 [3-1-9]</p>  <p>0.012 EQC: 8</p>	<p>3 206 [3-7-17]</p>  <p>0.208 EQC: 6</p>	<p>3 209 [2-4-8]</p>  <p>0.004 EQC: 41</p>	<p>3 210 [1-2-4]</p>  <p>0.984 EQC: 40</p>	<p>3 211 [1-2-5]</p>  <p>0.487 EQC: 7</p>
<p>3 212 [1-2-11]</p>  <p>0.158 EQC: 8</p>	<p>3 213 [1-3-12]</p>  <p>0.161 EQC: 8</p>	<p>3 216 [2-4-10]</p>  <p>0.005 EQC: 24</p>	<p>3 217 [2-1-3]</p>  <p>0.987 EQC: 40</p>	<p>3 219 [2-1-9]</p>  <p>0.162 EQC: 8</p>

<p>3 234 [2-1-3]</p>  <p>0.993 EQC: 42</p>	<p>3 235 [2-1-4]</p>  <p>0.406 EQC: 10</p>	<p>3 236 [2-6-14]</p>  <p>0.200 EQC: 6</p>	<p>3 237 [2-1-4]</p>  <p>0.122 EQC: 23</p>	<p>3 239 [2-1-3]</p>  <p>0.886 EQC: 40</p>
<p>3 240 [2-1-4]</p>  <p>0.982 EQC: 39</p>	<p>3 242 [2-5-9]</p>  <p>0.427 EQC: 7</p>	<p>3 243 [2-5-13]</p>  <p>0.189 EQC: 6</p>	<p>3 246 [3-5-11]</p>  <p>0.012 EQC: 41</p>	<p>3 247 [2-1-3]</p>  <p>0.973 EQC: 40</p>
<p>3 249 [2-5-8]</p>  <p>0.484 EQC: 7</p>	<p>3 250 [2-5-13]</p>  <p>0.087 EQC: 6</p>	<p>3 254 [3-1-2]</p>  <p>0.991 EQC: 7</p>	<p>3 255 [3-1-4]</p>  <p>0.447 EQC: 40</p>	<p>3 257 [3-6-9]</p>  <p>0.306 EQC: 7</p>
<p>3 259 [3-1-12]</p>  <p>0.019 EQC: 21</p>	<p>3 260 [3-5-7]</p>  <p>0.000 EQC: 23</p>	<p>3 262 [2-4-6]</p>  <p>0.897 EQC: 43</p>	<p>3 263 [2-4-14]</p>  <p>0.010 EQC: 15</p>	<p>3 264 [1-2-4]</p>  <p>0.913 EQC: 42</p>
<p>3 265 [1-2-5]</p>  <p>0.065 EQC: 10</p>	<p>3 266 [1-3-7]</p>  <p>0.496 EQC: 7</p>	<p>3 268 [8-1-2]</p>  <p>0.565 EQC: 44</p>	<p>3 271 [5-7-9]</p>  <p>0.045 EQC: 1</p>	<p>3 272 [5-3-1]</p>  <p>0.208 EQC: 7</p>
<p>3 277 [6-4-2]</p>  <p>0.329 EQC: 1</p>	<p>3 278 [3-2-4]</p>  <p>0.049 EQC: 3</p>	<p>3 283 [3-5-6]</p>  <p>0.070 EQC: 7</p>	<p>3 284 [3-5-9]</p>  <p>0.018 EQC: 6</p>	<p>3 285 [4-1-3]</p>  <p>0.015 EQC: 46</p>
<p>3 286 [8-6-4]</p>  <p>0.257 EQC: 1</p>	<p>3 291 [5-8-10]</p>  <p>0.000 EQC: 7</p>	<p>3 293 [3-1-2]</p>  <p>0.271 EQC: 7</p>		

D.2.2. Oniumreaktionen

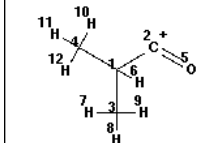
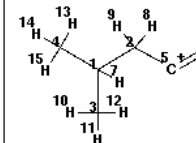
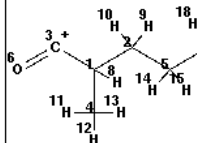
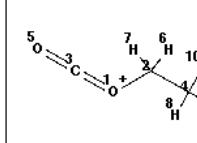
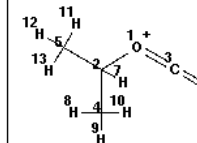
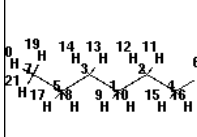
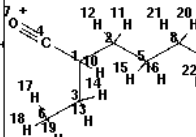
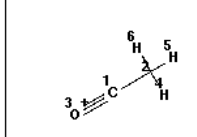
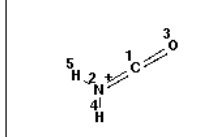
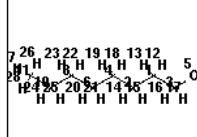
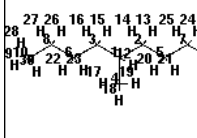
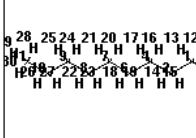
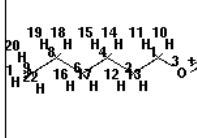
Beschriftung siehe Seite 253.

 6 2 [1-2-10-4] 0.000 EQC: 1	 6 3 [1-2-13-5] 0.999 EQC: 1	 6 4 [1-2-16-6] 0.969 EQC: 2	 6 5 [1-3-11-4] 0.448 EQC: 3	 6 6 [1-3-14-5] 0.956 EQC: 3	 6 7 [2-1-15-5] 0.975 EQC: 4
 6 9 [2-1-12-4] 0.894 EQC: 5	 6 10 [1-4-18-6] 0.499 EQC: 6	 6 11 [1-4-21-7] 0.756 EQC: 6	 6 12 [1-3-11-4] 0.579 EQC: 7	 6 13 [1-3-14-5] 0.758 EQC: 7	 6 14 [1-3-11-4] 0.717 EQC: 5
 6 15 [1-3-14-5] 0.926 EQC: 5	 6 16 [2-4-18-6] 0.755 EQC: 7	 6 17 [2-5-21-7] 0.863 EQC: 7	 6 18 [1-3-12-4] 0.468 EQC: 7	 6 19 [1-3-15-5] 0.799 EQC: 7	 6 24 [2-4-17-6] 0.857 EQC: 5
 6 25 [2-5-20-7] 0.997 EQC: 5	 6 26 [1-4-20-6] 0.088 EQC: 6	 6 27 [1-4-24-7] 0.705 EQC: 6	 6 28 [3-1-10-2] 0.983 EQC: 8	 6 30 [4-8-27-10] 0.982 EQC: 1	 6 31 [1-2-14-5] 0.983 EQC: 1
 6 32 [1-3-14-6] 0.179 EQC: 1	 6 33 [1-3-17-7] 0.896 EQC: 1	 6 34 [2-4-15-6] 0.000 EQC: 5	 6 35 [3-6-21-8] 0.948 EQC: 1	 6 36 [2-4-11-5] 0.697 EQC: 1	 6 37 [2-5-14-6] 0.854 EQC: 1
 6 40 [2-4-12-5] 0.208 EQC: 1	 6 41 [2-4-16-6] 0.998 EQC: 5	 6 42 [1-1-13-5] 0.998 EQC: 2	 6 43 [2-4-20-7] 0.604 EQC: 5	 6 45 [3-6-27-10] 0.52 EQC: 1	 6 46 [2-1-11-4] 0.999 EQC: 18
 6 47 [2-1-11-4] 0.999 EQC: 18	 6 48 [2-4-17-6] 0.626 EQC: 5	 6 49 [2-1-11-4] 0.999 EQC: 1	 6 50 [2-1-12-4] 0.977 EQC: 1	 6 51 [3-1-16-2] 0.998 EQC: 1	 6 52 [3-1-17-2] 0.973 EQC: 1
 6 54 [1-3-19-6] 0.843 EQC: 5	 6 55 [4-7-27-9] 0.903 EQC: 1	 6 56 [1-2-19-6] 0.999 EQC: 2	 6 57 [1-2-22-6] 0.919 EQC: 2	 6 58 [1-3-16-6] 0.877 EQC: 1	 6 59 [5-3-13-1] 0.959 EQC: 1

<p>6 60 [3-1-13-2]</p>  <p>0.068 EQC: 1</p>					
--	--	--	--	--	--

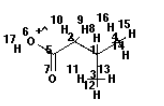
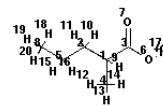
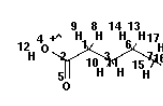
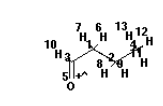
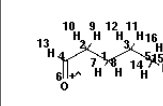
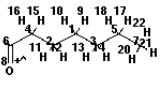
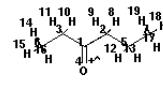
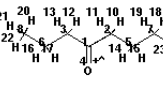
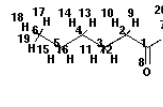
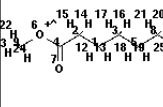
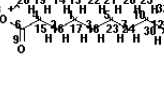
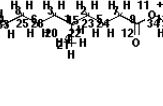
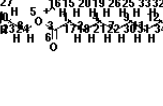
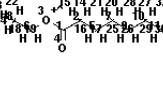
D.2.3. Carbonyleliminierungen

Beschriftung siehe Seite 253.

