

Konstruktionsalgorithmen bei molekularen Graphen und deren Anwendung*

Thomas Wieland[†]

Lehrstuhl II für Mathematik, Universität Bayreuth,
D-95440 Bayreuth

Überblick

Die mathematische Disziplin der Graphentheorie hat einen ihrer Ursprünge in der Beschreibung chemischer Substanzen. In der zweiten Hälfte des 19. Jahrhunderts wurde unter Mitwirkung von Liebig, Kekulé, Crum Brown u.a. ein mathematisches Modell für Moleküle entwickelt [12, 17], bei dem die Atome als Ecken und die Bindungen als Kanten eines Bindungsgraphen aufgefaßt werden und das schnell zum Standard etablierte.

Dieses Modell wurde in der Folgezeit von Mathematikern zunächst zur Herausbildung theoretischer Begriffe und zur Ableitung von Abzählformeln verwendet. Bedeutende Beiträge dazu lieferten H. Redfield [106] und G. Pólya [103].

Als später leistungsfähige Rechenmaschinen zur Verfügung standen, wurden auch konstruktive Probleme behandelt. Dabei lag das Zentrum des Interesses zumeist auf der Konstruktion aller Graphen zu gegebenen Bedingungen (s. etwa [30, 36, 52, 56]), also z.B. die Erzeugung aller Bindungsgraphen zu einer gegebenen Bruttoformel und weiteren Nebenbedingungen [53, 55, 96].

Wir wollen in dieser Arbeit weniger untersuchte Konstruktionsaufgaben diskutieren und dabei neue Methoden für das *Verbinden* von Graphen aufzeigen. In vielen praktischen Anwendungen hat sich nämlich gezeigt, daß die ungeheure Vielfalt von Strukturen, die von Graphengeneratoren erzeugt werden können, nur schwer handzuhaben ist und durch Zusatzinformationen eingeschränkt werden muß. Daneben gibt es auch eine Reihe von Erzeugungsproblemen in der mathematischen Chemie und in der Graphentheorie, die sich nicht auf Isomere zurückführen lassen.

Obwohl einige Begriffe im folgenden erst noch klar zu definieren sind, wollen wir kurz die drei Methoden der Verbindung von Graphen umreißen, die in dieser Arbeit beschrieben werden:

1. Verbindung zweier Graphen durch Hinzufügen von Kanten

Wir betrachten numerierte ungerichtete, schleifenfreie zusammenhängende Multigraphen. Unsere Methodik werden wir allgemein formulieren, so daß eine zusätzliche Beschriftung, etwa mit Atomtypen, nicht notwendig, aber auch nicht hinderlich ist.

*Diese Arbeit wurde von der Fakultät für Mathematik und Physik der Universität Bayreuth als Dissertation zur Erlangung des Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.) genehmigt. 1. Gutachter: Prof. Dr. R. Laue, 2. Gutachter: Prof. Dr. A. Kerber, Tag der Einreichung: 18. Oktober 1996, Tag des Kolloquiums: 20. Dezember 1996

[†]Jetzige Anschrift: Deutsches Fernerkundungsdatenzentrum, Deutsche Forschungsanstalt für Luft- und Raumfahrt e.V. (DLR), Oberpfaffenhofen, D-82234 Weßling, E-Mail: Thomas.Wieland@dlr.de

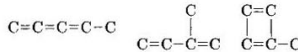
Wir setzen voraus, daß jede Ecke der Graphen neben ihrem aktuellen Grad noch einen maximalen Grad besitzt, den wir in Analogie zur Chemie als Valenz bezeichnen. Dann heißt die Konstruktionsaufgabe, der wir uns in Kapitel 3 widmen:

Bestimme alle nichtisomorphen Graphen, die sich durch Zusammenfassen der beiden Graphen und Hinzufügen von Kanten ergeben, so daß der Grad jeder Ecke höchstens so groß wie ihre Valenz ist.

Die Differenz zwischen dem aktuellen und dem maximalen Grad, die freien Valenzen, liefert uns die Anknüpfungspunkte, wie etwa bei folgendem Beispiel:



D.h. es sind alle wesentlich verschiedenen Möglichkeiten zu bestimmen, Einfach- und Mehrfachbindungen zwischen den Strukturen einzufügen. Bei diesem Beispiel gibt es 31, z.B.



(Sämtliche Lösungsgraphen werden in Abb. 3.1, S. 57, gezeigt.) Die Anzahlen an Lösungen wachsen allerdings sehr schnell und können bei einem Graphenpaar mit vier bzw. sieben Punkten bereits über einer Million liegen.

2. Verschmelzung zweier Graphen über alle gemeinsamen Teilgraphen

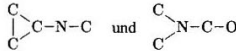
Auch hierbei betrachten wir Multigraphen mit und ohne Beschriftung. Die Konstruktionsaufgabe lautet:

Bestimme alle nichtisomorphen molekularen Graphen, die sich durch Verschmelzen zweier Graphen über alle ihre gemeinsamen Teilgraphen ergeben.

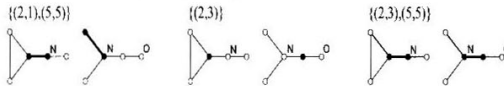
Dazu gehen wir wie folgt vor:

- i) Erzeugung aller nichtisomorphen gemeinsamen Teilgraphen (Kapitel 4)
- ii) Bestimmung aller nichtisomorphen Abbildungen der gefundenen Teilgraphen aufeinander und Konstruktion der aus diesen Abbildungen resultierenden Graphen (Kapitel 5)

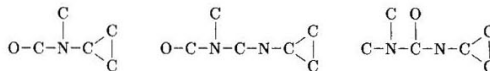
Bereits die Suche nach allen gemeinsamen Teilgraphen ist dabei ein eigenständiges Problem, das sowohl in der Graphentheorie als auch in der Chemie bereits mehrfach untersucht wurde. Wir werden einige dieser Ideen weiterentwickeln, eine durchgängige Formalisierung des Problems vorstellen und neuere mathematische Konstruktionsverfahren zur Suche einsetzen. Mit deren Hilfe sind wir dann in der Lage, beispielsweise die 16 nichtisomorphen gemeinsamen Teilgraphen von



zu bestimmen. Sie werden in Abb. 4.2, S. 69, gezeigt; drei davon sind:



Zu jedem gemeinsamen Teilgraphen wollen wir sodann alle Zuordnungsmöglichkeiten der Ecken und alle Verschmelzungen berechnen, die über diese Ecken unter Einhaltung der maximalen Grade möglich sind. Zu obigem Beispiel gibt es 12 derartige Lösungen; drei davon sind

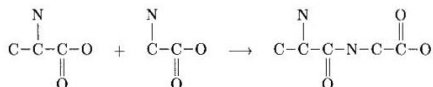


Diese und die weiteren zeigt auch Abb. 5.2 auf S. 95.

3. Verbindung über Verknüpfungsregeln unter Hinzunahme oder Wegfall von Ecken, in Nachbildung einer chemischen Reaktion

Bei molekularen Graphen gibt es aber auch den Fall, daß nicht alle möglichen Verbindungen eine Rolle spielen, sondern nur einige wenige, genau festgelegte. Solche Graphen ergeben sich z.B. im Verlauf einer chemischen Reaktion. Eine Syntax und einen Algorithmus zur redundanzfreien Konstruktion derartiger Moleküle werden wir in Kapitel 6 entwickeln.

Ein Beispiel ist die Ausbildung einer Peptidbrücke zwischen Aminosäuren:



Insbesondere wollen wir uns der Frage widmen, welche Möglichkeiten für die Anbindung einer Menge von Liganden an eine Gerüststruktur es gibt und wie diese konstruiert werden können.

Anwendungen in der Chemie

Bei den Anwendungen untersuchen wir in Kapitel 7 zunächst den iterativen Einsatz des Verschmelzungsverfahrens bei Problemen aus der Strukturauflklärung, der Suche nach quantitativen Struktur-Aktivitätsbeziehungen (QSAR) und der Wirkstoff-Forschung.

Ein weiteres Kapitel (8) ist einer derzeit populären Synthesemethodik der organischen Chemie gewidmet, der *kombinatorischen Chemie*. Grob gesprochen geht es dabei um die Anlagerung einer Menge von Bausteinsubstanzen an ein Gerüstmolekül – ein Problem, dessen konstruktive Aspekte wir ja bereits vorher diskutiert haben. Wir werden sowohl Verfahren zur geschickten Auswahl der Bausteine als auch zur Abzählung der Möglichkeiten der Anlagerungen vorstellen.

Im letzten Kapitel werden schließlich Einsatzmöglichkeiten der Verknüpfungs- und Verschmelzungsalgorithmen zur Verbesserung der Eingabe eines Isomergenerators vorgestellt.

Mathematische Hintergründe

Unser Ziel sind Algorithmen, die folgenden Anforderungen genügen:

- Sie konstruieren schnell, vollständig und redundanzfrei.
- Sie berücksichtigen Nebenbedingungen des Anwenders (sog. *Restriktionen*) so früh wie möglich.

Nur auf diese Weise ist auch die Relevanz für die Praxis gewährleistet.

Die oben angeführten Problemstellungen führen wir auf Operationen von Permutationsgruppen auf Mengen endlicher Abbildungen zurück. Das zentrale Werkzeug, das bei fast allen Konstruktionsalgorithmen eingesetzt wird, ist die *Ordnungstreue Erzeugung* von Repräsentanten von Gruppenoperationen. Dabei wird die Menge, auf der operiert wird, in invariante disjunkte Teilmengen zerlegt und die Transversale unter Ausnutzung einer Totalordnung sukzessive aufgebaut. Als neues Paradigma führen wir die simultane Ordnungstreue Erzeugung mehrerer Repräsentantensysteme ein.

Um den Aufwand der Verfahren weiter zu reduzieren, machen wir vom sog. *Homomorphieprinzip* Gebrauch. Bildet man nämlich ein Bahnenproblem auf ein leichter zu lösendes ab, etwa auf ein kleineres, so müssen nach diesem Prinzip nur noch Operationen der Urbilder der Stabilisatoren der gefundenen Repräsentanten auf deren Urbildmengen betrachtet werden. Neben der Aufspaltung von Bahnen beschreiben wir mit dem Homomorphieprinzip aber auch die Fusion von Bahnen.

Ein dritter zentraler Begriff sind *Doppelnebenklassen*. Bei der Betrachtung von Paaren von Graphen entsprechen die nichtisomorphen Lösungen der Konstruktionsprobleme oft Doppelnebenklassen nach den Automorphismengruppen der Graphen. Da aber auch diese als Bahnen von Gruppenoperationen beschrieben werden können, fügt sich dieser Begriff nahtlos in unsere übrige Darstellung ein.

Alle in dieser Arbeit vorgestellten Algorithmen wurden auch vom Autor in der Programmiersprache C++ objektorientiert implementiert und zum Programmsystem NEXTOR zusammengefaßt.

Die meisten der hier vorgestellten Verfahren sind neu und in dieser Form nicht in der Literatur zu finden. (Bei der Diskussion des Algorithmus zur Suche nach gemeinsamen Teilgraphen wird gesondert auf die in dieser Arbeit

entwickelten Neuerungen hingewiesen.) Es soll an dieser Stelle betont werden, daß viele der vorgestellten mathematischen Verfahren ganz allgemein und auf viele Arten diskreter Strukturen anwendbar sind. Die Konzentration auf Graphen und molekulare Graphen ist somit exemplarisch zu verstehen und ist durch die Anwendung motiviert, aber nicht durch die Methodik bedingt.

Inhaltsverzeichnis

Überblick	7
I Gruppen, Graphen und chemische Verbindungen	15
1 Gruppen und Nebenklassen	17
1.1 Gruppenoperationen und Symmetrieklassen	17
1.2 Nebenklassen und Verklebungen	20
1.3 Homomorphieprinzip und Ordnungstreue Erzeugung	22
2 Graphen und Moleküle	27
2.1 Grundlegende Definitionen	27
2.2 Datenstrukturen für Permutationsgruppen	31
2.3 Automorphismen von Teilgraphen	35
2.4 Molekulare Graphen	37
II Algorithmen	41
3 Verbindung zweier Graphen	43
3.1 Überblick	43
3.2 Aufspaltung nach dem Homomorphieprinzip	46
3.3 Ordnungstreue Erzeugung der Verbindungsmatrizen	49
3.4 Kanonisierung der verbundenen Graphen	53
3.5 Diskussion des Verfahrens	58
4 Erzeugung gemeinsamer Teilgraphen	61
4.1 Strategie des Algorithmus	61
4.2 Der Suchalgorithmus	67
4.3 Der Minimalitätstest	71
4.4 Diskussion des Verfahrens	75
4.4.1 Aufwand	75
4.4.2 Experimentelle Resultate	75
4.4.3 Anwendungen	76
4.5 Erweiterung auf mehrere Ausgangsgraphen	83
5 Verschmelzung zweier Graphen	87
5.1 Berechnung der Zuordnungen	87
5.2 Der Verschmelzungsalgorithmus	91
5.3 Experimentelle Resultate	96
5.4 Dreidimensionale Verschmelzungen	99
6 Regelbasierte Verbindung von molekularen Graphen	103
6.1 Allgemeine Verbindungssyntax	103
6.2 Mehrfachanlagerungen an eine Kernstruktur	105

III Anwendungen in der Chemie	107
7 Zielgerichtete stufenweise Konstruktion im Moleküldesign	109
7.1 Topologische Indices	109
7.2 Iterative Konstruktion	112
7.2.1 Allgemeiner Algorithmus	112
7.2.2 Strukturaufklärung mittels Spektreninterpretation	112
7.2.3 Inverses Problem der QSAR-Forschung	114
7.3 Design von Struktur- und Wirkstoffanaloga	116
7.3.1 Strukturanaloga über topologische Ähnlichkeiten	116
7.3.2 Wirkungsanaloga	117
8 Simulationen in der kombinatorischen Chemie	127
8.1 Hintergrund und Techniken	127
8.1.1 Einleitung	127
8.1.2 Auswahl der Bausteine	128
8.2 Aufbau einer kombinatorischen Bibliothek	134
8.2.1 Einstufige Verfahren	134
8.2.2 Mehrstufige Verfahren	135
8.3 Abzählen und Screenen einer Bibliothek	140
9 Verbesserung der Eingabe eines Isomergenerators	145
9.1 Makroatome und Goodlist	145
9.2 Vermeidung der Goodlist-Filterung	147
Literaturverzeichnis	150

Vorwort

Angeregt durch die Vorarbeiten, die von D. Moser [70, 98] und insbesondere Dr. R. Grund am Lehrstuhl II für Mathematik der Universität Bayreuth geleistet wurden [53, 54] und die zum Strukturgenerator **MOLGEN** [14, 131] führten, ergab sich der Wunsch nach einer Ausweitung der bestehenden Verfahren. Die Anforderungen entsprangen dabei zwei relativ unterschiedlichen Richtungen: Zum einen sollte ein theoretischer Beitrag im Rahmen der am Lehrstuhl untersuchten *konstruktiven Theorie diskreter Strukturen* [56, 69, 84, 85] geleistet werden, zum anderen brachte der durch den Vertrieb von **MOLGEN** in Gang gekommene intensive Kontakt mit einer Reihe von Chemikern verschiedener Forschungsrichtungen viele neue Fragestellungen aus der mathematischen Chemie mit sich. Somit wurde bei der Ausgestaltung dieser Arbeit der Versuch unternommen, beiden Anforderungen gleichmäßig Rechnung zu tragen, was zu einer Zweiteilung führte in einen theoretischen Teil, der verschiedene Algorithmen und ihre Hintergründe vorstellt, und einen anwendungsbezogenen Teil, der sich jedoch nicht damit begnügt, exemplarische Fälle der zuvor diskutierten Verfahren aufzuzeigen, sondern eigenständige Resultate vorzustellen versucht.

Denn auch die Mathematik steht heute, am Ende des zwanzigsten Jahrhunderts, vor neuen Herausforderungen: Die Menschen bringen mehr Erkenntnisse denn je zuvor hervor; ihr Wissen verdoppelt sich alle zehn Jahre. Daher können auch die einzelnen Fachdisziplinen ihre Grenzen nicht mehr in derselben Weise ziehen wie im 19. Jahrhundert, als sie sich etablierten. Kaum ein Wissenschaftler kann es sich heute noch leisten, sein Interesse nur auf seine Fachrichtung einzuschränken und andere völlig zu ignorieren. In den immer stärker fließenden Übergangszonen bilden sich neue Wissensgebiete heraus. So dürfen auch Abhandlungen, die die Grenzen zu überbrücken versuchen, ja die bemüht sind, den nicht immer einfachen Dialog zwischen den Fachsprachen zu verbessern, nicht nur unter dem Blickwinkel eines Fachgebietes gesehen, sondern müssen in ihrer Gesamtheit beurteilt werden.

Mein Dank gilt zunächst den Profs. R. Laue und A. Kerber, die mich an den Themenkreis heranführten, stets ein offenes Ohr für Fragen aller Art hatten und mir viel Freiraum für eigene Ideen ließen. Daneben danke ich allen Mitarbeitern an **MOLGEN** und am Lehrstuhl II für Mathematik, insbesondere Herrn Dr. C. Benecke, der neben vielen technische Hilfestellungen auch einige nützliche Programmroutinen beisteuerte, und Herrn Dipl.-Math. T. Grüner, der mir bei fast allen theoretischen Schwierigkeiten weiterhelfen konnte und dessen Implementierungen verschiedener Verfahren von großem Wert waren. Nicht zuletzt danke ich meiner Frau Andrea, deren Geduld und Verständnis diese Arbeit erst möglich gemacht haben.

Teil I

Gruppen, Graphen und chemische Verbindungen

Kapitel 1 Gruppen und Nebenklassen

1.1 Gruppenoperationen und Symmetrieklassen

Zunächst wollen wir die wichtigsten algebraischen Definitionen einführen. Vom Leser wird dabei lediglich erwartet, daß er mit elementaren Begriffen der Gruppentheorie vertraut ist. (Für Details und Beispiele sei auf [37] oder [67] verwiesen.)

1.1.1 Definition. *Es sei G eine multiplikativ geschriebene Gruppe und Ω eine nichtleere Menge. Eine Operation von G auf Ω (von rechts) ist eine Abbildung:*

$$\Omega \times G \rightarrow \Omega, (\omega, g) \mapsto \omega^g$$

mit

- $(\omega^g)^{g'} = \omega^{gg'} \quad \forall g, g' \in G \quad \forall \omega \in \Omega$ und
- $\omega^{id} = \omega$.

Man schreibt dafür kurz Ω_G . •

Gruppenoperationen liefern uns eine Reihe wichtiger Strukturen:

1.1.2 Definition. *Es operiere Ω_G . Seien $\omega \in \Omega$ und $\Delta \subseteq \Omega$.*

- $\omega^G := \{\omega^g \mid g \in G\}$ heißt die Bahn des Punktes ω .
- $\Omega//G := \{\omega^G \mid \omega \in \Omega\}$ heißt die Menge der Bahnen.
- $\mathcal{T}(\Omega//G)$ heißt eine Transversale der Bahnen mit $\Omega = \dot{\bigcup}_{t \in \mathcal{T}} t^G$, die sich aus der Äquivalenzklasseneigenschaft der Bahnen ergibt.
- $\Omega_g := \{\omega \in \Omega \mid \omega^g = \omega\}$ wollen wir die Menge der Fixpunkte von g nennen.
- $C_G(\Delta) := \{g \in G \mid \delta^g = \delta \quad \forall \delta \in \Delta\}$ heißt Zentralisator oder punktwise Stabilisator von Δ in G . Falls Δ einelementig ist, also $\Delta = \{\delta\}$, schreibt man auch $C_G(\delta)$.
- $N_G(\Delta) := \{g \in G \mid \delta^g \in \Delta \quad \forall \delta \in \Delta\}$ heißt Normalisator oder mengenweiser Stabilisator von Δ in G . •

Falls sowohl G als auch Ω endlich sind, nennen wir auch die Operation endlich. In dieser Arbeit werden wir ausschließlich endliche Gruppenoperationen betrachten.

Zudem halten wir fest, daß für einelementige Mengen $\Delta = \{\delta\}$ gilt, daß $C_G(\delta) = N_G(\delta)$.

1.1.3 Bemerkung Seien X und Y zwei nichtleere endliche Mengen. Dann kann man setzen:

$$Y^X := \{f \mid f: X \rightarrow Y\}$$

Operiert Y_H , so operieren G und H auf Y^X auf natürliche Weise:

- $Y^X \times G \rightarrow Y^X, (f, g) \mapsto f^g$ mit $f^g(x) = f(x^{g^{-1}})$,
- $Y^X \times H \rightarrow Y^X, (f, h) \mapsto \tilde{f}$ mit $\tilde{f}(x) = f(x)^h$ und
- $Y^X \times (H \times G) \rightarrow Y^X, (f, (h, g)) \mapsto \tilde{f}^g$.

Die Bahnen der Operationen von G , H und von $H \times G$ heißen *Symmetrieklassen von Abbildungen*.

Wir wollen nun $\underline{n} := \{1, 2, \dots, n\}$ setzen. Die Abbildungen $Y^{\underline{n}}$ lassen sich dann in Form einer *Liste* schreiben:

$$f = (f(1), f(2), \dots, f(n))$$

Damit können wir festlegen:

1.1.4 Definition. Sei $Y \subset \mathbb{N}$.

- Es heiÙe $f \leq f'$ genau dann, wenn der Bildvektor von f lexikographisch kleiner oder gleich dem Bildvektor von f' ist.
- Für $f \in M \subseteq Y^{\underline{n}}$ heiÙe es $f \leq M$ oder $f = \min(M)$, falls für alle $f' \in M$ gilt: $f \leq f'$.

Durch diese lexikographische Ordnung wird die Menge $Y^{\underline{n}}$ total geordnet.

1.1.5 Definition.

- $S_n := \{\pi \in \underline{n}^{\underline{n}} \mid \pi \text{ bijektiv}\}$ heißt *symmetrische Gruppe*.
- Die Elemente von S_n heißen *Permutationen*. *Untergruppen einer symmetrischen Gruppe nennt man Permutationsgruppen*.

Wir wollen nun einige Eigenschaften der symmetrischen Gruppe zusammenstellen. Für die Details sei wieder auf [67] verwiesen.

1.1.6 Bemerkungen

- $|S_n| = n!$.
- Für die Elemente $\pi \in S_n$ gibt es zwei Schreibweisen: Zum einen die Listenschreibweise

$$\begin{pmatrix} 1 & 2 & \dots & n \\ \pi(1) & \pi(2) & \dots & \pi(n) \end{pmatrix},$$

wobei die erste Zeile auch weggelassen werden kann, wenn Unklarheiten ausgeschlossen sind.

Ist π andererseits von der Form

$$\begin{pmatrix} i_1 & i_2 & \dots & i_{r-1} & i_r & i_{r+1} & \dots & i_n \\ i_2 & i_3 & \dots & i_r & i_1 & i_{r+1} & \dots & i_n \end{pmatrix},$$

so heißt π *r-Zyklus*, und man schreibt: $\pi = (i_1, i_2, \dots, i_r)$. Jede Permutation kann als Produkt paarweise disjunkter Zyklen geschrieben werden. Dabei bezeichnet $c(\pi)$ die Anzahl der Zyklen, und i_i die Länge des i -ten Zykels. Zyklen der Länge 2 heißen *Transpositionen*. Zyklen der Länge 1 werden i.a. nicht geschrieben. Für die identische Permutation schreiben wir auch $1 := (1)(2)\dots(n)$.

- Die geordneten Längen $\alpha_i(\pi)$, $\pi \in S_n$, der zyklischen Faktoren von π bilden eine eindeutig bestimmte (eigentliche) Partition von n , d.h. für alle i gilt $\alpha_i \geq \alpha_{i+1}$ und zudem $\sum_i \alpha_i = n$, in Zeichen $\alpha(\pi) := (\alpha_1(\pi), \dots, \alpha_{c(\pi)}(\pi)) \vdash n$. Dann heißt $\alpha(\pi)$ *Zykelpartition* von π .
- Definiere für $i \in \underline{n}$, $\pi \in S_n$:

$$a_i(\pi) := |\{j \mid \alpha_j(\pi) = i\}|$$

Dann nennt man das Tupel $a(\pi) := (a_1(\pi), \dots, a_n(\pi))$ den *Zykeltyp* von π . Dabei gilt: $\sum i \cdot a_i = n$, und man schreibt: $a \vdash n$.

1.1.7 Definition.

i) Ein m -Tupel natürlicher Zahlen $\lambda_i \in \mathbb{N}$ mit

$$\lambda := (\lambda_1, \lambda_2, \dots, \lambda_m) \quad \text{mit} \quad \sum_{i=1}^m \lambda_i = n$$

heißt uneigentliche Partition von n , kurz $\lambda \models n$.

ii) Sei \underline{n}^λ eine Mengenzersetzung von \underline{n} mit

$$\underline{n}_i^\lambda \subseteq \underline{n}, \quad \text{wobei} \quad |\underline{n}_i^\lambda| = \lambda_i \quad \forall i \in \underline{m} \quad \text{und} \quad \underline{n} = \bigcup_{i \in \underline{m}} \underline{n}_i^\lambda.$$

Der Normalisator von \underline{n}^λ

$$S(\underline{n}^\lambda) := S_{\underline{n}_1^\lambda} \oplus S_{\underline{n}_2^\lambda} \oplus \dots \oplus S_{\underline{n}_m^\lambda} = \{ \pi \in S_n \mid \pi \underline{n}_i^\lambda = \underline{n}_i^\lambda \forall i \in \underline{m} \}$$

heißt Young-Untergruppe.

Zu dieser Definition zwei

1.1.8 Bemerkungen (s.a. [67])

- i) Diese Definition der Mengenzersetzung ist recht allgemein, da sie auch leere Blöcke zuläßt: $\underline{n}_i^\lambda = \emptyset$ bei $\lambda_i = 0$.
- ii) Eine Younggruppe ist eine Untergruppe der symmetrischen Gruppe: $S(\underline{n}^\lambda) \leq S_n$. Zudem gilt:

$$S(\underline{n}^\lambda) \cong S_{\lambda_1} \times \dots \times S_{\lambda_m}$$

1.2 Nebenklassen und Verklebungen

In diesem Abschnitt wollen wir die grundlegenden Begriffe und Aussagen über Nebenklassen und Doppelnebenklassen vorstellen, die später eine wichtige Rolle spielen werden.

1.2.1 Definition. Sei G eine Gruppe und $A, B \leq G$ Untergruppen.

- Für jedes Element $g \in G$ heißt die Teilmenge

$$Ag := \{ag \mid a \in A\} \quad \text{bzw.} \quad gA := \{ga \mid a \in A\}$$

(die von g erzeugte) Rechts- bzw. Linksnebenklasse von A in G . Die Menge aller Rechtsnebenklassen wird mit $A \setminus G$, die der Linksnebenklassen mit G / A bezeichnet.

- Für jedes Element $g \in G$ heißt die Teilmenge

$$AgB := \{agb \mid a \in A, b \in B\}$$

(die von g erzeugte) Doppelnebenklasse von A und B in G . Man schreibt $A \setminus G / B$ für die Menge aller Doppelnebenklassen.

1.2.2 Bemerkung. Die soeben definierten Nebenklassen können auch als Bahnen von Operationen der Form

$$G \times U \rightarrow G, (g, (u_1, u_2)) \mapsto u_1 g u_2^{-1}$$

von Untergruppen $U \leq G \times G$ auf G aufgefaßt werden.

Für eine Untergruppe $A \leq G$ sind $A \times \{1\}$ und $\{1\} \times A$ Untergruppen von $G \times G$. Dabei gilt (s. [67]), daß $G // A \times \{1\} = A \setminus G$ und $G // \{1\} \times A = G / A$, was aus der Definition der Operation klar wird.

Ist nun $B \leq G$ eine weitere Untergruppe, so ist $A \times B \leq G \times G$, und es gilt:

$$G // A \times B = A \setminus G / B$$

was ebenfalls leicht ersichtlich ist.

Aber auch in anderer Weise ergeben sich Doppelnebenklassen als Bahnen einer Operation. Bei $A, B \leq G$ operiert B in natürlicher Weise auf den Nebenklassen von A :

$$(1.2.3) \quad A \setminus G \times B \rightarrow A \setminus G, (Ag, b) \mapsto Agb$$

Offensichtlich gilt $A \setminus G // B = A \setminus G / B$.

Ein wichtiges, elementares Resultat ist der Zusammenhang zwischen der Bahn eines Elements und der Nebenklasse seines Stabilisators: Ist G eine Gruppe, die auf einer Menge Ω operiert, so ist für jedes $\omega \in \Omega$ die Abbildung

$$(1.2.4) \quad \omega^G \rightarrow N_G(\omega) \setminus G, \omega^g \mapsto N_G(\omega)g$$

eine Bijektion.

Damit ergibt sich auch

1.2.5 Satz (Fundamentallemma). Sei G eine Gruppe, die auf einer Menge Ω transitiv operiert, und $U \leq G$ eine Untergruppe. Sei ferner $\omega_0 \in \Omega$ beliebig, aber fest gewählt. Dann ist die Abbildung

$$\Omega // U \rightarrow N_G(\omega_0) \setminus G / U, (\omega_0^g)^U \mapsto N_G(\omega_0)gU$$

eine Bijektion.

(Beweis siehe [85]).

Um die Bahnen der Operation von U auf Ω zu bestimmen, genügt es also, eine Transversale der Doppelnebenklassen $N_G(\omega_0) \setminus G / U$ zu berechnen, und umgekehrt. Dies ist ein wichtiges Hilfsmittel bei der Konstruktion diskreter Strukturen.

Ein weiterer wichtiger Satz für diese Arbeit ist das sog. *Verklebungslemma*. Hierin wird beschrieben, wie sich Objekte, die selbst Symmetrien beinhalten, auf alle wesentlich verschiedenen Weisen miteinander verbinden lassen. Um diesen vorzustellen, müssen wir noch den Begriff des Homomorphismus einführen.

1.2.6 Definition.

- Seien G und H Gruppen. Eine Abbildung $f : G \rightarrow H$ heißt Gruppenhomomorphismus, falls für alle $g_1, g_2 \in G$ gilt:

$$f(g_1 g_2) = f(g_1) f(g_2)$$

Sind die Mengen, zwischen denen eine Abbildung definiert ist, keine Gruppen, so verstehen wir in Verallgemeinerung dieser Definition unter einem Homomorphismus eine mit der auf den Mengen definierten Struktur verträgliche und strukturerhaltende Abbildung.

- Seien $(\Omega_1)_{G_1}$ und $(\Omega_2)_{G_2}$ zwei endliche Gruppenoperationen, und $f = (f_\Omega, f_G)$ sei ein Paar von Abbildungen, so daß $f_\Omega : \Omega_1 \rightarrow \Omega_2$ und $f_G : G_1 \rightarrow G_2$ ein Gruppenhomomorphismus ist. Dann heißt f Homomorphismus von Gruppenoperationen, falls für alle $\omega \in \Omega_1$ und $g \in G_1$ gilt:

$$f_\Omega(\omega^g) = (f_\Omega(\omega))^{f_G(g)}$$

- Ein Homomorphismus $f : G \rightarrow H$ heißt
 - Epimorphismus, falls f surjektiv ist,
 - Monomorphismus, falls f injektiv ist,
 - Isomorphismus, falls f bijektiv ist,
 - Automorphismus, falls f bijektiv und $G = H$ ist.
- Die Menge $\text{Ker}(f) := \{g \in G \mid f(g) = 1_H\}$ heißt Kern von f .

1.2.7 Satz (Verklebungslemma). Sei Ω eine Menge von Objekten, für die Isomorphismen definiert sind, und $\omega_1, \omega_2 \in \Omega$ zwei solche Objekte, die vermöge $f_0 : \omega_1 \rightarrow \omega_2$ isomorph sind. Sei weiterhin $A \leq \text{Aut}(\omega_1)$ eine Untergruppe der Automorphismen von ω_1 und $B \leq \text{Aut}(\omega_2)$. Dann operiert $A \times B$ auf $\text{Iso}(\omega_1, \omega_2)$, der Menge der Isomorphismen von ω_1 nach ω_2 , als

$$\text{Iso}(\omega_1, \omega_2) \times (A \times B) \rightarrow \text{Iso}(\omega_1, \omega_2), (f, (a, b)) \mapsto f^{(a,b)} = a^{-1} f b,$$

wobei $(a^{-1} f b)(x) = f(x^{a^{-1}})^b$ für $x \in \omega_1$ ist.

Ferner gilt:

i) $\text{Iso}(\omega_1, \omega_2) = f_0 \circ \text{Aut}(\omega_2)$.

ii) Die Abbildung

$$\begin{aligned} \text{Iso}(\omega_1, \omega_2) // A \times B &\rightarrow f_0(f_0^{-1} A f_0) \setminus \text{Aut}(\omega_2) / B, \\ (f_0 \sigma)^{(A \times B)} &\mapsto f_0(f_0^{-1} A f_0) \sigma B \end{aligned}$$

ist eine Bijektion.