<p>7 3 [5-2-1]</p>  <p>0.948 EQC: 1</p>	<p>7 4 [6-5-2]</p>  <p>0.834 EQC: 2</p>	<p>7 5 [6-3-1]</p>  <p>0.995 EQC: 1</p>	<p>7 8 [5-3-1]</p>  <p>0.972 EQC: 3</p>	<p>7 9 [6-3-1]</p>  <p>0.305 EQC: 3</p>
<p>7 12 [8-6-4]</p>  <p>0.998 EQC: 4</p>	<p>7 13 [7-4-1]</p>  <p>1.000 EQC: 5</p>	<p>7 14 [3-1-2]</p>  <p>0.252 EQC: 4</p>	<p>7 15 [3-1-2]</p>  <p>0.048 EQC: 6</p>	<p>7 17 [9-7-5]</p>  <p>0.509 EQC: 3</p>
<p>7 20 [11-9-7]</p>  <p>0.950 EQC: 2</p>	<p>7 21 [5-3-1]</p>  <p>0.057 EQC: 2</p>	<p>7 22 [7-5-3]</p>  <p>0.990 EQC: 3</p>		

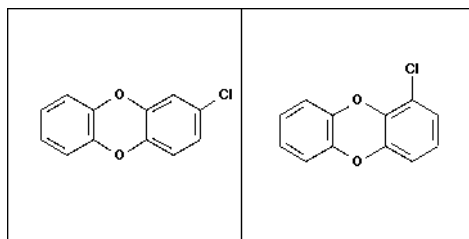
D.2.4. Mc Lafferty-Reaktionen

Beschriftung siehe Seite 253.

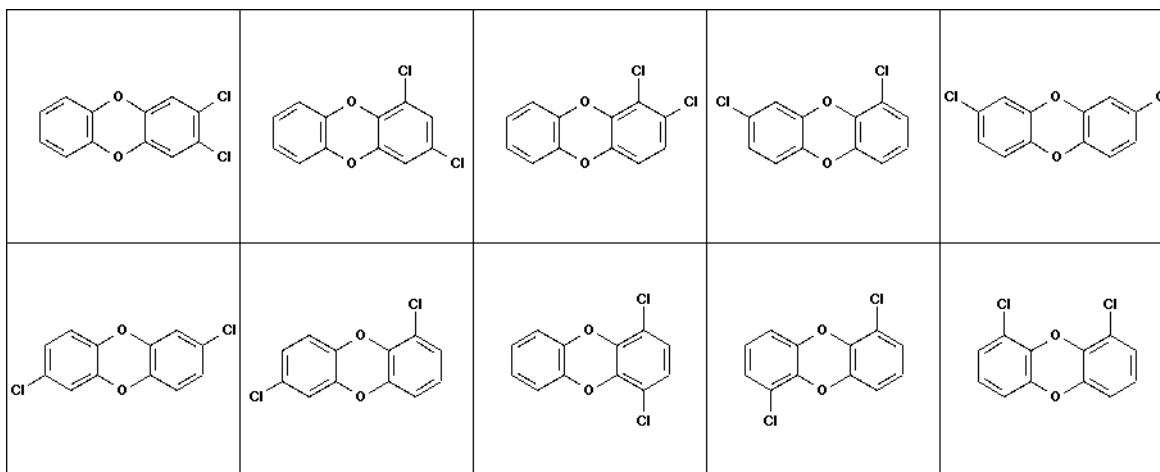
<p>9 2 [7-5-2-1-3-11]</p>  <p>0.544 EQC: 1</p>	<p>9 3 [7-3-1-2-5-15]</p>  <p>0.894 EQC: 1</p>	<p>9 4 [5-2-1-3-6-13]</p>  <p>0.856 EQC: 1</p>	<p>9 5 [5-3-1-2-4-11]</p>  <p>0.295 EQC: 2</p>	<p>9 6 [6-4-2-1-3-11]</p>  <p>0.870 EQC: 2</p>
<p>9 7 [8-6-4-2-1-9]</p>  <p>0.849 EQC: 2</p>	<p>9 9 [4-1-2-5-7-17]</p>  <p>0.679 EQC: 2</p>	<p>9 11 [4-1-3-6-8-20]</p>  <p>0.002 EQC: 2</p>	<p>9 12 [8-1-2-3-4-13]</p>  <p>0.805 EQC: 3</p>	<p>9 13 [7-4-2-1-3-16]</p>  <p>0.873 EQC: 1</p>
<p>9 15 [9-6-4-2-1-13]</p>  <p>0.787 EQC: 1</p>	<p>9 16 [12-9-7-5-2-16]</p>  <p>0.829 EQC: 1</p>	<p>9 17 [6-3-1-2-4-19]</p>  <p>0.800 EQC: 1</p>	<p>9 19 [4-1-2-5-7-20]</p>  <p>0.798 EQC: 1</p>	

E. Chlorierte Dioxine

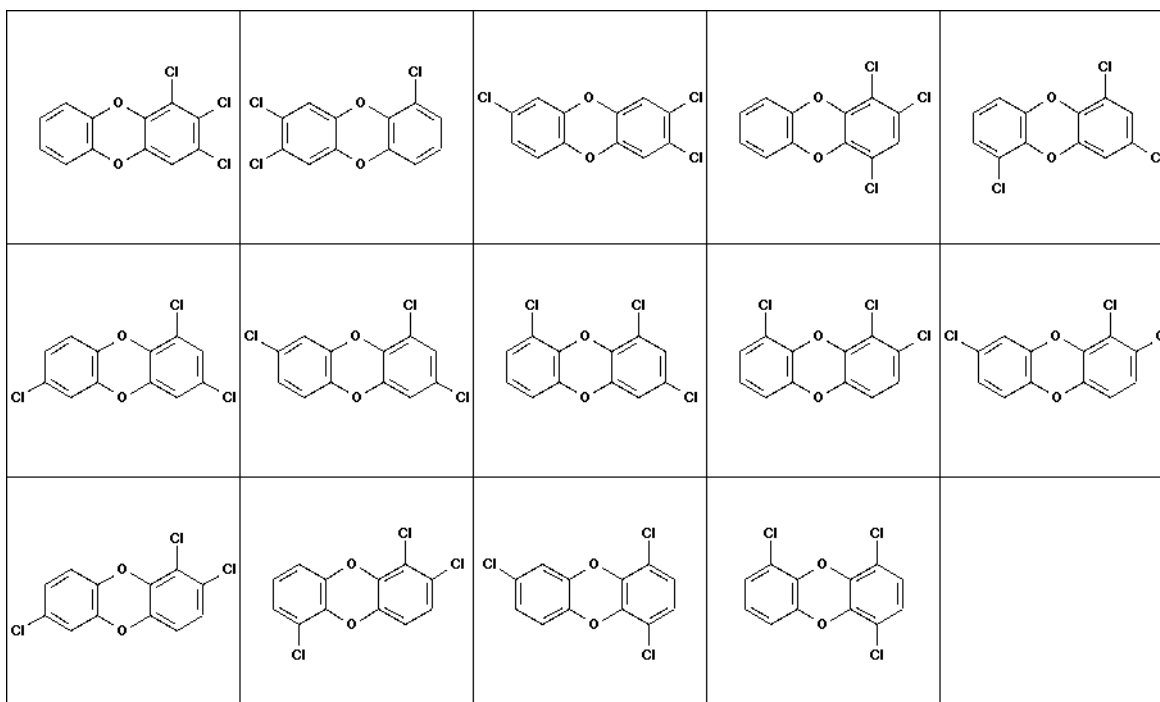
E.1. Monochlordioxine



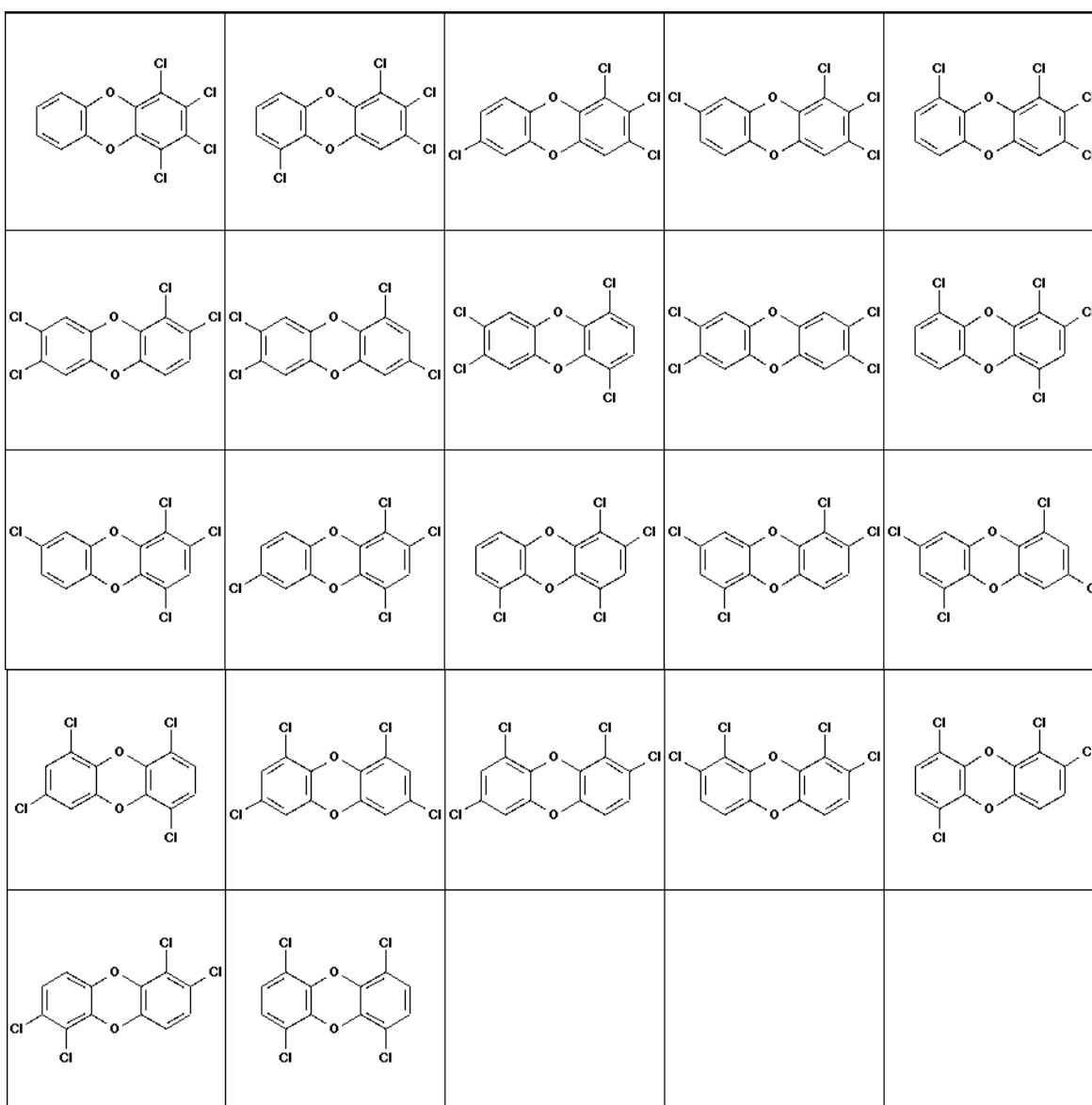
E.2. Dichlordioxine



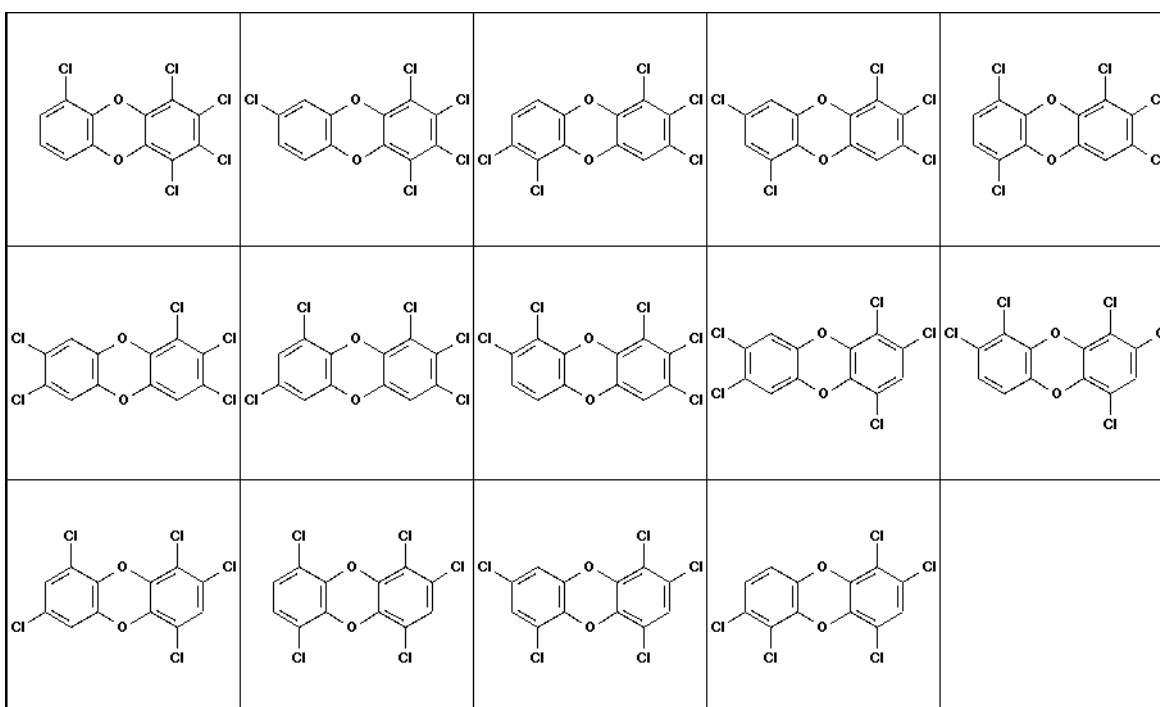
E.3. Trichlordioxine



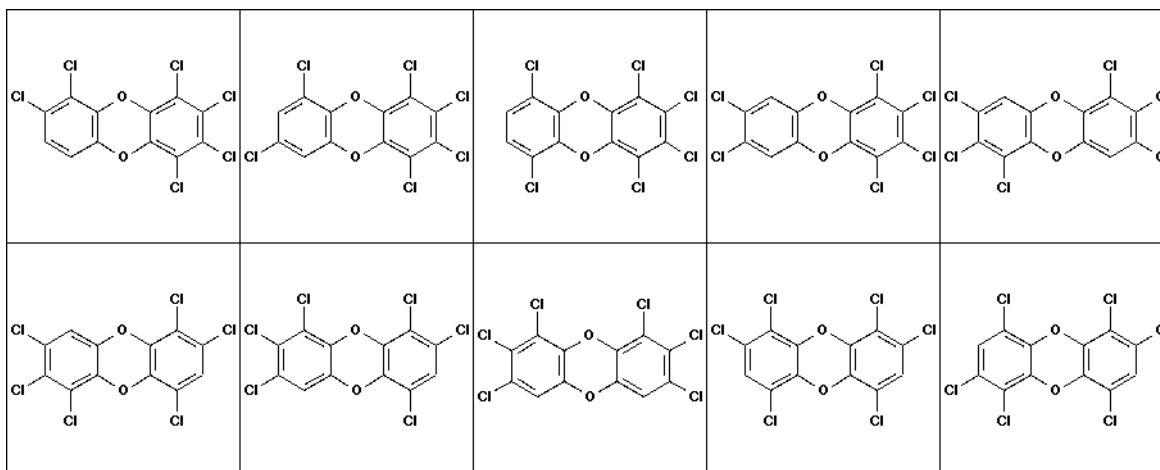
E.4. Tetrachlordioxine



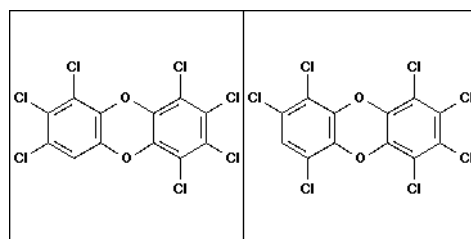
E.5. Pentachlordioxine



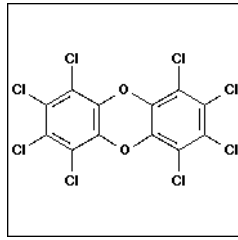
E.6. Hexachlordioxine



E.7. Heptachlordioxine



E.8. Octachlordioxin



F. Verwendete Regeldateien

F.1. Regeln für den Abbau von s-Triazinen mit EROS7

Die in Abschnitt 7.1.3.1 eingesetzten und im folgenden aufgeführten Tcl-Regeln wurden von T. Kostka geschrieben [62], um den Abbau von s-Triazinen zu simulieren. Sie sind hier aufgeführt, um eine ganze Regeldatei als Beispiel für Tcl-Regeln zu zeigen. Im Teil *RULE_INFO* der ersten Regel werden aus der KMAP-Datei trained.knet die Gewichte und aus der Datei test3-1+0.ctxeda die Datenpunkte vom Training mit KMAP [86] eingelesen. In der Reaktionsfunktion *FUNC* werden die Werte dieses Netzes abgefragt.

```
# Program rules.tcl

proc Tcl_rule {} {

#####
# Procedure : Tcl_rule
# Arguments :
#           irule - rule number
#           op - part of the used rule
#           sub_op - number of the model function
# Returns :
#           OK , BAD
#
#####

    global irule op sub_op rule_trace
    global name
    global kpl c_h2o c_h2
    global name date sconc kin_mode conversion_limit reaction_time
    global minimal_concentration
    global phases_p_r phase_contacts phprop comp_edus in_phase phase
    global attrib_1 center_1 rule_1
    global attrib_2 center_2 rule_2
    global attrib_3 center_3 rule_3
    global attrib_4 center_4 rule_4
    switch $irule {

        # Regelkopf

        GLOBAL {
            switch $op {
                # Einstellungen der Reaktoren, Phasen, Kinetik
                INIT_RULES {
                    set c_h2o 1.0
                    set c_h2 1.0
                    set name „nt_tria09.tcl“
                    putco name name
                    set date 27.06.1997
                    get temperature temp
                    print „temperature $temp“
                    putco date date
                    # reactors and phases
                    set in_phase(0) 0
                    set comp_edus(0) 1
                    set phases_p_r {1,0}
                    set phase_contacts(0) 0
                    set phprop(0) „$MONOMOLEC“
                    putco phases_per_reactor phases_p_r
                    putco phase_contacts phase_contacts
                }
            }
        }
    }
}
```

```

    putco phase_property phprop
    putco use_all_educts_together comp_edus
    putco input_phase_for_reactors in_phase
    set phase 1
    putco output_phase phase
    # kinetic
    set sconc(0) 0.10
    set conversion_limit 0.0
    set reaction_time 10368.e3
    set minimal_concentration 1.e-10
    set kin_mode(0) gear
    putco kinetic_model kin_mode
    putco conversion_limit conversion_limit
    putco reaction_time reaction_time
    putco minimal_concentration minimal_concentration
    put start_conc sconc
    # give rule variables to kernel
    set kp1 0.0
    put reactivity kp1
    print „test $test trace $trace“
    print „INIT_RULES called“
    return OK
}
# Vorbehandlung der Ausgangsmaterialien
PREP_ROOT {
    print „PREP_ROOT called . „
    return OK
}
# Vorbehandlung der Edukte
PREP_EDUCT {
    print „PREP_EDUCT called .“
    return OK
}
# Verteilungsfunktion
DISTRIBFUNC {
    return OK
}
# Abschlußfunktion
FINISH {
    return OK
}
default {
    return BAD
}
}
}

# Allgemein Hydrolyseregel

RULE_1 {
    switch $op {
    # Definition des Namen, der Attribute und der RSS
    RULE_INFO {
        set rule_1 „Allg. Hydrolyse“
        putco rule_name rule_1
        set attrib_1(0) NULL
        putco attributes attrib_1
        set center_1 {1,2,0,2,3,0,0}
        putco center_connectivity center_1
        set ff [net cpg init trained.knet test3-1+0.ctxeda]
        if {!$ff} {
            print „Rule 1: error during the net initialization „
        }
    }
    }
}