(Ein Beweis findet sich bei [85])

1.3 Homomorphieprinzip und Ordnungstreue Erzeugung

In der Mathematik diskreter Strukturen spielen Homomorphismen oft eine wichtige Rolle [67, 69, 83, 85]. Durch sie können Verfeinerungen und Vergrößerungen der Strukturbeschreibungen formal ausgedrückt werden. So gelingt es, komplizierte Probleme auf einfachere, d.h. leichter lösbare zurückzuführen, was effektivere Algorithmen zu entwickeln erlaubt. Etwa bei dem Problem der Berechnung einer Transversale der Bahnen einer Gruppenoperation bildet man durch einen Homomorphismus das Problem auf eine leichter konstruierbare Operation ab und kann die Stabilisatoren der Repräsentanten zur Vereinfachung der ursprünglichen Operation verwenden. Die Grundlage bildet der folgende

1.3.1 Satz (Homomorphieprinzip). *Seien G_1 und G_2 Gruppen, die auf Mengen Ω_1 bzw. Ω_2 operieren, und sei $\sigma = (\sigma_\Omega, \sigma_G)$ ein Epimorphismus von Gruppenoperationen. Dann gilt:*

- i) *Für jedes $\omega_2 \in \Omega_2$ und $g_2 \in G_2$ schneiden $\sigma_\Omega^{-1}(\omega_2)$ und $\sigma_\Omega^{-1}(\omega_2^{g_2})$ die gleichen Bahnen von G_1 auf Ω_1 .*
- ii) *Sind $\omega_2 \in \Omega_2$ und $\omega_1, \omega'_1 \in \sigma_\Omega^{-1}(\omega_2)$ mit $\omega_1^{g_1} = \omega'_1$ für ein $g_1 \in G_1$, so liegt g_1 im Urbild des Zentralisators von ω_2 :*

$$g_1 \in \sigma_G^{-1}(C_{G_2}(\omega_2)).$$

Zum besseren Verständnis der Zusammenhänge geben wir einen Beweis an (nach [85]):

- i) Seien $\omega_2 \in \Omega_2$ und $g_2 \in G_2$. Für $\omega_1 \in \sigma_\Omega^{-1}(\omega_2)$ ist $\sigma_\Omega(\omega_1) = \omega_2$, also auch $\sigma_\Omega(\omega_1)^{g_2} = \omega_2^{g_2}$. Nach Voraussetzung ist für $g \in \sigma_G^{-1}(g_2)$ damit $\sigma_\Omega(\omega_1^g) = \omega_2^{g_2}$, woraus $\omega_1^g \in \sigma_\Omega^{-1}(\omega_2^{g_2})$ folgt. Da sich dieser Schluß auch leicht umkehren läßt, folgt somit die Behauptung.
- ii) Seien $\omega_2 \in \Omega_2$ und $\omega_1, \omega'_1 \in \sigma_\Omega^{-1}(\omega_2)$ mit $\omega_1^{g_1} = \omega'_1$ für ein $g_1 \in G_1$. Somit gilt nach Voraussetzung $\sigma_\Omega(\omega_1)^{\sigma_G(g_1)} = \sigma_\Omega(\omega'_1)$. Da sowohl ω_1 als auch ω'_1 im Urbild von ω_2 liegen, ist $\omega_2^{\sigma_G(g_1)} = \omega_2$, d.h. $\sigma_G(g_1)$ liegt im Zentralisator von ω_2 . Der Schritt zum Urbild liefert dann $g_1 \in \sigma_G^{-1}(C_{G_2}(\omega_2))$.

□

1.3.2 Bemerkung. Die Anwendungen des Homomorphieprinzips bei der Berechnung von Transversalen können in zwei Klassen aufgeteilt werden (vgl. [85]):

- i) *Aufspaltung von Bahnen:* Wir nehmen an, es sei eine Transversale der Bahnen der Operation $(\Omega_2)_{G_2}$ im Bildraum gegeben und die Repräsentanten von $\Omega_1 // G_1$ im Bildraum gesucht. Dann müssen wir nach 1.3.1 zu allen $\omega_2 \in \mathcal{T}(\Omega_2 // G_2)$ nur die Operation von $\sigma_G^{-1}(C_{G_2}(\omega_2))$ auf $\sigma_\Omega^{-1}(\omega_2)$ betrachten. Die operierende Gruppe wird also (i.a. wesentlich) kleiner, was entsprechende Konstruktionsalgorithmen deutlich beschleunigt.

Besonders effektiv sind Algorithmen, die 1.3.1 sukzessive anwenden. Diese Technik bezeichnet man auch als *Surjektive Auflösung* (s. [67, 83]): Für $m \in \mathbb{N}$ und $i \in \underline{m}$ seien $(\Omega_i)_{G_i}$ endliche Gruppenoperationen, und für $i < m$ seien $\sigma^{(i)}$ Paare von Abbildungen wie in 1.3.1 (also etwa $\sigma_\Omega^{(i)} : \Omega_i \longrightarrow \Omega_{i-1}$). Dann erhält man eine Transversale von $\Omega_m // G_m$, indem man zunächst eine Transversale von $\Omega_1 // G_1$ und die Zentralisatoren ihrer Elemente bestimmt und dann jeweils Repräsentanten der Operation von $(\sigma_G^{(i+1)})^{-1}(C_{G_i}(\omega_i))$ auf $(\sigma_\Omega^{(i+1)})^{-1}(\omega_i)$ berechnet.

Beispiele dazu werden wir in 3.2.1 und 5.1.6 antreffen.

- ii) *Fusion von Bahnen:* Wir nehmen an, es sei eine Transversale der Bahnen der Operation $(\Omega_1)_{G_1}$ im Bildraum gegeben und die Repräsentanten von $\Omega_2 // G_2$ im Bildraum gesucht. Auf diese Situation trifft man meist dann, wenn man zur Bahnenberechnung Unterscheidungsmerkmale, wie etwa eine Klasseneinteilung, der Elemente verwendet hat, die eigentlich (d.h. im Bildraum) aufgehoben werden sollen.

Ein Beispiel dafür wird uns in 3.4 begegnen.

Trotz aller Aufspaltungs- und Fusionstechniken verschwindet das Problem der Bahnenberechnung nicht ganz, sondern wird nur verkleinert. Wir müssen also noch Methoden für das Erzeugen vollständiger Repräsentantensysteme von Bahnen von Gruppenoperationen bereitstellen.

Ein naiver Ansatz zur Berechnung einer $\mathcal{T}(\Omega // G)$ ist, alle $\omega \in \Omega$ zu durchlaufen und für jedes zu testen, ob es nicht ein $g \in G$ und ein bereits untersuchtes $\omega' \in \Omega$ gibt, so daß $\omega^g = \omega'$; gibt es diese nicht, so ist $\omega \in \mathcal{T}$. Diese

Vorgehensweise ist natürlich für die Praxis vollkommen unsinnig, da hier jedes neue Element mit allen bisherigen Repräsentanten verglichen werden muß.

Sehr oft gibt es auf Ω eine Ordnung, etwa die lexikographische wie in 1.1.4, die herangezogen werden kann. Da nur ein Vertreter jeder Bahn gesucht ist, kann man auch denjenigen nehmen, der bezüglich der Ordnung der größte ist, also $\omega \geq \omega^g$. Ein solcher Repräsentant ω heißt *kanonisch*.

Damit ergibt sich:

1.3.3 Algorithmus (Kanonischer Bahnenalgorithmus).

- i) Starte mit der leeren Menge $\mathcal{T} \leftarrow \emptyset$.
- ii) Berechne in *absteigender* Reihenfolge die Elemente $\omega \in \Omega$ und untersuche sie wie folgt:
 - Falls ein $g \in G$ existiert, so daß $\omega^g > \omega$, dann ist ω kein kanonischer Repräsentant.
 - Andernfalls setze: $\mathcal{T} \leftarrow \mathcal{T} \cup \{\omega\}$.

◇

Hierbei hängt der Test eines Kandidaten $\omega \in \Omega$ nur noch von G , aber nicht mehr von den bereits vorhandenen Repräsentanten ab. Allerdings muß noch ganz Ω durchlaufen werden.

Dieses Problem versucht das Prinzip der *Ordnungstreuen Erzeugung* nach R.C. READ [105] zu umgehen. Dabei wird Ω in invariante disjunkte Teilmengen zerlegt und die Transversale so sukzessive aufgebaut. Dies beschreibt folgender

1.3.4 Satz (Ordnungstreue Erzeugung). Sei Ω_G eine endliche Gruppenoperation, wobei Ω durch \leq total geordnet sei und eine disjunkte Zerlegung

$$\Omega = \bigcup_{i=1}^n \Omega_i$$

in invariante und nichtleere Teilmengen Ω_i existiere. Sei A ein Algorithmus, der für jedes $\omega \in \Omega$ entweder die leere Menge oder eine Menge $A(\omega) \subseteq \Omega$ in absteigender Reihenfolge erzeugt, so daß für die kanonischen Transversalen $\mathcal{T}_>^{(i)}$ von $\Omega_i // G$ für alle $i \in \{1, \dots, n-1\}$ die folgenden Bedingungen gelten:

•

$$\mathcal{T}_>^{(i+1)} \subseteq \bigcup_{\omega \in \mathcal{T}_>^{(i)}} A(\omega)$$

- Für alle $\omega_1, \omega_2 \in \mathcal{T}_>^{(i)}$ mit $\omega_1 < \omega_2$ gilt, daß aus $\omega_1 \in A(\omega'_1)$ und $\omega_2 \in A(\omega'_2)$ folgt, daß $\omega'_1 < \omega'_2$ ist.

Dann erhält man die gesuchte Transversale von $\Omega // G$ durch folgendes Vorgehen:

- i) Bestimme $\mathcal{T}_>^{(1)}$ und setze $\mathcal{T}_> \leftarrow \mathcal{T}_>^{(1)}$.
- ii) Für $i \in \{1, \dots, n-1\}$ bestimme alle $A(\omega^{(i)})$ für $\omega^{(i)} \in \mathcal{T}_>^{(i)}$, berechne $\mathcal{T}_>^{(i+1)}$ durch Vereinigung der $A(\omega^{(i)})$ und Entfernung aller nichtkanonischen Elemente und setze

$$\mathcal{T}_> \leftarrow \mathcal{T}_> \cup \mathcal{T}_>^{(i+1)}.$$

Häufig gilt für den Algorithmus A zusätzlich, daß es für jedes $\omega \in \mathcal{T}_>^{(i+1)}$ genau ein $\omega' \in \mathcal{T}_>^{(i)}$ gibt mit $\omega \in A(\omega')$. Dadurch wird die Zahl überflüssiger Kandidaten, die zu testen sind, weiter klein gehalten.

Wir betrachten nun die Situation bei Symmetrieklassen von Abbildungen (s. 1.1.3). Eine endliche Gruppe G operiere auf einer endlichen Menge X . Ist Y eine weitere endliche Menge, so operiert G auch auf Y^X . Gemäß 1.1.4 existiert auf Y^X eine lexikographische Ordnung. Ohne Einschränkung können wir von

$$Y = \{0, \dots, m-1\} \quad \text{und} \quad X = \{0, \dots, n-1\}$$

mit $m, n \in \mathbb{N}$ ausgehen und folgenden Satz formulieren:

1.3.5 Satz. Sei $f \in \mathcal{T}_>(Y^X // G)$ und $f_1 \in Y^X$ ein Anfangsstück von f , d.h. es existiert ein $t \leq n$ mit

$$f_1(j) = \begin{cases} f(j) & \text{für } j < t \\ 0 & \text{für } j \geq t \end{cases}$$

Dann ist auch $f_1 \in \mathcal{T}_>(Y^X // G)$.

Beweis: Wir führen den Beweis indirekt und nehmen an, f_1 sei nicht kanonisch, also $f_1 \notin \mathcal{T}_>(Y^X // G)$. Dann existiert ein $g \in G$ mit $f_1^g > f_1$, d.h. es gibt auch ein $\tilde{t} < n$, so daß

$$\begin{aligned} f_1^g(j) &= f_1(j) & \text{für } j < \tilde{t} \text{ und} \\ f_1^g(\tilde{t}) &> f_1(\tilde{t}). \end{aligned}$$

Es gilt, daß $\tilde{t} < t$, da sonst unter der Permutation g wegen $f_1(j) = 0$ für $j \geq t$ auch f_1^g nicht größer als f_1 werden könnte.

Wir unterscheiden nun nach der Wirkung von g :

- i) g vertauscht Plätze $k < \tilde{t}$ und $k' > \tilde{t}$ (d.h. $k^{g^{-1}} = k'$) mit $f_1(k) = f_1(k') = 0$, wobei aber $f(k') > 0$ ist. (Gibt es mehrere solcher Paare (k, k') , so wählen wir das mit kleinstem k .) Nach Voraussetzung ist dann

$$f^g(j) = f_1^g(j) = f_1(j) = f(j)$$

für $j < k$ und

$$f(k) = f_1(k) = f_1^g(k) = 0, \text{ aber } f^g(k) = f(k') > 0,$$

also $f(k) < f^g(k)$. Das bedeutet $f < f^g$, im Widerspruch zur vorausgesetzten Maximalität von f .

- ii) In allen anderen Fällen ist $f^g(j) = f(j) \forall j < \tilde{t}$ und

$$f^g(\tilde{t}) = f_1^g(\tilde{t}) > f_1(\tilde{t}) = f(\tilde{t}),$$

was wiederum ein Widerspruch zur Maximalität von f ist. □

Im Fall $m = 1$ können wir die Elemente von Y^X auch als Teilmengen von X auffassen. Ein Anfangsstück f_1 ist dann eine Teilmenge $f_1 \subset f$ mit $f_1 < f$ (s. [85]).

Satz 1.3.5 zeigt uns, wie 1.3.4 algorithmisch eingesetzt wird: Es genügt, Anfangsstücke lexikographisch zu erweitern, ohne daß sie nochmals auf ihre Maximalität getestet werden müssen. In der Umkehrung bedeutet dies, daß, falls das Anfangsstück bereits nicht kanonisch ist, es auch durch die Belegung der weiteren Plätze nicht zu einem kanonischen Repräsentanten werden kann.

Bei jedem Setzen eines Wertes im Kandidaten muß man bei der Ordnungstreuen Erzeugung auf Maximalität testen. Eine Abbildung, die letztlich als maximal erkannt wird, hat dann schon eine Reihe von Tests hinter sich. Sind an die Ergebnisse noch zusätzliche Nebenbedingungen gestellt, so war dieser Aufwand u.U. umsonst.

Eine Möglichkeit liegt darin, beim Füllen des Kandidaten nur die Nebenbedingungen zu überprüfen und nur in bestimmten Schritten (oder ausschließlich ganz am Ende) einen Maximalitätstest durchzuführen. Ein Nachteil an dieser Vorgehensweise ist sicher, daß dabei der Fall eintreten kann, daß man einige Zeit mit nicht-maximalen Abbildungen voranschreitet. Um dies zu verhindern – oder zumindest zu lindern –, verwendet man die Schlußfolgerungen, die sich aus einem nicht bestandenen Maximaltest ziehen lassen:

1.3.6 Lemma. Sei $f \in Y^X$ und $g \in G$ mit $f^g > f$, d.h.

$$\exists t < n - 1 : f^g(j) = f(j) \quad \forall j < t, \quad f^g(t) > f(t)$$

Sei $y := \max\{x^g \mid x < t, f^g(x) > 0\}$ und $z := \max\{j^g, y\}$. Dann gilt für jedes $f' \in Y^X$ mit $f'(j) = f(j)$ für $j \leq z$:

$$(f')^g > f'.$$

(Ein Beweis für eine allgemeinere Form dieser Beziehung findet sich in [127, 128], so daß hier darauf verzichtet wird.)

Dieses Lemma ermöglicht es, eine ganze Reihe von Schritten zurückzugehen, um dann wieder zu einer kanonischen Abbildung zu gelangen. Da der Maximaltest aber i.a. bereits beim ersten g abbricht, bei dem f größer wird, wird so nicht immer der größtmögliche Rückwärtsschritt erreicht.

Wir wollen nun noch eine Erweiterung des Paradigmas der Ordnungstreuen Erzeugung vorstellen. Man kann nämlich über 1.3.4 und 1.3.5 auch zu einem Verfahren gelangen, das die Transversalen zu mehreren Gruppenoperationen simultan berechnet.

1.3.7 Korollar (Simultane Ordnungstreue Erzeugung). Sei G eine endliche Gruppe, die auf einer endlichen Kette $(X_i)_{i \in \underline{n}}$ von Mengen mit $n \in \mathbb{N}$,

$$|X_i| = i \quad \text{und} \quad X_i \subset X_{i+1} \text{ für } 1 \leq i \leq n-1$$

operiert, und Y eine weitere endliche Menge. Sei A ein Algorithmus, der für alle $i \in \{1, \dots, n-1\}$ für jedes $f \in Y^{X_i}$ die leere Menge oder eine Menge $A(f) \subseteq Y^{X_{i+1}}$ in absteigender Reihenfolge erzeugt, so daß für die Transversalen $\mathcal{T}^{(i)}$ der Bahnen $Y^{X_i} // G$ die folgenden Bedingungen gelten:

- $\mathcal{T}^{(i+1)} \subseteq \bigcup_{f \in \mathcal{T}^{(i)}} A(f)$.
- Für alle $f_1, f_2 \in \mathcal{T}^{(i+1)}$ mit $f_1 < f_2$ gilt, daß aus $f_1 \in A(f'_1)$ und $f_2 \in A(f'_2)$ folgt, daß $f'_1 < f'_2$ ist.
- Für alle $f \in \mathcal{T}^{(i)}$ gilt, daß $\tilde{f}(x) = f(x)$ für alle $x \in X_i$, $\tilde{f} \in A(f)$.

Dann erhält man Transversalen $\mathcal{T}^{(i)}$ der Bahnen $Y^{X_i} // G$ durch folgendes Vorgehen:

- i) Bestimme $\mathcal{T}^{(1)}$.
- ii) Für $i \in \{1, \dots, n-1\}$ bestimme für $f^{(i)} \in \mathcal{T}^{(i)}$ alle $A(f^{(i)})$ und berechne $\mathcal{T}^{(i+1)}$ durch Vereinigung der $A(f^{(i)})$ und Entfernung aller nichtkanonischen Elemente.

Auf diese Weise läßt sich nicht nur eine Transversale von $Y^X // G$, sondern gleichzeitig von allen $Y^{X_i} // G$ mit $i \in \underline{n}$ berechnen. Ein Beispiel für diese Vorgehensweise werden wir in 4.2.3 antreffen.

Kapitel 2 Graphen und Moleküle

2.1 Grundlegende Definitionen

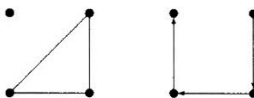
In diesem Abschnitt wollen wir die grundlegenden Definitionen aus der Graphentheorie vorstellen, auf denen wir später aufbauen werden (nach [67, 117]).

2.1.1 Definition.

- i) Sei V eine nichtleere Menge und $E \subseteq V \times V$ eine Relation. Dann heißt das Paar $G = (V, E)$ ein Graph. Die Elemente von V heißen dabei Ecken von G , die Elemente von E Kanten von G .
- ii) Ein Graph G heißt ungerichtet, falls E symmetrisch ist, d.h. mit $E(v_1, v_2)$ gilt auch $E(v_2, v_1)$, sonst gerichtet.
- iii) Zwei Ecken $u, v \in V$ heißen adjazent (oder verbunden), falls die Relation $E(u, v)$ besteht.

In dieser Arbeit wollen wir nur endliche Graphen behandeln, d.h. V sei endlich. Es ist aber prinzipiell auch möglich, unendliche Graphen zu untersuchen. Zudem wollen wir uns auf schleifenfreie Graphen beschränken. Da V endlich ist, sind die Elemente von V nummeriert, also auch der Graph. Wir werden auf diese Tatsache später zurückkommen.

2.1.2 Beispiel. Die folgenden Abbildungen zeigen einen ungerichteten und einen gerichteten Graphen (ohne die Nummerierung ihrer Ecken). Die Richtung wird dadurch angegeben, daß die Kanten als Pfeile gezeichnet sind:



◇

2.1.3 Definition.

- i) Ein Graph $T = (V', E')$ heißt Teilgraph von $G = (V, E)$, falls $V' \subseteq V$ und $E' \subseteq E \cap V' \times V'$.
- ii) Unter einem Kantenzug der Länge k in einem Graphen G von der Ecke u zur Ecke u_{k+1} wollen wir eine Folge von Ecken (u_1, \dots, u_{k+1}) verstehen, für die gilt:

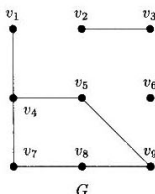
$$\forall i \in \{1, \dots, k\} : E(u_i, u_{i+1}).$$

- iii) Ein Kantenzug heißt offen, falls $u_1 \neq u_{k+1}$, sonst geschlossen.
- iv) Ein Pfad ist ein offener Kantenzug, in dem alle Ecken paarweise verschieden sind.

- v) Ein Kreis (oder Ring) ist ein geschlossener Kantenzug, in dem alle Ecken bis auf die Randpunkte paarweise verschieden sind. Ein Graph, der keinen Kreis enthält, heißt kreisfrei.
- vi) Ein Graph $G(V, E)$ heißt zusammenhängend, falls für alle Ecken $u, v \in V$ ein Pfad von u nach v existiert.
- vii) Einen maximalen zusammenhängenden Teilgraph von G nennen wir Zusammenhangskomponente.

Ein zusammenhängender Graph hat somit nur eine Zusammenhangskomponente. Zudem bemerken wir ohne Beweis (s. [117, Satz 2.3.3]), daß jeder Graph die Vereinigung seiner Zusammenhangskomponenten ist.

2.1.4 Beispiel. Wir wollen diese Begriffe nun an einem Beispiel erläutern:



Die Ecken $(v_1, v_4, v_5, v_7, v_8, v_9)$ bilden einen zusammenhängenden Teilgraphen T . Insgesamt besteht G aus drei Zusammenhangskomponenten; neben T sind dies (v_2, v_3) und (v_6) . G enthält den Kreis $(v_4, v_5, v_8, v_9, v_4)$. Der Kantenzug (v_1, v_4, v_7) bildet einen Pfad. ◊

2.1.5 Definition.

- i) Ein kreisfreier, zusammenhängender Graph heißt Baum.
- ii) Einen Baum, bei dem eine Ecke ausgezeichnet ist, nennt man Wurzelbaum, die ausgezeichnete Ecke dann Wurzel. Ist der Baum ein gerichteter Graph, dann darf die Wurzel w nur Anfangspunkt von Kanten sein, d.h. es gibt kein $v \in V$, so daß die Relation $E(v, w)$ besteht.
- iii) Ist ein Graph G nicht zusammenhängend und jede seiner Zusammenhangskomponenten ein Baum, so heißt G Wald.

Es ist klar, daß in einem Baum zu je zwei Ecken $u, v \in V$ genau ein Pfad von u nach v existiert.

2.1.6 Bemerkung. Da wir im folgenden Graphen mit den gruppentheoretischen Methoden des vorangegangenen Kapitels untersuchen und konstruieren wollen, formen wir die (zweistellige) Kantenrelation aus 2.1.1 in eine Abbildung (also eine dreistellige Relation mit eindeutiger Zuordnung) um.

Sei $G = (V, E)$ ein (ungerichteter*) numerierter Graph mit $|V| = p$ Punkten. Dann entspricht seine Kantenrelation der Abbildung

$$\gamma : \underline{p}^{[2]} \rightarrow \{0, 1\},$$

wobei $\underline{p}^{[2]}$ alle zweielementigen Teilmengen von \underline{p} seien und $\gamma(\{i, j\}) = 1$, falls $E(v_i, v_j)$ erfüllt ist, und $\gamma(\{i, j\}) = 0$ sonst.

Somit können wir ohne Einschränkung G mit γ identifizieren.

In Erweiterung dieser Bemerkung können wir folgenden, für Anwendungen in der Chemie besonders wichtigen Begriff definieren:

* Für gerichtete Graphen, die wir im folgenden kaum benötigen werden, betrachtet man statt der Teilmengen $\underline{p}^{[2]}$ geordnete Paare.

2.1.7 Definition. Ein numerierter m -Graph (Multigraph) mit p Punkten ist eine Abbildung

$$\gamma : \underline{p}^{[2]} \rightarrow \{0, 1, \dots, m\}$$

Wird einem Punktepaar $\{i, j\}$ der Wert r zugeordnet, so bedeutet dies, daß die Punkte durch eine r -fache Kante miteinander verbunden sind. (Ist $r = 0$, sind die Punkte unverbunden.)

Der Einfachheit halber setzen wir oft $m := \{0, \dots, m - 1\}$.

Die S_p operiert auf den Punkten und induziert damit eine Operation auf den Graphen:

$$(2.1.8) \quad m^{\underline{p}^{[2]}} \times S_p \rightarrow m^{\underline{p}^{[2]}}, (\gamma, \pi) \mapsto \gamma^\pi$$

Dabei ist das Bild die Abbildung $\gamma^\pi = \gamma^\pi$ mit $\gamma^\pi(\{i, j\}) = \gamma(\{i^{\pi^{-1}}, j^{\pi^{-1}}\})$.

Von besonderer Bedeutung sind die Permutationen, welche die Bindungsverhältnisse des Graphen unverändert lassen:

2.1.9 Definition. Der Stabilisator eines numerierten Multigraphen $\gamma \in m^{\underline{p}^{[2]}}$ heißt Automorphismengruppe von γ :

$$Aut(\gamma) := C_{S_p}(\gamma) = \{\pi \in S_p \mid \gamma^\pi = \gamma\}$$

2.1.10 Bemerkung. Da die Ecken aus \underline{p} auf natürliche Weise angeordnet sind, lassen sich Graphen und Multigraphen auch in Form von Matrizen $A \in \overline{\mathbb{N}}_{p,p}^{\overline{\mathbb{N}}}$ darstellen. Dabei setzen wir

$$a_{ij} = \gamma(\{i, j\})$$

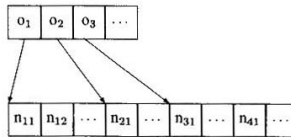
Die so festgelegte Matrix heißt *Adjazenzmatrix* und wird sehr häufig verwendet, um Graphen in einem Computer zu speichern.

Bei ungerichteten Graphen ist die Adjazenzmatrix symmetrisch, so daß es genügt, nur die obere oder untere Dreiecksmatrix zu speichern.

Die S_p operiert auch auf den $p \times p$ -Matrizen, nämlich durch gleichzeitiges Vertauschen von Zeilen und Spalten. Somit ist für $\pi \in S_p$ dann $f^\pi = f$ äquivalent zu $A^\pi = A$, und die Automorphismen von f bilden gerade den Stabilisator $C_{S_p}(A)$.

Adjazenzmatrizen sind zweifelsohne eine sehr einfache und leicht verwendbare Form zum Speichern von Graphen, aber sie werden zuweilen uneffizient bei größeren Punktezahlen, da sie dann oft nur sehr dünn besetzt sind. Eine andere Möglichkeit, die Kanten eines Graphen darzustellen, sind *Nachbarschaftslisten*, wo nur die Nachbarn jeder einzelnen Ecke (und der Grad der Kante bei Multigraphen) gespeichert werden.

Auch bei dem hierzu implementierten Programmsystem NEXTOR wurde (meistens) die Darstellung als Nachbarschaftsliste gewählt. Dabei werden alle Nachbarn in einer linearen Liste abgelegt, und ein zweiter Vektor enthält die Position der jeweils ersten Nachbarn der Ecken. (Mehr Details, etwa zur Umrechnung der Darstellungen, finden sich in [126].)



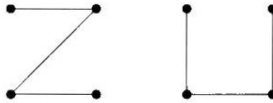
In vielen Anwendungen wird die Numerierung der Ecken aber nicht benötigt, ja sie ist sogar überflüssig. Beispielsweise bei Molekülen, die wir später auch als Graphen beschreiben werden, darf die Darstellung nicht von der Numerierung der Atome abhängen. Daher dürfen wir Graphen nur bis auf Isomorphie unterscheiden.

2.1.11 Definition.

- Zwei Graphen γ und γ' mit p Punkten heißen isomorph (in Zeichen $\gamma \simeq \gamma'$), falls es ein $\pi \in S_p$ gibt mit $\gamma^\pi = \gamma'$.
- Eine Isomorphieklassse numerierter Graphen, d.h. eine Bahn von S_p , heißt unnumerierter Graph Γ .

Man kann einen unnummerierten Graphen dadurch veranschaulichen, daß man einen Repräsentanten einer Isomorphieklasse wählt und die Numerierung entfernt.

2.1.12 Beispiel. Folgende nummerierte Graphen sind isomorph und daher Repräsentanten desselben Graphen:



◇

2.2 Datenstrukturen für Permutationsgruppen

Permutationsgruppen $G \leq S_n$ haben wegen $|S_n| = n!$ schon für relativ kleine Grade n oft eine so große Ordnung, daß sie nicht mehr in Form aller ihrer Elemente in einem Rechner gespeichert werden können. Wir benötigen daher Datenstrukturen, die es uns erlauben, solche G knapp und effizient zu speichern und trotzdem alle Elemente rekonstruieren zu können. Zwei derartige Konzepte, wie sie auch in der Implementation der in dieser Arbeit beschriebenen Algorithmen, im Programmsystem NEXTOR, verwendet wurden, wollen wir im folgenden vorstellen.

Wir wollen zunächst die Gruppe über die Transversalen bestimmter Linksnebenklassen darstellen. Dieses Konzept geht zurück auf C.C. SIMS [114]. Dazu betrachten wir ohne Einschränkung wieder die Operation von $G \leq S_n$ auf $\Omega := \underline{n}$.

2.2.1 Definition. Sei $\pi \in S_n$. Wir kürzen die punktwisen Stabilisatoren für $k \leq n$ ab als

$$G^{(k)} := C_G(\{\pi(1), \dots, \pi(k)\})$$

und setzen $G^{(0)} := G$. Die sich daraus ergebende Untergruppenkette

$$G = G^{(0)} \geq G^{(1)} \geq \dots \geq G^{(n-1)} = \{1\}$$

heißt Simskette oder Zentralisator-kette von G bezüglich π . Dabei heißt π bzw. die Folge $(\pi(1), \dots, \pi(n))$ auch Basis von G . •

2.2.2 Lemma. Sei $\{g_{k,1}, \dots, g_{k,l(k)}\}$ jeweils eine Transversale der Linksnebenklassen $G^{(k-1)}/G^{(k)}$ für $k \in \{1, \dots, n-1\}$, wobei $l(k)$ der Index von $G^{(k)}$ in $G^{(k-1)}$ sei. Dann existiert für jedes $g \in G$ eine eindeutige Darstellung

$$g = \prod_k^{n-1} g_{k,j_k} \quad \text{mit } 1 \leq j_k \leq l(k)$$

und für die Ordnung der Gruppe gilt:

$$|G| = \prod_k^{n-1} l(k)$$

(Beweis siehe [114].)

2.2.3 Beispiel. Wir betrachten die symmetrische Gruppe vom Grad 4:

$$\begin{aligned} S_4 = \{ & 1, (12), (13), (14), (23), (24), (34), \\ & (12)(34), (13)(24), (14)(23), \\ & (123), (124), (132), (134), (142), (143), (234), (243), \\ & (1234), (1243), (1324), (1342), (1423), (1432) \} \end{aligned}$$

Die entsprechenden Zentralisatoren zur Basis $(1, 2, 3, 4)$ sind:

$$\begin{aligned} G^{(0)} &= S_4 \\ G^{(1)} &= \{1, (23), (24), (34), (234), (243)\} \\ G^{(2)} &= \{1, (34)\} \\ G^{(3)} &= \{1\} \end{aligned}$$

Mögliche Transversalen sind

$$\begin{aligned} \mathcal{T}(G^{(0)}/G^{(1)}) &= \{1, (12), (13), (14)\} \\ \mathcal{T}(G^{(1)}/G^{(2)}) &= \{1, (23), (24)\} \\ \mathcal{T}(G^{(2)}/G^{(3)}) &= \{1, (34)\} \end{aligned}$$

Damit erhalten wir bspw. die Darstellungen $(123) = (12)(23)$ und $(1324) = (13)(24)(34)$. ◇

Wir benötigen also zum Speichern einer Permutationsgruppe in Form einer Simskette neben der Basis nur die Repräsentanten der Linksnebenklassen.

Oft haben wir die Situation, daß die Gruppe G nicht gegeben ist, sondern erst berechnet werden soll. Dabei erweist sich die Datenstruktur der Simskette, wie wir später sehen werden, als sehr hilfreich. Sie ist allerdings nicht die sparsamste Möglichkeit, eine Permutationsgruppe zu speichern.

2.2.4 Bemerkung. Nach 1.2.4 können die Bahn $\pi(k)^{G^{(k)}}$ und die Nebenklasse $G^{(k-1)}/G^{(k)}$ für $k \in \{1, \dots, n-1\}$ bijektiv aufeinander abgebildet werden. Ist nun ein $g \in S_n$ gegeben, so können wir mit Hilfe der Simskette leicht entscheiden, ob g in G liegt oder nicht. Ist bereits $\pi(1)^g$ nicht in $\pi(1)^{G^{(1)}}$, so gilt auch $g \notin G$. Andernfalls existiert ein $g_1 \in \mathcal{T}(G^{(1)}/G^{(2)})$ mit $\pi(1)^{g_1^{-1}} = \pi(1)$. Diesen Prozeß können wir solange fortsetzen, bis wir feststellen, daß $g \notin G$, oder bis wir die Darstellung von g aus 2.2.2 gefunden haben.

Sehen wir uns einen Zwischenschritt an: Sei $\omega := \pi(k)^g$. Dann gilt, daß $\omega^{G^{(k-1)}} \subseteq \pi(k)^{G^{(k-1)}}$, falls es ein $g_k \in G^{(k-1)}$ gibt mit $\pi(k)^{g_k} \in \omega^{G^{(k-1)}}$. Daher genügt es eigentlich, für die Bahn $\omega^{G^{(k-1)}}$ nur diese Permutation g_k zu speichern. Falls es einen Punkt $\bar{\omega} \in \omega^{G^{(k-1)}}$ gibt, der von allen Punkten der Bahn aus erreicht werden kann, müssen wir uns nur zusätzlich merken, daß $\bar{\omega}^{g_k^{-1}} = \pi(k)$.

Diese Überlegungen führen uns zu einer sehr kompakten Darstellung, den *markierten Verzweigungen* nach M. JERRUM [63].

2.2.5 Definition. Sei $n \in \mathbb{N}$ und $G \leq S_n$ eine Permutationsgruppe, $\pi \in S_n$ sowie

$$G \geq G^{(1)} \geq \dots \geq G^{(n-2)} \geq \{1\}$$

eine Simskette von G bezüglich π .

- Ein Wald $B = (\underline{n}, E)$ aus gerichteten Bäumen, bei dem die Kanten die Form $(\pi(i), \pi(j))$ mit $i < j$ haben, heißt *Verzweigung (branching)* auf \underline{n} bezüglich π .
- Ein Tripel $B = (\underline{n}, E, \sigma)$ heißt *markierte Verzweigung (labeled branching)* für G bezüglich π , falls gilt:
 - i) (\underline{n}, E) ist eine Verzweigung.
 - ii) $\sigma : E \rightarrow S_n$, $(\pi(i), \pi(j)) \mapsto \sigma_{i,j}$ ist eine Kantenbeschriftung der Verzweigung. Dabei heißt $\sigma_{i,j}$ *Kantenmarke von $\pi(j)$ in B und $\pi(i)$ Vater von $\pi(j)$* .
 - iii) Für alle $i, j \in \underline{n}$ gilt: $\sigma_{i,j} \in G^{(i)}$ und $\pi(i)^{\sigma_{i,j}} = \pi(j)$.