```

```

    return OK
}
# Überprüfung der Bedingungen für die RSS
CONSTR {
  switch $sub_op {
  GLOBAL {
    return OK
  }
  # für das Eduktensemble
  0 {
    if {$trace>2} {
      print „Rule 1: Checking total ensemble „
      print „trace=$trace“
    }
    set blt [prop E_N_ATOMS na]
    if {!$blt} {
      print „error getting property E_N_ATOMS“
    }
    set blt [prop E_LAST_ATOM il]
    if {!$blt} {
      print „error getting property E_LAST_ATOM „
    }
    set sar 0
    for {set i 1} {$i<=$il} {incr i} {
      set blt [prop A_AROMATIC $i aromat]
      if {$blt} {
        if {$aromat} {incr sar}
      }
    }
    set ar $sar
    if {$ar!=6&&$na>7} {return BAD}
    return OK
  }
  # für das erste Atom in der RSS
  1 {
    if {$trace>2} {
      print „Checking atom 1 rule 1 „
    }
    set atom [center 1]
    set blt [prop A_ELEMENT $atom elem]
    if {!$blt} {
      print „error getting property A_ELEMENT $atom“
    }
    if
    {$selem!=6&&$selem!=8&&$selem!=7&&$selem!=9&&$selem!=17&&$selem!=35&&$selem!=35&&$selem!=16}
    {return BAD}
    return OK
  }
  # für das zweite Atom in der RSS
  2 {
    if {$trace>2} {
      print „Checking atom 2 rule 1 „
    }
    set atom [center 2]
    set blt [prop A_ELEMENT $atom elem]
    if {!$blt} {
      print „error getting property A_ELEMENT $atom“
    }
    if {$selem!=6} {return BAD}
    return OK
  }
  3 {
    if {$trace>2} {

```

```

    print „Checking atom 3 rule 1 „
  }
  set atom1 [center 1]
  set atom2 [center 2]
  set atom3 [center 3]
  set blt [prop A_ELEMENT $atom1 elem1]
  if {!$blt} {
    print „error getting property A_ELEMENT $atom1“
  }
  set blt [prop A_ELEMENT $atom2 elem2]
  if {!$blt} {
    print „error getting property A_ELEMENT $atom2“
  }
  set blt [prop A_ELEMENT $atom3 elem3]
  if {!$blt} {
    print „error getting property A_ELEMENT $atom1“
  }
  if {$elem3!=6&&$elem3!=8&&$elem3!=1&&$elem3!=16&&$elem3!=7} {return BAD}
  set blt [prop A_ENSIG $atom2 ensig2]
  if {!$blt} {
    print „error getting property A_ENSIG $atom2“
  }
  set blt [prop A_NEIGHBORS $atom1 ng]
  if {!$blt} {
    print „error getting property A_NEIGHBORS $atom1“
  }
  set l1 [array size ng]
  set nhneigh1 0
  for {set i 0} {$i<$l1} {incr i} {
    set blt [prop A_ELEMENT $ng($i) j]
    if {!$blt} {
      print „error getting property A_ELEMENT $ng($i)“
    }
    if {$j!=1} {incr nhneigh1}
  }
  set blt [prop B_BOORD $atom1 $atom2 bor1]
  if {!$blt} {
    print „error getting property B_BOORD $atom1 $atom2“
  }
  set blt [prop B_BOORD $atom2 $atom3 bor2]
  if {!$blt} {
    print „error getting property B_BOORD $atom2 $atom3“
  }
  set blt [prop A_AROMATIC $atom1 ar1]
  if {!$blt} {
    print „error getting property A_AROMATIC $atom1“
  }
  set blt [prop A_AROMATIC $atom2 ar2]
  if {!$blt} {
    print „error getting property A_AROMATIC $atom2“
  }
  if {$trace>3} {
    print „Rule 1: e1=$elem1 e3=$elem3 ensig=$ensig2 bor1=$bor1 ar1=$ar1
ar2=$ar2 nhneigh1=$nhneigh1 „
  }
  if {$elem1==7&&$elem3==6&&$ar2!=0} {return BAD}
  if {$elem1==8&&$bor1>1} {return BAD}
  if {$bor1==1&&$elem1==7&&$elem3==8&&$ensig2<10.1} {return BAD}
  if {$elem1==7&&$elem3==7&&$ar2!=0&&$nhneigh1>1} {return BAD}
  if {$elem1==6&&$elem2==6} {return BAD}
  if {$ar1!=0} {return BAD}
  if {$elem1==8&&$nhneigh1==1} {return BAD}
  if {$elem1==7&&$ar2==0&&$elem3!=8&&$elem3!=7} {return BAD}

```

```

    return OK
}
default {
    print „illegal number of atoms“
    return BAD
}
}
}
# Reaktionsfunktion mit Festlegung der reaktivität
FUNC {
    # Eduktfunktion
    if {$strace>2} {
        print „Rule 1: FUNC called „
    }
    set atom1 [center 1]
    set atom2 [center 2]
    set atom3 [center 3]
    set ff [net cpg set educt]
    if {!$ff} {
        print „Rule 1: net cpg set educt failed.“
    }
    # Durchführung der Reaktion
    set h1 [new_atom 1 elsys]
    set h2 [new_atom 1 elsys]
    set o [new_atom 8 elsys]
    change_bond_order $atom2 $atom1 -1
    change_bond_order $h1 $o 1
    change_bond_order $h2 $atom1 1
    change_bond_order $o $atom2 1
    # Produktfunktion
    set fff [net cpg set product]
    if {!$fff} {
        print „net cpg set product failed.“
    }
    if {$ff&&$fff} {
        set f1 [net cpg get R1]
        if {$f1} {
            set ar [array exist R1]
            if {$ar} {
                set dim [array size R1]
                for {set i 0} {$i<$dim} {incr i} {
                    print „R1[$i]=$R1[$i]“
                }
            } else {
                print „R1=$R1“
            }
        } else {
            print „error getting the net result“
        }
    } else {
        print „error setting the net values“
    }
    # Überprüfung, ob wirklich alles paßt
    set blt [prop A_ELEMENT $atom1 elem]
    if {!$blt} {
        print „error getting property A_ELEMENT $atom1“
    }
    set blt [prop A_ELEMENT $atom3 elem3]
    if {!$blt} {
        print „error getting property A_ELEMENT $atom3“
    }
    set blt [prop B_BOORD $atom1 $atom2 bor1]
    if {!$blt} {

```

```

    print „error getting property B_BOORD $atom1 $atom2“
  }
set blt [prop A_AROMATIC $atom2 ar2]
if {!$blt} {
  print „error getting property A_AROMATIC $atom2“
}
if {$selem==7&&$ar2==0&&$bor1==0&&$elem3!=8&&$elem3!=7} {return BAD}
# konjugiere pi-Systeme
set blt [prop A_ELECSYSS $atom2 esys2]
if {!$blt} {
  print „error getting property A_ELECSYSS $atom2“
}
set blt [prop A_ELECSYSS $o esyso]
if {!$blt} {
  print „error getting property A_ELECSYSS $o“
}
set picount1 0
set dim [array size esys2]
for {set i 0} {$i<$dim} {incr i} {
  set blt [prop EL_IS_SIGMA $esys2($i) is_sigma]
  if {!$blt} {
    print „error getting property EL_IS_SIGMA $esys2($i)“
  }
  if {!$is_sigma} {
    set pis2 $esys2($i)
    incr picount1
  }
}
set picount2 0
set dim [array size esyso]
for {set i 0} {$i<$dim} {incr i} {
  set blt [prop EL_IS_SIGMA $esyso($i) is_sigma]
  if {!$blt} {
    print „error getting property EL_IS_SIGMA $esyso($i)“
  }
  if {!$is_sigma} {
    set piso $esyso($i)
    incr picount2
  }
}
if {$picount1&&$picount2} {
  set blt [combine_elsys $pis2 $piso es_pi]
  if {!$blt} {
    print „combine_elsys $pis2 $piso“
  }
  print „combine c-o“
}
# the end
set kp1 [expr 5.e-9*$c_h2o]
set symmetry_factor [expr $symmetry_factor*2]
if {$trace>2} {
  print „symmetry_factor=$symmetry_factor“
}
return OK
}
default {
  return BAD
}
}
}
}

```

Reaktionstyp der reduktiven Dealkylierung

```

RULE_2 {
  switch $op {
    # Name, Attribute, RSS
    RULE_INFO {
      set rule_2 „Reduktive Dealkylierung“
      putco rule_name rule_2
      set attrib_2(0) NULL
      putco attributes attrib_2
      set center_2 {1,2,0,0}
      putco center_connectivity center_2
      return OK
    }
    # Bedingungen der RSS
    CONSTR {
      switch $sub_op {
        GLOBAL {
          return OK
        }
        0 {
          if {$trace>2} {
            print „Checking total ensembles rule 2 „
          }
          set blt [prop E_LAST_ATOM il]
          if {!$blt} {
            print „error getting property E_LAST_ATOM „
          }
          set sar 0
          for {set i 1} {$i<=$il} {incr i} {
            set blt [prop A_AROMATIC $i aromat]
            if {$blt} {
              if {$aromat} {incr sar}
            }
          }
          set ar $sar
          if {$ar!=6} {return BAD}
          return OK
        }
        1 {
          if {$trace>2} {
            print „Checking atom 1 rule 2“
          }
          set atom [center 1]
          set blt [prop A_ELEMENT $atom elem]
          if {!$blt} {
            print „error getting property A_ELEMENT $atom“
          }
          if {$elem!=7} {return BAD}
          set blt [prop A_AROMATIC $atom ar]
          if {!$blt} {
            print „error getting property A_AROMATIC $atom“
          }
          if {$sar!=0} {return BAD}
          return OK
        }
        2 {
          if {$trace>2} {
            print „Checking atom 2 rule 2“
          }
          set atom [center 2]
          set blt [prop A_ELEMENT $atom elem]
          if {!$blt} {
            print „error getting property A_ELEMENT $atom“
          }
        }
      }
    }
  }
}

```

```

    if {$selem!=6} {return BAD}
    set blt [prop A_AROMATIC $atom ar]
    if {!$blt} {
        print „error getting property A_AROMATIC $atom“
    }
    if {$sar!=0} {return BAD}
    return OK
}
default {
    print „illegal number of atoms“
}
}
}
# Reaktionsfunktion
FUNC {
    # Eduktfunktion
    if {$strace>2} {
        print „FUNC called Rule 2“
    }
    # Durchführung der Reaktion
    set atom1 [center 1]
    set atom2 [center 2]
    set h1 [new_atom 1 elsys]
    set h2 [new_atom 1 elsys]
    change_bond_order $atom2 $atom1 -1
    change_bond_order $h1 $atom1 1
    change_bond_order $h2 $atom2 1
    # Produktfunktion
    set blt [prop A_NEIGHBORS $atom2 ng]
    if {!$blt} {
        print „error getting property A_AROMATIC $atom“
    }
    set ll [array size ng]
    set nhneigh 0
    for {set i 0} {$i<$ll} {incr i} {
        set blt [prop A_ELEMENT $ng($i) j]
        if {!$blt} {
            print „error getting property A_AROMATIC $atom“
        }
        if {$j!=1} {incr nhneigh}
    }
    set kp1 [expr $c_h2*4.42e-8*3./(2.+$nhneigh)]
    set symmetry_factor [expr $symmetry_factor*2]
    if {$strace>2} {
        print „symmetry_factor=$symmetry_factor kp1=$kp1“
    }
    return OK
}
default {
    return BAD
}
}
}

```

kombinierte Hydrolyse/Debarboxylierung für die Abbauprodukte der Cyanursäure

```

RULE_4 {
    switch $op {
        # Name, Attribute, RSS
        RULE_INFO {
            set rule_4 „Decarboxylierung“
            putco rule_name rule_4
            set attrib_4(0) NULL
        }
    }
}

```

```

putco attributes attrib_4
set center_4 {1,2,0,2,3,0,3,4,0,4,5,0,5,6,0,4,7,0,0}
putco center_connectivity center_4
return OK
}
# Bedingungen der RSS
CONSTR {
  switch $sub_op {
    GLOBAL {
      return OK
    }
    0 {
      return OK
    }
    1 {
      if {$strace>2} {
        print „Checking atom 1 rule 4 „
      }
      set atom [center 1]
      set blt [prop A_ELEMENT $atom elem]
      if {!$blt} {
        print „error getting property A_ELEMENT $atom“
      }
      if {$selem!=7} {return BAD}
      return OK
    }
    2 {
      if {$strace>2} {
        print „Checking atom 2 rule 4 „
      }
      set atom [center 2]
      set blt [prop A_ELEMENT $atom elem]
      if {!$blt} {
        print „error getting property A_ELEMENT $atom“
      }
      if {$selem!=6} {return BAD}
      return OK
    }
    3 {
      if {$strace>2} {
        print „Checking atom 3 rule 4 „
      }
      set atom3 [center 3]
      set atom2 [center 2]
      set blt [prop A_ELEMENT $atom3 elem]
      if {!$blt} {
        print „error getting property A_ELEMENT $atom3“
      }
      if {$selem!=7} {return BAD}
      set blt [prop B_BOORD $atom2 $atom3 bor2]
      if {!$blt} {
        print „error getting property B_BOORD $atom2 $atom3“
        return BAD
      }
      if {$bor2<2} {return BAD}
      return OK
    }
    4 {
      if {$strace>2} {
        print „Checking atom 4 rule 4 „
      }
      set atom4 [center 4]
      set blt [prop A_ELEMENT $atom4 elem]

```

```

    if {!$blt} {
        print „error getting property A_ELEMENT $atom4“
    }
    if {$elem!=6} {return BAD}
    return OK
}
5 {
    if {$strace>2} {
        print „Checking atom 5 rule 4 „
    }
    set atom5 [center 5]
    set blt [prop A_ELEMENT $atom5 elem]
    if {!$blt} {
        print „error getting property A_ELEMENT $atom5“
    }
    if {$elem!=8} {return BAD}
    return OK
}
6 {
    if {$strace>2} {
        print „Checking atom 6 rule 4 „
    }
    set atom6 [center 6]
    set blt [prop A_ELEMENT $atom6 elem]
    if {!$blt} {
        print „error getting property A_ELEMENT $atom6“
    }
    if {$elem!=1} {return BAD}
    return OK
}
7 {
    if {$strace>2} {
        print „Checking atom 7 rule 4 „
    }
    set atom4 [center 4]
    set atom7 [center 7]
    set blt [prop A_ELEMENT $atom7 elem]
    if {!$blt} {
        print „error getting property A_ELEMENT $atom7“
    }
    if {$elem!=8} {return BAD}
    set blt [prop B_BOORD $atom4 $atom7 bor]
    if {!$blt} {
        print „error getting property B_BOORD $atom4 $atom7“
        return BAD
    }
    if {$bor<2} {return BAD}
    return OK
}
}
default {
    print „illegal number of atoms“
    return BAD
}
}
}
# Reaktionsfunktion
FUNC {
    # Eduktfunktion
    if {$strace>2} {
        print „FUNC called Rule 4“
    }
    set c1 [center 1]
    set c2 [center 2]

```



```

        if {$saromat} {incr sar}
    }
}
set ar $sar
if {$sar!=6} {return BAD}
return OK
}
1 {
  if {$strace>2} {
    print „Checking atom 1 rule 3 „
  }
  set atom [center 1]
  set blt [prop A_ELEMENT $atom elem]
  if {!$blt} {
    print „error getting property A_ELEMENT $atom“
  }
  if {$selem!=7} {return BAD}
  return OK
}
2 {
  if {$strace>2} {
    print „Checking atom 2 rule 3 „
  }
  set atom2 [center 2]
  set atom1 [center 1]
  set blt [prop A_ELEMENT $atom2 elem]
  if {!$blt} {
    print „error getting property A_ELEMENT $atom“
  }
  if {$selem!=6} {return BAD}
  set blt [prop B_BOORD $atom1 $atom2 bor1]
  if {!$blt} {
    print „error getting property B_BOORD $atom1 $atom2“
  }
  set blt [prop A_ENSIG $atom2 esig]
  if {!$blt} {
    print „error getting property A_ENSIG $atom2“
  }
  set blt [prop A_POLARIZ $atom2 apol]
  if {!$blt} {
    print „error getting property A_ELEMENT $atom2“
  }
  # if {$bor1<2} {return BAD}
  if {$esig<11.44||$esig>11.46} {return BAD}
  if {$apol<5.9||$apol>6.1} {return BAD}
  return OK
}
3 {
  if {$strace>2} {
    print „Checking atom 3 rule 3 „
  }
  set atom3 [center 3]
  set blt [prop A_ELEMENT $atom3 elem]
  if {!$blt} {
    print „error getting property A_ELEMENT $atom3“
  }
  if {$selem!=8} {return BAD}
  return OK
}
4 {
  if {$strace>2} {
    print „Checking atom 4 rule 3 „
  }
}

```

```

        set atom4 [center 4]
        set blt [prop A_ELEMENT $atom4 elem]
        if {!$blt} {
            print „error getting property A_ELEMENT $atom3“
        }
        if {$elem!=1} {return BAD}
        return 0
    }
    default {
        print „illegal number of atoms“
        return BAD
    }
}
# Reaktionsfunktion
FUNC {
    # Eduktfunktion
    if {$trace>2} {
        print „FUNC called rule 3 „
    }
    set atom1 [center 1]
    set atom2 [center 2]
    set atom3 [center 3]
    set atom4 [center 4]
    # Durchführung der Reaktion
    set h1 [new_atom 1 elsys]
    set h2 [new_atom 1 elsys]
    set o [new_atom 8 elsys]
    change_bond_order $h1 $o 1
    change_bond_order $atom1 $atom2 -2
    change_bond_order $atom3 $atom4 -1
    change_bond_order $h2 $atom1 1
    change_bond_order $atom1 $atom4 1
    change_bond_order $o $atom2 1
    change_bond_order $atom2 $atom3 1
    # Produktfunktion
    set kp1 [expr 1.e-6*$c_h2o]
    if {$trace>2} {
        print „symmetry_factor=$symmetry_factor kp1=$kp1“
    }
    return OK
}
default {
    return BAD
}
}

# end of the rules
default {
    print „illegal rule number“
    return BAD
}
}

# Eine zweite Regelfunktion

proc Test_rule {} {
    global_name

```

```

print „hier koennten weitere Regeln stehen“
return BAD
}

```

F.2. Regeln für die Simulation von Massenspektren mit EROS7

Im folgenden sind die Regeln in C++ angegeben, die für die Simulation von Massenspektren inklusive der Wasserstoffumlagerung verwendet wurden. Daß das bei der Wasserstoffumlagerung entstehende Radikal nicht mit einem möglichen, benachbarten π -System konjugiert wird, ist Absicht, da so auf den Mechanismus zur Verhinderung der doppelten Wasserstoffumlagerung auf eine funktionelle Gruppe verzichtet werden konnte, da die α -Spaltungen und Wasserstoffumlagerungen nur für nicht konjugierte Radikale zugelassen wurden. Dies ist ein einfacher Weg, dies für die verwendeten Beispiele zu erreichen, besitzt aber keine allgemeine Gültigkeit. Da diese Regeln ohnehin nur dazu geeignet sind, zu zeigen, daß so Massenspektren simuliert werden können, aber aus mehreren Gründen (siehe 6.10) nicht breiter eingesetzt werden können, wurde auf eine diesbezügliche Verbesserung verzichtet.

Wenn diese Ergänzung gewünscht wird, müßten die α -Spaltung und die Wasserstoffumlagerung auch für konjugierte Radikale ergänzt werden und entstehende Radikale nach Möglichkeit konjugiert werden. Für die Umlagerungszähler würden im Regelteil INIT_RULES entsprechende genealogische Variablen angemeldet, die auch zur Hashcodeberechnung herangezogen werden. Im Teil PREP_ROOT werden sie initialisiert und alle Reaktionstypen außer der Wasserstoffumlagerung werden sie zurücksetzen. Eine Wasserstoffumlagerung wird nur erlaubt, wenn der Zähler das Maximum von eins noch nicht erreicht hat und dann inkrementiert.