Eine markierte Verzweigung heißt *vollständig*, wenn zusätzlich gilt:

- iv) Falls $\omega \in \pi(i)^{G^{(i)}}$, dann existiert ein gerichteter Pfad in B von $\pi(i)$ nach ω .

Eine vollständige markierte Verzweigung repräsentiert damit ganz G , d.h. es gilt $\langle \sigma(E) \rangle = G$.

Der Existenzsatz für markierte Verzweigungen wird hier ohne Beweis angegeben – ein ausführlicher Beweis findet sich bei Jerrum [63].

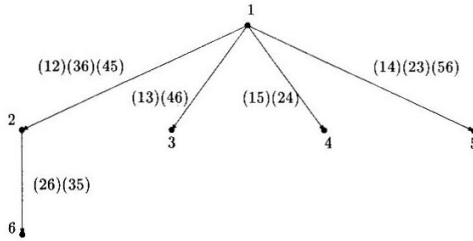
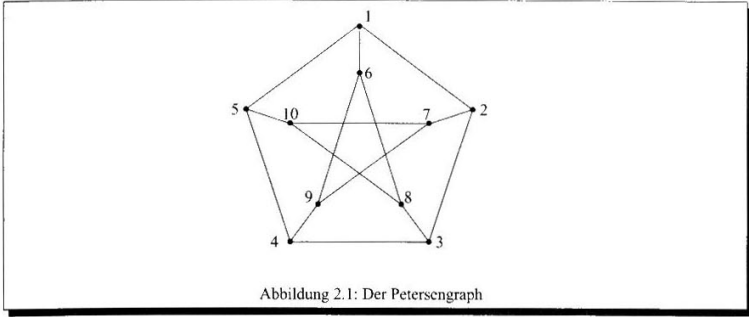
2.2.6 Satz. Sei $G \leq S_n$ eine Permutationsgruppe. Dann existiert zu jedem $\pi \in S_n$ eine vollständige markierte Verzweigung für G bzgl. π .

2.2.7 Beispiele.

- i) Als Beispiel betrachten wir die Diedergruppe D_6 . Mögliche Transversalen (bzgl. der identischen Anordnung) der Nebenklassen in der Simskettendarstellung sind:

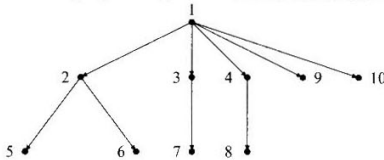
$$\begin{aligned} \mathcal{T}(G^{(0)}/G^{(1)}) &= \{1, (12)(36)(45), (13)(46), (14)(23)(56), \\ &\quad (15)(24), (16)(25)(34)\} \\ \mathcal{T}(G^{(1)}/G^{(2)}) &= \{1, (26)(35)\} \\ \mathcal{T}(G^{(2)}/G^{(3)}) &= \{1\} \end{aligned}$$

Die vollständige markierte Verzweigung für D_6 (bzgl. der identischen Anordnung) ist dann:



ii) Ein etwas größeres Beispiel ist die Automorphismengruppe A des sog. *Petersengraphs* (s. etwa [95]) in Abb. 2.1, welche die Ordnung 120 hat.

Die vollständige markierte Verzweigung zu A bzgl. der identischen Anordnung hat folgende Form:



Die Kantenmarken sind hier zugunsten der Übersichtlichkeit weggelassen. Sie lauten:

$$\begin{aligned}
 \sigma_{12} &= (12)(35)(67)(8, 10) \\
 \sigma_{13} &= (13)(45)(68)(9, 10) \\
 \sigma_{14} &= (14)(23)(69)(78) \\
 \sigma_{19} &= (19)(24)(57) \\
 \sigma_{1,10} &= (1, 10, 6, 7)(2589)(34) \\
 \sigma_{25} &= (25)(34)(7, 10)(89) \\
 \sigma_{26} &= (265)(3, 8, 10, 7, 9, 4) \\
 \sigma_{37} &= (37)(49)(56)(8, 10) \\
 \sigma_{48} &= (48)(56)(9, 10)
 \end{aligned}$$

2.2.8 Bemerkung. Für die Darstellung einer Permutationsgruppe $G \leq S_n$ als Linksnebenklassentransversalen

$$\mathcal{T}(G^{(0)}/G^{(1)}), \dots, \mathcal{T}(G^{(n-2)}/G^{(n-1)})$$

benötigt man insgesamt $n(n+1)/2$ Permutationen. Geht man von der (etwas großzügigen, aber realistischen) Annahme aus, daß eine Permutation durch einen n -dimensionalen Vektor im Rechner dargestellt wird, so beträgt der Aufwand für das Abspeichern einer einzelnen Permutation $\mathcal{O}(n)$. Der Gesamtaufwand für das Speichern einer Gruppe in der Form 2.2.2 ist somit $\mathcal{O}(n^2)$; der Test aus 2.2.4, ob eine gegebene Permutation ein Element der Gruppe ist, benötigt einen Zeitaufwand von $\mathcal{O}(n^2)$.

Bei einer markierten Verzweigung müssen (höchstens) $n-1$ Kantenbeschriftungen gespeichert werden. Unter obiger Annahme ist der Gesamtaufwand für das Speichern einer Gruppe in der Form 2.2.5 damit $\mathcal{O}(n^2)$.

Wir betrachten nun wieder den Test, ob eine gegebene Permutation g Element von G ist. Dabei kann die Methode aus 2.2.4 analog übernommen werden. Im schlimmsten Fall muß man einen Pfad der Länge $n-1$ entlanggehen und in $\mathcal{O}(n^2)$ Zeit das Produkt der Kantenbeschriftungen bilden, um eine Permutation in $G^{(i-1)}$ zu finden, die $\pi(i)$ auf $\pi(j)$ abbildet. Das würde die Zeitkomplexität für den Test gegenüber der Simskettendarstellung auf $\mathcal{O}(n^3)$ erhöhen. Um diesen Nachteil zu beseitigen, markiert man zusätzlich die Ecken der Verzweigung mit Abbildungen $\tau: \underline{n} \rightarrow S_n$ gemäß folgenden Regeln:

- i) Führt zu $\pi(j)$ keine Kante hin, so ist $\tau_j := \text{id}$.
- ii) Führt die (eindeutig bestimmte) Kante $(\pi(i), \pi(j))$ zu $\pi(j)$ hin, so ist $\tau_j := \tau_i \sigma_{ij}$. Damit ist τ_j das Produkt der Kantenmarken von der Wurzel der Zusammenhangskomponente bis $\pi(j)$.

Diese Abbildungen τ_j nennt man *Eckenmarken*. Dadurch wird die Zeitkomplexität für das Testen auf Mitgliedschaft auf $\mathcal{O}(n^2)$ vermindert. Allerdings benötigt man nun (höchstens) $2n+1$ Permutationen zum Abspeichern einer vollständigen markierten Verzweigung.

Wir wollen nun untersuchen, wie eine vollständige markierte Verzweigung aus einer Simskette berechnet werden kann.

2.2.9 Algorithmus (Markierte Verzweigung aus einer Simskette). Sei $n \in \mathbb{N}$ und $G \leq S_n$ eine Permutationsgruppe, $\pi \in S_n$ sowie

$$G \geq G^{(1)} \geq \dots \geq G^{(n-2)} \geq \{1\}$$

eine Simskette von G bezüglich π . Sei ferner $i \in \{1, \dots, n-2\}$ und B_{i+1} eine vollständige markierte Verzweigung für $G^{(i+1)}$.

Durchlaufe alle $g \in \mathcal{T}(G^{(i)}/G^{(i+1)})$ und führe für jedes g durch:

- i) Wähle j so, daß $\pi(j) = \pi(i)^g$.
- ii) Falls zu $\pi(j)$ bislang keine Kante hinführt, so füge eine neue Kante $(\pi(i), \pi(j))$ ein und setze $\sigma_{ij} := g$.

Passe nun die Eckenmarken entsprechend an. Dann ist das so konstruierte B_i eine vollständige markierte Verzweigung für $G^{(i)}$. ▽

(Die Richtigkeit dieses Verfahrens wird in [63] bewiesen.)

2.2.10 Bemerkung. Wichtig für die praktische Anwendung markierter Verzweigungen ist auch, die Basis π wechseln zu können, d.h. eine markierte Verzweigung zu $\pi' = (\pi(1), \dots, \pi(r-1), \pi(s), \pi(r), \pi(r+1), \dots, \pi(s-1), \pi(s+1), \dots, \pi(n))$ zu berechnen, in der die Basis zyklisch von r bis s verschoben wurde. Die dazu notwendigen Algorithmen sind in [22, 56] beschrieben, so daß hier auf Details verzichtet wird. Im Programmsystem NEXTOR wird die Implementierung aus [56] verwendet.

2.3 Automorphismen von Teilgraphen

Bei der Diskussion verschiedener graphentheoretischer Algorithmen im zweiten Teil dieser Arbeit werden wir gelegentlich auf die Situation treffen, daß aus der Automorphismengruppe eines Graphen nur die Abbildungen betrachtet werden sollen, die einen gegebenen Teilgraphen permutieren. Dieses Problem wollen wir daher an dieser Stelle allgemein untersuchen.

Gegeben sei ein numerierter Multigraph $\gamma \in m^{\mathbb{P}^{[2]}}$ mit p Ecken, wobei $m := \{0, 1, \dots, m-1\}$ mit $m, p \in \mathbb{N}$. Ferner sei

$$(2.3.1) \quad \eta \in m^{\mathbb{T}^{[2]}} \text{ mit } T \subseteq \mathbf{p}, \forall i, j \in T : \eta(\{i, j\}) = \gamma(\{i, j\}).$$

(Dies ist die Übertragung des Teilgraphenbegriffs aus 2.1.3.i auf Multigraphen; abkürzend schreiben wir dafür $\eta \subseteq \gamma$.)

Weiterhin wollen wir die Automorphismengruppe von γ bezeichnen als $A := \text{Aut}(\gamma)$. Für die Automorphismen π , die η fest lassen, gilt sicher: $\pi \in N_A(\eta)$. Nun kann es zwei $\pi, \pi' \in N_A(\eta) \subseteq A$ geben, mit $\pi \downarrow \eta = \pi' \downarrow \eta$ aber $\pi \neq \pi'$. Daher müssen wir die Restklassen nach dem Zentralisator von η betrachten. Dabei stellen wir fest:

2.3.2 Satz. *Die Automorphismen eines numerierten Multigraphen γ mit Automorphismengruppe A , die einen Teilgraphen $\eta \subseteq \gamma$ invariant lassen, induzieren auf η eine Automorphismengruppe E isomorph zu*

$$N_A(\eta)/C_A(\eta).$$

Beweis: Wir bezeichnen die Einschränkung von $N_A(\eta)$ auf η als f , d.h.

$$f : N_A(\eta) \rightarrow \text{Aut}(\eta),$$

wobei $E := \text{Im}(f)$ die induzierte Permutationsgruppe sei.

Diese Einschränkung ist offensichtlich ein Homomorphismus. Sein Kern, also die Elemente, die auf die Identität aus E abgebildet werden, ist gerade der punktweise Stabilisator $C_A(\eta)$, denn nur seine Permutationen lassen η völlig unverändert. Damit folgt nach dem Homomorphiesatz, daß die Menge der Restklassen $N_A(\eta)/C_A(\eta)$ eine zu $\text{Im}(f) = E$ isomorphe Gruppe bildet, was die Behauptung zeigt. \square

2.3.3 Bemerkung. Um den Algorithmus für die Berechnung der Teilgraphen vorstellen zu können, benötigen wir noch ein Verfahren, das eine vollständige markierte Verzweigung aus einer Menge von Erzeugern berechnet. Ein solches wird ebenfalls bei [63] beschrieben und bei [81] ausführlich aufgearbeitet. Daher wollen wir hier nur die wesentlichen Aspekte kurz ansprechen.

Gegeben sei eine Menge von Erzeugern $\{g_1, \dots, g_m\}$, die eine Permutationsgruppe $G \leq S_k$ mit $k > m$ erzeugen, und eine Basispermutation $\pi \in S_k$. Zunächst wird dann eine markierte Verzweigung ohne Kanten erzeugt. Anschließend durchläuft man alle $i \in \underline{k}$ und führt dabei jeweils die folgenden Operationen durch:

- i) Man berechnet eine Transversale $\mathcal{T}(G^{(i-1)}/G^{(i)})$ und fügt diese in die markierte Verzweigung ein, ohne die zuvor eingefügten zu beeinträchtigen.
- ii) Aus dieser Transversale und den Erzeugern von $G^{(i-1)}$ wird dann die Menge der sog. Schreier-Erzeuger von $G^{(i)}$ bestimmt, die anschließend durch ein „Siebverfahren“ (Sifting) zu einem Erzeugendensystem von $G^{(i)}$ mit höchstens $k-1$ Elementen reduziert wird.

Nach Durchführung dieses Verfahrens erhält man eine vollständige markierte Verzweigung, die $G = \langle g_1, \dots, g_m \rangle$ repräsentiert.

Nun können wir die Teilgraphenautomorphismen berechnen:

2.3.4 Algorithmus (Teilgraphenautomorphismen). Gegeben sei ein numerierter Multigraph $\gamma \in m\mathbb{E}^{|\mathbf{p}|}$ mit p Ecken, ein Teilgraph $\eta \subseteq \gamma$ über $T = \{t_1, \dots, t_r\} \subseteq \mathbf{p}$, und die Automorphismengruppe $\text{Aut}(\gamma)$ als vollständige markierte Verzweigung B zur Basis $\pi \in S_p$.

- i) Wechsle mit 2.2.10 die Basis von B zu $\tilde{\pi}$ so, daß anschließend für $j \in \mathbf{r}$ gilt: $\tilde{\pi}(j) = t_j$.
- ii) Initialisiere die Erzeugermenge S mit $S \leftarrow \emptyset$. Durchlaufe alle $j \in \mathbf{r}$ führe jeweils durch:
 - Führt zu t_j eine Kante (etwa von t_i) und ist $\sigma_{ij} \in N_{\text{Aut}(\gamma)}(\eta)$, so setze $S \leftarrow S \cup \{\tilde{\sigma}_{ij}\}$, wobei $\tilde{\sigma}_{ij}$ aus σ_{ij} durch Einschränkung auf T entstehe.
- iii) Berechne die von S erzeugte markierte Verzweigung nach 2.3.3.

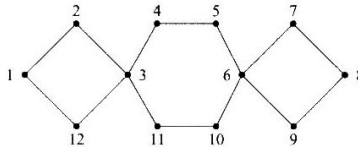
▽

Wir wollen noch kurz den Aufwand für dieses Verfahren untersuchen. Ausführliche Beweise zu den einzelnen Angaben finden sich in der zitierten Literatur.

2.3.5 Bemerkung.

- Der Aufwand für das Wechseln der Basis beträgt $\mathcal{O}(p^2)$ [22, 56]. Dies muß im schlimmsten Fall r mal durchgeführt werden, so daß der Gesamtaufwand für Schritt i) $\mathcal{O}(rp^2)$ ist.
- In Schritt ii) wird die Verzweigung linear durchlaufen; der Aufwand ist also $\mathcal{O}(r)$.
- Das Berechnen einer markierten Verzweigung aus einer Erzeugermenge in Schritt iii) braucht $\mathcal{O}(r^5)$ Zeit [63]. Im praktischen Einsatz in NEXTOR sind die Teilgraphen allerdings meist nicht allzu groß und verfügen zudem auch nur über eine kleine, oft sogar triviale, Automorphismengruppe.

2.3.6 Beispiel. Gegeben sei folgender Graph



Seine Automorphismengruppe (Ordnung 16) hat als vollständige markierte Verzweigung (bzgl. der identischen Anordnung) nur vier Kanten mit den Beschriftungen

$$\begin{aligned} \sigma_{18} &= (18)(27)(36)(45)(9,12)(10,11) \\ \sigma_{2,12} &= (2,12) \\ \sigma_{4,11} &= (4,11)(5,10) \\ \sigma_{79} &= (79) \end{aligned}$$

Betrachten wir die Teilmenge von Ecken $T := \{3, 4, 5, 6, 10, 11\}$. Hier zeigt sich der Unterschied zwischen $\text{Aut}(\eta^{(T)})$ und $N_A(\eta^{(T)})/C_A(\eta^{(T)})$ bzw. E wie in 2.3.2. Der Teilgraph η hat, für sich genommen, $\text{Aut}(\eta^{(T)}) = D_6$ (s. 2.2.7 i), während E nur eine Untergruppe mit Index 3 davon ist. Bei der Durchführung des Algorithmus 2.3.4 muß zunächst die Basis zu

$$\tilde{\pi} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 3 & 4 & 5 & 6 & 10 & 11 & 1 & 2 & 7 & 8 & 9 & 12 \end{pmatrix}$$

gewechselt werden. Die Erzeugendenmenge S ergibt sich (in neuer Numerierung) dann als

$$S = \{(14)(23)(56), (26)(35)\}$$

◇

2.4 Molekulare Graphen

In diesem Abschnitt wollen wir den Bezug der bisher definierten mathematischen Strukturen zur naturwissenschaftlichen Anwendung, insbesondere in der Chemie herstellen.

Seitdem 1864 Alexander CRUM BROWN Moleküle mit Hilfe von Graphen beschrieben hat, besteht eine enge Verbindung zwischen Chemie und Graphentheorie. Das Isomerieproblem war auch die Motivation für die Entwicklung der algebraischen Kombinatorik; eine der bekanntesten Arbeiten ist dabei die von G. PÓLYA [103]. Weitere geschichtliche Hintergründe finden sich etwa bei [17].

Es ist anzumerken, daß längst nicht für alle Moleküle eine einfache graphentheoretische Beschreibung möglich ist. Die Atome als Ecken aufzufassen, ist relativ universell. Bei den Typen möglicher Bindungen gibt es allerdings große Unterschiede. Ionen- und Metallbindungen lassen sich ebenso wie Komplexbindungen kaum als Kanten interpretieren. Daher beschränken sich die graphentheoretischen Modelle der Moleküle meist auf die *organische Chemie*; in diesem Bereich, der sich mit der Chemie der Kohlenstoffverbindungen beschäftigt, betrachtet man vorwiegend sog. *kovalente* Bindungen (s. etwa [82]), die sich eindeutig als Verknüpfung zwischen zwei Atomen und daher auch als Kanten eines Graphen darstellen lassen.

Zunächst wollen wir die graphentheoretische Beschreibung von Molekülen definieren.

2.4.1 Definition. Sei $\gamma \in \mathfrak{m}^{\mathbb{P}^2}$ ein numerierter, schleifenfreier Multigraph mit p Ecken. Dann heißt

$$d_i(\gamma) = \sum_{j=1}^p \gamma(\{i, j\})$$

Grad der Ecke i . Den Vektor $d(\gamma) = (d_1(\gamma), \dots, d_p(\gamma))$ bezeichnet man als Gradvektor des Graphen γ . •

Für andere Arten von Graphen läßt sich der Grad entsprechend definieren. Bei einem schlichten Graph entspricht der Grad einer Ecke der Anzahl der Kanten, auf denen diese Ecke liegt.

Graphen, bei denen gilt $d_1 = \dots = d_p$, heißen *regulär* (wie etwa der Petersengraph in Abb. 2.1).

Stellt man einen Graphen mit Hilfe einer Adjazenzmatrix A dar (s. 2.1.10), so entspricht der Grad von i gerade der Summe der Einträge der i -ten Zeile: $d_i(A) = \sum_j a_{ij}$.

2.4.2 Beispiel. Für den Graphen aus Beispiel 2.1.4 gilt: $d_1 = d_2 = d_3 = 1$, $d_4 = 3$, $d_5 = d_7 = d_8 = d_9 = 2$, $d_6 = 0$. ◊

2.4.3 Definition. Sei $\Psi = \{\psi_1, \dots, \psi_n\}$ eine endliche Menge und $\gamma \in \mathfrak{m}^{\mathbb{P}^2}$ ein Graph. Dann heißt eine Abbildung

$$\beta: \mathbb{P} \rightarrow \Psi \quad \text{mit} \quad \beta(i) = \beta(j) \Rightarrow d_i = d_j \quad \forall i, j \in \mathbb{P}$$

Beschriftung (der Ecken) von γ . •

Hierbei kann Ψ eine Menge von Farben, Nummern oder Symbolen sein. Die Bedingung stellt sicher, daß gleichbeschriftete Ecken auch gleichen Grad haben.

2.4.4 Definition. Ein Paar (γ, β) aus einem numerierten, schleifenfreien, ungerichteten, zusammenhängenden Multigraphen $\gamma \in \mathfrak{m}^{\mathbb{P}^2}$ und einer Beschriftung $\beta \in \Psi^{\mathbb{P}}$ heißt molekularer Graph. •

Eine einfache Kante entspricht einer σ -Bindung, eine Doppelkante einer $\sigma\pi$ -Bindung und eine Dreifachkante einer $\sigma\pi^2$ -Bindung [82]. Höhere Bindungen sind in der organischen Chemie selten, lassen sich aber entsprechend ausdrücken.

Bei molekularen Graphen ist meist $\Psi = \{C, H, O, N, \dots\}$, also die Elementsymbole des Periodensystems, oder der Einfachheit halber $\Psi = \underline{n}$ bei n verschiedenen Elementen.

2.4.5 Bemerkung. Eine Sonderrolle nehmen Wasserstoffatome ein. Da Wasserstoffe nur eine Bindung haben können, im Graph also Ecken vom Grad 1 sind, sie aber gleichzeitig fast überall vorkommen, werden sie meist weggelassen bzw. nicht als Ecken sondern als zusätzliche Beschriftung aufgefaßt. Man spricht dann von *wasserstoffreduzierten* molekularen Graphen. Auch im folgenden werden wir uns meist dieser Darstellung bedienen.

Allerdings ist es im allgemeinen nicht erforderlich, die Anzahl der H-Atome an einem anderen Atom explizit anzugeben. Fast alle in der organischen Chemie vorkommenden Elemente haben einen festen Eckengrad, den man als *Valenz* bezeichnet. Diese ist beispielsweise bei Kohlenstoff 4, bei Stickstoff 3 und bei Sauerstoff 2. Hat ein Atom i im Graphen einen Grad d_i , der kleiner als seine Valenz val_i ist, so geht man, wenn nichts anderes angegeben ist, davon aus, daß für die Anzahl h_i der Wasserstoffe gilt:

$$h_i = val_i - d_i$$

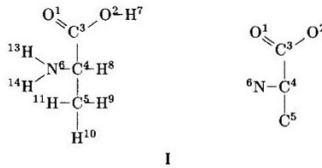
also die nichtgebundenen Valenzen durch H-Atome aufgefüllt sind.

Bei wasserstoffreduzierten Graphen ist die Bedingung $\beta(i) = \beta(j) \Rightarrow d_i = d_j$ aus 2.4.3 durch

$$\beta(i) = \beta(j) \Rightarrow val_i = val_j$$

zu ersetzen.

2.4.6 Beispiel. Die natürliche Aminosäure *L-Alanin* hat folgenden molekularen Graphen (links mit, rechts ohne Wasserstoffe):



I

Wir betrachten nur den wasserstoffreduzierten Graphen. Seine Adjazenzmatrix ist

$$\begin{pmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

und der Beschriftungsvektor ist $\beta = (O, O, C, C, C, N)$.

◇

Die symmetrische Gruppe S_p operiert natürlich auch auf den Abbildungen β ; daher können wir in Analogie zu 2.1.9 und 2.1.11 definieren:

2.4.7 Definition.

i) Der Normalisator eines molekularen Graphen (γ, β) heißt Automorphismengruppe von (γ, β) :

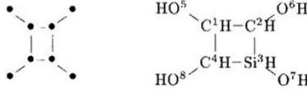
$$Aut(\gamma, \beta) := C_{S_p}(\gamma, \beta) = \{\pi \in S_p \mid \gamma^\pi = \gamma \wedge \beta^\pi = \beta\}$$

ii) Zwei molekulare Graphen (γ, β) und (γ', β') mit je p Punkten heißen isomorph, falls es ein $\pi \in S_p$ gibt mit $\gamma^\pi = \gamma'$ und $\beta^\pi = \beta'$.

iii) Zwei Ecken $i, j \in \underline{p}$ eines molekularen Graphen (γ, β) heißen topologisch äquivalent, falls $j \in Aut(\gamma, \beta)(i)$ gilt.

•

2.4.8 Beispiel. Wir betrachten (als Graph und als Molekül) 1,2,3,4-Tetrahydroxysilacyclobutan



II

Hat C den Typ 1 und Si den Typ 2, so kann die Beschriftung durch den Vektor $(1, 1, 2, 1, 1, 1, 1, 1)$ dargestellt werden. Die Automorphismengruppe des Graphen hat die Ordnung 8 und lautet (läßt man Fixpunkte weg):

$$\begin{aligned}
 &1 \\
 &(24)(68) \\
 &(12)(34)(56)(78) \\
 &(1234)(5678) \\
 &(13)(57) \\
 &(13)(24)(57)(68) \\
 &(14)(23)(58)(67) \\
 &(1432)(5876)
 \end{aligned}$$

Wären alle Beschriftungen gleich, wie etwa bei Tetrahydroxycyclobutan, so wäre dies auch die Automorphismengruppe des Moleküls. Wegen des Siliziums aber fallen eine Reihe von Abbildungen weg. So sind etwa die Spiegelungen an den Seitenhalbierenden gegenüberliegender Seiten keine selbsterhaltenden Abbildungen mehr. Als Automorphismengruppe des molekularen Graphen bleibt lediglich

$$\begin{aligned}
 &1 \\
 &(24)(68)
 \end{aligned}$$

◇

Für die Berechnung der Automorphismengruppe werden zunächst die Knoten des Graphen in Automorphismenpartitionen eingeteilt, d.h. es kann danach keine Automorphismen geben, die Knoten verschiedener Partitionen aufeinander abbilden. Die Klassifikation erfolgt in zwei Stufen:

2.4.9 Algorithmus (Knotenklassifizierung).

- i) Bilde jeweils die Summe über die dritten Potenzen der Zeileneinträge der Adjazenzmatrix. (Diese Gewichtsfunktion liefert eine eindeutige Gewichtung von Knoten bis zum Grad 7 und einer maximalen Kantenvielfachheit von 6.) Die ursprüngliche Einteilung der Knoten nach ihren Atomtypen wird zusätzlich in Betracht gezogen.
- ii) Iteriere einen daraus resultierenden klassenbezogenen linearen Gewichtsvektor so lange, bis keine weitere Verfeinerung mehr gefunden werden kann.

▽

Die Automorphismengruppe selbst wird nun mit Hilfe einer Sims-Kette (s. 2.2.2) berechnet und gespeichert. Für die gesuchte Automorphismengruppe heißt dann das Verfahren (Weitere Details finden sich in [85, 126]):

2.4.10 Algorithmus (Berechnung der Automorphismengruppe).

- i) Durchlaufe alle möglichen Stabilisatoren, d.h. alle $i \in \{1, \dots, p - 1\}$ (bei p Punkten) absteigend und dafür alle $j \in \{i + 1, \dots, p\}$.
- ii) Liegen i und j in derselben durch 2.4.9 erhaltenen Klasse, so untersuche, ob die Permutation der Zeilen i und j der Adjazenzmatrix eine zu dieser Matrix äquivalente liefert. Ist dies der Fall, gehe zu iii); sonst fahre mit den nächsten Knoten im Durchlauf fort.
- iii) Untersuche rekursiv, ob die Permutation, die durch Multiplikation der Transposition (i, j) mit einem der bereits gefundenen Stabilisator-Transversalen-Elemente entsteht, auch ein Automorphismus ist. Verläuft dieser Test positiv, speichere diese Information. Setze anschließend den Durchlauf fort.

▽

Teil II
Algorithmen

Kapitel 3 Verbindung zweier Graphen

3.1 Überblick

Der Hauptteil dieser Arbeit ist konstruktiven Problemen der Graphentheorie gewidmet. Dabei wollen wir allerdings nicht die häufig untersuchten Probleme wie die Konstruktion aller Graphen zu gegebenen Bedingungen (s. etwa [30, 36, 52, 56]) diskutieren, sondern neue Methoden für das *Verbinden* von Graphen finden. In vielen praktischen Anwendungen hat sich nämlich gezeigt, daß die ungeheure Vielfalt von Strukturen, die von Graphengeneratoren erzeugt werden können, nur schwer handzuhaben ist und durch Zusatzinformationen eingeschränkt werden muß. In diesem und in den folgenden Kapiteln wollen wir insbesondere drei Methoden der Verbindung von Graphen entwickeln:

- i) Verbindung durch Hinzufügen von Kanten zwischen den Graphen
- ii) Verschmelzung über alle gemeinsamen Teilgraphen (wobei bereits die Teilgraphensuche ein eigenständiges Problem ist)
- iii) Verbindung über Verknüpfungsregeln unter Hinzunahme oder Wegfall von Ecken, in Nachbildung einer chemischen Reaktion

Der Schwerpunkt der Anwendungen, die wir dabei betrachten und immer im Auge behalten wollen, liegt auf der organischen Chemie (vgl. S. 37), wo diese Verfahren besonders gut eingesetzt werden können. Unser Ziel sind Algorithmen, die folgenden Anforderungen genügen:

- Sie konstruieren schnell, vollständig und redundanzfrei.
- Sie berücksichtigen Nebenbedingungen des Anwenders (sog. *Restriktionen*) so früh wie möglich.

Zunächst wollen wir also ermitteln, auf wieviele wesentlich verschiedene Arten zwei Graphen miteinander verbunden werden können. Dies ist ein Teilproblem, das bei einer Reihe von Konstruktionsaufgaben auftaucht, aber auch als eigenständige Methode gewinnbringend eingesetzt werden kann. Eine Untersuchung in der hier vorgewonnenen Art konnte aber in der Literatur nicht gefunden werden.

Wir betrachten numerierte ungerichtete, schleifenfreie Multigraphen $\gamma_1 \in m^{\mathbb{P}^2}$ und $\gamma_2 \in m^{\mathbb{Q}^2}$ mit $m, p, q \in \mathbb{N}$, wobei m o.E. für beide gleich gewählt werden kann. Unsere Methodik wollen wir allgemein formulieren, so daß eine zusätzliche Beschriftung nicht notwendig, aber auch nicht hinderlich ist.

Wir setzen voraus, daß jede Ecke i von γ_k neben ihrem aktuellen Grad $d_i(\gamma_k)$ noch einen maximalen Grad besitzt, den wir in Analogie zu 2.4.5 als $val_i(\gamma_k)$ bezeichnen. Es ist klar, daß gilt:

$$d_i(\gamma_k) \leq val_i(\gamma_k) \leq m$$

für alle $i \leq p$ bzw. $i \leq q$.

Damit stehen wir vor folgendem

3.1.1 Konstruktionsproblem. Gegeben seien zwei Multigraphen $\gamma_1 \in m^{\mathbb{P}^2}$ und $\gamma_2 \in m^{\mathbb{Q}^2}$. Bestimme alle nichtisomorphen Graphen, die sich durch Zusammenfassen von γ_1 und γ_2 und Hinzufügen einer Kantenmenge ϵ ergeben, für die gilt:

$$ij \in \epsilon : \mathbb{P} \times \mathbb{Q} \rightarrow \{0, 1, \dots, m\}.$$

$$ii) d_i(\gamma_1) + \epsilon(i, j) < val_i(\gamma_1) \text{ und } d_j(\gamma_2) - \epsilon(i, j) \leq val_j(\gamma_2) \text{ f\"ur alle } (i, j) \in \underline{p} \times \underline{q}$$

Eine Erzeugung aller ϵ in dieser Form w\u00e4re zwar m\u00f6glich; es ergeben sich dabei aber schnell so gro\u00dfe Anzahlen von L\u00f6sungen, da\u00df es f\u00fcr die Anwendung kaum noch brauchbar ist. Aus deren Sicht ist es w\u00fcnschenswert, nicht nur die Zahl der L\u00f6sungen \u00fcberschaubar zu halten, sondern auch den Ort der Verbindung zwischen den Teilen restringieren zu k\u00f6nnen. Zu diesem Zweck f\u00fchren wir zus\u00e4tzliche fiktive Ecken ein, die dann die Verbindung zum Gegenpart bewerkstelligen sollen.

3.1.2 Bemerkung (Einf\u00fcgen fiktiver Ecken). Gegeben sei $\gamma \in m\mathbb{P}^{[2]}$. Setze $f^{(\gamma)} := \sum_{i=1}^p (val_i - d_i)$. Dann entsteht der erweiterte Graph

$$\hat{\gamma} : \{1, \dots, p + f^{(\gamma)}\}^{[2]} \rightarrow \{0, \dots, m\},$$

indem jede Ecke $i \in \underline{p}$ mit $val_i - d_i$ neuen bivalenten Ecken mit jeweils einer einwertigen Kante verbunden wird. Ist γ mit $\beta \in \Psi\mathbb{P}$ beschriftet, so sei $\hat{\beta}(i)$ f\u00fcr $i > p$ ein Symbol $\phi \notin \Psi$.

Diese Vorgehensweise l\u00e4\u00dft die Maximalzahl der M\u00f6glichkeiten zu. Der Anwender hat aber jederzeit die Option, die Position und Anzahl der fiktiven Ecken vorzugeben, um sich auf bestimmte L\u00f6sungen zu beschr\u00e4nken. Auf diese Weise erhalten wir ein sehr flexibles Konzept, das in dieser Hinsicht andere Methoden wie die direkte Erzeugung der Multigraphen oder die Reduktion auf schlichte Graphen mit dem Homomorphieprinzip [16, 85] hinter sich l\u00e4\u00dft.

Wenn wir im folgenden von erweiterten Graphen sprechen, so meinen wir damit stets *irgendeine* Erweiterung, nicht notwendig die maximale. Dabei ist dann $f^{(\gamma)}$ die tats\u00e4chliche Zahl fiktiver Ecken.

Die recht formale Prozedur der Erweiterung veranschaulichen wir an einem

3.1.3 Beispiel. Wir betrachten die Verbindung Cyclopropen, und zwar den wasserstoffreduzierten Graphen. Da hierbei alle Beschriftungen gleich sind, sind sie unerheblich. Der H\u00f6chstgrad (die Valenz) der Ecken ist 4.