Die Regeln wurden überdies so geschrieben, daß die Wasserstoffumlagerungen an- und abgeschaltet werden können. Dies geschieht durch den Aufrufparameter `-V`. Mit `-Vhuml#1` übergibt EROS7 eine *int*-Variable unter dem Namen *huml* an die Regeln. Diese testen, ob diese Variable übergeben wurde, welchen Wert sie hat und aktiviert bzw. deaktiviert abhängig davon den Reaktionstyp der Wasserstoffumlagerung. Mit dieser Technik lassen sich Steuerungsmechanismen in die Regeln einbauen. Hierfür können beliebige Variablen der Typen *int* und *double* eingesetzt werden.

```

#include <iostream.h>
#include „rules.h“
#include „rh_ilist.h“
#include „nn_bpg.h“
#include <math.h>

int  get_iprop      (rx_eifc *eifc, const char* prop_name, int idx1=0, int idx2=0);
double IP          (rx_eifc *eifc, int iels);
int  lp_at_pis     (rx_eifc *eifc, int iel);
int  deconjugate_pis(rx_eifc *eifc, int atom1, int atom2);

RULEOK rules_cc (ostream &print, rx_eifc *eifc, rx_datex *val, int irule, OP op, int
sub_op, int test, int trace) {

static double      prob=0.;
static int const  *level=0;
static int        radic_els=0;
static const int  *huml=0;
static int        def_huml=0;

static nn_bpg nn_alpha, nn_onium, nn_co;

```

```

switch(irule) {
case GLOBAL:
  switch(op) {
  case INIT_RULES: {
    val->putco(„name“, „MS-rules for EROS7“);
    val->putco(„date“, „23.07.97“);
    static const int    input_phase[]={ 0 };
    static const int phases_of_reactor[]={ 1, 0 };
    static const int    phase_property[]={ MONOMOLEC };
    static const int    phase_contacts[]={ 0 };
    static const int    input_together[]={ 0 };
    val->putco(„input_phase_for_reactors“, input_phase);
    val->putco(„phases_per_reactor“,      phases_of_reactor);
    val->putco(„phase_property“,          phase_property);
    val->putco(„phase_contacts“,          phase_contacts);
    val->putco(„use_all_educts_together“, input_together);
    val->putco(„output_phase“,            1);
    static const char *const kin_mode[]={ „prob_kin“ };
    val->putco(„kinetic_model“,            kin_mode);
    val->putco(„minimal_concentration“,    1.e-5);
    val->put(„reactivity“,                  &prob);
    val->get(„level“,                       &level);
    val->get(„huml“,                         &huml);
    if (!(*huml)) huml=&def_huml;
    if (!nn_alpha.init(„rnnrul3x.dat“)) return BAD;
    if (!nn_onium.init(„rnnrul6zz.dat“)) return BAD;
    if (!nn_co.init(„rnnrul7z.dat“)) return BAD;
    return OK;}
  case PREP_ROOT:
    return OK;
  case PREP_EDUCT: {
    radic_els = 0;
    int iel, nel, nelec;
    eifc->prop(nel, „E_N_ELECSYSS“);
    for (iel = 1; iel<= nel; iel++) {
      eifc->prop(nelec, „EL_ELECTRONS“, iel);
      if (nelec%2) {radic_els = iel; break;}
    }
    if (*level == 1) return OK;
    int charge=get_iprop(eifc, „E_CHARGE“);
    if (charge == 0) return BAD;
    return OK;}
  case DISTRIBFUNC:
    return OK;
  case FINISH:
    return OK;
  default:
    return BAD;
  }
}

case 1:
  switch(op) {
  case RULE_INFO: {
    static const char *const attrib[]={ „rearrangement“, NULL };
    static const int    center[]={ 0 };
    val->putco(„rule_name“,                „Ionisation“);
    val->putco(„attributes“, attrib);
    val->putco(„center_connectivity“, center);
    return OK;}
  case CONSTR:

```

```

switch(sub_op) {
case 0: {
    if (get_iprop(eifc, „E_CHARGE“) return BAD;
    return OK;}
default:
    return BAD;
}
}
case FUNC: {
int iel, nel, nelec, igr, sgr;
rh_ilist<int> hgroup_idx;
vector<int> hgroup;
eifc->prop(nel, „E_N_ELECSYSS“);
for (iel = 1; iel <= nel; iel++) {
    eifc->set_active_ens(1);
    if (!get_iprop(eifc, „EL_IS_SIGMA“, iel)) {
        eifc->prop(hgroup, „EL_HGROUP“, iel);
        igr = hgroup[0];
        sgr = hgroup[1];
        eifc->prop(nelec, „EL_ELECTRONS“, iel);
        if ((!hgroup_idx.at_idx(igr)) && (nelec)) {
            if (!lp_at_pis(eifc, iel)) {
                prob = lell * exp(-2.*IP(eifc, iel));
                eifc->change_electrons(iel, -1);
                eifc->symmetry_factor = sgr; // bzw. = sgr
print << „Ioni: prob = „ << prob << endl;
                eifc->save_active_ens_as_product();
                hgroup_idx.append(igr);
            }
        }
    }
}
return OK;}
default:
    return BAD;
}
}

case 2:
switch(op) {
case RULE_INFO: {
    static const char *const attrib[]={ NULL };
    static const int center[]={ 1, 2, 0, 2, 3, 0, 0 };
    val->putco(„rule_name“, „Alpha cleavage“);
    val->putco(„attributes“, attrib);
    val->putco(„center_connectivity“, center);
    return OK;}
case CONSTR:
    switch(sub_op) {
    case 0: {
        if (!get_iprop(eifc, „E_RADICAL“) return BAD;
        vector<int> atoms;
        if (!radic_els) return BAD;
        eifc->prop(atoms, „EL_ATOMS“, radic_els);
        if (atoms.size() > 1) return BAD;
        return OK;}
    case 1:
        if (!get_iprop(eifc, „A_RADICCENTER“, eifc->center(1))) return BAD;
        return OK;
    case 2: {
        vector<int> els;
        eifc->prop(els, „N_ELECSYSS“, eifc->center(1), eifc->center(2));
        if (els.size() > 2) return BAD;
        if ((els[0] == radic_els) || (els[els.size()] == radic_els)) return BAD;

```

```

        return OK;}
    case 3: {
        //if (get_iprop(eifc, „A_ELEMENT“, eifc->center(3)) == 1) return BAD;
        vector<int> els;
        eifc->prop(els, „N_ELECSYSS“, eifc->center(2), eifc->center(3));
        if (els.size() > 2) return BAD;
        return OK;}
    default:
        return BAD;
}
case FUNC: {
    int issys, c1, c2, c3, iel1, iel2, imol, elem;
    double chrg;
    c1 = eifc->center(1);
    c2 = eifc->center(2);
    c3 = eifc->center(3);
    nn_alpha.set_vals(0, eifc);
    issys = deconjugate_pis(eifc, c2, c3);
    if (!issys) return BAD;
    eifc->split_elsy(issys, c2, c3, iel1, iel2, 1);
    eifc->combine_elsys(radic_els, iel1, es_pi);
    eifc->prop(imol, „A_MOLECULE“, c1);
    if (!get_iprop(eifc, „M_CHARGE“, imol)) return BAD;
    nn_alpha.set_vals(1, eifc);
    prob = nn_alpha.get_val();
    eifc->prop(elem, „A_ELEMENT“, c3);
    eifc->prop(chrg, „A_CHARGECENTER“, c1);
    if ((elem == 1) && (chrg < 0.1)) prob *= 0.06;
print << „alpha: prob = „ << prob << endl;
    if (prob < 0.05) return BAD;
    return OK;}
    default:
        return BAD;
}

```

```

case 3:
    switch(op) {
    case RULE_INFO: {
        static const char *const attrib[]={ NULL };
        static const int      center[]={ 1, 2, 0, 3, 4, 0, 0 };
        val->putco(„rule_name“,      „Onium reaction“);
        val->putco(„attributes“, attrib);
        val->putco(„center_connectivity“, center);
        return OK;}
    case CONSTR:
        switch(sub_op) {
        case 0:
            if (get_iprop(eifc, „E_RADICAL“)) return BAD;
            return OK;
        case 1: {
            int c1, elem, i, n, nel, nat, iel;
            double chrg;
            vector<int> els, atoms;
            c1 = eifc->center(1);
            eifc->prop(elem, „A_ELEMENT“, c1);
            if ((elem == 1) || (elem == 6)) return BAD;
            eifc->prop(chrg, „A_CHARGECENTER“, c1);
            if (chrg <= 0.) return BAD;
            eifc->prop(els, „A_ELECSYSS“, c1);
            n = els.size();
            for (i=0; i<n; i++) {
                iel = els[i];
            }
        }
        }
    }
}

```

```

        if (!get_iprop(eifc, „EL_IS_SIGMA“, iel)) {
            eifc->prop(atoms, „EL_ATOMS“, iel);
            nat = atoms.size();
            eifc->prop(nel, „EL_ELECTRONS“, iel);
            if ((nat > 1) && (nel > 1)) return OK;
        }
    }
    return BAD;}
case 2: {
    int c1=eifc->center(1), c2=eifc->center(2);
    if (get_iprop(eifc, „A_ELEMENT“, c2) != 6) return BAD;
    vector<int> els;
    eifc->prop(els, „N_ELECSYSS“, c1, c2);
    if (els.size() != 1) return BAD;
    return OK;}
case 3: {
    int c1=eifc->center(1), c2=eifc->center(2), c3=eifc->center(3), idist,
idist2;
    if (get_iprop(eifc, „A_ELEMENT“, c3) != 6) return BAD;
    eifc->prop(idist, „N_DIST_INDEX“, c2, c3);
    if (idist > 4) return BAD;
    eifc->prop(idist2, „N_DIST_INDEX“, c1, c3);
    if (idist2 < idist) return BAD;
    return OK;}
case 4:
    if (get_iprop(eifc, „A_ELEMENT“, eifc->center(4)) != 1) return BAD;
    return OK;
default:
    return BAD;
}
case FUNC: {
    int c1, c2, c3, c4;
    c1 = eifc->center(1);
    c2 = eifc->center(2);
    c3 = eifc->center(3);
    c4 = eifc->center(4);
    nn_onium.set_vals(0, eifc);
    eifc->change_bond_order(c1, c2, -1);
    eifc->change_bond_order(c3, c4, -1);
    eifc->change_bond_order(c1, c4, 1);
    eifc->change_bond_order(c2, c3, 1);
    nn_onium.set_vals(1, eifc);
    prob = 0.5 * nn_onium.get_val();
print << „onium: prob = „ << prob << endl;
    if (prob < 0.05) return BAD;
    return OK;}
default:
    return BAD;
}

```

```

case 4:
    switch(op) {
    case RULE_INFO: {
        static const char *const attrib[]={ NULL };
        static const int center[]={ 1, 2, 0, 2, 3, 0, 0 };
        val->putco(„rule_name“, „Carbonyl elimination“);
        val->putco(„attributes“, attrib);
        val->putco(„center_connectivity“, center);
        return OK;}
    case CONSTR:
        switch(sub_op) {
        case 0:

```

```

        if (radic_els) return BAD;
        return OK;
    case 1: {
        double chrg;
        int elem = get_iprop(eifc, „A_ELEMENT“, eifc->center(1));
        if ((elem != 8) && (elem != 7)) return BAD;
        eifc->prop(chrg, „A_CHARGECENTER“, eifc->center(1));
        if (chrg <= 0.) return BAD;
        return OK;}
    case 2: {
        int c1 = eifc->center(1), c2 = eifc->center(2);
        if (get_iprop(eifc, „A_ELEMENT“, c2) != 6) return BAD;
        vector<int> els;
        eifc->prop(els, „N_ELECSYSS“, c1, c2);
        if (els.size() != 3) return BAD;
        return OK;}
    case 3:
        if (get_iprop(eifc, „A_ELEMENT“, eifc->center(3)) == 1) return BAD;
        return OK;
    default:
        return BAD;
    }
}
case FUNC: {
    int issys, c2, c3, iel1, iel2;
    c2 = eifc->center(2);
    c3 = eifc->center(3);
    nn_co.set_vals(0, eifc);
    issys = deconjugate_pis(eifc, c2, c3);
    if (!issys) return BAD;
    eifc->split_elsy(issys, c2, c3, iel1, iel2, 2);
    nn_co.set_vals(1, eifc);
    prob = 0.5 * nn_co.get_val();
print << „co: prob = „ << prob << endl;
    if (prob < 0.05) return BAD;
    return OK;}
    default:
        return BAD;
    }
}

case 5:
    switch(op) {
    case RULE_INFO: {
        static const char *const attrib[]={ NULL };
        static const int      center[]={ 1,2,0, 2,3,0, 3,4,0, 4,5,0, 5,6,0, 0 };
        val->putco(„rule_name“,      „McLafferty reaction“);
        val->putco(„attributes“, attrib);
        val->putco(„center_connectivity“, center);
        return OK;}
    case CONSTR:
        switch(sub_op) {
        case 0: {
            if (!radic_els) return BAD;
            vector<int> atoms;
            eifc->prop(atoms, „EL_ATOMS“, radic_els);
            if (atoms.size() != 1) return BAD;
            return OK;}
        case 1: {
            int c1 = eifc->center(1), elem, i, n;
            eifc->prop(elem, „A_ELEMENT“, c1);
            if (elem != 8) return BAD;
            vector<int> els;
            eifc->prop(els, „A_ELECSYSS“, c1);

```

```

    n = els.size();
    for (i=0; i<n; i++) {
        if (els[i] == radic_els) return OK;
    }
    return BAD;}
case 2: {
    vector<int> els;
    eifc->prop(els, „N_ELECSYSS“, eifc->center(1), eifc->center(2));
    if (els.size() != 2) return BAD;
    return OK;}
case 3:
    if (get_iprop(eifc, „A_ELEMENT“, eifc->center(3)) != 6) return BAD;
    return OK;
case 4:
    return OK;
case 5:
    return OK;
case 6:
    if (get_iprop(eifc, „A_ELEMENT“, eifc->center(6)) != 1) return BAD;
    return OK;
default:
    return BAD;
}
case FUNC: {
    int i, n, c1, c2, c3, c4, c5, c6, ipis, iel1, iel2, iel3, iel4, nhn, issys;
    vector<int> v;
    c1 = eifc->center(1);
    c2 = eifc->center(2);
    c3 = eifc->center(3);
    c4 = eifc->center(4);
    c5 = eifc->center(5);
    c6 = eifc->center(6);
    nhn = 0;
    eifc->prop(v, „A_NEIGHBORS“, c5);
    n = v.size();
    for (i=0; i<n; i++) {
        if (get_iprop(eifc, „A_ELEMENT“, v[i]) != 1) nhn++;
    }
    issys = deconjugate_pis(eifc, c3, c4);
    if (!issys) return BAD;
    eifc->prop(v, „N_ELECSYSS“, c1, c2);
    n = v.size();
    for (i=0; i<n; i++) {
        if (!get_iprop(eifc, „EL_IS_SIGMA“, v[i])) ipis = v[i];
    }
    eifc->prop(v, „N_ELECSYSS“, c5, c6);
    eifc->split_elsy(v[0], c5, c6, iel1, iel2, 1);
    eifc->split_elsy(issys, c3, c4, iel3, iel4, 1);
    eifc->combine_elsys(radic_els, iel2, es_sigma);
    eifc->combine_elsys(iel1, iel4, es_pi);
    eifc->combine_elsys(ipis, iel3, es_pi);
    prob = 0.8361;
    if (nhn == 1) prob = 0.23;
print << „ml: prob = „ << prob << endl;
    if (prob < 0.05) return BAD;
    return OK;}
default:
    return BAD;
}
}
case 6:
    switch(op) {

```

```

case RULE_INFO: {
    static const char *const attrib[]={ „rearrangement“, NULL };
    static const int      center[]={ 2, 3, 0, 0 };
    val->putco(„rule_name“,      „Hydrogen rearrangement“);
    val->putco(„attributes“, attrib);
    val->putco(„center_connectivity“, center);
    return OK;}
case CONSTR:
    switch(sub_op) {
    case 0: {
        if (!(*huml)) return BAD;
        if (!radic_els) return BAD;
        vector<int> atoms;
        eifc->prop(atoms, „EL_ATOMS“, radic_els);
        if (atoms.size() != 1) return BAD;
        return OK;}
    case 1: {
        int c1 = eifc->center(1), i, n;
        vector<int> els;
        eifc->prop(els, „A_ELECSYSS“, c1);
        n = els.size();
        for (i=0; i<n; i++) {
            if (els[i] == radic_els) return OK;
        }
        return BAD;}
    case 2: {
        int idist, c1 = eifc->center(1), c2 = eifc->center(2);
        eifc->prop(idist, „N_DIST_INDEX“, c1, c2);
        if (get_iprop(eifc, „A_ELEMENT“, c1) == 6) {
            if ((idist < 3) || (idist > 5)) return BAD;
        } else {
            if (idist != 2) return BAD;
        }
        return OK;}
    case 3:
        if (get_iprop(eifc, „A_ELEMENT“, eifc->center(3)) != 1) return BAD;
        return OK;
    default:
        return BAD;
    }
case FUNC: {
    int iel1, iel2, c1, c2, c3, elem, idist;
    vector<int> els;
    c1 = eifc->center(1);
    c2 = eifc->center(2);
    c3 = eifc->center(3);
    elem = get_iprop(eifc, „A_ELEMENT“, c1);
    eifc->prop(idist, „N_DIST_INDEX“, c1, c2);
    eifc->prop(els, „N_ELECSYSS“, c2, c3);
    eifc->split_elsy(els[0], c2, c3, iel1, iel2, 1);
    eifc->combine_elsys(radic_els, iel2, es_sigma);
    if (elem == 6) {
        switch(idist) {
        case 3:
            prob = 0.1;
            break;
        case 4:
            prob = 0.3;
            break;
        case 5:
            prob = 0.05;
            break;
        default:

```

```

        return BAD;
    }
    } else {
        prob = 0.5;
    }
    eifc->symmetry_factor = 1; // oder nicht?
print << „h-uml: prob = „ << prob << endl;
    return OK;}
    default:
        return BAD;
    }

default:
    return BAD;

}

}

// -----

int get_iprop(rx_eifc *eifc, const char* prop_name, int idx1, int idx2) {
    int ivar=0;
    eifc->prop(ivar, prop_name, idx1, idx2);
    return ivar;
}

double IP(rx_eifc *eifc, int iel) {
    return 10.;
}

int lp_at_pis(rx_eifc *eifc, int iel) {
    int iel2, iat, nels, nat, i, j, nelec;
    if (get_iprop(eifc, „EL_IS_SIGMA“, iel)) return 0;
    vector<int> atoms, elsys, atoms2;
    eifc->prop(atoms, „EL_ATOMS“, iel);
    nat = atoms.size();
    if (nat == 1) return 0;
    for (i=0; i<nat; i++) {
        iat = atoms[i];
        eifc->prop(elsys, „A_ELECSYSS“, iat);
        nels = elsys.size();
        for (j=0; j<nels; j++) {
            iel2 = elsys[j];
            if (!get_iprop(eifc, „EL_IS_SIGMA“, iel2)) {
                eifc->prop(atoms2, „EL_ATOMS“, iel2);
                eifc->prop(nelec, „EL_ELECTRONS“, iel2);
                if ((nelec)&&(atoms2.size()==1)) return 1;
            }
        }
    }
    return 0;
}

int deconjugate_pis(rx_eifc *eifc, int atom1, int atom2) {
    int issys=0, i, n, iel1, iel2;
    vector<int> elsys, etyp;
    eifc->prop(elsys, „N_ELECSYSS“, atom1, atom2);
    n = elsys.size();
    etyp.resize(n);

```

```

for (i=0; i<n; i++) {
    etyp[i] = get_iprop(eifc, „EL_IS_SIGMA“, elsys[i]);
}
for (i=0; i<n; i++) {
    if (etyp[i]) {
        issys = elsys[i];
    } else {
        if (eifc->split_elsy(elsys[i], atom1, atom2, iel1, iel2, -1)) return 0;
    }
}
return issys;
}