III

Hier ist $f = 4$. Als maximale Erweiterung ergibt sich



Die erweiterte Adjazenzmatrix hat dann die Form

$$\left(\begin{array}{ccc|cccc} 0 & 2 & 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 1 & 0 & 0 \\ \hline 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array} \right)$$

(Die Linien trennen die urpr\u00fcnglichen von den fiktiven Ecken.) ◇

Die Symmetrie der Graphen \u00fcbertr\u00e4gt sich auf ihre Erweiterungen. Allerdings sollen Abbildungen, die eine fiktive Ecke auf eine urspr\u00fcngliche (oder umgekehrt) transportieren, nicht als selbsterhaltend gelten. Wir setzen daher f\u00fcr $\gamma \in m\mathbb{P}^{[2]}$

$$(3.1.4) \quad \text{Aut}(\hat{\gamma}) = N_C(\{p + 1, \dots, p + f^{(\gamma)}\}) \text{ mit } C = C_{S_{p + f^{(\gamma)}}}(\hat{\gamma})$$

Entsprechend auch f\u00fcr beschriftete Graphen.

Mit Hilfe des Begriffs der Graphenerweiterung haben wir unser Problem 3.1.1 transformiert. Gesucht sind n\u00e4mlich nun Abbildungen

$$(3.1.5) \quad \begin{aligned} & \hat{\epsilon} : \{1, \dots, f^{(\gamma_1)}\} \times \{1, \dots, f^{(\gamma_2)}\} \rightarrow \{0, 1\} \\ & \text{mit } \sum_{\substack{i=1 \\ f^{(\gamma_2)}}}^{f^{(\gamma_1)}} \hat{\epsilon}(i, j) \leq 1 \quad \forall j \in \{1, \dots, f^{(\gamma_2)}\} \\ & \sum_{j=1} \hat{\epsilon}(i, j) \leq 1 \quad \forall i \in \{1, \dots, f^{(\gamma_1)}\}. \end{aligned}$$

Bei der Konstruktion dieser Matrizen müssen wir auch die Symmetrie der zu verbindenden Graphen berücksichtigen. Da wir nur die fiktiven Ecken betrachten, genügt es, uns auf die Automorphismen der von ihnen gebildeten Teilgraphen $\hat{\eta}_k$ (mit $k = 1, 2$) zu beschränken. Nach Satz 2.3.2 sind deren Gruppen isomorph* zu

$$N_{\text{Aut}(\hat{\gamma}_k)}(\hat{\eta}_k) / C_{\text{Aut}(\hat{\gamma}_k)}(\hat{\eta}_k).$$

Wir wollen wiederum ihre isomorphen Bilder in S_{f_k} mit $f_k := f^{(\gamma_k)}$ betrachten, welche wir im folgenden als A_1 und A_2 bezeichnen werden.

Klar ist, daß A_1 auf den Zeilen und A_2 auf den Spalten von $\hat{\epsilon}$ operieren. Somit operiert $A_{1,2}$, die isomorphe Einbettung von $A_1 \times A_2$ in $S_{f_1 + f_2}$, die sich durch entsprechende Ummumerierung ergibt, auf der ganzen Matrix $\hat{\epsilon}$. Damit haben wir eine neue Fassung unseres Problems gefunden:

3.1.6 Lemma (Transformiertes Konstruktionsproblem). *Um alle nichtisomorphen Verbindungen zweier Multigraphen $\gamma_1 \in \mathfrak{m}^{\mathbb{P}^{[2]}}$ und $\gamma_2 \in \mathfrak{m}^{\mathbb{Q}^{[2]}}$ wie in 3.1.1 an den Stellen, die durch die Erweiterungen $\hat{\gamma}_1$ und $\hat{\gamma}_2$ gegeben sind, zu konstruieren, müssen höchstens alle Repräsentanten der Operation von $A_{1,2}$ auf der Menge der Abbildungen $\hat{\epsilon}$ von der Form 3.1.5 erzeugt werden.*

Dabei weist das Wort „höchstens“ auf die fehlende Eindeutigkeit hin, d.h. mehrere Abbildungen $\hat{\epsilon}$ können zum gleichen Verbindungsgraphen führen. Mit diesem Problem werden wir uns in Abschnitt 3.4 beschäftigen.

*Allerdings ist dies ein Sonderfall, denn per Definition ist $N_{\text{Aut}(\hat{\gamma}_k)}(\hat{\eta}_k) = \text{Aut}(\hat{\gamma}_k)$.

3.2 Aufspaltung nach dem Homomorphieprinzip

Durch die Umformung haben wir unser Konstruktionsproblem auf die Erzeugung eines Repräsentantensystems von Bahnen einer Gruppenoperation zurückgeführt. Bevor wir die eigentliche Generierung besprechen, wollen wir nach Möglichkeiten zur Vereinfachung suchen. Hierzu bietet sich die in 1.3.2.1 vorgestellte Technik der surjektiven Auflösung an.

Wir betrachten wieder Abbildungen

$$\tilde{\epsilon} : \underline{\mathbf{f}}_1 \times \underline{\mathbf{f}}_2 \rightarrow \{0, 1\}$$

von der Form 3.1.5. Die Gruppe A_2 operiert auf der Menge $\underline{\mathbf{f}}_2$ und zerlegt diese in r Bahnen X_1, \dots, X_r . Daraus ergeben sich Mengen von Abbildungen

$$\begin{aligned} E_1 &= \{ \tilde{\epsilon}_1 \mid \tilde{\epsilon}_1 : \underline{\mathbf{f}}_1 \times X_1 \rightarrow \{0, 1\} \} \\ &\vdots \\ E_{r-1} &= \{ \tilde{\epsilon}_{r-1} \mid \tilde{\epsilon}_{r-1} : \underline{\mathbf{f}}_1 \times (X_1 \cup \dots \cup X_{r-1}) \rightarrow \{0, 1\} \} \\ E_r &= \{ \tilde{\epsilon}_r \mid \tilde{\epsilon}_r : \underline{\mathbf{f}}_1 \times (X_1 \cup \dots \cup X_{r-1} \cup X_r) \rightarrow \{0, 1\} \} \end{aligned}$$

Es ist klar, daß sich diese durch sukzessive Abbildungen

$$\sigma_\epsilon^{(k)} : E_k \longrightarrow E_{k-1}$$

epimorph aufeinander transportieren lassen. Ebenso können wir auch die jeweiligen Einschränkungen von A_2 als Epimorphismen

$$\sigma_A^{(k)} : A_2 \downarrow \bigcup_{\kappa=1}^k X_\kappa \longrightarrow A_2 \downarrow \bigcup_{\kappa=1}^{k-1} X_\kappa$$

auffassen. Zusammen bilden die Paare $\sigma^{(k)} = (\sigma_\epsilon^{(k)}, \sigma_A^{(k)})$ also Epimorphismen von Gruppenoperationen, wobei die Verträglichkeit mit den Operationen wiederum klar ist. Somit sind wir in der Situation von 1.3.1 bzw. 1.3.2.1, was folgende Vorgehensweise rechtfertigt:

3.2.1 Algorithmus (Surjektive Auflösung der Operation).

- i) Berechne eine Transversale der Operation von $A_1 \times (A_2 \downarrow X_1)$ auf E_1 unter den Nebenbedingungen 3.1.5.
- ii) Führe das folgende rekursive Verfahren, beginnend mit Tiefe 1, durch:

- (a) Bestimme in Tiefe k für jeden Repräsentanten $\tilde{\epsilon}_{k-1}$ aus der Tiefe $k-1$ eine Transversale der Operation von $A_1 \times C_A(\tilde{\epsilon}_{k-1})$ mit $A = A_2 \downarrow \bigcup_{\kappa=1}^k X_\kappa$ auf

$$\{ \hat{\theta} \in E_k \mid \hat{\theta}(i, j) = \tilde{\epsilon}_{k-1}(i, j) \forall 1 \leq i \leq f_1, j \in X_1 \cup \dots \cup X_{k-1} \}$$

unter den Nebenbedingungen 3.1.5.

- (b) Überprüfe, falls vorgegeben, die weiteren Nebenbedingungen. Sind diese erfüllt und ist $k < r$, so fahre in Tiefe $k+1$ fort.

▽

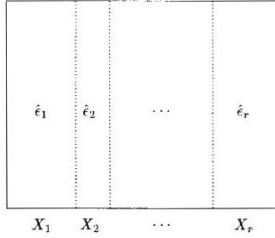
Einen Algorithmus für Schritt ii-a werden wir in 3.3.2 vorstellen.

Als „weitere Nebenbedingungen“ kommen in Frage:

- Anzahl aller Einsen in der Matrix
- Bedingungen an den Graphen, der sich durch das Einsetzen von $\tilde{\epsilon}_k$ und Rückbildung der Erweiterung ergibt: Größe der Ringe, verbotene Teilstrukturen etc.

Sieht man die rekursive Aufspaltung als Baum, so bedeutet der Test auf die weiteren Nebenbedingungen, daß u.U. ganze Äste abgeschnitten werden können, sofern ein Kandidat diese nicht erfüllt. Im Gegensatz zum Füllen der gesamten Matrix haben wir auf diese Weise eine Methode gefunden, bei der nicht erst vollständige Kandidaten untersucht werden, sondern bereits in einem frühen Stadium die Konstruktion unerwünschter Lösungen vermieden werden kann. Dies erfüllt auch unsere Forderung nach frühestmöglicher Berücksichtigung von Anwenderrestriktionen (vgl. S. 43).

Anschaulich gesprochen haben wir also die zu berechnende Matrix $\tilde{\epsilon}$ in Streifen zerlegt, etwa:



(wobei sich i. a. die Streifenstruktur erst durch Umordnung bzgl. der Bahnen zeigt). Unsere Vorgehensweise bedeutet dann, daß jeweils ein Streifen unter der Operation des Stabilisators der bisher gefüllten Streifen in A_2 belegt wird.

3.2.2 Bemerkung. Bislang haben wir in diesem Abschnitt die Aufspaltungstechnik nur auf die Spalten angewandt. Sie läßt sich natürlich genauso auf die Zeilen anwenden, wodurch die zu konstruierende Matrix in Kästchen zerlegt wird. Aus Gründen der Übersichtlichkeit wollen wir hier aber auf weitere Details verzichten und uns weiter auf Streifen beschränken. In der Implementation indessen kann diese Technik eingesetzt werden.

3.2.3 Beispiel. Wir betrachten die folgenden molekularen Graphen.



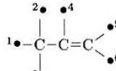
IV: Ethen

V: Propen

Als vollständige Erweiterungen ergeben sich



IV'



V'

Als induzierte Automorphismengruppen der fiktiven Ecken erhält man:

$$A_1 = \{1, (12), (34), (12)(34), (13)(24), (14)(23), (1324), (1423)\}$$

$$A_2 = \{1, (12), (13), (23), (56), (12)(56), (13)(56), (23)(56), (123), (132), (123)(56), (132)(56)\}$$

Die Bahnen unter A_2 sind dabei $\{1, 2, 3\}$, $\{4\}$ und $\{5, 6\}$.

◇

Aufgrund der Nebenbedingungen 3.1.5. daß für jedes feste $i \in \underline{f}_1$ bzw. jedes feste $j \in \underline{f}_2$ es höchstens ein k geben darf, so daß $\hat{\epsilon}(i, k) = 1$ bzw. $\hat{\epsilon}(k, j) = 1$, ist die Berechnung der Abbildungen auf den einzelnen Streifen nicht voneinander unabhängig. Es muß vielmehr beim ordnungstreuen Erzeugen der Paare (i, j) mit $\hat{\epsilon}(i, j) = 1$ überprüft werden, ob sich in der entsprechenden Zeile und Spalte nicht bereits ein Paar befindet, das auf 1 abgebildet wird. Um dies auf einfache Weise zu berücksichtigen, legen wir Obergrenzen für die Zeilen der zu erzeugenden Matrix $\hat{\epsilon}$ fest. Dabei sei

- $C_i^{(k)}$ die Obergrenze der Zeile i im Streifen k ,
- $\hat{\epsilon}_k$ der aktuelle Repräsentant, der bis zum Streifen $k - 1$ gefüllt sei.
- $A_2^{(k)} := A_2 \downarrow (X_1 \cup \dots \cup X_k)$,
- $x_{k,i}$ das i -te Element von X_k .

Es gilt: $\zeta_i^{(k)} \in \{0, 1\}$ für alle entsprechenden i, k .

Die Zeilenobergrenzen schränken aber auch die Möglichkeiten für die Automorphismen der Zeilen ein. Es können dann nur noch Zeilen als austauschbar angesehen werden, die dieselbe Zeilenobergrenze haben. Als Gruppe ergibt sich damit im Streifen k :

$$A_1^{\zeta^{(k)}} := N_{A_1}(\{i \mid \zeta_i^{(k)} = 0\}) \cap N_{A_1}(\{i \mid \zeta_i^{(k)} = 1\}) = \{\pi \in A_1 \mid \zeta_j^{(k)} = \zeta_{i^\pi}^{(k)}, j = i^\pi \forall i \in \underline{f_1}\}$$

Die Festlegung dieser Werte für den Streifen k geschieht folgendermaßen:

3.2.4 Festlegung der Vorgabewerte für die Ordnungstreue Erzeugung im Streifen k .

- Ist $k = 1$, so führe durch:

- i) Setze $\zeta_i^{(1)} = 1$ für $i \in \underline{f_1}$.
- ii) Setze als operierende Gruppe $A_1 \times A_2^{(1)}$.

- Ansonsten führe durch:

- i) Ist $\sum_{\kappa=1}^{k-1} \sum_{j=1}^{|\mathbf{X}_\kappa|} \hat{\epsilon}_k(i, x_{\kappa,j}) = 1$ für $i \in \underline{f_1}$, so setze $\zeta_i^{(k)} = 0$; ansonsten setze $\zeta_i^{(k)} = 1$.
- ii) Setze als operierende Gruppe $A_1^{\zeta^{(k)}} \times C_{A_2^{(k)}}(\hat{\epsilon}_{k-1})$.

▽

3.3 Ordnungstreue Erzeugung der Verbindungsmatrizen

In diesem Abschnitt wollen wir die Methoden zur Erzeugung der Verbindungsmatrizen 3.1.5 von fiktiven Ecken zweier Graphen vorstellen. Es handelt sich dabei um die Anwendung der Ordnungstreuen Erzeugung 1.3.4 auf die restringierte Konstruktion von $\{0, 1\}$ -Matrizen, wobei wir uns in der Situation 1.3.5, also bei Symmetrieklassen von Abbildungen, befinden.

Wir betrachten das Füllen eines Streifen $k \in \mathbf{r}$ (vgl. Abschnitt 3.2, dessen weitere Bezeichnungen wir gleichfalls übernehmen). Initialisiert man den Streifen mit 0, so hat man alle Plätze zu finden, auf denen eine 1 erlaubt ist, d.h. alle Paare, die auf einen Wert ungleich 0 abgebildet werden können. Die folgende Funktion überprüft ein Paar $(i, j) \in \mathbf{f}_1 \times X_k$ darauf. (Beim Fall $k = 1$ sind die Summationen entsprechend reduziert zu verstehen.)

3.3.1 Algorithmus (Test, ob $\hat{\epsilon}_k(i, j) = 1$ sein darf).

- i) Ist $\sum_{\kappa=1}^{k-1} \sum_{\nu=1}^{|X_\kappa|} \hat{\epsilon}_k(i, x_{\kappa,\nu}) + \sum_{\nu=1}^{j-1} \hat{\epsilon}_k(i, x_{k,\nu}) = c_i^{(k)}$ oder $\sum_{\mu=1}^{i-1} \hat{\epsilon}_k(\mu, j) = 1$, so gib „negativ“ zurück.
 ii) Gib „positiv“ zurück.

▽

Bei den Abbildungen

$$\hat{\epsilon} : \mathbf{f}_1 \times \mathbf{f}_2 \rightarrow \{0, 1\}$$

ordnen wir Paare (i, j) aus den beiden Komponenten wie üblich lexikographisch an. Damit können wir stets vom „nächstgrößeren Paar“ sprechen.

Wir verwenden folgende Bezeichnungen:

$$\hat{\epsilon}^{(k)} := \sum_{\mu=1}^{f_1} \sum_{\nu=1}^k \sum_{\nu=1}^{|X_\nu|} \hat{\epsilon}_k(\mu, x_{\nu,\nu}), \text{ die Summe aller Einträge im aktuellen Kandidaten } \hat{\epsilon}_k$$

i Index für die Paare, denen 1 zugeordnet wird
 $\omega_l \in \mathbf{f}_1 \times X_k$, das l -te Paar, das auf 1 abgebildet wird

Der Algorithmus ruft eine Funktion zum Testen der Kanonizität eines Kandidaten. Diese werden wir im Anschluß besprechen.

3.3.2 Algorithmus (Ordnungstreues Füllen des k -ten Streifens).

- i) Initialisiere den aktuellen Kandidaten mit $\hat{\epsilon}_k(i, j) \leftarrow 0$ für alle $i \in \mathbf{f}_1$ und $j \in X_k$. Bestimme die Vorgabewerte für diesen Streifen mit 3.2.4.
 ii) Setze $l = \hat{\epsilon}^{(k)} + 1$. Setze ω_l auf das größte Paar (i, j) des Streifens, für das $\hat{\epsilon}_k(i, j) = 1$; gibt es ein solches nicht, so setze ω_l auf das kleinste Paar (i, j) des Streifens.
 iii) Solange ω_l nicht das letzte Paar aus $\mathbf{f}_1 \times X_k$ überschritten hat, führe durch:
 (a) Setze ω_l auf das nächstgrößere Paar.
 (b) Überprüfe mit 3.3.1, ob ω_l zulässig ist.
 (c) Ist das Ergebnis von 3.3.1 negativ, so gehe zu Schritt iii).
 (d) Setze $\hat{\epsilon}_k(\omega_l) \leftarrow 1$.
 (e) Teste den Kandidaten $\hat{\epsilon}_k$, ob er kanonisch ist. Ist dies nicht der Fall, so setze $\hat{\epsilon}_k(\omega_l) \leftarrow 0$ und gehe zu Schritt iii).
 (f) Somit ist eine neue Lösung gefunden. Teile dies der aufrufenden Prozedur 3.2.1 mit.
 (g) Gehe zu Schritt ii).
 iv) Ist $\hat{\epsilon}^{(k)} = 0$, so konnte kein Paar mehr auf 1 gesetzt werden. Beende daher die Funktion.
 v) Setze das zuletzt auf 1 gesetzte Paar auf 0 zurück. Setze $l = \hat{\epsilon}^{(k)} + 1$ und gehe zu Schritt iii).

▽

3.3.3 Bemerkung. Die Überprüfung 3.3.1 auf Zulässigkeit liefert häufig nicht nur ein Ergebnis für das gegebene Paar. Vielmehr läßt sich damit oft eine ganze Spalte oder Zeile als unzulässig erkennen. Die Ausnutzung dieser Informationen führt zu einer Beschleunigung des Verfahrens.

3.3.4 Beispiel. Wir kommen auf Beispiel 3.2.3 zurück und betrachten die Füllung des dritten Streifens. Die Matrix sei bereits folgendermaßen gefüllt:

$$\left(\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \middle| \begin{array}{c} \cdot \\ \cdot \\ \cdot \\ \cdot \end{array} \right)$$

Damit ist $\zeta_1^{(3)} = 1$, $\zeta_2^{(3)} = 0$, $\zeta_3^{(3)} = 0$ und $\zeta_4^{(3)} = 0$. Als operierende Gruppe erhalten wir $\{1, (34)\} \times \{1, (56)\}$. Man sieht leicht ein, daß es dann vier nichtisomorphe Möglichkeiten gibt, den Streifen zu füllen:

$$\left(\begin{array}{cc} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{array} \right), \quad \left(\begin{array}{cc} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{array} \right), \quad \left(\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{array} \right), \quad \left(\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{array} \right)$$

◇

Schritt iii-e des Algorithmus 3.3.2 erfordert eine Methode zur Überprüfung, ob der aktuelle Kandidat maximal unter der operierenden Gruppe ist. Ein solcher Test auf Kanonizität (d.h. Minimalität oder Maximalität, in unserem Fall Maximalität) ist eine effektive Methode, um zu vermeiden, daß isomorphe Objekte generiert werden, und er wird bei vielen Verfahren zur Erzeugung diskreter Strukturen eingesetzt (vgl. etwa [53, 56, 69, 85] und 1.3.3). Da dabei u.U. eine Vielzahl möglicher Permutationen des Objektes zu untersuchen ist, muß für diesen *Maximalitätstest* ein schneller Algorithmus gefunden werden, der mit möglichst geringem Aufwand über die Kanonizität eines Kandidaten entscheidet.

Dazu verwenden wir ein rekursives Verfahren, in dem wir von der Darstellung der Automorphismengruppen als markierte Verzweigungen (s. 2.2.5) Gebrauch machen. Wir erinnern daran, daß gilt:

$$(3.3.5) \quad C_G(\{\pi(1), \dots, \pi(k)\}) = \langle \sigma_{i,j} \mid i > k \wedge j > k \rangle$$

wobei $G \leq S_n$ eine Permutationsgruppe und $\pi \in S_n$ eine Anordnung der Basiselemente ist.

Sei nun $\hat{\epsilon}_k$ ein Kandidat, der auf Maximalität getestet werden soll. Der naive Ansatz ist, alle σ aus der operierenden Gruppe

$$G := A_1^{(k)} \times C_{A_2^{(k)}}(\hat{\epsilon}_{k-1})$$

zu durchlaufen und auf $\hat{\epsilon}_k^\sigma > \hat{\epsilon}_k$ zu testen. Allerdings ist dieser Ansatz meist mit sehr großem Aufwand verbunden, so daß er für die Praxis nicht geeignet ist. Sinnvoller ist es, die Tatsache auszunutzen, daß die Kandidaten lexikographisch absteigend erzeugt werden, denn es werden ja stets die kleinstmöglichen Paare gesucht, die auf 1 abgebildet werden können.

Wir betrachten die Paare

$$\Omega := \{\omega_1, \dots, \omega_l\} \subseteq \mathbf{f}_1 \times X_k \text{ mit } \hat{\epsilon}_k(\omega_i) = 1 \ \forall i \in \mathbb{1},$$

also all diejenigen, die auf 1 abgebildet werden. Die Ordnungstreu der Erzeugung erlaubt es, sich beim Überprüfen der Maximalität ab dem zweiten Element in Ω auf den Zentralisator der vorangegangenen zu beschränken.

Wir berechnen sodann die Bahnen ω_i^G bzw. $\omega_i^{C_G(\{\omega_1, \dots, \omega_{i-1}\})}$. Dazu wenden wir jeden Erzeuger auf jedes (bisher gefundene) Bahnelement an und speichern nicht nur die Bahnen, sondern auch die Permutationen, die die einzelnen Bahnelemente ω auf ω_i transportieren, also die $\sigma \in C_G(\{\omega_1, \dots, \omega_{i-1}\})$ mit $\omega^\sigma = \omega_i$. Diese bezeichnet man auch als *fusionierende Elemente*.

3.3.6 Lemma. Sei $\sigma_t^{(i)}$ das fusionierende Element, welches das t -te Element von $\omega_i^{C_G(\{\omega_1, \dots, \omega_{i-1}\})}$ auf $\omega_i \in \Omega$ abbildet. Dann ist

$$\left\{ \sigma_t^{(i)} \mid t = 1, \dots, |\omega_i^{C_G(\{\omega_1, \dots, \omega_{i-1}\})}| \right\}$$

eine Transversale der Nebenklassen $C_G(\{\omega_1, \dots, \omega_{i-1}\})/C_G(\{\omega_1, \dots, \omega_i\})$.

Beweis: Klar. □

Somit haben wir über die Zentralisatoren der fusionierenden Elemente der ω_i eine Möglichkeit der Darstellung von G im Sinne von 2.2.2 gefunden. Durch eine rekursive Funktion, die die Gruppenelemente der Reihe nach anwendet, können wir nun den Kanonizitätstest durchführen; zuvor initialisieren wir noch die Datenstrukturen.

- $C_G^{(t)}$:= $C_G(\{\omega_1, \dots, \omega_{t-1}\})$ mit $C_G^{(1)} = G$.
- O_t ist die Bahn von ω_t unter $C_G^{(t)}$.
- $\phi_t(i)$ ist die Permutation (das fusionierende Element), die i auf ω_t abbildet.
- σ ist der aktuelle Erzeuger von $C_G^{(t)}$ nach 3.3.5.
- ξ ist das aktuelle Bahnelement, womit alle (bisher gefundenen) Elemente einer Bahn durchlaufen werden.

3.3.7 Algorithmus (Initialisierung des Maximalitätstests).

- i) Durchlaufe alle $\omega_t \in \Omega$ und führe jeweils durch:
 - (a) Initialisiere $O_t \leftarrow \{\omega_t\}$.
 - (b) Setze das aktuelle Bahnelement $\xi \leftarrow \omega_t$ und $\phi_t(\omega_t) \leftarrow id$.
 - (c) Durchlaufe alle Erzeuger $\sigma \in C_G^{(t)}$ und führe jeweils durch:
 - Ist $\xi^\sigma \notin O_t$, so füge es hinzu: $O_t \leftarrow O_t \cup \{\xi^\sigma\}$ und setze $\phi_t(\xi^\sigma) \leftarrow \phi_t(\xi) \cdot \sigma$.
 - (d) Gibt es noch ein weiteres $\xi \in O_t$, so gehe damit zu Schritt c.
 - (e) Wechsle die Basis von $C_G^{(t)}$ so, daß eine Darstellung für $C_G^{(t+1)}$ wie in 3.3.5 entsteht.
- ii) Rufe den rekursiven Test 3.3.8 mit Kandidat Ω und Tiefe 1 auf und gib dessen Ergebnis zurück.

▽

Die rekursive Testfunktion hat als Argumente die aktuelle Tiefe t und eine zu überprüfende Matrix, dargestellt durch die Paare $\Theta = \{\theta_1, \dots, \theta_t\}$ mit Eintrag 1. Der lexikographische Vergleich der Matrizen läßt sich somit auch als Vergleich solcher Paarektoren verstehen.

3.3.8 Algorithmus (Rekursiver Maximalitätstest von Θ in Tiefe t).

- i) Ist

$$\Theta \cap \bigcup_{\omega_t < \omega_{t+1}} \omega^{C_G^{(t)}} \neq \emptyset,$$

so kann ein Paar mit Eintrag 0 zwischen diesem ω_t und dem nächsten auf ein Paar mit Eintrag 1 abgebildet werden. Da dadurch eine größere Matrix entsteht, kann mit „negativ“ zurückgesprungen werden.

- ii) Durchlaufe alle $\xi \in O_t$ und führe durch:
 - (a) Berechne $\Theta^{\phi_t(\xi)}$.
 - (b) Ist $\Theta^{\phi_t(\xi)} > \Omega$, so gib „negativ“ zurück.
 - (c) Ist $t < l$, rufe die nächste Tiefe des Tests mit $\Theta^{\phi_t(\xi)}$ und $t + 1$ auf. Gibt dieser „negativ“ zurück, so gib ebenfalls „negativ“ zurück.
- iii) Gib „positiv“ zurück.

▽

3.3.9 Beispiel. In Weiterführung von 3.3.4 betrachten wir den Maximalitätstest von

$$\begin{array}{c} 5 \quad 6 \\ \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \end{array}$$

also des dritten Streifens mit den Spalten 5 und 6. Wir haben bereits festgestellt, daß hier $G = \{1, (34)\} \times \{1, (56)\}$. Es ist $\Omega = \{(2, 5), (3, 6)\}$ und $C_G^{(2)} = \{1, (34)\} \times \{1\}$. Damit erhalten wir aus 3.3.7:

$$\begin{aligned} O_1 &= \{(2, 5), (2, 6)\}, & \phi_1((2, 6)) &= (1, (56)) \\ O_2 &= \{(3, 6), (4, 6)\}, & \phi_2((4, 6)) &= ((34), 1) \end{aligned}$$

Die Nulleinträge auf den Plätzen (2, 6) und (3, 5) können unter den Gruppen nicht auf Plätze mit 1 abgebildet werden, so daß die Bedingung 3.3.8.i in keinem Fall erfüllt ist. Im rekursiven Test gelangt man dann zu folgenden Situationen (wobei die Einrücktiefe die Rekursionstiefe widerspiegelt):

- Es ist $\Theta = \{(2, 5), (3, 6)\}$. Das erste Element aus O_1 ist (2, 5), womit sich $\Theta^{\phi_1((2,5))} = \Omega$ ergibt.
 - In dieser Tiefe haben wir wieder $\Theta = \{(2, 5), (3, 6)\}$. Das erste Element von O_2 ist (3, 6), was uns zu $\Theta^{\phi_2((3,6))} = \Omega$ führt.
 - Mit dem nächsten Element (4, 6) aus der Bahn erhalten wir $\Theta^{\phi_2((4,6))} = \{(2, 5), (4, 6)\} < \Omega$. Somit hat der Kandidat in dieser Tiefe bestanden.
- Als zweites und letztes Element von O_1 haben wir (2, 6). Daraus ergibt sich $\Theta^{\phi_1((2,6))} = \{(2, 6), (3, 5)\} < \Omega$.
 - An die zweite Tiefe wird $\Theta = \{(2, 6), (3, 5)\}$ übergeben. Neben der Identität aus $\phi_2((3, 6))$ erhalten wir außerdem $\Theta^{\phi_2((4,6))} = \{(2, 6), (4, 5)\} < \Omega$. Damit führt auch dieses Θ zu einem positiven Ergebnis.

Insgesamt hat also der Kandidat in allen Tiefen bestanden und ist folglich kanonisch.

Die Gesamtheit aller Lösungen zu diesem Beispiel stellen wir in 3.4.10 mit Abb. 3.1 auf Seite 57 vor. ◇

3.3.10 Bemerkung. Die hier beschriebene Vorgehensweise ist zwar schon recht effektiv; bei einer Computerimplementation sollten aber noch einige Punkte beachtet werden, die eine teilweise erhebliche Beschleunigung bewirken können:

- i) Die Berechnung der Bahnen in 3.3.7 ist ein aufwendiger Schritt. Um zu vermeiden, daß er zu oft wiederholt wird, kann man die Ermittlung der Bahnen gleich beim Füllen der Matrix vornehmen. Dazu bestimmt und speichert man beim Zuweisen der auf 1 abzubildenden Paare in 3.3.2 die Bahn des jeweiligen Paares.
- ii) Auch ein noch so schneller Maximaltest kann das Problem überflüssiger Kandidaten nicht vermeiden helfen. Denn trotz der Ordnungstreue der Erzeugung kann es vorkommen, daß es viele Möglichkeiten für das Setzen einer 1 gibt, die alle auf ihre Maximalität untersucht werden müssen. Hier sorgt Lemma 1.3.6 für Abhilfe. Nach einem negativ ausgefallenen Test läßt sich damit die Position ermitteln, die – bildet man sie auf 1 ab – frühestens zu einem kanonischen Kandidaten führen kann.
Noch ausgefeiltere Methoden wie in [56] bestimmen daraus zudem eine Liste von möglichen weiteren Plätzen, auf die man die Untersuchung dann gänzlich einschränken kann.
- iii) Bei molekularen Graphen herrscht sehr oft wenig bis keine Symmetrie vor, hat also die Automorphismengruppe nur wenige Elemente oder ist trivial. Auch wenn über die Äquivalenz fiktiver Ecken häufig wieder etwas mehr Symmetrie ins Spiel kommt (s. 3.1.4), trifft man beim Füllen der Streifen in sehr vielen Fällen auf triviale operierende Gruppen. Bei diesen erübrigt sich dann ein Kanonizitätstest, denn jeder erzeugte Kandidat ist dabei maximal. Daher sollte die Implementierung dafür eine besondere Behandlung vorsehen, um keine überflüssigen Test aufzurufen.

3.4 Kanonisierung der verbundenen Graphen

In den letzten beiden Abschnitten haben wir gezeigt, wie zwei erweiterte Graphen auf alle wesentlich verschiedenen Weisen miteinander verbunden werden können, da ja nach 3.1.6 diese Verbindungen den mit 3.2.1 und 3.3.2 konstruierten Abbildungen \tilde{e} entsprechen.

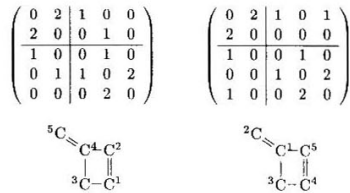
Nach dem Verbinden wollen wir die fiktiven Ecken nun wieder entfernen. Es handelt sich um die Umkehrung der Erweiterung 3.1.2. Dazu lassen wir die fiktiven Ecken weg und setzen dort eine k -fache Kante, wo zwei ursprüngliche Ecken über k fiktive miteinander gebunden sind. (Diese Vorgehensweise ergibt sich ganz natürlich, so daß auf eine formale Beschreibung verzichtet werden kann.)

Die Adjazenzmatrix (vgl. 2.1.10) des so entstehenden Graphen setzt sich aus vier Blöcken zusammen:

$$(3.4.1) \quad M := \left(\begin{array}{c|c} T_1 & V \\ \hline V^T & T_2 \end{array} \right)$$

Dabei sei T_1 die Adjazenzmatrix des ersten Ausgangsgraphen, T_2 die des zweiten und V die Verbindungsmatrix, die durch die Rückbildung aus \tilde{e} entsteht. Durch die Konstruktion und den dabei durchgeführten Maximalitätstest haben wir sichergestellt, daß nur solche V erzeugt werden, die unter der Operation von $\text{Aut}(T_1)$ auf den Zeilen und $\text{Aut}(T_2)$ auf den Spalten nichtisomorph sind. Allerdings ist damit noch nicht erreicht, daß auch alle verbundenen Graphen nichtisomorph sind. Denn nach 2.1.8 operiert auf diesen eine größere Gruppe, nämlich $S_{|T_1|+|T_2|}$. Somit sind die erhaltenen V nicht wirklich kanonisch; wir wollen sie daher als *semikanonisch* bezeichnen (in Anlehnung an [53, 57]). Ihre Kanonizität muß erst noch durch einen weiteren Test verifiziert werden.

3.4.2 Beispiel. Bei der Verbindung von Ethen und Propen, den Strukturen **IV** und **V** aus 3.2.3, werden die folgenden beiden Lösungen mit 3.2.1 und 3.3.2 generiert:



Obwohl die Verbindungsblöcke der beiden Matrizen unter der entsprechenden Gruppenoperation in verschiedenen Bahnen liegen, sind die resultierenden Gesamtgraphen offensichtlich isomorph (vermittelt der Permutation (14)(25)).

◇

Graphen, die zwar semikanonisch aber nicht kanonisch sind, treten vor allem bei strukturellen Ähnlichkeiten der Ausgangsgraphen auf, wie es auch aus dem Beispiel ersichtlich ist. Ihr Anteil ist dann am größten, wenn beide Ausgangsgraphen bereits isomorph sind.

Wir wollen nun die Kanonisierung, d.h. die Eliminierung der isomorphen semikanonischen Graphen, formal und algorithmisch beschreiben. Dazu seien wieder $\gamma_1 \in m\mathbb{P}^{[2]}$ und $\gamma_2 \in m\mathbb{A}^{[2]}$ unsere Ausgangsgraphen (evtl. noch mit Beschriftungen β_1 und β_2).

Unter der *Einschränkung* $\gamma \downarrow \{i_1, \dots, i_r\}$ eines Graphen γ wollen wir den Teilgraphen im Sinne von 2.3.1 auf der (bei Bedarf unnummerierten) Eckenteilmenge $\{1, \dots, r\}$ verstehen. Mit Hilfe dieser Schreibweise können wir festlegen:

$$\Omega := \left\{ \gamma \in m\mathbb{P}^{[p+q]} : (\gamma \downarrow \mathbf{p}) \simeq \gamma_1 \wedge (\gamma \downarrow \{p+1, \dots, p+q\}) \simeq \gamma_2 \right\}$$

Darauf operiert S_{p+q} in gewohnter Weise. Somit stellt Ω die Menge aller verbundenen Graphen aus γ_1 und γ_2 dar. Deren Adjazenzmatrix hat stets die Struktur 3.4.1.

Die Konstruktion der Verbindungsmatrizen nach 3.2.1 und 3.3.2 entspricht also der Bestimmung eines vollständigen Repräsentantensystems der Bahnen

$$(3.4.3) \quad \Omega // C_{S_{p+q}}(\{1, \dots, p\}) \cap C_{S_{p+q}}(\{p+1, \dots, p+q\}),$$

wogegen eigentlich

$$\Omega // S_{p+q}$$

gesucht ist¹. In der Sprechweise des Homomorphieprinzips (vgl. 1.3.2.ii) müssen wir also eine Fusion der Bahnen durchführen. Algorithmisch bedeutet dies, daß wir jeden Repräsentanten von 3.4.3 darauf untersuchen müssen, ob es ein Element einer Transversale der Nebenklassen

$$S_{p+q} / (C_{S_{p+q}}(\{1, \dots, p\}) \cap C_{S_{p+q}}(\{p+1, \dots, p+q\}))$$

gibt, durch das dieser auf einen bereits erzeugten Repräsentanten abgebildet wird.

Der Durchlauf durch die Gruppenelemente ist rekursiv organisiert, analog zu 2.4.10. Um eine allzu technische Beschreibung zu vermeiden, wollen wir die Berechnung der Gruppenelemente nicht explizit angeben.

Für die Vorstellung des Verfahrens benötigen wir wieder ein paar Bezeichnungen (Weitere Schreibweisen übernehmen wir aus den Abschnitten 3.2 und 3.3.):

- ω $\in \Omega$ ist der zu testende Repräsentant.
- \hat{e} ist die Verbindungsmatrix zu ω über die fiktiven Ecken
- $\pi \in S_{p+q}$ ist die Laufvariable für die zu testenden Permutationen
- $\varphi(\hat{e}, \pi) \in \{0, 1\}^{\mathfrak{E} \times \mathfrak{E}}$ bildet \hat{e} so ab, wie es einer Vertauschung mittels π der tatsächlichen und der Fortsetzung auf die fiktiven Ecken im erweiterten Graphen entspricht.
- $e^{(k)} := \sum_{\mu=1}^{f_1} \sum_{\nu=1}^{|X_k|} \hat{e}(\mu, x_{k,\nu})$, die Summe aller Einträge im k -ten Streifen.
- $e_{\pi}^{(k)} := \sum_{\mu=1}^{f_1} \sum_{\nu=1}^{|X_k|} \varphi(\hat{e}, \pi)(\mu, x_{k,\nu})$, die entsprechende Summe bei $\varphi(\hat{e}, \pi)$.

Wir überprüfen, ob ein ω durch π auf ein bereits erzeugtes ω' transportiert wird. Bei einer rein lexikographisch absteigenden Erzeugung ließe sich dies als $\omega^{\pi} > \omega$ ausdrücken. Da aber in 3.3.2 keine Matrizen mit einer festen Zahl von Einträgen ungleich Null erzeugt werden, sondern diese variiert, müssen wir neben der Maximalität noch einige weitere Fälle unterscheiden.

3.4.4 Algorithmus (Kanonizitätstest von ω).

- i) Initialisiere π mit der Identität in S_{p+q} .
- ii) Durchlaufe alle π aus einer Transversale von

$$S_{p+q} / (C_{S_{p+q}}(\{1, \dots, p\}) \cap C_{S_{p+q}}(\{p+1, \dots, p+q\}))$$

und führe jeweils durch:

- (a) Ist $(\omega^{\pi} \downarrow \mathfrak{p}) \neq \gamma_1$ oder $(\omega^{\pi} \downarrow \{p+1, \dots, p+q\}) \neq \gamma_2$, so fahre mit dem nächsten π in Schritt ii) fort.
- (b) Durchlaufe die Streifen von \hat{e} , also alle $k \in \mathfrak{E}$, und führe jeweils durch:

1. Ist $e^{(k)} = 0$ und $e_{\pi}^{(k)} = 0$, so fahre mit dem nächsten Streifen in Schritt ii)-(b) fort.
2. Ist $e^{(k)} = 0$ und $e_{\pi}^{(k)} \neq 0$, so kann $\varphi(\hat{e}, \pi)$ kein bereits berechneter Repräsentant sein. Fahre daher in diesem Fall mit dem nächsten π in Schritt ii) fort.
3. Ist $e^{(k)} \neq 0$ und $e_{\pi}^{(k)} = 0$, so ist ω nicht kanonisch. Brich daher den Test mit „negativ“ ab.
4. Durchlaufe alle $i \in \mathfrak{E}_1$ und für jedes i alle $j \in \{1, \dots, |X_k|\}$. Für jedes Paar (i, j) führe durch:
 - A. Ist $\hat{e}(i, j) < \varphi(\hat{e}, \pi)(i, j)$, so unterscheide: Ist $e^{(k)} \neq e_{\pi}^{(k)}$ und ist (i, j) größer als das größte Paar in \hat{e} , das auf 1 abgebildet wird, so würde $\varphi(\hat{e}, \pi)$ vom Erzeugungsalgorithmus erst nach \hat{e} generiert; fahre daher in diesem Fall mit dem nächsten π in Schritt ii) fort. Andernfalls wurde $\varphi(\hat{e}, \pi)$ bereits vorher erzeugt. Folglich ist ω nicht kanonisch; brich daher den Test mit „negativ“ ab.
 - B. Ist $\hat{e}(i, j) > \varphi(\hat{e}, \pi)(i, j)$, so unterscheide: Ist $e^{(k)} \neq e_{\pi}^{(k)}$ und ist (i, j) größer oder gleich dem größten Paar in $\varphi(\hat{e}, \pi)$, das auf 1 abgebildet wird, so wurde $\varphi(\hat{e}, \pi)$ bereits vorher erzeugt, womit ω sich als nicht kanonisch erweist; brich daher den Test mit „negativ“ ab. Andernfalls würde $\varphi(\hat{e}, \pi)$ vom Erzeugungsalgorithmus erst nach \hat{e} generiert; fahre daher in diesem Fall mit dem nächsten π in Schritt ii) fort.

¹Waren γ_1 und γ_2 beschriftet, so können wir uns hier auf die Young-Untergruppe von S_{p+q} beschränken, die diese Beschriftungen erhält.

iii) Gib „positiv“ zurück.

∇

3.4.5 Beispiel. Bei den Strukturen IV und V aus 3.2.3 wird als 17. Lösung der linke Graph aus 3.4.2 mit der Adjazenzmatrix

$$\omega = \left(\begin{array}{cc|cc} 0 & 2 & 1 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 \end{array} \right) \text{ und } \tilde{\epsilon} = \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right)$$

erzeugt. Beim Durchlaufen der Permutationen trifft man auch auf $\pi = (14)(25)$. Durch dessen Anwendung erhält man

$$\omega^\pi = \left(\begin{array}{cc|cc} 0 & 2 & 1 & 0 & 1 \\ 2 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 2 \\ 1 & 0 & 0 & 2 & 0 \end{array} \right) \text{ bzw. } \varphi(\tilde{\epsilon}, \pi) = \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right)$$

Die ersten Streifen stimmen überein. Für den zweiten Streifen gilt $e^{(2)} = 1$, aber $e_\pi^{(2)} = 0$ (Fall ii-b-3). Daher kann ω verworfen werden.

◇

3.4.6 Bemerkung. Der einzig ungeklärte Punkt im Verfahren 3.4.4 ist die Isomorphieüberprüfung in Schritt ii-a. Dort müssen wir untersuchen, ob $\{\omega^\pi \downarrow \mathbf{p}\} \simeq \gamma_1$ oder $\{\omega^\pi \downarrow \{p+1, \dots, p+q\}\} \simeq \gamma_2$. Ein Rückgriff auf die Definition 2.1.11 der Graphenisomorphie ist algorithmisch wenig sinnvoll, da dies ein Durchlaufen der kompletten S_p bzw. S_q bedeuten würde. Wir verwenden daher eine kanonische Numerierung [15, 94, 97] der Ausgangs- und der eingeschränkten Graphen (in der Implementierung aus [11]). Dabei wird eine eindeutige Numerierung der Ecken eines Graphen (oder molekularen Graphen) bestimmt, d.h. aus jeder Isomorphieklasse eines Graphen wird ein kanonischer Repräsentant ermittelt. Die Zurückführung auf die kanonische Numerierung erleichtert den Isomophietest erheblich: Es genügt nun, die Graphen kanonisch zu numerieren und dann auf Gleichheit zu testen.

Insgesamt ist das Verfahren 3.4.4 mit einem hohen Aufwand verbunden. Um diesen zumindest teilweise zu reduzieren, wollen wir eine sog. *Hash-Technik* verwenden (s. etwa [111]).

Dabei handelt es sich um eine Methode für den direkten Zugriff auf gespeicherte Objekte mittels einer zusätzlichen Tabelle. Durch eine arithmetische Funktion wird jedem Objekt eine i.a. natürliche Zahl zugeordnet, welche die Tabellenadresse repräsentiert. Verschiedenen Objekten sollten verschiedene Adressen zugeordnet werden; allerdings besteht stets die Möglichkeit, daß zwei oder mehr Objekte auf dieselbe Adresse fallen. Man spricht in einem solchen Fall von einer *Kollision*.

Bei uns sind die Objekte numerierte bzw. numerierte beschriftete Graphen. Zur Berechnung der Hash-Funktion ermitteln wir zunächst die kanonische Numerierung des Graphen und bilden dann eine gewichtete Summe über die Einträge der Adjazenzmatrix:

$$(3.4.7) \quad h(\gamma) = \left(\sum_{i,j=1}^p w(i,j) \cdot \gamma(i,j) \right) \bmod M$$

Dabei sei w eine Gewichtsfunktion und M die Größe der Tabelle. Es empfiehlt sich, für M eine Primzahl zu wählen, da sonst bei der Bildung des Restwertes unterschiedliche Teilinformationen nivelliert werden können. Auch für w können Primzahlgewichte gewählt werden. Zudem läßt sich eine Hashfunktion alternativ auch über Grapheninvarianten definieren [9, 18]. Allgemein sollte aber Wert darauf gelegt werden, daß jeder Wert der Hashfunktion mit derselben Wahrscheinlichkeit angenommen wird.

Diese Technik setzen wir gemäß der folgenden Überlegung ein:

3.4.8 Lemma. Ein semikanonischer Kandidat γ ist kanonisch, falls bisher noch kein anderer mit demselben Hashwert $h(\gamma)$ erzeugt wurde.

Beweis: Klar.

∎

Daher können wir in diesen Fällen den Maximalitätstest 3.4.4 vermeiden. In der Implementation in NEXTOR wird dazu ein Bitvektor der Größe M angelegt; ist die Adresse $h(\gamma)$ für einen Kandidaten γ unbesetzt, so wird dieser als kanonisch akzeptiert und die Adresse auf „belegt“ gesetzt. Insbesondere bei den ersten Kandidaten kann so einige Male der Kanonizitätstest ausgelassen werden.

Das Hashing eröffnet uns aber auch die Möglichkeit, auf den Test 3.4.4 ganz zu verzichten. Dazu speichern wir alle bisherigen Lösungen (in kanonischer Numerierung) in mehreren verketteten Listen, und zwar eine für jede Adresse der Hashtabelle.

Um dies im Zusammenhang zu beschreiben, bezeichnen wir die Hashtabelle mit

$$\mathcal{H} : \underline{M} \rightarrow \mathcal{P}(m^{\mathbb{P} \pm \mathbb{Q}^{21}}), \quad i \mapsto \mathcal{H}_i$$

Dabei befindet sich an der Position i die Menge \mathcal{H}_i aller Lösungen γ mit Hashwert $h(\gamma) = i$.

3.4.9 Algorithmus (Erzeugung kanonischer Repräsentanten mittels Hashing).

- i) Initialisiere $\mathcal{H}_i \leftarrow \emptyset$ für alle $i \in \underline{M}$.
- ii) Erzeuge den nächsten semikanonischen Graphen γ mit den Verfahren 3.2.1 und 3.3.2.
- iii) Nummeriere γ auf kanonische Weise.
- iv) Ist $\mathcal{H}_{h(\gamma)} = \emptyset$, so ist γ kanonisch; speichere es daher durch $\mathcal{H}_{h(\gamma)} \leftarrow \{\gamma\}$ und gehe zu Schritt vii.
- v) Durchlaufe alle $\theta \in \mathcal{H}_{h(\gamma)}$. Falls es ein θ gibt, so daß $\theta = \gamma$, dann ist γ keine Lösung; gehe daher zu Schritt vii.
- vi) Da bis hierher kein zu γ identischer Repräsentant gefunden wurde, ist γ kanonisch; speichere es daher durch $\mathcal{H}_{h(\gamma)} \leftarrow \mathcal{H}_{h(\gamma)} \cup \{\gamma\}$.
- vii) Solange es noch weitere semikanonische Graphen gibt, gehe zu Schritt ii.

▽

Wir haben also zwei alternative Verfahren zur Verfügung, um das Problem der Isomorphie semikanonischer Repräsentanten zu beheben. Neben stilistischen Aspekten (eine rein kombinatorische Lösung gegenüber einer datentechnischen), die in der Anwendung eine geringe Rolle spielen, sind in der Praxis im wesentlichen zwei Punkte entscheidend:

- *Zeitaufwand.* Dabei liegt die Hash-Methode 3.4.9 klar vorn. Sie ist – je nach Problem – etwa 8 - 20 mal schneller als die Verifizierung über den Test 3.4.4.
- *Speicherbedarf.* Der Kanonizitätstest greift nicht auf bisher ermittelte Lösungen zurück. Bei seiner Verwendung ist somit kein zusätzlicher Speicher erforderlich; alle Repräsentanten können sofort auf den Massenspeicher ausgelagert (oder benutzerspezifisch weiterbehandelt) werden. Die Hash-Technik hat dagegen einen relativen hohen Speicherbedarf, da sie alle Lösungen speichern muß. Steht nicht ausreichend Hauptspeicher zur Verfügung, so kann die Hashtabelle auch auf einem Massenspeicher angelegt werden.

In Abwägung der Vor- und Nachteile und unter Berücksichtigung der heutzutage zur Verfügung stehenden Speicherkapazitäten muß dem Hash-Verfahren 3.4.9 klar der Vorzug gegenüber dem durch den Kanonizitätstest 3.4.4 gestützten gegeben werden. Bereits bei kleinen, besonders aber bei größeren Fällen ist die von letzterem benötigte Zeit nicht mehr vertretbar. Aus diesem Grund ist auch die VorgabeEinstellung in NEXTOR die Hash-Methode.

3.4.10 Beispiel. Zum Abschluß zeigen wir in Abb. 3.1 noch alle 31 kanonischen Lösungen des Verknüpfungsproblems der Strukturen IV und V aus Beispiel 3.2.3, das uns bis hierher begleitet hat. Die Platzierung der Graphen wurde automatisch mit dem entsprechenden Modul aus **MOLGEN** erstellt (s. etwa [14, 112]). ◇

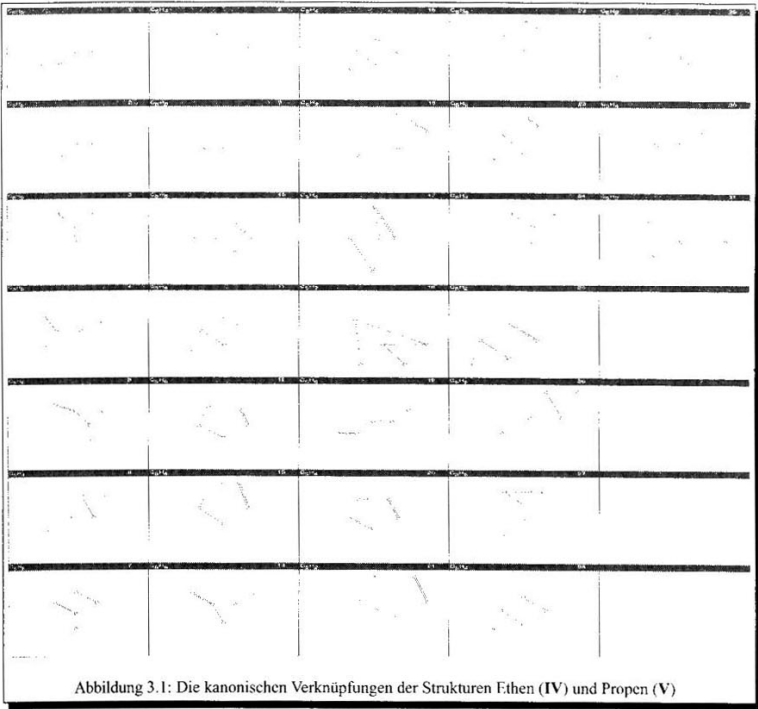


Abbildung 3.1: Die kanonischen Verknüpfungen der Strukturen Fithen (IV) und Propen (V)

3.5 Diskussion des Verfahrens

In diesem Abschnitt wollen wir den Algorithmus zur Verbindung zweier Graphen diskutieren, d.h. seinen Aufwand theoretisch und anhand praktischer Tests betrachten. Mögliche Anwendungen werden wir in späteren Kapiteln vorstellen.

Eine Aufwandsanalyse für den hier eingesetzten Verfahrenstyp wurde in [58] und [85] gegeben, auf die wir uns hier beziehen. Wir unterscheiden nach der Aufwandsreduktion, die durch den Einsatz des Homomorphieprinzips erreicht wurde, und dem Aufwand zur Ordnungstreuen Erzeugung.

Wir betrachten die Situation aus Abschnitt 3.2. Die Epimorphismen

$$\sigma_{\tilde{e}}^{(k)} : E_k \twoheadrightarrow E_{k-1} \quad \text{und} \quad \sigma_A^{(k)} : A_2 \downarrow \bigcup_{\kappa=1}^k X_\kappa \twoheadrightarrow A_2 \downarrow \bigcup_{\kappa=1}^{k-1} X_\kappa$$

lösen das Konstruktionsproblem im Sinne von 1.3.2.i surjektiv auf. $(\sigma_{\tilde{e}}^{(k)})^{-1}$ ist dabei die Abbildung, die den Streifen k anfügt. Somit ist der Betrag $|(\sigma_{\tilde{e}}^{(k)})^{-1}(\tilde{e}_{k-1})|$ beschränkt durch die Anzahl der Möglichkeiten, wie Paare $(i, j) \in \underline{f}_1 \times X_k$ durch \tilde{e}_k unter Berücksichtigung von 3.1.5 auf 1 abgebildet werden können. Diese wiederum summiert sich über die Anzahl solcher Paare:

$$b_k := \sum_{t=1}^{\min(f_1, |X_k|)} \frac{1}{t!} \prod_{\nu=1}^t (f_1 - \nu + 1) (|X_k| - \nu + 1)$$

Damit gilt:

$$|(\sigma_{\tilde{e}}^{(k)})^{-1}(\tilde{e}_{k-1})| \leq b_k \quad \forall \tilde{e}_{k-1} \in E_{k-1}, k \in \{2, \dots, r\}$$

Ein Algorithmus berechne für jede Untergruppe $U_k \leq A_1 \times A_2^{(k)}$ und jede Menge $D_k \subseteq E_k$ auf der U_k operiert, eine Transversale $\mathcal{T}(D_k // U_k)$ zusammen mit $N_{U_k}(\tilde{e}_k)$ für alle $\tilde{e}_k \in \mathcal{T}(D_k // U_k)$ in der Zeit $\alpha_k(|D_k|)$, wobei $\alpha_k : \mathbb{N} \rightarrow \mathbb{N}$ eine monoton wachsende Funktion sei. Damit erhalten wir:

3.5.1 Satz. *Das schrittweise Berechnen von $\mathcal{T}(E_r // A_1 \times A_2)$ unter den Nebenbedingungen 3.1.5 mit Hilfe der Technik der Surjektiven Auflösung (3.2.1) verkürzt die Zeit von maximal $\alpha_r(|E_r|)$ auf maximal*

$$\alpha_1(|E_1|) + \sum_{k=2}^r \alpha_k(b_k) \cdot c_k \cdot |E_k // A_1 \times A_2^{(k)}|,$$

wobei c_k die maximale Zeit sei, um $(\sigma_{\tilde{e}}^{(k)})^{-1}(\tilde{e}_{k-1})$ für ein beliebiges $\tilde{e}_{k-1} \in E_{k-1}$ zu berechnen.

Beweis: Klar. □

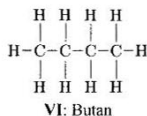
Hierbei wurde der Algorithmus, mit dem die Transversale tatsächlich berechnet werden soll, als *black box* verwendet, und sein Aufwand nur durch die Funktion α_k repräsentiert. Wir haben daher auch lediglich eine relative Komplexität erhalten. Dies fügt sich andererseits aber sehr gut in unser Konzept der allgemeinen Objektorientierung ein, deren Kennzeichen ja gerade eine Kapselung und Trennung der einzelnen Verfahren ist.

Eine allgemeine Komplexitätsanalyse des Verfahrens der Ordnungstreuen Erzeugung 1.3.4 liegt bislang nicht vor [57, 85] und kann auch im Rahmen dieser Arbeit nicht gegeben werden. Die Abhängigkeit von vielerlei Faktoren wie den Größen der Gruppen, Untergruppen und Normalisatoren, dem jeweiligen Verhältnis der Anzahlen von Kandidaten und Repräsentanten usw. macht eine derartige Analyse selbst zu einem äußerst komplizierten Problem. Die Wirksamkeit dieses Verfahrens konnte aber bereits in vielen Anwendungsbeispielen unter Beweis gestellt werden [53, 56, 69, 85].

Wir wollen daher nun die praktische Leistungsfähigkeit der Algorithmen anhand einer Reihe systematischer Rechenbeispiele aufzeigen. Leider ist die Darstellungsmöglichkeit der Ergebnisse sehr begrenzt, da schon bei verhältnismäßig wenig Atomen die Menge nichtisomorpher molekularer Graphen „explodiert“. Nachdem wir bereits in 3.4.10 alle Ergebnisgraphen zu einem konkreten Beispiel aufgelistet haben, werden wir uns hier auf die Angabe von Anzahlen beschränken, wobei zu bemerken ist, daß immer wirklich alle Graphen *konstruiert* und nicht nur abgezählt wurden.

Um das Verhalten des Verfahrens bei einem möglichst breiten Spektrum von Graphen zu untersuchen, wurden eine Teststruktur mit allen 5.338 Isomeren der Formel C_iH_j mit $1 \leq i \leq 7$ und $2 \leq j \leq 16$ verknüpft. Die Menge dieser Isomere wurde mit dem Programm **MOLGEN** erzeugt, das am Lehrstuhl II für Mathematik der Universität Bayreuth entwickelt wurde [13, 14, 53, 54, 131].

Es wurde stets mit der Maximalzahl fiktiver Ecken verknüpft. Als Vergleichsgraph wurden folgende Struktur gewählt:



Aufgrund der Vielfalt der Fälle ist es nicht möglich, alle Anzahlen im einzelnen anzugeben. Daher zeigt Tabelle 3.1 auch nur jeweils die arithmetischen Mittelwerte zu einer Isomerenfamilie; im einzelnen sind dies:

- \bar{n} Arithmetisches Mittel der Anzahlen von Verknüpfungsgraphen
- n_{\max} Höchste Anzahl von Verknüpfungsgraphen beim Verknüpfen von Butan (VI) mit allen Isomeren
- n_{\min} Kleinste Anzahl von Verknüpfungsgraphen
- σ Standardabweichung von n
- \bar{t} Mittlere CPU-Zeit für die Erzeugung aller Verknüpfungsgraphen in Sekunden
- $\frac{\bar{t}}{n}$ Mittlere CPU-Zeit pro Repräsentant in Sekunden
- \bar{c} Arithmetisches Mittel der Anzahlen von semikanonischen Graphen
- $\frac{\bar{c}}{c}$ Mittleres Verhältnis der Anzahlen von kanonischen zu semikanonischen Graphen

Die Resultate wurden auf einer DEC 3000/600 AXP Workstation mit 333 MHz ermittelt.

Ein Vergleich mit anderen Ansätzen ist aufgrund der Neuartigkeit der hier vorgestellten Problemlösung leider nicht möglich. Es wird aber deutlich, daß die Kombination von Surjektiver Auflösung und Ordnungstreuer Erzeugung ein recht leistungsfähiges Verfahren ergeben hat.

Formel	Isomere	\bar{n}	n_{max}	n_{min}	σ	\bar{t}	$\frac{t}{n}$	\bar{c}	$\frac{c}{z}$
C ₁ H ₄	1	31,00	31	31	0,00	0,07	0,00226	31,00	1,000
C ₂ H ₂	1	8,00	8	8	0,00	0,01	0,00125	8,00	1,000
C ₂ H ₄	1	59,00	59	59	0,00	0,10	0,00169	59,00	1,000
C ₂ H ₆	1	184,00	184	184	0,00	0,44	0,00239	215,00	0,856
C ₃ H ₂	2	8,00	8	8	0,00	0,01	0,00187	8,00	1,000
C ₃ H ₄	3	81,00	106	59	19,30	0,11	0,00143	81,00	1,000
C ₃ H ₆	2	510,50	790	231	279,50	0,74	0,00170	578,00	0,883
C ₃ H ₈	1	1237,00	1237	1237	0,00	2,41	0,00195	1999,00	0,619
C ₄ H ₂	7	8,00	8	8	0,00	0,02	0,00250	8,00	1,000
C ₄ H ₄	11	92,18	174	31	43,93	0,14	0,00168	92,18	1,000
C ₄ H ₆	9	768,22	1041	215	240,33	0,94	0,00132	808,33	0,950
C ₄ H ₈	5	3253,60	6594	1411	1809,80	4,65	0,00154	4066,40	0,800
C ₄ H ₁₀	2	4422,50	5526	3319	1101,50	11,90	0,00255	9401,00	0,470
C ₅ H ₂	21	9,14	12	8	1,81	0,02	0,00226	9,14	1,000
C ₅ H ₄	40	119,28	294	59	51,83	0,18	0,00162	119,28	1,000
C ₅ H ₆	40	1100,95	2026	215	537,13	1,42	0,00140	1140,53	0,965
C ₅ H ₈	26	6176,62	12880	1599	3133,00	8,31	0,00142	7022,27	0,880
C ₅ H ₁₀	10	21184,60	44561	6747	11481,14	35,53	0,00176	29339,30	0,722
C ₅ H ₁₂	3	29789,67	45563	2982	19054,36	72,18	0,00306	53194,00	0,560
C ₆ H ₂	85	9,27	12	8	1,86	0,02	0,00244	9,27	1,000
C ₆ H ₄	185	135,31	294	59	57,42	0,22	0,00175	135,31	1,000
C ₆ H ₆	217	1372,37	3507	212	761,62	1,86	0,00146	1395,34	0,984
C ₆ H ₈	159	9277,69	31442	659	5340,60	12,62	0,00142	9889,28	0,938
C ₆ H ₁₀	77	39358,92	98752	3916	21891,30	60,12	0,00153	46301,78	0,850
C ₆ H ₁₂	25	105869,24	269935	8202	65180,74	201,27	0,00193	137456,68	0,770
C ₆ H ₁₄	5	161606,40	252001	48633	87638,88	382,51	0,00223	240017,40	0,673
C ₇ H ₂	356	10,07	12	8	2,00	0,03	0,00261	10,07	1,000
C ₇ H ₄	920	156,46	294	31	65,23	0,27	0,00183	156,46	1,000
C ₇ H ₆	1230	1713,86	6000	215	922,57	2,55	0,00155	1732,32	0,989
C ₇ H ₈	1031	12758,45	56464	1644	7812,51	18,47	0,00149	13235,46	0,964
C ₇ H ₁₀	575	63444,54	234715	3825	39464,16	100,72	0,00158	68942,65	0,920
C ₇ H ₁₂	222	210726,51	560260	8208	130376,11	440,44	0,00195	240579,44	0,876
C ₇ H ₁₄ *	37	382153,54	1149116	40947	260610,40	1090,13	0,00247	435643,49	0,877
C ₇ H ₁₆ *	6	346545,50	846536	93846	239520,10	1076,04	0,00282	414582,67	0,836

* Wegen beschränkter Rechenzeit und -kapazität konnten bei C₇H₁₄ nur 37 von 56 und bei C₇H₁₆ nur 6 von 9 Isomeren untersucht werden.

Tabelle 3.1: Rechenergebnisse bei der Erzeugung von Verknüpfungsgraphen von Butan (VT) und den Isomeren zu C_iH_j mit 1 ≤ i ≤ 7 und 2 ≤ j ≤ 16

Kapitel 4 Erzeugung gemeinsamer Teilgraphen

4.1 Strategie des Algorithmus

Eine Möglichkeit, zwei molekulare Graphen miteinander zu verbinden, ist es, alle gemeinsamen Teilstrukturen zu suchen und die Graphen mittels dieser, sofern möglich, zu verschmelzen. Dieser Ansatz, der bisher für Moleküle nur rudimentär ausgeführt wurde [133], erlaubt den Aufbau einer u.U. großen Zahl von Strukturen aus fest vorgegebenen Bausteinen. Wir betrachten also folgendes

4.1.1 Konstruktionsproblem. *Gegeben seien zwei molekulare Graphen γ_1, γ_2 . Bestimme alle nichtisomorphen molekularen Graphen, die sich durch Verschmelzen von γ_1 und γ_2 über alle ihre gemeinsamen Teilstrukturen ergeben.*

Da die Methoden sehr allgemein sind, sprechen wir im folgenden meist nur von Graphen statt von molekularen Graphen; die Beschriftung spielt nur am Rande eine Rolle. Alle weiteren Strukturen werden gleichfalls so abstrakt wie möglich gehalten, um die Vorgabe unnötiger Voraussetzungen zu vermeiden.

Die Grundidee ist folgende:

4.1.2 Verschmelzungsstrategie. Die Konstruktion der Verschmelzungsstrukturen zweier Graphen γ_1 und γ_2 wird in zwei Schritten durchgeführt:

- i) Erzeugung aller nichtisomorphen gemeinsamen Teilgraphen von γ_1 und γ_2
- ii) Bestimmung aller nichtisomorphen Abbildungen der gefundenen Teilgraphen aufeinander und Konstruktion der aus diesen Abbildungen resultierenden Graphen

Dabei ist der erste Schritt auch unabhängig vom Ziel der Verschmelzung ein wichtiges Problem in Graphentheorie [4, 47, 86, 92] und Chemie [10, 27, 31, 80, 93, 99, 121]. Daher ist dieses Kapitel dem Problem der Erzeugung gemeinsamer Teilgraphen gewidmet. Zunächst erläutern wir die globale Strategie des Verfahrens, bevor wir auf die Teilprobleme wie Teilgraphensuche, Minimalitätstest usw. eingehen. Einen für Schritt ii) (die Verschmelzung) geeigneten Algorithmus wollen wir im nächsten Kapitel vorstellen.

Wir beginnen mit einer Formalisierung des Problems:

4.1.3 Definition. *Seien $\gamma_1 \in \mathcal{M}^{\mathbf{p}^{[2]}}$ und $\gamma_2 \in \mathcal{M}^{\mathbf{q}^{[2]}}$ numerierte ungerichtete, schleifenfreie Multigraphen*. Gibt es zwei Teilgraphen $\eta_1 \in \mathcal{M}^{\mathbf{T}_1^{[2]}}$ und $\eta_2 \in \mathcal{M}^{\mathbf{T}_2^{[2]}}$, wobei $\eta_1 \subseteq \gamma_1$ und $\eta_2 \subseteq \gamma_2$ sowie $|\mathbf{T}_1| = |\mathbf{T}_2| = k \leq \min(p, q)$, so daß $\eta_1 \simeq \eta_2$, also ein kantenerhaltender Isomorphismus $\sigma : \mathbf{T}_1 \rightarrow \mathbf{T}_2$ existiert, so heißt das Paar (η_1, η_2) gemeinsamer k -Teilgraph von γ_1 und γ_2 .*

Die Menge aller gemeinsamen k -Teilgraphen von γ_1 und γ_2 bezeichnen wir als $P_k(\gamma_1, \gamma_2)$.

*Die Gleichheit von m in beiden Graphen ist wieder o.B.d.A.

Die Automorphismengruppen der jeweiligen Graphen operieren auch auf den Teilgraphen. Die Abbildungen vertauschen zum einen die Ecken der Teilgraphen untereinander; zum anderen bilden sie einen Teilgraphen auch auf einen isomorphen innerhalb des gegebenen Graphen ab. Denn ein Teilgraph kann natürlich mehrmals vorkommen. Diese Aspekte müssen bei der Konstruktion aller nichtisomorphen gemeinsamen Teilgraphen in Betracht gezogen werden. Somit entspricht die Menge der nichtisomorphen gemeinsamen k -Teilgraphen den Bahnen

$$P_k(\gamma_1, \gamma_2) // \text{Aut}(\gamma_1) \times \text{Aut}(\gamma_2)$$

Um alle nichtisomorphen gemeinsamen Teilgraphen zu erhalten, müssen wir also

$$(4.1.4) \quad \bigcup_{k=1}^{\min(p,q)-1} \mathcal{T}(P_k(\gamma_1, \gamma_2) // \text{Aut}(\gamma_1) \times \text{Aut}(\gamma_2))$$

bestimmen. Dabei können die einzelnen $P_k(\gamma_1, \gamma_2)$ ebenso wie die ganze Menge auch leer sein. Wir lassen die Vereinigung nur bis $\min(p, q) - 1$ laufen, um zu verhindern, daß ein ganzer Graph als gemeinsamer Teilgraph identifiziert wird. Die Definition 4.1.3 schließt diesen Fall nicht aus.