```

Neuronale Netze zu diesen Regeln

Der Inhalt der folgenden drei Dateien sind die Gewichte, Skalierungen und Parameter für die Backpropagationnetze. Zum möglichst leichten Umstieg von EROS6 auf EROS7 wurde am Dateiformat für die Backpropagationnetze nur die Angabe der Namen der physikochemischen Parameter geändert. */NN_PARM* ist die Angabe der Parameter in VERGIL und wird von EROS7 nicht gelesen und ist nur aus Kompatibilitätsgründen enthalten. Die restlichen Abschnitte der Datei für die Backpropagationnetze sind identisch. Für EROS7 sind die unter */NN_PARM7* angegebenen Eigenschaften maßgebend. Die vier Zahlen dahinter sind die zwei Indizes, die auf null gesetzt sind, wenn sie nicht benötigt werden, eine Zahl für das Ensemble, von dem die Eigenschaft genommen werden soll (0 = Edukt, 1 = Produkt bzw. erstes Zwischenprodukt usw.). Die letzte gibt den Datentyp an (0 = ganzzahlig / *int*, 1 = Fließkommazahl / *double*). */NN_ARCHT*, */NN_CONNS* und */NN_SCAL* haben das gleiche Format wie die Netze von EROS6. */NN_ARCHT* folgt eine Zeile mit der Zahl der versteckten Schichten (hidden layers) und in der nächsten Zeile die Zahl der Neuronen für jede Schicht ohne Bias beginnend mit der Eingabeschicht. */NN_SCAL* gibt nach der fortlaufenden Numerierung die minimalen und maximalen Werte der Eigenschaft während des Trainings an. Die Eigenschaften zwischen diesen Werten werden von den Eingabeneuronen in einen Wertebereich von null bis eins skaliert. Diese Skalierung ist für alle Inputneuronen angegeben und wird von der Skalierung der Ausgabeneuronen gefolgt. Hier ist der Bereich angegeben, in den die Werte zwischen null und eins skaliert werden. Unter */NN_CONNS* sind sämtliche Gewichte des Netzes angegeben und zwar schichtenweise, beginnend mit den Gewichten zwischen der Eingabeschicht und der ersten versteckten Schicht. Die Gewichte kommen dann für ein Netz mit einem hidden layer und einem Ausgabeneuron in der Reihenfolge: input1 - hidden1, input2 - hidden1, ..., bias - hidden1, input1 - hidden2, ..., bias - hidden_last, hidden1 - output, ..., bias - output.

Netz für die α -Spaltung:

```

/NN_ARCHT      2    2
1
  12   3    1
/NN_PARM       16   16
PARAMETER EDUKT:
1    ATOM 1 ENSIG *
2    ATOM 1 FREEEL *
3    ATOM 1 NEIGHBOR *
4    BOND 2 3 TBDE *
5    ENSEMBLE EDELTAHF *
STOP
PARAMETER PRODUKT:
6    ATOM 2 APOLARIZABILITY *
7    ATOM 2 ENPI *

```

8 ATOM 2 QSIG *
9 ATOM 3 APOLARIZABILITY *
10 ATOM 3 QSIG *
11 BOND 2 1 PSTAB *
12 ENSEMBLE EDELTAHF *
END

/NN_PARM7 12 12
A_ENSIG 1 0 0 1
A_FREEEL 1 0 0 0
A_NEIGHBOR 1 0 0 0
B_TBDE 2 3 0 1
E_DELTAHF 0 0 0 1
A_POLARIZ 2 0 1 1
A_ENPI 2 0 1 1
A_QSIG 2 0 1 1
A_POLARIZ 3 0 1 1
A_QSIG 3 0 1 1
B_PSTAB 2 1 1 1
E_DELTAHF 0 0 1 1

/NN_SCAL 13 13
1 8.968300 15.304200
2 1.000000 5.000000
3 1.000000 3.000000
4 -92.822502 480.858002
5 257.174805 945.775330
6 2.125000 9.201000
7 6.641700 10.931600
8 -0.055000 0.552900
9 0.386000 6.782400
10 -0.252800 0.000000
11 0.000000 7.455200
12 -15.910600 1213.183960
13 0.000000 0.995000

/NN_CONNS 43 43
1.139383
1.473840
1.395796
-1.247808
-0.440537
-1.299575
-1.419531
3.151312
-2.142884
0.546493
-6.229643
-3.132250
-0.539434
-0.682565
-4.904122
2.436973
-0.518647
5.592857
4.013568
3.796338
2.633508
4.288496
-2.034354
0.601260
-3.548747
-6.780631
-0.523646
6.964950
-3.849618

```

-0.455457
-0.303718
 2.028105
-3.101313
 1.117309
 4.066958
 0.729223
-2.652695
-2.812138
-0.675825
 6.165740
-6.543190
-10.318967
 5.170370
/END      0      0

```

Netz für die Oniumreaktion:

```

/NN_ARCHT      2      2
 1
 11      2      1
/NN_PARM      15      15
PARAMETER EDUKT:
 1      ATOM 1 ENPI *
 2      ATOM 1 FREEEL *
 3      ATOM 1 NONHNEIGHBOR *
 4      ATOM 1 QPI *
 5      ATOM 2 NONHNEIGHBOR *
 6      BOND 1 2 DENSIG *
 7      BOND 1 2 DQSIG *
 8      ENSEMBLE EDELTAHF *
STOP
PARAMETER PRODUKT:
 9      ATOM 1 APOLARIZABILITY *
10      ATOM 1 LPSTAB *
11      ATOM 4 QTOT *
END
/NN_PARM7      11      11
A_ENPI      1 0 0 1
A_FREEEL      1 0 0 0
A_NONHNEIGHBOR 1 0 0 0
A_QPI      1 0 0 1
A_NONHNEIGHBOR 2 0 0 0
B_DENSIG      1 2 0 1
B_DQSIG      1 2 0 1
E_DELTAHF      0 0 0 1
A_POLARIZ      1 0 1 1
A_LPSTAB      1 0 1 1
A_QTOT      4 0 1 1
/NN_SCAL      12      12
 1      9.455500      11.616000
 2      0.000000      2.000000
 3      2.000000      3.000000
 4      0.100800      0.242300
 5      2.000000      4.000000
 6      0.088200      2.725800
 7      -0.474800      0.011600
 8      -98.436996      656.533691
 9      2.559000      7.090800
10      0.000000      10.057100
11      -0.119200      -0.091800

```

```

12      0.000000      0.999000
/NN_CONNS      27      27
-0.631653
-2.841740
 3.425206
-1.077220
-0.205757
 3.651624
-3.584734
 0.577689
-2.491549
-0.240271
 1.744900
 0.944103
-0.749909
 2.701843
 5.886912
-2.439176
 3.877181
-3.218500
-9.040503
-5.702037
-0.892885
 6.837634
 2.773469
-2.080549
-9.147391
 5.189584
10.084118
/END      0      0

```

Netz für die Carbonyleliminierung:

```

/NN_ARCHT      2      2
1
 2  1  1
/NN_PARM      6      6
PARAMETER EDUKT:
 1      ATOM 2  ENPI *
 2      BOND 3 2  PSTAB *
STOP
PARAMETER PRODUKT:
END
/NN_PARM7      2      2
A_ENPI  2 0 0 1
B_PSTAB 3 2 0 1
/NN_SCAL      3      3
 1      9.305900      10.108800
 2      0.000000      7.876400
 3      0.048000      1.000000
/NN_CONNS      5      5
-3.467265
 2.447763
 4.537941
-3.087231
 6.941255
/END      0      0

```

G. Publikationen

- [1] J. Gasteiger, W. Hanebeck, K.-P. Schulz, S. Bauerschmidt, R. Höllering
Automatic Analysis and Simulation of Mass Spectra
in *Computer-Enhanced Analytical Spectroscopy*, Vol. 4
C. L. Wilkins, (Hrsg.)
Plenum Press, New York, 1993, 97-133
- [2]* K.-P. Schulz, S. Bauerschmidt, R. Höllering, J. Gasteiger
Automatic Elucidation of Reactions in the Mass Spectrometer
in AIP conference proceedings **330**: E.C.C.C. 1
F. Bernardi, J. L. Rivail (Hrsg.)
American Institute of Physics, Woodbury, NY, 1995, 726-733
- [3]* R. Höllering, J. Gasteiger, and W. Hanebeck
Simulation of Mass Spectra of Organic Compounds by Massimo
in AIP conference proceedings **330**: E.C.C.C. 1
F. Bernardi, J. L. Rivail (Hrsg.)
American Institute of Physics, Woodbury, NY, 1995, 782
- [4]* L. Steinhauer, R. Höllering, S. Bauerschmidt, T. Kostka, J. Gasteiger,
Kinetic Calculations by the Simulation of Organic Reactions,
ZAMM (Zeitschrift für angewandte Mathematik und Mechanik), 1997, eingereicht
http://www.uni-regensburg.de/Fakultaeten/nat_Fak_I/Mennicken/gamm97/s20/steinhauer/steinhauer.html

Die mit * gekennzeichneten Publikationen sind Teil der Arbeit.

H. Lebenslauf

Vor- und Zuname: Robert Höllering

Geburtsdatum und -ort: 06.04.1963 in München

Eltern: Eduard Höllering und Ingeborg Höllering, geb. Oetl

Staatsangehörigkeit: deutsch

Familienstand: ledig

Schulbildung: 1969-1973 Grundschule an der Samberger Straße in München
1973-1976 Thomas-Mann-Gymnasium München
1976-1983 Klenze-Gymnasium München

Wehrdienst: Oktober 1983 bis Dezember 1984

Studium: November 1984 bis März 1992
Studium der Chemie an der Technischen Universität München
Oktober 1991 bis März 1992
Diplomarbeit bei Prof. Dr. J. Gasteiger

Promotion: Begonnen im April 1992 bei Prof. Dr. J. Gasteiger
an der Technischen Universität München,
Februar 1994 Wechsel mit dem Doktorvater
an die Friedrich-Alexander-Universität Erlangen-Nürnberg