Für die eigentliche Konstruktion werden in der Literatur mehrere, recht ähnliche Verfahren angegeben (s. etwa [4, 10, 31, 47, 80, 86, 92]), die aber z.T. auch heuristische Vereinfachungen verwenden. Meist handelt es sich nur um Anpassung des Basisalgorithmus an spezielle Aufgabenstellungen. Eine grundlegend abweichende Idee findet sich nicht – vermutlich aufgrund der Struktur und der Komplexität des Problems.

Wir stützen uns hier auf die Vorarbeiten von V. Golender und A. Rozenblit [47] sowie G. Levi [86] und setzen zur Konstruktion die simultane Ordnungstreu Erzeugung nach 1.3.7 ein. Dabei stellen insbesondere der Minimaltest in 4.3 sowie verschiedene Detaillösungen (etwa die Berücksichtigung von Substrukturen mit freien Valenzen 4.1.11.ii oder der Zusammenhangstest in 4.2.2.ii) echte Neuentwicklungen dar.

Zudem begnügen sich die in der angegebenen Literatur beschriebenen Algorithmen meistens damit, den (oder die) größten gemeinsamen Teilgraphen zu finden. Im Hinblick auf unsere Anwendung zur Verschmelzung werden wir aber Wert auf die Erzeugung aller nichtisomorphen gemeinsamen Teilgraphen legen. Auch dies – zusammen mit der Einbettung in die Theorie der Automorphismengruppen und der Ordnungstremen Erzeugung – stellt eine Neuerung dar.

Schließlich zeigt das Studium der Literatur auch, daß für die Beweise der verwendeten Tatsachen stets auf Referenzen verwiesen wird, in denen die Sätze aber auch wieder nur vorgestellt werden. Somit stellt die hier zu findende durchgängige Beweisführung einen weiteren Aspekt unserer Aufarbeitung dar.

Will man alle gemeinsamen nichtisomorphen k -Teilgraphen zu zwei Graphen γ_1 und γ_2 bestimmen, so braucht man beim Vergleich jeder Ecke mit jeder (nach [86])

$$\frac{p! q!}{k! (p-k)! (q-k)!}$$

Operationen, wobei p die Zahl der Ecken von γ_1 und q die von γ_2 seien. Dieser hohe Aufwand kann deutlich verringert werden, indem man das Problem auf eine Suche nach vollständigen Teilgraphen eines Graphen zurückführt.

4.1.5 Definition. Ein Graph $\gamma \in m^{\mathbb{E}^{[2]}}$ heißt vollständig, falls jede Ecke mit jeder verbunden ist, also für alle $i, j \in \mathbf{p}$ mit $i \neq j$ gilt: $\gamma(\{i, j\}) \neq \emptyset$.

Bei vollständigen Graphen mit p Punkten ist also die Zahl der Kanten $\frac{p(p-1)}{2}$.

4.1.6 Beispiel. Die vollständigen schlichten Graphen mit bis zu 7 Punkten zeigt Abb. 4.1. ◇

Nun wollen wir untersuchen, welche Ecken für die Identifizierung gemeinsamer Teilgraphen überhaupt in Frage kommen. Seien dazu im folgenden $\gamma_1 \in m^{\mathbb{E}^{[2]}}$ und $\gamma_2 \in m^{\mathbb{E}^{[2]}}$ nummerierte ungerichtete m -Graphen.

4.1.7 Definition.

i) Ein Paar (i, j) von Ecken mit $i \in \mathbf{p}$ und $j \in \mathbf{q}$ heißt korrespondierendes Eckenpaar, falls

(a) γ_1 und γ_2 mit β_1 bzw. β_2 beschriftet sind und $\beta_1(i) = \beta_2(j)$ gilt,

(b) γ_1 und γ_2 unbeschriftet sind und $d_1(\gamma_1) = d_j(\gamma_2)$ gilt.

ii) Die Menge aller korrespondierenden Eckenpaare bezeichnen wir als

$$M_C(\gamma_1, \gamma_2) \subseteq \mathbf{p} \times \mathbf{q}.$$

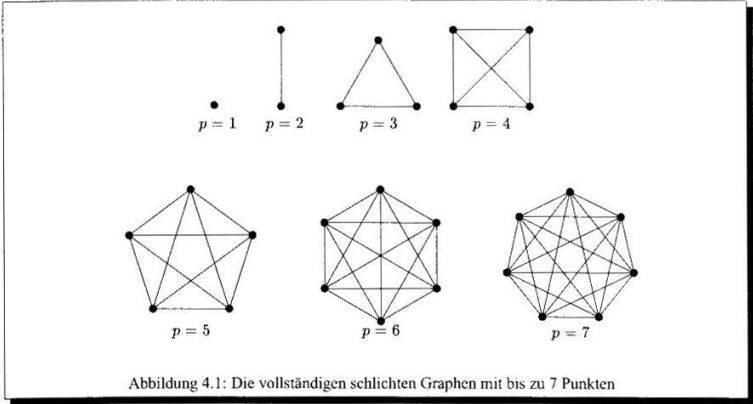


Abbildung 4.1: Die vollständigen schlichten Graphen mit bis zu 7 Punkten

Es ist klar, daß sich gemeinsame Teilgraphen stets nur aus korrespondierenden Eckenpaaren zusammensetzen. Zudem ist zu bemerken, daß sich auf M_G eine Ordnung ergibt, wenn man die Eckenpaare lexikographisch vergleicht.

4.1.8 Bemerkung. Bei molekularen Graphen, die bei unserer Betrachtung von besonderer Bedeutung sind, ist die Bedingung der gleichen Beschriftung schärfer als die der gleichen Valenz. Es steht auch frei, ob man für die Korrespondenz die Gleichheit der Anzahl der an dem jeweiligen Atom gebundenen Wasserstoffatome voraussetzen will oder nicht. Für die hier beabsichtigte Anwendung zur Verschmelzung ist es sinnvoll, die Wasserstoffe zu ignorieren; dann können auch Teilgraphenatome mit unterschiedlichen Wasserstoffanzahlen übereinandergelegt werden.

Es ist sogar so, daß jede Art von Beschriftung, die invariant unter der Automorphismengruppe ist, für die Festlegung der Korrespondenz herangezogen werden kann.

4.1.9 Definition.

i) Zwei korrespondierende Eckenpaare (i, j) und (i', j') mit $i \neq i'$ und $j \neq j'$ heißen kompatibel, falls $\gamma_1(\{i, i'\}) = \gamma_2(\{j, j'\})$.

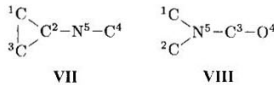
ii) Die Abbildung $\tilde{\chi} : M_G^{[2]} \rightarrow \{0, 1\}$ mit

$$\tilde{\chi}((i, j), (i', j')) = \begin{cases} 1 & \text{falls } (i, j) \text{ und } (i', j') \text{ kompatibel} \\ 0 & \text{sonst} \end{cases}$$

heißt Kompatibilitätsgraph. Seine Adjazenzmatrix bezeichnet man auch als Kompatibilitätsmatrix.

4.1.10 Beispiele.

i) Im folgenden werden wir als Beispiel folgende molekulare Graphen betrachten:



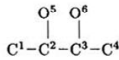
Dabei sind die korrespondierenden Eckenpaare:

$$\{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3), (4, 1), (4, 2), (4, 3), (5, 5)\}$$

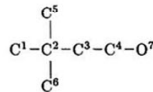
Die Kompatibilitätsmatrix hat also 13 Zeilen und Spalten. Sie lautet:

	1	1	1	2	2	2	3	3	3	4	4	4	5
(1,1)	0	0	0	0	0	0	0	0	0	0	1	1	0
(1,2)	0	0	0	0	0	0	0	0	0	1	0	1	0
(1,3)	0	0	0	0	0	0	0	0	0	1	1	0	0
(2,1)	0	0	0	0	0	0	0	0	0	0	1	1	1
(2,2)	0	0	0	0	0	0	0	0	0	0	1	0	1
(2,3)	0	0	0	0	0	0	0	0	0	1	1	0	1
(3,1)	0	0	0	0	0	0	0	0	0	0	1	1	0
(3,2)	0	0	0	0	0	0	0	0	0	1	0	1	0
(3,3)	0	0	0	0	0	0	0	0	0	1	1	0	0
(4,1)	0	1	1	0	1	1	0	1	1	0	0	0	1
(4,2)	1	0	1	1	0	1	1	0	1	0	0	0	1
(4,3)	1	1	0	1	1	0	1	1	0	0	0	0	1
(5,5)	0	0	0	1	1	1	0	0	0	1	1	1	0

- ii) Es folgt noch ein Beispiel, das auf den ersten Blick nur unwesentlich größer erscheint, aber eine bereits doppelt so große Kompatibilitätsmatrix hat. Dies zeigt, daß diese Matrizen schnell eine beachtliche Dimension erreichen können (maximal natürlich $p \cdot q$). Wir werden später auf diese Strukturen zurückkommen, da hier eine etwas größere Symmetrie (d.h. größere Automorphismengruppe) vorliegt als bei noch kleineren Graphen.



IX: Buta-2,3-diol



X: 2,2-Dimethylbutan-4-ol

Es ergeben sich folgende korrespondierende Eckenpaare:

$$\{(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), (5, 7), (6, 7)\}$$

Die Kompatibilitätsmatrix zeigt Tab. 4.1.

◇

Weitere Beispiele folgen.

4.1.11 Bemerkung. Die Art und Weise, wie Kompatibilität definiert wird, bestimmt die Form der Ergebnisse der folgenden Algorithmen wesentlich. So ist die Definition 4.1.9 gut geeignet, wenn vollständige Graphen gegeben und exakte Teilstrukturen gesucht sind. Es gibt aber auch noch weitere Möglichkeiten:

- i) Betrachtet man (i, j) und (i', j') als kompatibel, falls $d_{i,i'} = d_{j,j'}$, wobei $D = (d_{m,n})$ die Abstandsmatrix[†] sei, so können bei der Suche nach gemeinsamen Teilstrukturen Atome unterschiedlicher Beschriftung übersprungen werden. Dies führt zu Teilgraphen, die zuweilen dem bloßen Auge plausibler erscheinen als die exakt abgeleiteten, die aber für unsere Zwecke unbrauchbar sind.
- ii) Besteht die Eingabe selbst aus Substrukturen, also aus Molekülen, die an einigen Atomen noch freie Valenzen besitzen, welche sich in einer später darauf anzuwendenden Applikation noch verbinden, so dürfen bei der Bestimmung der Kompatibilität nicht nur die bestehenden Kanten in Betracht gezogen werden. Vielmehr

[†]Die Einträge der Abstandsmatrix D zu einem Graphen sind definiert als: d_{ij} ist die Länge des kürzesten Pfades von i nach j .

	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3	4	4	4	4	4	4	5	6			
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	7			
(1,1)	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	0	0	1	1	1	1	1	1			
(1,2)	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	1		
(1,3)	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1	1	1	0	0	0	1	1	1	1		
(1,4)	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	1	1	1	1	0	0	1	1	0	0		
(1,5)	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1		
(1,6)	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	1	1	0	1	0	1	1	1	0	1	1		
(2,1)	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1	0	1	
(2,2)	1	0	1	0	1	1	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0	1	0	0	0	1	
(2,3)	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	1	1	0	1	
(2,4)	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	1	1	0	
(2,5)	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	1	0	1	0	
(2,6)	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	1	
(3,1)	0	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
(3,2)	0	0	0	1	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0
(3,3)	1	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
(3,4)	1	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
(3,5)	1	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
(3,6)	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
(4,1)	0	0	1	1	1	1	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	
(4,2)	0	0	0	1	0	0	0	0	0	1	0	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	1	
(4,3)	1	0	0	0	1	1	1	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	
(4,4)	1	1	0	0	1	1	1	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
(4,5)	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	
(4,6)	1	0	1	1	0	1	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	
(5,7)	1	1	1	0	1	1	0	0	0	1	0	0	1	1	1	0	1	1	1	1	1	1	0	1	1	0	0	
(6,7)	1	1	1	0	1	1	1	1	0	1	1	0	0	0	1	0	0	1	1	0	0	1	1	0	1	1	0	0

Tabelle 4.1: Kompatibilitätsmatrix zu den Strukturen IX und X

ist es nötig, auch die potentiellen Kanten, also diejenigen zu berücksichtigen, die sich mit Hilfe von freien Valenzen schließen können.

Daher setzt man in diesem Fall, ausgehend vom wasserstoffreduzierten molekularen Graphen (vgl. 2.4.5): (i, j) und (i', j') mit $i \neq i'$ und $j \neq j'$ sind genau dann kompatibel, wenn

- $\gamma_1(\{i, i'\}) = \gamma_2(\{j, j'\})$ oder
- $\gamma_1(\{i, i'\}) < \gamma_2(\{j, j'\})$ und

$$\gamma_2(\{j, j'\}) - \gamma_1(\{i, i'\}) < \min(FV_{\gamma_1}(i), FV_{\gamma_1}(i'))$$

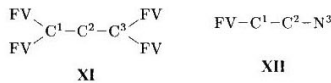
oder

- $\gamma_1(\{i, i'\}) > \gamma_2(\{j, j'\})$ und

$$\gamma_1(\{i, i'\}) - \gamma_2(\{j, j'\}) < \min(FV_{\gamma_2}(j), FV_{\gamma_2}(j'))$$

Dabei bezeichnet $FV_{\gamma_n}(k)$ die Anzahl der freien Valenzen der Ecke k im Graphen γ_n .

Als Beispiel betrachten wir folgende Strukturen:



Die korrespondierenden Eckenpaare sind: (1, 1), (1, 2), (2, 1), (2, 2), (3, 1), (3, 2).

Als Kompatibilitätsmatrix erhält man mit obiger Erweiterung (die sich dadurch ergebenden Kompatibilitäten sind hervorgehoben):

	1	1	2	2	3	3
	1	2	1	2	1	2
(1, 1)	0	0	0	1	0	1
(1, 2)	0	0	1	0	1	0
(2, 1)	0	1	0	0	0	1
(2, 2)	1	0	0	0	1	0
(3, 1)	0	1	0	1	0	0
(3, 2)	1	0	1	0	0	0

Bei den Paaren (1, 2) und (3, 1) etwa ist $\gamma_1(\{1, 3\}) = 0$ (sie sind unverbunden) und $\gamma_2(\{2, 1\}) = 1$ (eine Einfachbindung). Sowohl 1 als auch 3 in γ_1 haben aber 2 freie Valenzen; damit ist die Differenz der Bindungen kleiner als die zur Verfügung stehenden freien Valenzen. Da also die fehlende Bindung noch vermittels der freien Valenzen geschlossen werden könnte, gelten diese Paare als kompatibel.

- iii) Wie erwähnt, erzeugt das hier vorgestellte Verfahren Teilgraphen. Man kann es aber auch so modifizieren, daß gemeinsame *Untergraphen* generiert werden. Als Untergraph $\eta \preceq \gamma \in m^{\mathbf{p}^{[2]}}$ bezeichnen wir einen Graphen, für den gilt:

$$\eta \in m^{T^{[2]}} \text{ mit } T \subseteq \mathbf{p}, \forall i, j \in T : \eta(\{i, j\}) \leq \gamma(\{i, j\}).$$

Das bedeutet, daß η nicht alle Bindungen von γ haben muß. Um gemeinsame Untergraphen zu erzeugen, nimmt man als Eckenmenge von $\bar{\chi}$ neben den Eckenpaaren noch Paare von Kanten von γ_1 und γ_2 hinzu und definiert Kompatibilität entsprechend [80, 99]. Der Rest des Algorithmus kann in unveränderter Form verwendet werden.

Die objektorientierte Implementierung der vorgestellten Datenstrukturen und Algorithmen mittels C++ in NEX-TOR ermöglicht es, Definitionen u.ä. modular auszutauschen und anzupassen, ohne den gesamten Code ändern zu müssen.

4.2 Der Suchalgorithmus

Unser Suchverfahren gründet sich auf folgenden

4.2.1 Satz. *Zwei numerierte ungerichtete m -Graphen γ_1 und γ_2 haben genau dann einen gemeinsamen k -Teilgraphen, wenn ihr Kompatibilitätsgraph $\tilde{\chi}$ einen vollständigen k -Teilgraphen enthält.*

Beweis:

\Rightarrow Sei $(\eta_1, \eta_2) \in P_k(\gamma_1, \gamma_2)$. Nach Definition 4.1.3 gibt es einen kantenerhaltenden Isomorphismus $\sigma : T_1 \rightarrow T_2$, der die Ecken aufeinander abbildet, etwa $\sigma(i_r) = j_s$ für $r \in \underline{k}$ bei $T_1 = \{i_1, \dots, i_k\}$ und $T_2 = \{j_1, \dots, j_k\}$. Da es sich bei σ um einen Isomorphismus handelt, müssen alle (i_r, j_s) mit $r \in \underline{k}$ korrespondierende Eckenpaare im Sinne von 4.1.9 sein. Damit ist $T_1 \times T_2 \subseteq M_C(\gamma_1, \gamma_2)$, was wiederum bedeutet, daß $T_1 \times T_2$ einen Teilgraphen von $\tilde{\chi}$ definiert.

Da σ kantenerhaltend ist, gilt für alle $r, s \in \underline{k}$ mit $r \neq s$

$$\gamma_1(\{i_r, i_s\}) = \gamma_2(\{\sigma(i_r), \sigma(i_s)\}) = \gamma_2(\{j_r, j_s\})$$

Damit sind alle (i_r, j_s) zu allen (i_s, j_s) mit $r \neq s$ kompatibel. Für den entsprechenden Teilgraphen gilt:

$$\tilde{\chi}(\{i_r, j_r\}, \{i_s, j_s\}) = 1 \quad \forall r, s \in \underline{k}, r \neq s;$$

er ist also vollständig.

\Leftarrow $\tilde{\chi}$ enthalte den vollständigen k -Teilgraph mit den Ecken $(i_1, j_1), \dots, (i_k, j_k)$. Die Projektionen auf die erste bzw. zweite Komponente dieser Paare bestimmen je einen Teilgraphen von γ_1 und γ_2 , etwa durch

$$\begin{aligned} \eta_1 : T_1^{[2]} &\rightarrow m, & (i_r, i_s) &\mapsto \gamma_1(\{i_r, i_s\}) \\ \eta_2 : T_2^{[2]} &\rightarrow m, & (j_r, j_s) &\mapsto \gamma_2(\{j_r, j_s\}), \end{aligned}$$

wobei $T_1 := \{i_1, \dots, i_k\}$ und $T_2 := \{j_1, \dots, j_k\}$.

Die Abbildung $\sigma : T_1 \rightarrow T_2, i_r \mapsto j_r$ ist sicherlich bijektiv. Wegen der Vollständigkeit des Teilgraphen von $\tilde{\chi}$ gilt, daß (i_r, j_r) und (i_s, j_s) für alle $r \neq s$ kompatibel sind, also

$$\gamma_1(\{i_r, i_s\}) = \gamma_2(\{j_r, j_s\}) \quad \forall r, s \in \underline{k}, r \neq s.$$

Da zudem alle (i_r, j_r) als korrespondierend vorausgesetzt sind, ist σ somit ein kantenerhaltender Isomorphismus, und damit $(\eta_1, \eta_2) \in P_k(\gamma_1, \gamma_2)$, also ein gemeinsamer Teilgraph von γ_1 und γ_2 . □

Unser Verfahren zum Auffinden aller vollständigen Teilgraphen ist ein sog. *Backtrack*-Verfahren [111], das die vollständigen Teilgraphen in lexikographisch aufsteigender Reihenfolge bezüglich der Ordnung der Eckenpaare in M_C erzeugt.

Jeder vollständige Teilgraph entspricht zwar einem gemeinsamen Teilgraphen, aber diese können auch isomorph sein. Daher ist es notwendig, im Sinne von 1.3.3 zu testen, ob der gefundene Teilgraph minimal unter der Operation der Automorphismengruppen $Aut(\gamma_1)$ und $Aut(\gamma_2)$ ist (vgl. 4.1.4). Diesen *Minimaltest* wollen wir zunächst als *black box* verwenden und ihn im nächsten Abschnitt diskutieren.

Wir verwenden in der Beschreibung des Algorithmus folgende Bezeichnungen:

- l ist die aktuelle Tiefe im Suchbaum.
- k ist die Nummer der kleinsten Ecke von $\tilde{\chi}$, die in Tiefe l hinzugefügt werden kann.
- V ist das Feld, in dem der aktuelle Teilgraph von $\tilde{\chi}$ gespeichert wird.
- w ist die Ecke von $\tilde{\chi}$, die die Wurzel des aktuellen Suchbaums ist.
- X_l Zum Auffinden der vollständigen Teilgraphen von $\tilde{\chi}$ erweitern wir unseren Suchbaum nur um solche Ecken, die mit allen bisherigen (von der Wurzel des Suchbaums aus gesehen) eine Kante in $\tilde{\chi}$ haben. Wir bezeichnen daher mit X_l die Menge der Ecken, die in Tiefe l hinzugefügt werden kann.
- $U_{\tilde{\chi}}(x) := \{y \in M_C \mid \tilde{\chi}(x, y) = 1\}$, bezeichnet also die Menge der Eckenpaare, die zu einem gegebenen Paar kompatibel sind.
- \mathcal{P} ist die Menge aller Lösungen.

4.2.2 Bemerkung. Teil unserer generellen Strategie ist es zudem, mögliche Restriktionen, die der Benutzer an die Ergebnisse hat, so früh wie möglich zu berücksichtigen, um die Konstruktion überflüssiger Lösungen zu vermeiden (vgl. S. 43). Da i.a. der Minimaltest mehr Zeit als das Überprüfen der Restriktionen in Anspruch nimmt, wird dieser nur durchgeführt, wenn sich die Lösung mit den Vorgaben in Einklang befindet.

Mögliche Restriktionen sind:

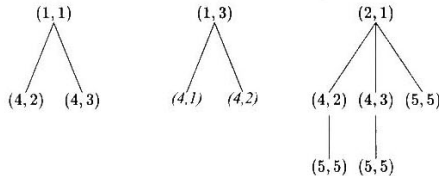
- Die Angabe einer maximalen Größe der zu suchenden Teilstrukturen. Bei einer solchen Vorgabe muß der Suchbaum nur bis zur entsprechenden Tiefe verfolgt werden.
- Die Forderung, nur zusammenhängende Teilgraphen zu bestimmen. Diese besonders wichtige Restriktion kann aber nicht so leicht zu einem Abschneiden des Suchbaumes verwendet werden, da eine später zu V hinzukommende Ecke u.U. erst den Zusammenhang herstellen kann. Mit folgender Überlegung gelingt ein Abschneiden allerdings in einer Reihe von Fällen: Sind keine der potentiellen weiteren Ecken der gemeinsamen Teilgraphen (also etwa kein i mit $(i, j) \in X_l$, denn es genügt, diese Eigenschaften für einen der beiden Ausgangsgraphen γ_r zu überprüfen) mit einer Ecke aus V (einem i' mit $(i', j') \in V$) oder einem Nachbarn einer Ecke aus V (einem i'' mit der Eigenschaft: $\exists (i', j') \in V : \gamma_1(\{i', i''\}) \neq 0$) gebunden (d.h. haben alle i mit $(i, j) \in X_l$ von $\{i' \mid \exists j' : (i', j') \in V\}$ in γ_1 mindestens Abstand 2), so kann in einer größeren Tiefe daraus kein zusammenhängender gemeinsamer Teilgraph mehr entstehen und der Ast des Suchbaums muß nicht weiter verfolgt werden.

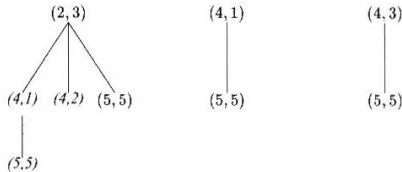
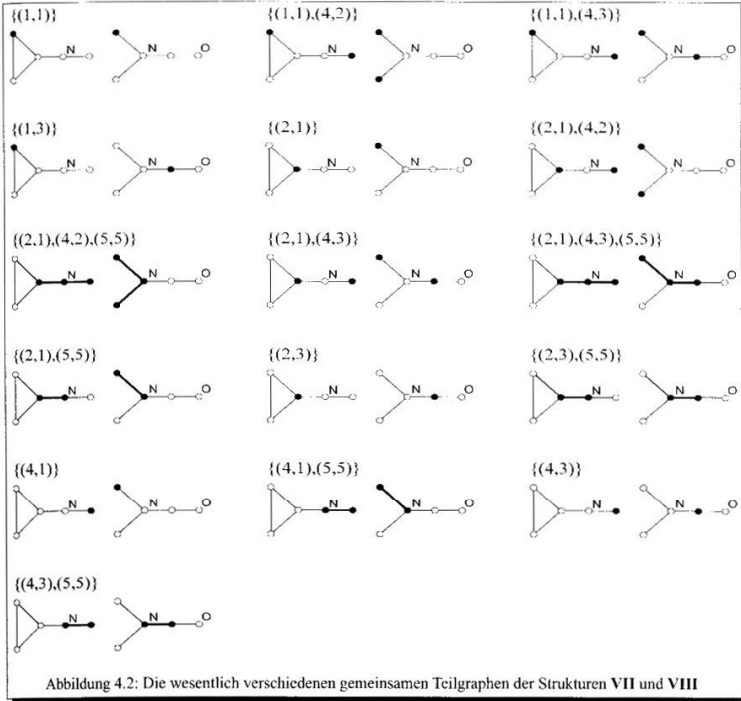
4.2.3 Algorithmus (Teilgraphensuche). Initialisiere $\mathcal{P} \leftarrow \emptyset$. Durchlaufe alle $w \in M_C$ und führe jeweils durch:

- Solange w nicht minimal ist, so nimm das nächste Element aus M_C als Wurzel.
- Setze $\mathcal{P} \leftarrow \mathcal{P} \cup \{V\}$ bei $V = \{w\}$.
- Beginne den Suchbaum von der Wurzel w aus: $l \leftarrow 1, X_l \leftarrow U_{\bar{x}}(w)$. Setze als Startwert für die Auswahl der Ecken $k \leftarrow w$.
- Solange $l \geq 0$ ist, führe durch:
 - Suche ein $x \in X_l$ mit $x > k$ (bezüglich der Ordnung in M_C). Gibt es kein solches, so gehe zu Schritt f.
 - Dieses x ist nun eine neue Ecke, also erhöhe $l \leftarrow l + 1$ und setze $v_l \leftarrow x$.
 - Erfüllt die Menge $V = \{w, v_1, \dots, v_l\}$ die gegebenen Restriktionen und ist sie minimal, so ist sie eine neue Lösung: $\mathcal{P} \leftarrow \mathcal{P} \cup \{V\}$.
 - Setze den Startwert $k \leftarrow x$ und schränke die potentiellen weiteren Ecken auf die Nachbarn von x ein, die auch mit allen bisherigen verbunden sind: $X_l \leftarrow X_{l-1} \cap U_{\bar{x}}(x)$.
 - Sollen nur zusammenhängende Lösungen erzeugt werden, so überprüfe die Bedingung aus 4.2.2.ii und gehe zu Schritt a, falls die Bedingung erfüllt ist, der Minimaltest, sofern durchgeführt, nicht negativ war und die vorgegebene Tiefe nicht bereits erreicht ist. Sind auch nichtzusammenhängende Lösungen erlaubt, so gehe zu Schritt a, falls der Minimaltest, sofern durchgeführt, nicht negativ war und die vorgegebene Tiefe nicht bereits erreicht ist.
 - Führe einen Backtrack-Schritt durch: $k \leftarrow v_l, l \leftarrow l - 1$.

▽

4.2.4 Beispiel. Für die Strukturen VII und VIII aus 4.1.10.i ergeben sich folgende Suchbäume.





Dabei sind die Kandidaten, die sich mit den kursiv hervorgehobenen Ecken ergeben, nicht minimal. Somit ist die Lösungsmenge:

$$\mathcal{P} = \{ \{(1,1)\}, \{(1,1), (4,2)\}, \{(1,1), (4,3)\}, \{(1,3)\}, \\ \{(2,1)\}, \{(2,1), (4,2)\}, \{(2,1), (4,2), (5,5)\}, \\ \{(2,1), (4,3)\}, \{(2,1), (4,3), (5,5)\}, \{(2,1), (5,5)\}, \\ \{(2,3)\}, \{(2,3), (5,5)\}, \}$$

$$\{(4, 1)\}, \{(4, 1), (5, 5)\}, \{(4, 3)\}, \{(4, 3), (5, 5)\}$$

Abbildung 4.2 veranschaulicht diese Lösungen.

Bei diesem Beispiel sind fast alle vom Algorithmus erzeugten vollständigen Teilgraphen nichtisomorph und daher Lösungen. Dies liegt in der geringen Symmetrie der Ausgangsgraphen begründet. Bei stärker symmetrischen Strukturen reduziert sich die Anzahl der Lösungen entsprechend. \diamond

4.3 Der Minimalitätstest

In diesem Abschnitt wollen wir ein Verfahren beschreiben, mit dem die Kandidaten, die mit dem Algorithmus 4.2.3 zur Teilgraphensuche erzeugt werden, auf ihre Minimalität getestet werden.

Im Gegensatz zur Neuerzeugung von Graphen ist bei der Teilgraphensuche die Anzahl der möglichen Permutationen, die zu untersuchen sind, wesentlich geringer. Es handelt sich dabei nämlich „nur“ um die Automorphismen der Ausgangsgraphen, wobei aber auch deren Anzahl groß werden kann.

Im folgenden seien wieder $\gamma_1 \in m^{\mathbb{Z}^{(2)}}$ und $\gamma_2 \in m^{\mathbb{Z}^{(2)}}$ numerierte ungerichtete m -Graphen. Wir haben bereits nach 4.1.7 festgestellt, daß die Ordnung in \mathbf{p} und \mathbf{q} eine (lexikographische) Ordnung auf $M_C(\gamma_1, \gamma_2)$ induziert. Und auch die Operationen der Automorphismengruppen übertragen sich entsprechend:

4.3.1 Lemma. *Die Operationen von $Aut(\gamma_1)$ auf \mathbf{p} und $Aut(\gamma_2)$ auf \mathbf{q} induzieren eine Operation von $Aut(\gamma_1) \times Aut(\gamma_2)$ auf $M_C(\gamma_1, \gamma_2)$.*

Beweis: Daß es sich bei der Anwendung von $Aut(\gamma_1) \times Aut(\gamma_2)$ auf $M_C \subseteq \mathbf{p} \times \mathbf{q}$ um eine Operation handelt, ist bis auf die Abgeschlossenheit klar. Seien also $(i, j) \in M_C$ und $(\pi, \sigma) \in Aut(\gamma_1) \times Aut(\gamma_2)$. Wir unterscheiden wie in Definition 4.1.7. i:

- (a) Ist γ_1 und γ_2 mit β_1 bzw. β_2 beschriftet, so gilt nach 2.4.7: $\beta_1 \pi(i) = \beta_1(i) = \beta_2(j) = \beta_2 \sigma(j)$ und damit $(i, j)^{(\pi, \sigma)} \in M_C$. Fordert man für die Korrespondenz noch die Gleichheit der Anzahlen der Wasserstoffatome, so ist auch dies bei topologisch äquivalenten Ecken erfüllt.
- (b) Bei unbeschrifteten Graphen haben topologisch äquivalente Ecken auch gleichen Grad, was wiederum bedeutet, daß $(i, j)^{(\pi, \sigma)} \in M_C$ ist.

□

Da wir in Algorithmus 4.2.3 nicht direkt auf die Ecken der beiden Ausgangsgraphen zugreifen, sondern dies über eine zusätzliche Datenstruktur, den Kompatibilitätsgraphen, erledigen, müssen wir noch den Zusammenhang der jeweiligen Automorphismengruppen herstellen:

4.3.2 Lemma. *Die Automorphismengruppe des Kompatibilitätsgraphen ist das direkte Produkt der Automorphismengruppen der Ausgangsgraphen:*

$$Aut(\gamma_1) \times Aut(\gamma_2) = Aut(\tilde{\chi}).$$

Beweis: Seien $(i_1, j_1), (i_2, j_2) \in M_C(\gamma_1, \gamma_2)$ kompatible Eckenpaare. Dann gilt:

$$\begin{aligned} & (\pi, \sigma) \in Aut(\gamma_1) \times Aut(\gamma_2) \\ \Leftrightarrow & \gamma_1 \left(\{i_1 \pi^{-1}, i_2 \pi^{-1}\} \right) = \gamma_1 \pi(\{i_1, i_2\}) = \gamma_1(\{i_1, i_2\}) = \\ & \gamma_2(\{j_1, j_2\}) = \gamma_2 \sigma(\{j_1, j_2\}) = \gamma_2 \left(\{j_1 \sigma^{-1}, j_2 \sigma^{-1}\} \right) \\ \Leftrightarrow & \tilde{\chi}^{(\pi, \sigma)}(\{(i_1, j_1), (i_2, j_2)\}) = \tilde{\chi} \left(\{(i_1 \pi^{-1}, j_1 \sigma^{-1}), (i_2 \pi^{-1}, j_2 \sigma^{-1})\} \right) = \\ & \tilde{\chi}(\{(i_1, j_1), (i_2, j_2)\}) \\ \Leftrightarrow & (\pi, \sigma) \in Aut(\tilde{\chi}) \end{aligned}$$

Sind (i_1, j_1) und (i_2, j_2) nicht kompatibel, so zeigt die zweite Zeile, daß dann auch $(i_1, j_1)^{(\pi, \sigma)}$ und $(i_2, j_2)^{(\pi, \sigma)}$ nicht kompatibel sind. Da diese Eckenpaare ansonsten beliebig waren, folgt die Behauptung.

□

Um also minimale vollständige Teilgraphen von $\tilde{\chi}$ zu erhalten, genügt es, auf Minimalität bezüglich $Aut(\gamma_1) \times Aut(\gamma_2)$ zu testen.

Im folgenden setzen wir der Einfachheit halber

$$A := Aut(\gamma_1), B := Aut(\gamma_2).$$

Beim Durchlauf in Algorithmus 4.2.3 wird zunächst die Minimalität der Wurzel des Suchbaums untersucht. Da bei der Berechnung der Automorphismengruppe in 2.4.10 bereits die Bahnen ebenfalls gespeichert werden, ist dazu nur zu untersuchen, ob für $w = (i, j)$ gilt:

$$w = (\min A(i), \min B(j))$$

Für den Test längerer Vektoren verwenden wir ein rekursives Verfahren, das analog zu 3.3.7 und 3.3.8 arbeitet. Sei dazu $V := \{v_1, \dots, v_k\}$ die Eckenmenge von $\tilde{\chi}$, die auf Minimalität getestet werden soll. Wir benutzen wieder die Tatsache, daß die Kandidaten kanonisch (hier lexikographisch aufsteigend) erzeugt werden, und überprüfen in Rekursionstiefe l , ob v_l minimal in seiner Bahn unter $C_{A \times B}(\{v_1, \dots, v_{l-1}\})$ ist oder ob sich ein anderes Element v_r mit $r > l$, „dazwischen schieben“ kann. Um letztere Gruppe zu bestimmen, wird obige Bemerkung zusammen mit einem evtl. Wechsel der Basis (s. 2.2.10) verwendet. Indem man jeden Erzeuger auf jedes (bisher gefundene) Bahnelement anwendet, erhält man sowohl die Bahn von v_l als auch die fusionierenden Elemente, die die einzelnen Bahnelemente auf v_l transportieren (vgl. 3.3.6).

Die Bahnenberechnung erledigt ein vorgeschalteter Schritt, zu dessen Beschreibung wir neben V einige weitere Bezeichnungen benötigen:

- $v_i^{(r)}$ ist die r -Komponente von v_i mit $r = 1, 2$.
- $C_G^{(l)} := C_G(\{v_1^{(r)}, \dots, v_{l-1}^{(r)}\})$ mit $G = A$ bzw. $G = B$ bei $r = 1, 2$ und $C_A^{(0)} = A$ bzw. $C_B^{(0)} = B$.
- x ist das aktuelle Bahnelement, womit alle (bisher gefundenen) Elemente einer Bahn durchlaufen werden.
- $\phi_i^{(r)}(i)$ ist die Permutation (das fusionierende Element), die i auf $v_i^{(r)}$ abbildet mit $r = 1, 2$.
- $O_i^{(r)}$ ist die Bahn von $v_i^{(r)}$ unter $C_G^{(l)}$ mit $G = A$ bzw. $G = B$ bei $r = 1, 2$.
- σ ist der aktuelle Erzeuger von $C_G^{(l)}$ nach 3.3.5.

4.3.3 Algorithmus (Bahnenberechnung für den Minimaltest).

- i) Durchlaufe alle $v_l \in V$ und führe jeweils durch:
 - (a) Initialisiere $O_i^{(1)} \leftarrow \{v_i^{(1)}\}$, $O_i^{(2)} \leftarrow \{v_i^{(2)}\}$.
 - (b) Setze das aktuelle Bahnelement $x \leftarrow v_i^{(1)}$ und $\phi_i^{(1)}(v_i^{(1)}) \leftarrow id_A$.
 - (c) Durchlaufe alle Erzeuger $\sigma \in C_A^{(l)}$ und führe jeweils durch:
 - Ist $x^\sigma \notin O_i^{(1)}$, so füge es hinzu: $O_i^{(1)} \leftarrow O_i^{(1)} \cup \{x^\sigma\}$ und setze $\phi_i^{(1)}(x^\sigma) \leftarrow \phi_i^{(1)}(x) \cdot \sigma$.
 - (d) Gibt es noch ein weiteres $x \in O_i^{(1)}$, so gehe damit zu Schritt c.
 - (e) Führe die Schritte b-d analog für B und $v_i^{(2)}$ durch.
 - (f) Ist $v_l \neq (\min O_i^{(1)}, \min O_i^{(2)})$, so kann V kein minimaler Repräsentant sein. Brich daher den Test mit dem Ergebnis „negativ“ ab.
 - (g) Wechsle die Basis von $C_A^{(l)}$ und $C_B^{(l)}$ so, daß eine Darstellung für $C_A^{(l+1)}$ und $C_B^{(l+1)}$ wie in 3.3.5 entsteht.
- ii) Rufe den rekursiven Test 4.3.4 mit Tiefe l und Kandidat V auf und gib dessen Ergebnis zurück.

▽

Der rekursive Test hat als Argumente die aktuelle Tiefe l und eine Eckenmenge $U \subset M_C$, die durch Anwendung der fusionierenden Elemente entsteht, wie sie in 4.3.3 ermittelt wurden. Dabei setzt man bei $\pi \in A$ und $\sigma \in B$

$$v_i^{(\pi, \sigma)} = ((v_i^{(1)})^\pi, (v_i^{(2)})^\sigma) \text{ und } V^{(\pi, \sigma)} = \{v_1^{(\pi, \sigma)}, \dots, v_k^{(\pi, \sigma)}\}$$

Dann lautet das Verfahren:

4.3.4 Algorithmus (Rekursiver Minimaltest von U in Tiefe l).

- i) Durchlaufe alle $x \in O_i^{(1)}$ und führe jeweils durch:
 - Durchlaufe alle $y \in O_i^{(2)}$ und führe jeweils durch:
 - (a) Wende die fusionierenden Elemente von x und y auf U an: $U' := U^{(\phi_i^{(1)}(x), \phi_i^{(2)}(y))}$.
 - (b) Ist $U' < V$, so ist V kein minimaler Repräsentant; gib daher „negativ“ zurück.
 - (c) Ist $l < k$ und $C_A^{(l+1)} \neq \{id_A\}$ oder $C_B^{(l+1)} \neq \{id_B\}$, so rufe die nächste Tiefe des Tests mit U' und $l + 1$ auf. Gib dieser „negativ“ zurück, so gib ebenfalls „negativ“ zurück.
- ii) Gib „positiv“ zurück.

▽

4.3.5 Bemerkung. Mit dem in 4.3.4 angegebenen Test wird vermieden, daß zwei isomorphe Teilgraphen von $\bar{\chi}$ erzeugt werden. Ein weiteres Problem bleibt aber bestehen: In der bislang beschriebenen Form ist es möglich, daß der Algorithmus zwei vollständige Teilgraphen von $\bar{\chi}$ erzeugt, die aber als gemeinsame Teilgraphen von γ_1 und γ_2 völlig identisch sind. Das ist kein Widerspruch zu den Sätzen 4.2.1 oder 4.3.2, sondern liegt allein in der Tatsache begründet, daß wir zwar *Mengen* erzeugen und vergleichen wollen, in Wirklichkeit (und anders ist es auch in einem Rechner nicht machbar) generieren wir aber *Felder*, also geordnete Mengen.

Die lexikographische Erzeugung sichert lediglich, daß die Kandidaten in der ersten Komponente aufsteigend sind. Bei nicht monoton aufsteigenden zweiten Komponenten indessen können sich gleiche gemeinsame Teilgraphen ergeben. Bei Strukturen VII und VIII aus 4.1.10.i und 4.2.4 etwa sind $\{(2, 1), (4, 3)\}$ und $\{(2, 3), (4, 1)\}$ sicher nicht isomorph unter der Operation der Automorphismengruppe. Allerdings stellen beide denselben gemeinsamen Teilgraphen dar.

Aus diesem Grund haben wir dem Minimaltest noch eine zusätzliche Überprüfung vorgeschaltet. Dabei bezeichnen wir als *Normalform* eine monoton aufsteigende Anordnung in der ersten Komponente und die lexikographisch kleinste Anordnung in der zweiten Komponente, so daß sich ein vollständiger Teilgraph von $\bar{\chi}$ ergibt. Sollte also ein Kandidat in der zweiten Komponente nicht monoton aufsteigend sein, so wird er verworfen, falls er nicht in Normalform ist. (In unserem Beispiel ist dies $\{(2, 1), (4, 3)\}$.) Denn nur in diesem Fall können wir sicher sein, daß der Kandidat in dieser Form bereits erzeugt und getestet wurde bzw. in einem generell nicht-minimalen Ast des Suchwaldes liegt. Ist er in Normalform, so wird er auf seine Minimalität unter der Operation der Automorphismengruppe mit 4.3.4 untersucht.

Dieses Normalform-Kriterium vermeidet also, daß identische gemeinsame Teilgraphen generiert werden. Um es zum Abschneiden des Suchbaumes zu verwenden, sind aber noch weitere Verfeinerungen erforderlich, die im Rahmen dieser Arbeit leider nicht möglich waren.

4.3.6 Beispiel. Wir kommen auf die Strukturen IX und X aus Beispiel 4.1.10.ii zurück. Diese haben die Automorphismengruppen

$$\begin{aligned} A &= \{1, (14)(23)(56)\} \\ B &= \{1, (15), (16), (56), (156), (165)\} \end{aligned}$$

i) Zunächst soll $V = \{(1, 2), (2, 5)\}$ auf Minimalität getestet werden. Wir führen dazu Algorithmus 4.3.3 durch.

- (a) Wir stellen fest: $O_1^{(1)} = \{1, 4\}$, $\phi_1^{(1)}(4) = (14)(23)(56)$, also $v_1^{(1)} = \min O_1^{(1)}$. Für die zweite Komponente gilt: $O_1^{(2)} = \{2\}$ und somit $v_1^{(2)} = \min O_1^{(2)}$.
- (b) Für v_2 erhalten wir: $O_2^{(1)} = \{2\}$ und $O_2^{(2)} = \{1, 5, 6\}$ mit $\phi_2^{(2)}(1) = (15)$ und $\phi_2^{(2)}(6) = (56)$. Folglich ergibt sich:

$$v_2 = (2, 5) > (\min O_2^{(1)}, \min O_2^{(2)}) = (2, 1)$$

Damit ist V kein minimaler Repräsentant.

ii) Ein weiterer Kandidat sei $V = \{(1, 1), (3, 5)\}$. Bei der Initialisierung mit 4.3.3 ergibt sich:

- (a) $O_1^{(1)} = \{1, 4\}$, $\phi_1^{(1)}(4) = (14)(23)(56)$. Außerdem ist $O_1^{(2)} = \{3\}$.
- (b) $O_2^{(1)} = \{1, 5, 6\}$ mit $\phi_1^{(2)}(5) = (15)$ und $\phi_1^{(2)}(6) = (156)$ sowie $O_2^{(2)} = \{5, 6\}$ mit $\phi_2^{(2)}(6) = (56)$.

Danach wird der rekursive Test 4.3.4 mit V aufgerufen (Die Einrücktiefe spiegelt die Rekursionstiefe wider.):

- $U' = V$
 - $U' = V$ und unter $\phi_2^{(2)}(6)$: $U' = \{(1, 1), (3, 6)\} > V$, also bestanden.
- Unter der fusionierenden Permutation $\phi_1^{(2)}(5) = (15)$ ergibt sich $U' = \{(1, 5), (3, 1)\} > V$.
 - $U' = \{(1, 5), (3, 1)\}$ oder $\{(1, 6), (3, 1)\}$, beide wieder größer als V .
- Wir erhalten $U' = \{(1, 5), (3, 6)\} > V$ mit der Permutation $\phi_1^{(2)}(6) = (156)$.
 - $U' = \{(1, 5), (3, 6)\}$ oder $\{(1, 6), (3, 5)\}$ – abermals bestanden.

- Wendet man zusätzlich noch $\phi_1^{(1)}(4)$ an, so ergeben sich auch ausschließlich U' , die größer als V sind.

⋮

Der Kandidat besteht in allen Rekursionstiefen, ist also ein minimaler Repräsentant.

Insgesamt gibt es zu diesen beiden Strukturen 52 nichtisomorphe gemeinsame Teilgraphen.

◇

C_iH_j	0	2	4	6	8	10	12	14	16
1	—	—	1	—	—	—	—	—	—
2	1	1	1	1	—	—	—	—	—
3	1	2	3	2	1	—	—	—	—
4	3	7	11	9	5	2	—	—	—
5	6	21	40	40	26	10	3	—	—
6	19	85	185	217	159	77	25	5	—
7	50	356	920	1230	1031	575	222	56	9

Tabelle 4.2: Anzahlen von Kohlenstoffisomeren, erzeugt mit **MOLGEN**. (Der Zeilenindex bezeichnet die Anzahl der Kohlenstoffatome, der Spaltenindex die der Wasserstoffatome.)

4.4 Diskussion des Verfahrens

In diesem Abschnitt wollen wir den Algorithmus zur Bestimmung aller gemeinsamen Teilgraphen diskutieren, seinen Aufwand theoretisch und anhand praktischer Tests betrachten und mögliche Anwendungen vorstellen.

4.4.1 Aufwand

Bei den Vorberechnungen ist es einfach, einen Aufwand für den schlechtesten Fall anzugeben:

- i) Die Erstellung der Menge $M_C(\gamma_1, \gamma_2)$ der korrespondierenden Eckenpaare mit $\gamma_1 \in m^{E^{[2]}}$ und $\gamma_2 \in m^{E^{[2]}}$ braucht $\mathcal{O}(pq)$ Operationen.
- ii) Für die Aufstellung des Kompatibilitätsgraphen $\bar{\chi}$ ergibt sich ein Aufwand von $\mathcal{O}(p^2q^2)$.

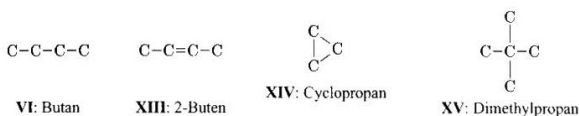
Die weiteren Schritte erfordern eine zusätzliche Erläuterung: Die meisten in der Informatik untersuchten Probleme gehören zur Klasse der *nichtdeterministisch-polynomiale (NP) Probleme*. Das bedeutet, daß diese Probleme auf einer nichtdeterministisch arbeitenden Maschine in polynomialer Zeit gelöst werden können. (Zu Details der Komplexitätstheorie siehe [45].) Eine weitere Klasse, die eine Teilmenge der NP-Probleme bilden, sind die *NP-vollständigen* Probleme. Alle NP-vollständigen Probleme lassen sich polynomial in alle anderen Probleme dieser Klasse transformieren. Daher würde das Auffinden eines Algorithmus, der ein NP-vollständiges Problem in polynomialer Zeit löst, bedeuten, daß alle NP-vollständigen Probleme in polynomialer Zeit lösbar sind. Leider wurde ein solcher Algorithmus bislang nicht gefunden, und man vermutet, daß dies prinzipiell nicht möglich ist.

Zu den bekanntesten NP-vollständigen Problemen gehört neben dem Problem des Handlungsreisenden und der Teilgraphenisomorphie auch das Auffinden maximaler vollständigen Teilgraphen, welche man auch als *Cliquen* bezeichnet [45, Problem GT19]. Das wiederum heißt, daß Algorithmus 4.2.3 ebenfalls nichtpolynomial arbeitet. Bei der Suche nach gemeinsamen Teilgraphen wird die Analyse – wie auch Levi in [86] betont – noch dadurch erschwert, daß sowohl die Zahl der Ecken als auch die der Kanten von $\bar{\chi}$ sehr variieren können. Allerdings sind wir durch die bei uns verwendete Ordnungstreue Erzeugung und die damit verbundene Möglichkeit, ganze Suchbäume „abzuschneiden“, gegenüber reinen Backtrack-Suchverfahren im Vorteil.

4.4.2 Experimentelle Resultate

Auch hier haben wir wie in Abschnitt 3.5 zur Sicherstellung eines möglichst breiten Spektrums von Graphen alle 5.418 Isomere der Formel C_iH_j mit $0 \leq i \leq 7$ und $0 \leq j \leq 16$ herangezogen. (Tabelle 4.2 zeigt die Anzahlen der Isomere im einzelnen.) Diese wurden mit Hilfe von **MOLGEN** (s. S. 59) erzeugt und anschließend mit mehreren Teststrukturen auf gemeinsame Teilgraphen verglichen.

Als Vergleichsgraphen wurden folgende Strukturen untersucht:



Wir ordnen die Ergebnisse nach der Anzahl der Kanten im Kompatibilitätsgraphen, einer Größe, die für die Komplexität des Verfahrens, wie oben bemerkt, eine entscheidende Rolle spielt. Tabellen 4.3, 4.4, 4.5 und 4.6 zeigen dazu jeweils folgende Werte:

$ E(\bar{\chi}) $	Anzahl der Kanten des Kompatibilitätsgraphen
\bar{n}	Arithmetisches Mittel der Anzahlen gemeinsamer Teilgraphen
n_{\max}	Höchste Anzahl gemeinsamer Teilgraphen beim Vergleich mit allen Isomeren
n_{\min}	Kleinste Anzahl gemeinsamer Teilgraphen
σ	Standardabweichung
\bar{t}	Mittlere CPU-Zeit für die Durchführung von Algorithmus 4.2.3 in Sekunden
$\frac{\bar{t}}{n}$	Mittlere CPU-Zeit pro Repräsentant in Sekunden

Ein $|E(\bar{\chi})|$ - \bar{n} -Diagramm für diese drei Beispiele findet sich in Abb. 4.3. Die „Zacken“ mancher Kurven erklären sich aus dem häufigen bzw. seltenen Auftreten der verglichenen Strukturen in den Isomerenfamilien. Ein $|V(\bar{\chi})|$ - \bar{n} -Diagramm (hinsichtlich der Zahl der korrespondierenden Eckenpaare) böte tendenziell dasselbe Bild; daher wird darauf hier verzichtet.

Die Resultate wurden auf einer DEC 3000/600 AXP Workstation mit 333 MHz ermittelt; im $|E(\bar{\chi})|$ - \bar{t} -Diagramm für obige Beispiele in Abb. 4.4 sehen wir, daß der Zeitaufwand sehr stark vom jeweiligen Problem abhängt, so daß die Rechenzeiten auch hinter dem theoretischen exponentiellen Wachstum (vgl. 4.4.1) zurückbleiben können. Diesen Eindruck verstärkt auch die Betrachtung der Rechenzeiten pro Resultat in Abb. 4.5 und der mittleren Rechenzeiten, aufgetragen gegenüber den Anzahlen von Lösungen, in Abb. 4.6. (Die größere Steigung für Struktur XV, Dimethylpropan, erklärt sich durch die höhere Symmetrie dieses Graphen, was zu relativ weniger Lösungen führt und damit wiederum den Zeitaufwand pro Lösung erhöht.)

Ein Vergleich mit anderen Arbeiten ist nur sehr schwer durchzuführen. Das hat mehrere Gründe:

- Vielfach herrscht ja der Wunsch vor, nur die größte gemeinsame Teilstruktur zu finden. Nach deren Auffinden bricht das Verfahren ab [10, 27, 92].
- Zum Teil werden auch heuristische Vereinfachungen verwendet, etwa die Einschränkung auf planare Graphen [10] oder Abstandsmatrizen [47].
- Viele Autoren geben keine expliziten Rechenzeiten an [4, 27, 86, 92, 99] oder verwenden Maschinen, mit denen ein Vergleich schwerfällt [31, 47, 80].
- Auf der anderen Seite haben auch einige Gruppen mit dem Problem der Isomorphie von Lösungen zu kämpfen, die sie nach eigenen Angaben nicht in allen Fällen oder nur unzureichend schnell erkennen können [10, 31].
- Außerdem macht es einen großen Unterschied, ob man bei der Definition der korrespondierenden Eckenpaare eine Gleichheit der Wasserstoffe voraussetzt oder nicht (vgl. 4.1.8). Im letzten Fall, den wir den Berechnungen dieses Abschnitts zugrunde gelegt haben, werden die Kompatibilitätsmatrizen um ein vielfaches größer und dementsprechend die Rechenzeiten. Bei den Strukturen IX und X aus 4.1.10.ii beispielsweise hat $\bar{\chi}$ mit Berücksichtigung der Wasserstoffe nur 8 Ecken, ohne aber 26.

Allgemein läßt sich daher nur sagen, daß die hier vorgestellte Lösung angesichts des NP-vollständigen Charakter des Problems, dessen Auswirkungen sich bei allen Ansätzen niederschlagen, durchaus konkurrenzfähig ist. Auf Besonderheiten wurde bereits in Abschnitt 4.1, Seite 62f., eingegangen.

4.4.3 Anwendungen

Das Problem des Auffindens gemeinsamer Teilgraphen hat neben seiner Berechtigung als abstraktes graphentheoretisches Problem eine Reihe von Anwendungen. Obgleich im Rahmen dieser Arbeit davon nur wenige untersucht werden konnten, sollte das modulare Konzept der hier vorgestellten Algorithmen und ihrer Implementation eine Anpassung an die konkrete Aufgabenstellung relativ leicht ermöglichen.

Mögliche Anwendungen (mit Schwergewicht auf der Chemie) sind unter anderem:

$ E(\bar{x}) $	\bar{n}	n_{\max}	n_{\min}	σ	\bar{t}	$\frac{\bar{t}}{\bar{n}}$
0	0,00000	0	0	0,00000	0,00000	0,00000
6	5,66667	7	3	1,88562	0,00000	0,00000
12	8,60000	12	5	2,87054	0,00200	0,00029
18	8,00000	9	5	1,73205	0,00500	0,00078
24	11,71429	19	5	4,02543	0,00500	0,00048
30	14,06667	21	5	4,62553	0,00667	0,00044
36	12,25000	17	5	3,99218	0,00875	0,00083
42	21,11111	29	14	5,03200	0,01278	0,00060
48	23,93103	33	10	6,40545	0,01448	0,00062
54	26,87234	37	7	7,60352	0,01830	0,00070
60	19,30435	30	5	7,07374	0,01478	0,00076
66	27,82143	41	15	7,98236	0,01821	0,00067
72	34,30000	47	8	10,47330	0,02427	0,00074
78	39,92953	55	13	10,77898	0,02973	0,00076
84	41,89815	60	7	12,37863	0,03389	0,00083
90	34,05000	61	8	13,06513	0,03175	0,00097
96	40,90625	58	26	10,56929	0,03156	0,00079
102	49,29304	66	20	12,40385	0,04055	0,00085
108	58,04131	74	19	12,66767	0,04911	0,00086
114	63,02259	83	23	14,60803	0,05534	0,00090
120	65,29125	89	24	16,50119	0,06059	0,00095
126	57,21530	93	11	19,57260	0,05762	0,00105

Tabelle 4.3: Rechenergebnisse bei der Suche nach gemeinsamen Teilgraphen von Butan (VI) und den Isomeren zu C_4H_8 mit $0 \leq i \leq 7$ und $0 \leq j \leq 16$

$ E(\bar{x}) $	\bar{n}	n_{\max}	n_{\min}	σ	\bar{t}	$\frac{\bar{t}}{\bar{n}}$
0	0,00000	0	0	0,00000	0,00000	0,00000
6	6,17391	9	2	1,94829	0,00087	0,00012
12	7,40000	10	3	2,06794	0,00382	0,00051
18	9,20755	11	3	1,94928	0,00516	0,00058
24	10,02077	12	3	2,22466	0,00641	0,00065
30	11,17419	13	3	2,05207	0,00736	0,00067
36	11,87238	14	3	2,48249	0,00863	0,00075
42	12,67367	15	3	2,60079	0,00975	0,00078
48	13,24068	16	4	3,03031	0,01096	0,00085
54	13,47179	17	3	3,37456	0,01195	0,00091
60	13,46040	18	3	3,71122	0,01287	0,00099
66	12,93333	19	7	3,81750	0,01367	0,00111
72	11,34286	20	3	4,10535	0,01314	0,00119
78	10,22222	13	7	1,98762	0,01333	0,00131
84	4,00000	4	4	0,00000	0,00000	0,00000

Tabelle 4.4: Rechenergebnisse bei der Suche nach gemeinsamen Teilgraphen von Cyclopropan (XIV) und den Isomeren zu C_3H_4 mit $0 \leq i \leq 7$ und $0 \leq j \leq 16$

1. Verschmelzung molekularer Graphen

Im nächsten Kapitel werden als eine wesentliche und neue Anwendung des Teilgraphensuchverfahrens die Verschmelzung molekularer Graphen vorstellen. Genauer gesagt handelt es sich dabei um die Bestimmung aller nichtisomorphen Abbildungen der gefundenen Teilgraphen aufeinander und Konstruktion der aus diesen Abbildungen resultierenden Graphen. Auf konkrete Anwendungen werden wir in späteren Kapiteln eingehen.

2. Erkennung von NMR-Verschiebungen

Die Interpretation von ^{13}C -NMR Spektren basiert meistens auf dem Vergleich des Spektrums einer unbekanntem mit einer großen Referenzmenge von Daten bekannter Strukturen. Dabei wird die Erkennung der größten gemeinsamen Teilgraphen zum direkten Vergleich der chemischen Shift-Werte verwendet, was eine automatische Analyse der Inkremente sowie die Suche nach Fehlenträgen in Spektrendatenbanken ermöglicht [27, 28].

$ E(\bar{x}) $	\bar{x}	n_{\max}	n_{\min}	σ	\bar{z}	$\frac{z}{\bar{z}}$
0	0,00000	0	0	0,00000	0,00000	0,00000
2	3,00000	3	3	0,00000	0,00000	0,00000
4	3,00000	3	3	0,00000	0,00000	0,00000
6	4,00000	4	4	0,00000	0,00000	0,00000
8	6,50000	7	6	0,50000	0,00000	0,00000
10	8,66667	11	7	1,69967	0,00000	0,00000
12	8,00000	12	4	4,00000	0,00000	0,00000
14	8,00000	8	8	0,00000	0,00000	0,00000
20	5,66667	6	5	0,47140	0,00333	0,00067
22	10,71429	13	8	1,48461	0,00429	0,00038
24	12,00000	18	4	4,28528	0,00545	0,00044
26	14,11111	21	9	3,95655	0,00556	0,00038
28	13,20000	20	6	4,44522	0,00400	0,00025
30	10,00000	12	8	2,00000	0,00500	0,00042
40	9,83333	15	4	4,21966	0,00500	0,00060
42	18,71429	31	9	5,50448	0,01000	0,00057
44	22,02500	31	9	6,41673	0,01250	0,00058
46	23,62500	34	9	6,64713	0,01300	0,00055
48	24,11538	33	10	7,08371	0,01423	0,00062
50	23,50000	36	8	8,16395	0,01200	0,00057
52	17,33333	23	8	6,64998	0,01000	0,00045
66	15,68421	28	6	6,07889	0,01263	0,00077
68	29,50588	48	12	9,56787	0,01929	0,00067
70	34,10270	52	12	10,37123	0,02303	0,00069
72	36,66359	53	6	10,41411	0,02507	0,00071
74	37,49057	53	10	10,48021	0,02522	0,00068
76	37,15584	53	15	10,40238	0,02597	0,00071
78	33,88000	54	10	12,03435	0,02560	0,00077
80	27,00000	37	14	7,87401	0,02000	0,00074
98	30,62000	64	8	12,18998	0,02700	0,00097
100	47,56461	74	17	14,55863	0,03854	0,00084
102	53,18043	76	10	14,06414	0,04223	0,00082
104	56,50813	77	19	13,14718	0,04469	0,00081
106	57,26091	78	21	13,22699	0,04593	0,00082
108	56,99130	76	19	13,64245	0,04654	0,00084
110	54,90090	76	23	14,23386	0,04626	0,00087
112	51,39286	76	12	15,08982	0,04464	0,00089
114	40,55556	75	21	16,93196	0,03889	0,00104

Tabelle 4.5: Rechenergebnisse bei der Suche nach gemeinsamen Teilgraphen von Buten (XIII) und den Isomeren zu C_iH_j mit $0 \leq i \leq 7$ und $0 \leq j \leq 16$

3. Interpretation von Massenspektren

Der Ansatz, ein selbstlernendes System zur Interpretation von Massenspektren zu verwenden, liefert zu einem unbekanntem Spektrum eine Liste von Referenzmolekülen, die mit der unbekanntem Verbindung verschiedene Teilstruktureigenschaften gemeinsam haben sollten. Ein paarweiser Vergleich dieser Referenzmoleküle kann dann eingesetzt werden, um die maximale Teilstruktur höchster Frequenz zu finden [31].

4. Syntheseplanung, Klassifikation von Reaktionen

Für die Planung einer Synthese ist es hilfreich, das gewünschte Produkt mit leicht verfügbaren Ausgangsverbindungen (Edukten) zu vergleichen, um mögliche molekulare Bausteine und damit einen effektiven Syntheseweg zu finden. Dabei spielen gemeinsame Teilgraphen von Edukten und Produkten eine wichtige Rolle [31, 32].

Dieser Zusammenhang kann auch zur Klassifikation von Reaktionen eingesetzt werden. Dabei betrachtet man die Bindungen, die im Zuge der Reaktion gebrochen oder gebildet werden. Diese lassen sich leicht ermitteln, sobald die gemeinsamen (maximalen) Teilstrukturen bekannt sind [93].

5. Wirkstoffdesign

Beim Design und der Synthese von neuen Wirkstoffen mit einer erwünschten pharmakophorischen Aktivität kann ein Vergleich der Strukturen bekannter Wirkstoffe eine Liste der gemeinsamen Fragmente liefern. Dies

$ E(\bar{\chi}) $	\bar{n}	n_{\max}	n_{\min}	σ	\bar{z}	$\frac{\bar{z}}{\sigma}$
0	0,00000	0	0	0,00000	0,00000	0,00000
8	4,50000	6	3	1,50000	0,00000	0,00000
12	6,00000	6	6	0,00000	0,00000	0,00000
16	6,00000	6	6	0,00000	0,00500	0,00083
20	9,00000	9	9	0,00000	0,00000	0,00000
24	4,00000	4	4	0,00000	0,00500	0,00125
28	7,00000	7	7	0,00000	0,00000	0,00000
32	5,50000	7	4	1,50000	0,00000	0,00000
36	8,33333	10	7	1,24722	0,00667	0,00081
40	8,37500	14	5	2,86956	0,00625	0,00066
44	10,83333	15	8	2,91071	0,00833	0,00078
48	9,85714	15	4	3,04390	0,00714	0,00085
52	11,16667	15	8	3,02306	0,00833	0,00075
56	9,00000	12	6	3,00000	0,01000	0,00125
60	7,66667	10	4	2,62467	0,00667	0,00070
68	11,00000	11	11	0,00000	0,01500	0,00136
72	12,71429	19	9	2,91373	0,01429	0,00113
76	14,92857	21	11	3,23661	0,01571	0,00103
80	15,13043	22	4	4,87500	0,02174	0,00162
84	17,00000	25	9	4,04969	0,02267	0,00134
88	18,00000	26	10	4,41378	0,02407	0,00136
92	17,66667	26	11	4,48631	0,02714	0,00158
96	19,15385	24	9	4,34734	0,02769	0,00152
100	15,60000	22	6	5,27636	0,02400	0,00156
104	16,33333	22	11	4,49691	0,03333	0,00214
108	6,00000	6	6	0,00000	0,01000	0,00167
116	12,00000	14	11	1,22474	0,01750	0,00150
120	15,07143	26	5	5,22992	0,02286	0,00159
124	20,45455	31	10	6,26240	0,03273	0,00163
128	21,93421	36	8	6,91173	0,03526	0,00164
132	24,62712	38	6	7,01548	0,04212	0,00175
136	25,57931	41	10	7,24228	0,04503	0,00178
140	26,45270	40	11	7,05608	0,04845	0,00186
144	27,57273	41	7	7,38605	0,05273	0,00195
148	27,20290	39	13	6,92104	0,05449	0,00203
152	27,16216	42	11	7,52119	0,05514	0,00202
156	24,66667	36	8	7,87753	0,05167	0,00212
160	24,00000	32	16	5,51362	0,05400	0,00222
168	6,00000	6	6	0,00000	0,01000	0,00167
176	21,00000	22	19	1,22474	0,04500	0,00214
180	23,55000	39	14	7,12373	0,04950	0,00213
184	29,77612	45	14	8,11268	0,06015	0,00205
188	33,31795	48	13	8,99068	0,06897	0,00210
192	36,27475	56	16	8,92217	0,07800	0,00218
196	38,25076	58	7	9,03856	0,08505	0,00225
200	39,78744	60	15	9,30318	0,09226	0,00234
204	40,90244	63	14	9,58099	0,09811	0,00242
208	41,66967	63	17	9,49801	0,10230	0,00248
212	42,26636	62	17	9,61668	0,10617	0,00253
216	41,43612	60	17	10,13474	0,10740	0,00261
220	41,87629	60	17	9,85802	0,11134	0,00269
224	38,13793	58	8	10,42783	0,10724	0,00284
228	36,77778	56	22	10,79895	0,10444	0,00285

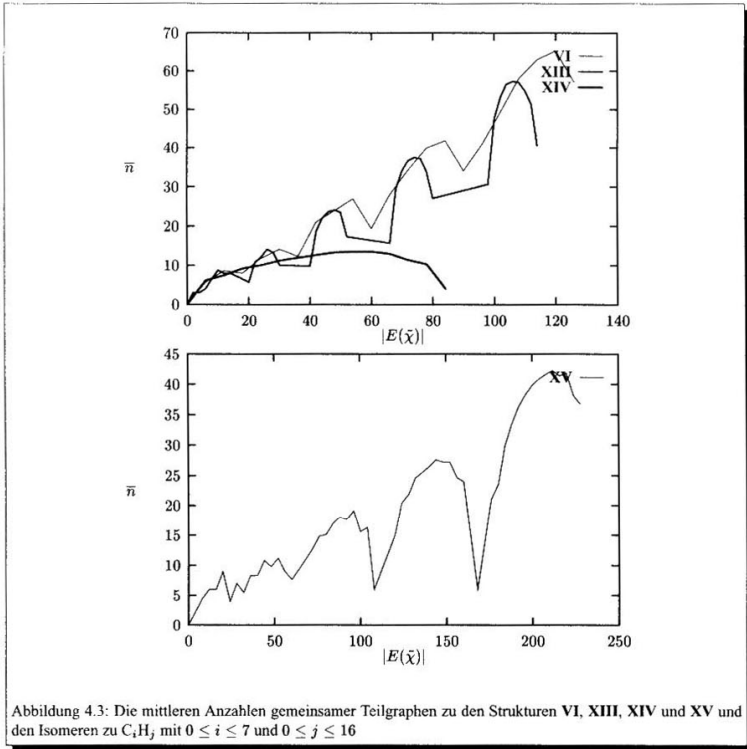
Tabelle 4.6: Rechenergebnisse bei der Suche nach gemeinsamen Teilgraphen von Dimethylpropan (XV) und den Isomeren zu C_7H_{16} mit $0 \leq i \leq 7$ und $0 \leq j \leq 16$

kann einen Hinweis auf die aktive Teilstruktur geben, die für die Wirkung verantwortlich ist [116].

6. Ähnlichkeit von Molekülen

Die quantitative Analyse der Ähnlichkeit von Molekülen ist von großer Bedeutung bei molekularen Datenbanken, der Wirkstoffuntersuchung, der Vorhersage physiochemischer Eigenschaften und der Modellierung von Rezeptorschnittstellen. Eine Möglichkeit, Ähnlichkeit zu definieren, besteht in der Betrachtung gemeinsamer Teilgraphen. Sind $\gamma_1 \in m^{p^1}$ und $\gamma_2 \in m^{q^2}$ Graphen (der Einfachheit halber unbeschriftet, was aber hierbei keine entscheidende Rolle spielt), so setzt man

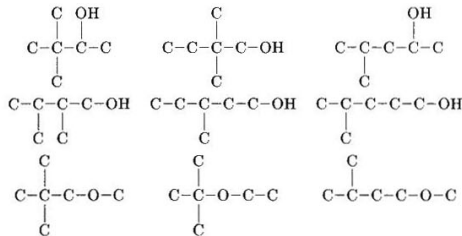
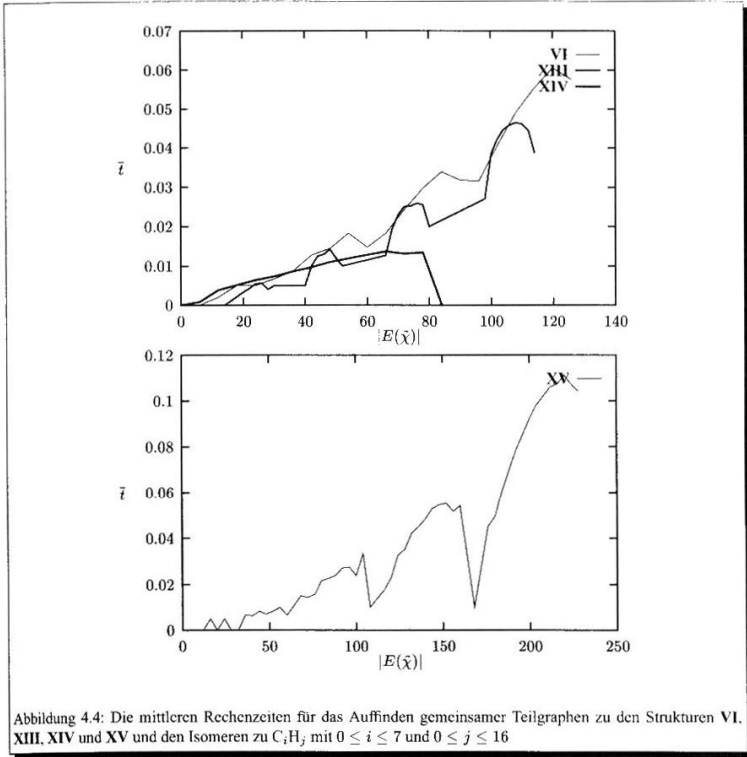
$$\mu(\gamma_1, \gamma_2) = \max\{k \mid k \in \{1, \dots, \min(p, q)\} \wedge P_k(\gamma_1, \gamma_2) \neq \emptyset\}.$$



Dann läßt sich der Abstand der Graphen definieren als [64, 118]

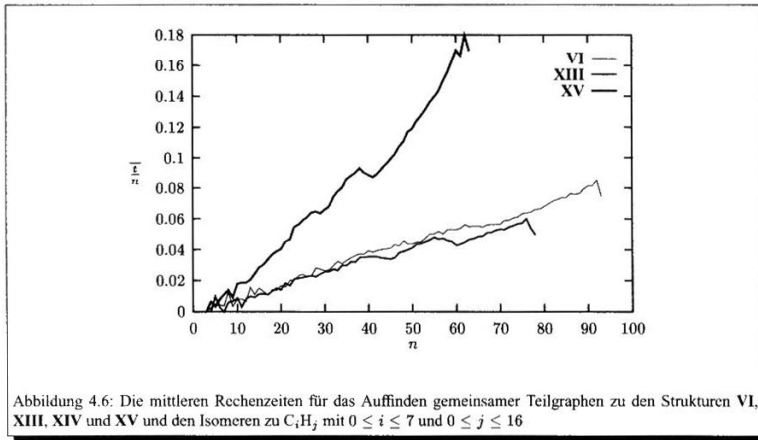
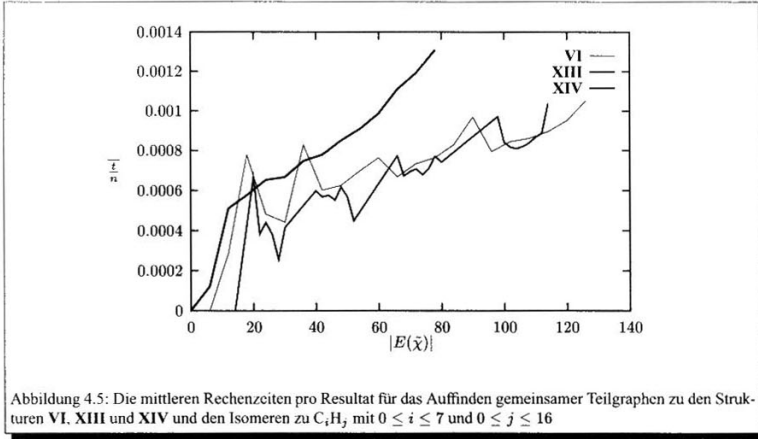
$$d(\gamma_1, \gamma_2) = |\gamma_1| + |\gamma_2| - 2\mu(\gamma_1, \gamma_2) = p + q - 2\mu(\gamma_1, \gamma_2).$$

Damit bestimmt d eine Metrik auf der Menge aller bis auf Isomorphie bestimmten Multigraphen. Auf diese Weise lassen sich dann auch Umgebungen von molekularen Graphen definieren. Je kleiner der Abstand, also desto ähnlicher gelten die Moleküle. So umfaßt etwa die Umgebung mit max. Abstand 2 von 2,2-Dimethylbutan-4-ol (Struktur X) in der Menge aller 31 Isomere zu $C_6H_{14}O$ folgende 9 molekulare Graphen:



7. Bilderkennung

Im Rahmen von Systemen mit künstlicher Intelligenz zur Bilderkennung verwendet man die Teilgraphensu-



che, um Objekte zu identifizieren, die mit einer Videokamera aufgenommen wurden [2, 4].

4.5 Erweiterung auf mehrere Ausgangsgraphen

Während die Suche nach gemeinsamen Teilgraphen zweier Graphen ein mehrfach untersuchtes Problem ist, gibt es nur wenige Ansätze zum Auffinden gemeinsamer Teilgraphen in mehreren Graphen. Dabei läßt man entweder einen vorher gefundenen Teilgraphen so wachsen, daß er stets in allen Ausgangsstrukturen enthalten ist [121], oder man führt diese Aufgabe auf den zweielementigen Fall zurück [10].

Die Allgemeinheit des hier entwickelten Verfahrens erlaubt es, dieses auf natürliche Weise auf die Anwendung für mehrere Ausgangsgraphen zu erweitern.

4.5.1 Definition. Gegeben seien n Graphen $\gamma_1 \in m^{\mathbf{p}_1^{[2]}}$, \dots , $\gamma_n \in m^{\mathbf{p}_n^{[2]}}$, wobei wieder m o.E. gleich sei. Gesucht ist nun ein n -Tupel (η_1, \dots, η_n) mit $\eta_i \in m^{\mathbf{T}_i^{[2]}}$, $\eta_i \subseteq \gamma_i$. Zusätzlich seien $T_1 := \dots = T_n = T$ und $\forall i, j \in \mathbf{n} : \eta_i \simeq \eta_j$. Ein solches Tupel wollen wir gemeinsamen k -Teilgraphen von $\gamma_1, \dots, \gamma_n$ nennen.

Damit überträgt sich auch die Formulierung 4.1.4 des Suchproblems sowie die Definition 4.1.7 korrespondierender Eckentupel entsprechend. Die Menge aller korrespondierenden Eckentupel bezeichnen wir wieder mit

$$M_C(\gamma_1, \dots, \gamma_n) \subseteq \mathbf{p}_1 \times \dots \times \mathbf{p}_n$$

4.5.2 Definition.

i) Zwei korrespondierende Eckentupel (i_1, \dots, i_n) und (i'_1, \dots, i'_n) heißen kompatibel, falls

$$\gamma_1(\{(i_1, i'_1)\}) = \dots = \gamma_n(\{(i_n, i'_n)\}).$$

ii) Die Abbildung $\tilde{\chi} : M_C^{[2]} \rightarrow \{0, 1\}$ mit

$$\tilde{\chi}(\{(i_1, \dots, i_n), (i'_1, \dots, i'_n)\}) = \begin{cases} 1 & \text{falls } (i_1, \dots, i_n) \text{ und } (i'_1, \dots, i'_n) \\ & \text{kompatibel} \\ 0 & \text{sonst} \end{cases}$$

heißt Kompatibilitätsgraph.

Wir haben das Problem der Teilgraphensuche in mehreren Graphen wieder auf eine Suche nach vollständigen Graphen in $\tilde{\chi}$ zurückgeführt, denn Satz 4.2.1 überträgt sich ganz entsprechend. Wir können also Algorithmus 4.2.3 in dieser Form anwenden, wobei wir übrigens auch das Zusammenhangsargument 4.2.2.ii beibehalten können, müssen aber den Minimaltest modifizieren.

Zunächst ist klar, daß die Lemmata 4.3.1 und 4.3.2 entsprechend auch für den Fall $n > 2$ gelten. Setzt man $A_i := \text{Aut}(\gamma_i)$, so kann man mit Bezeichnungen analog zu den in 4.3.3 eingeführten die erste Stufe des Minimaltests formulieren:

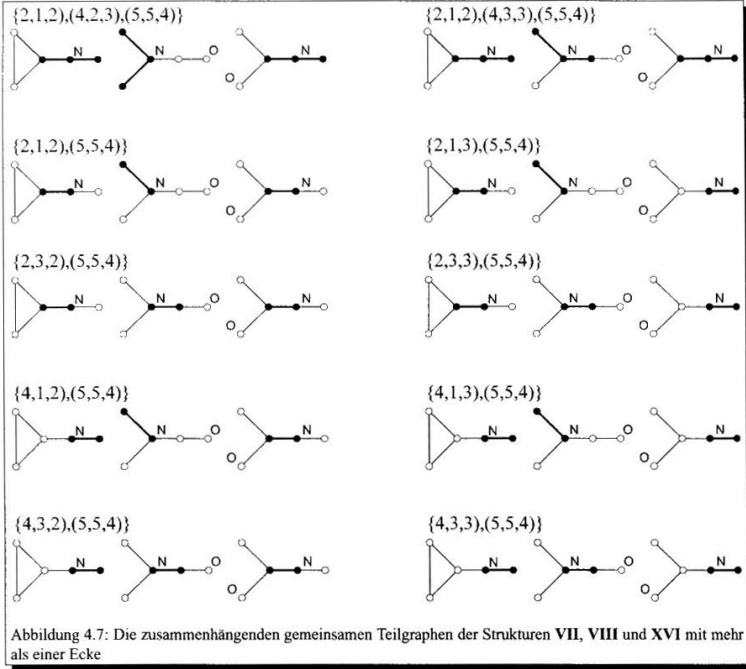
4.5.3 Algorithmus (Bahnenberechnung für den Minimaltest).

i) Durchlaufe alle $v_i \in V$ und führe jeweils durch:

- (a) Initialisiere $O_i^{(i)} \leftarrow \{v_i^{(i)}\}$ für alle $i \in \mathbf{n}$.
- (b) Setze das aktuelle Bahnelement $x \leftarrow v_i^{(1)}$ und $\phi_i^{(1)}(v_i^{(1)}) \leftarrow id_{A_i}$.
- (c) Durchlaufe alle Erzeuger $\sigma \in C_{A_i}^{(i)}$ und führe jeweils durch:
 - Ist $x^\sigma \notin O_i^{(1)}$, so füge es hinzu: $O_i^{(1)} \leftarrow O_i^{(1)} \cup \{x^\sigma\}$ und setze $\phi_i^{(1)}(x^\sigma) \leftarrow \phi_i^{(1)}(x) \cdot \sigma$.
- (d) Gibt es noch ein weiteres $x \in O_i^{(1)}$, so gehe damit zu Schritt c.
- (e) Führe die Schritte b-d analog für A_2, \dots, A_n und $v_i^{(2)}, \dots, v_i^{(n)}$ durch.
- (f) Ist $v_i \neq (\min O_i^{(1)}, \dots, \min O_i^{(n)})$, so kann V kein minimaler Repräsentant sein. Brich daher den Test mit dem Ergebnis „negativ“ ab.
- (g) Wechsle die Basis der $C_{A_i}^{(i)}$ bis $C_{A_n}^{(i)}$ so, daß eine Darstellung für $C_{A_1}^{(i+1)}, \dots, C_{A_n}^{(i+1)}$ wie in 3.3.5 entsteht.

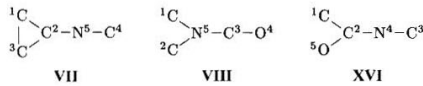
ii) Rufe den rekursiven Test mit Tiefe 1 und Kandidat V auf und gib dessen Ergebnis zurück.

▽



Im rekursiven Test werden dann alle Bahnen verschachtelt und daraus die fusionierenden Elemente angewandt (wieder in Analogie zu 4.3.4). Auch das Normalformkriterium aus 4.3.5 kann entsprechend übernommen werden.

4.5.4 Beispiel. Wir betrachten wieder die Strukturen VII und VIII aus 4.1.10.i und nehmen noch eine dritte Struktur hinzu:



Hierzu gibt es 37 korrespondierende Eckentripel. Wir beschränken uns diesmal auf die zusammenhängenden Lösungen. Diese sind:

$$\begin{aligned}
 \mathcal{P} = & \{ \{(1, 1, 1)\}, \{(1, 1, 2)\}, \{(1, 1, 3)\}, \{(1, 3, 1)\}, \\
 & \{(1, 3, 2)\}, \{(1, 3, 3)\}, \{(2, 1, 1)\}, \{(2, 1, 2)\}, \\
 & \{(2, 1, 2), (4, 2, 3), (5, 5, 4)\}, \{(2, 1, 2), (4, 3, 3), (5, 5, 4)\}, \\
 & \{(2, 1, 2), (5, 5, 4)\}, \{(2, 1, 3)\}, \{(2, 1, 3), (5, 5, 4)\}, \\
 & \{(2, 3, 1)\}, \{(2, 3, 2)\}, \{(2, 3, 2), (5, 5, 4)\},
 \end{aligned}$$

$\{(2, 3, 3)\}, \{(2, 3, 3), (5, 5, 4)\},$
 $\{(4, 1, 1)\}, \{(4, 1, 2)\}, \{(4, 1, 2), (5, 5, 4)\},$
 $\{(4, 1, 3)\}, \{(4, 1, 3), (5, 5, 4)\}, \{(4, 3, 1)\},$
 $\{(4, 3, 2)\}, \{(4, 3, 2), (5, 5, 4)\}, \{(4, 3, 3)\}, \{(4, 3, 3), (5, 5, 4)\}$

Abbildung 4.7 zeigt die Lösungen mit mehr als einer gemeinsamen Ecke.

◇

Kapitel 5 Verschmelzung zweier Graphen

5.1 Berechnung der Zuordnungen

In diesem Kapitel wollen wir die Details des in 4.1.1 und 4.1.2 erwähnten Verfahrens zur Verschmelzung von Graphen über alle gemeinsamen Teilgraphen vorstellen. Wir nehmen dabei engen Bezug zu den Definitionen aus Abschnitt 4.1. Seien wieder $\gamma_1 \in m^{\mathbb{E}^{[2]}}$ und $\gamma_2 \in m^{\mathbb{A}^{[2]}}$ Multigraphen. (Die Allgemeinheit der Darstellung erfordert es nicht, daß wir uns auf molekulare Graphen einschränken.)

Wir betrachten nun ein festes Paar gemeinsamer Teilgraphen, wie in 4.1.3 definiert. Dort wurde nur vorausgesetzt, daß es irgendeinen kantenerhaltenden Isomorphismus $\sigma : T_1 \rightarrow T_2$ gibt. Es ist offensichtlich, daß dieser nicht notwendig eindeutig ist. Daher müssen wir noch die wesentlich verschiedenen Zuordnungen bestimmen.* Dies liefert uns folgender

5.1.1 Satz. *Ist $(\eta_1, \eta_2) \in m^{T_1^{[2]}} \times m^{T_2^{[2]}}$ ein gemeinsamer k -Teilgraph der Graphen γ_1 und γ_2 und $\sigma_0 : T_1 \rightarrow T_2$ ein kantenerhaltender Isomorphismus, so entsprechen der Menge aller wesentlich verschiedenen Zuordnungen zwischen η_1 und η_2 die Doppelnebenklassen*

$$\sigma_0(\sigma_0^{-1}A\sigma_0) \backslash \text{Aut}(\eta_2) / B,$$

wobei A und B die auf Permutationsgruppen abgebildeten Teilgraphenautomorphismengruppen von η_1 und η_2 sind.

Beweis: Die Automorphismen von γ_1 und γ_2 , die die Teilgraphen invariant lassen, induzieren nach 2.3.2 Automorphismengruppen isomorph zu

$$N_{\text{Aut}(\gamma_1)}(\eta_1) / C_{\text{Aut}(\gamma_1)}(\eta_1) \quad \text{und} \quad N_{\text{Aut}(\gamma_2)}(\eta_2) / C_{\text{Aut}(\gamma_2)}(\eta_2).$$

Diese bilden wir mit geeigneten Homomorphismen auf Permutationsgruppen $A \leq \text{Aut}(\eta_1)$ und $B \leq \text{Aut}(\eta_2)$ ab. Mit $\sigma_0 : T_1 \rightarrow T_2$ als Referenzisomorphismus haben wir genau die Situation des Verklebungslemmas 1.2.7, welches die Behauptung liefert. □

Verschmelzungen molekularer Graphen lassen sich ganz analog konstruieren. Es ist dazu nur jeweils die Automorphismengruppe $\text{Aut}(\gamma)$ durch die des molekularen Graphen $\text{Aut}(\gamma, \beta)$ (vgl. 2.4.7) zu ersetzen.

Die in 4.1.2 erwähnte Konstruktion der aus den Zuordnungen zwischen den Teilgraphen resultierenden Graphen stellt kein Konstruktionsproblem im eigentlichen Sinne mehr dar. Vielmehr handelt es sich dabei um geeignetes Anpassen und Füllen der Datenstrukturen sowie um die Anwendung von Kriterien, welche Art von Verschmelzung zulässig bzw. erwünscht ist. Diesen Gesichtspunkten ist der nächste Abschnitt gewidmet.

Wir wollen nun zunächst ein Verfahren zur Berechnung einer Transversalen der Doppelnebenklassen aus 5.1.1 vorstellen. Dabei bedienen wir uns der Darstellung von Permutationen als Matrizen. Es sei dabei

$$M_n := \left\{ A \in \{0, 1\}^{n \times n} \mid \sum_{\mu=1}^n a_{\mu,j} = \sum_{\nu=1}^n a_{i,\nu} = 1 \quad \forall i, j \in \underline{n} \right\}$$

* Diese Rechnung ist natürlich nur notwendig, wenn $|T_1| > 1$, d.h. der gemeinsame Teilgraph aus mehr als nur einer isolierten Ecke besteht.

die Menge aller *Permutationsmatrizen*. Es existiert eine Bijektion

$$\phi : S_n \xrightarrow{\sim} M_n, \pi \mapsto A \text{ mit } a_{i,j} = \delta_{i\pi(j)},$$

wobei δ die Kroneckersche Delta-Funktion mit

$$\delta_{i,j} = \begin{cases} 1 & \text{falls } i = j \\ 0 & \text{sonst} \end{cases}$$

sei. Wir können uns daher auf die Berechnung geeigneter Permutationsmatrizen zurückziehen.

Es ist offensichtlich, daß die Multiplikation zweier Permutationen von links, etwa $\sigma \cdot \pi$ einer eben solchen Matrixmultiplikation $\phi(\sigma)\phi(\pi)$ und auch einer Vertauschung der Zeilen von $\phi(\pi)$ entspricht. Analog kommt eine Multiplikation $\pi \cdot \sigma$ von rechts einer Spaltenvertauschung von $\phi(\pi)$ gleich.

Das bringt uns zu folgender Umformung:

5.1.2 Lemma. *Die Berechnung eines vollständigen Repräsentantensystems der Doppelnebenklassen*

$$\sigma_0(\sigma_0^{-1}A\sigma_0) \setminus \text{Aut}(\eta_2) / B$$

aus Satz 5.1.1 ist äquivalent zur Bestimmung einer Transversalen der Operation

$$\phi(\text{Aut}(\eta_2)) // \sigma_0(\sigma_0^{-1}A\sigma_0) \times B$$

Wir haben also unser Konstruktionsproblem in ähnlicher Weise auf die Bestimmung von Repräsentanten einer Menge restringierter Abbildungen zurückgeführt wie in 3.1.6. In der Tat sind beide Probleme so eng verwandt, daß wir die in 3.3 vorgestellten Methoden auch für unsere hiesige Doppelnebenklassenberechnung verwenden können. Eine Gegenüberstellung soll dies verdeutlichen:

	Graphenverbindungen Problem 3.1.5	Teilgraphenzuordnungen Problem 5.1.2
Abb.	$\hat{\epsilon} : \underline{f}_1 \times \underline{f}_2 \rightarrow \{0, 1\}$	$A : \underline{n} \times \underline{n} \rightarrow \{0, 1\}$
Nebenbed.	$\sum_{i=1}^{f_1} \hat{\epsilon}(i, j) \leq 1 \forall j \in \underline{f}_2$ $\sum_{j=1}^{f_2} \hat{\epsilon}(i, j) \leq 1 \forall i \in \underline{f}_1$	$\sum_{\mu=1}^n a_{\mu,j} = \sum_{\nu=1}^n a_{i,\nu} = 1 \forall i, j \in \underline{n}$ $A \in \phi(\text{Aut}(\eta_2))$
Gruppen	$N_{\text{Aut}(\hat{\gamma}_1)}(\hat{\eta}_1) / C_{\text{Aut}(\hat{\gamma}_1)}(\hat{\eta}_1)$ $N_{\text{Aut}(\hat{\gamma}_2)}(\hat{\eta}_2) / C_{\text{Aut}(\hat{\gamma}_2)}(\hat{\eta}_2)$	$(N_{\text{Aut}(\gamma_1)}(\eta_1) / C_{\text{Aut}(\gamma_1)}(\eta_1))\sigma_0$ $N_{\text{Aut}(\gamma_2)}(\eta_2) / C_{\text{Aut}(\gamma_2)}(\eta_2)$

Der wesentliche Unterschied liegt somit in den Nebenbedingungen. Während in 3.1.5 auch eine unterschiedliche Zahl von Paaren, die auf 1 abgebildet werden, zulässig ist, haben wir es bei 5.1.2 mit einer festen Zahl von Einsen zu tun – und mit einer Beschränkung auf Elemente aus $\phi(\text{Aut}(\eta_2))$.

Wir benötigen also eine Funktion analog zu 3.3.1, die überprüft, ob ein Paar ω zulässig ist. Sei dazu A der aktuelle Kandidat. Es gibt dabei jetzt vier mögliche Rückgabewerte:

- 0: Die Position ist nicht erlaubt.
- 1: Die Position ist erlaubt, der Kandidat ist aber auch mit dieser Eins noch unvollständig.
- 2: Die Position ist erlaubt, der Kandidat erfüllt alle Nebenbedingungen.
- 3: Die Position ist nicht erlaubt, der Kandidat kann auch nicht mehr zu einer zulässigen Abbildung erweitert werden.

Diesen Aspekten trägt folgende Funktion Rechnung. Wir betrachten dabei und im folgenden wieder die lexikographische Anordnung der Paare.

5.1.3 Algorithmus (Test, ob $a_{ij} = 1$ sein darf).

- i) Existiert ein $\mu < i$ mit $\sum_{\nu=1}^n a_{\mu\nu} = 0$, so gib 3 zurück. (Eine vor i liegende Zeile konnte nicht besetzt werden.)
- ii) Ist $\sum_{\nu=1}^n a_{i\nu} = 1$, so gib 0 zurück. (Die aktuelle Zeile i ist bereits gefüllt.)
- iii) Ist $i < n$ und kann A bei $a_{ij} = 1$ so erweitert werden, daß $\phi^{-1}(A) \in \text{Aut}(\eta_2)$ ist, so gib 1 zurück.
- iv) Ist $i = n$ und (i, j) die letztmögliche Position für $a_{ij} = 1$, so daß $\phi^{-1}(A) \in \text{Aut}(\eta_2)$ ist, so gib 2 zurück.
- v) Gib 0 zurück.

▽

In Analogie zu 3.3.2 werden folgende Bezeichnungen verwendet:

$$\alpha := \sum_{\mu=1}^n \sum_{\nu=1}^n a_{\mu\nu}, \text{ die Summe aller Einträge im aktuellen Kandidaten } A$$

l Index für die Paare, denen 1 zugeordnet wird

$\omega_l \in \underline{n} \times \underline{n}$, das l -te Paar, das auf 1 abgebildet wird

Damit können wir den Algorithmus zum Ordnungstrennen Erzeugen der Doppelnebenklassenrepräsentanten formulieren:

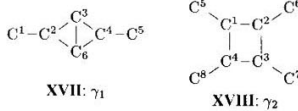
5.1.4 Algorithmus (Ordnungstreu Erzeugung aller Zuordnungen).

- i) Initialisiere den aktuellen Kandidaten mit $a_{ij} = 0$ für alle $i, j \in \underline{n}$.
- ii) Setze $l = \alpha + 1$. Setze ω_l auf das kleinste Paar (i, j) , falls $\alpha = 0$, bzw. auf das größte Paar (i, j) , für das $a_{ij} = 1$.
- iii) Solange ω_l nicht das letzte Paar aus $\underline{n} \times \underline{n}$ überschritten hat, führe durch:
 - (a) Setze ω_l auf das nächstgrößere Paar.
 - (b) Überprüfe mit 5.1.3, ob ω_l zulässig ist. Ist das Ergebnis 0, so gehe zu Schritt iii. Ist das Ergebnis 3, so gehe zu Schritt iv.
 - (c) Setze $a_{\omega_l} = 1$.
 - (d) Teste den Kandidaten A , ob er kanonisch ist. Ist dies nicht der Fall, so setze $a_{\omega_l} = 0$ und gehe zu Schritt iii.
 - (e) War 2 das Ergebnis von 5.1.3, so ist eine neue Lösung gefunden; gib diese aus oder speichere sie.
 - (f) Gehe zu Schritt ii.
- iv) Ist $\alpha = 0$, so konnte kein Paar auf 1 gesetzt werden. Beende daher die Funktion.
- v) Setze das zuletzt auf 1 gesetzte Paar auf 0 zurück. Setze $l = \alpha + 1$ und gehe zu Schritt iii.

▽

Auf eine Beschreibung des Maximalitätstests verzichten wir hier, da dieser identisch mit 3.3.8 abläuft.

5.1.5 Beispiel. Da die bislang betrachteten Strukturen nur annähernd triviale Beispiele liefern, wollen wir hier folgende molekularen Graphen untersuchen (vgl. auch 2.4.8):



Die Automorphismengruppen dieser Graphen lauten:

$$\begin{aligned} \text{Aut}(\gamma_1) &= \{1, (15)(24), (36), (15)(24)(36)\} \\ \text{Aut}(\gamma_2) &= \{1, (12)(34)(56)(78), (1234)(5678), (13)(24)(57)(68), \\ &\quad (13)(57), (14)(23)(58)(67), (1432)(5876), (24)(68)\} \end{aligned}$$

Dabei ist $\{(1, 1), (2, 5), (4, 3), (5, 7)\}$, einer der mit 4.2.3 erzeugten gemeinsamen Teilgraphen, was der Zuordnung $(1, 2, 4, 5) \rightarrow (1, 5, 3, 7)$ entspricht. Es ergeben sich für die Teilstruktur



folgende (auf **4** umnummerierte) operierende Gruppen:

$$\begin{aligned} \text{Aut}(\eta_2) &= \{1, (12), (12)(34), (13)(24), \\ &\quad (1324), (14)(23), (1423), (34)\} \\ (N_{\text{Aut}(\gamma_1)}(\eta_1)/C_{\text{Aut}(\gamma_1)}(\eta_1))\sigma_0 &= \{1, (14)(23)\} \\ N_{\text{Aut}(\gamma_2)}(\eta_2)/C_{\text{Aut}(\gamma_2)}(\eta_2) &= \{1, (13)(24)\} \end{aligned}$$

Aus diesen Daten berechnet Algorithmus 5.1.4 drei Doppelnebenklassen mit folgenden Matrixrepräsentanten (darunter die daraus jeweils resultierenden Zuordnungen):

$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
1 2 4 5	1 2 4 5	1 2 4 5
↓ ↓ ↓ ↓	↓ ↓ ↓ ↓	↓ ↓ ↓ ↓
1 5 3 7	1 5 7 3	5 1 3 7

◇

5.1.6 Bemerkung. Auch bei diesem Verfahren ist eine Aufspaltung des Erzeugungsproblems mit Hilfe des Homomorphieprinzips 1.3.1 möglich. Man muß allerdings beachten, daß es sich hierbei nur um die zweite Stufe einer Berechnung handelt, dessen erste – die Teilgraphensuche – bereits mit einem beträchtlichen Aufwand verbunden ist. Daher sind die zu lösenden Doppelnebenklassenprobleme meist von einer Größe, die eine weitergehende mathematische Methodik aufgrund des damit verbundenen Overheads nicht rechtfertigt; somit wurde in der Implementation in NEXTOR auch darauf verzichtet.

Das in 5.1.4 beschriebene Verfahren ist nicht nur für unser Zuordnungsproblem zu gebrauchen. Es handelt sich dabei vielmehr um eine universelle Methode zur Berechnung von allgemeinen Doppelnebenklassen. Bisherige (s. etwa [23, 51, 127]) setzen oft voraus, daß eine der beiden Untergruppen eine Young-Gruppe war; diese Einschränkung besteht hier nicht. Bei einer so allgemeinen Verwendung sollte eine Aufspaltung über das Homomorphieprinzip aber wieder in Betracht gezogen werden.

5.2 Der Verschmelzungsalgorithmus

Durch die bisher beschriebenen Verfahren sind wir bereits fast in der Lage, Verschmelzungsstrukturen gemäß der Strategie 4.1.2 zu konstruieren. In diesem Abschnitt wird nun beschrieben, wie die eigentlichen Verschmelzungen berechnet werden. Dazu setzen wir voraus, daß für jede Ecke in den Ausgangsgraphen eine Valenz im Sinne von 2.4.5 festgelegt ist. Dies bedeutet aber noch nicht, daß wir generell eine Beschriftung wie etwa bei molekularen Graphen fordern.

Seien wieder $\gamma_1 \in m^{\mathbb{N}^{[2]}}$ und $\gamma_2 \in m^{\mathbb{N}^{[2]}}$ Graphen und $(\eta_1, \eta_2) \in m^{\mathbb{T}_1^{[2]}} \times m^{\mathbb{T}_2^{[2]}}$ ein gemeinsamer k -Teilgraph. Ferner wollen wir mit $d_i^{(t)}$ und $val_i^{(t)}$ den Grad bzw. die Valenz der i -ten Ecke des Graphen γ_t mit $t = 1, 2$ bezeichnen.

Es ist offensichtlich, daß nicht jeder gemeinsame Teilgraph bzw. nicht jede sich daraus ergebende Zuordnung auch zu einem Verschmelzungsgraphen führen kann. Es müssen insbesondere die Beschränkungen berücksichtigt werden, die die Valenzen der einzelnen Ecken auferlegen.

Wir unterscheiden dabei, ob der gemeinsame Teilgraph aus einer oder aus mehreren Ecken besteht. Im ersten Fall ist es einfach, eine entsprechende Regel zu formulieren:

5.2.1 Kriterium. γ_1 und γ_2 können nur dann über den gemeinsamen 1-Teilgraphen (η_1, η_2) mit $\mathbb{T}_1 = \{i\}$ und $\mathbb{T}_2 = \{j\}$ verschmolzen werden, wenn

$$val_i^{(1)} \geq d_i^{(1)} + d_j^{(2)} \quad \wedge \quad val_j^{(2)} \geq d_i^{(1)} + d_j^{(2)}.$$

Sind γ_1 und γ_2 mit β_1 und β_2 beschriftet, so muß zusätzlich $\beta_1(i) = \beta_2(j)$ gelten.

Beweis: Klar. □

Besteht der gemeinsame Teilgraph aus mehreren Ecken, so können nicht einfach nur die Valenzen und Grade betrachtet werden. Es müssen vielmehr die Kanten innerhalb des gemeinsamen Graphen aus der Summation der Grade herausgehalten, sprich davon subtrahiert werden.

5.2.2 Kriterium. γ_1 und γ_2 können nur dann über den gemeinsamen k -Teilgraphen (η_1, η_2) mit $k > 1$ sowie $\mathbb{T}_1 = \{i_1, \dots, i_k\}$ und $\mathbb{T}_2 = \{j_1, \dots, j_k\}$ unter der Zuordnung $\sigma(i_\kappa) = j_\kappa$ verschmolzen werden, wenn gilt:

$$val_{i_\kappa}^{(1)} \geq d_{i_\kappa}^{(1)} + d_{j_\kappa}^{(2)} - \bar{d}_\kappa \quad \wedge \quad val_{j_\kappa}^{(2)} \geq d_{i_\kappa}^{(1)} + d_{j_\kappa}^{(2)} - \bar{d}_\kappa \quad \forall \kappa \in \underline{k}.$$

Dabei sei

$$\bar{d}_\kappa := \sum_{\substack{\nu=1, \dots, k \\ \gamma_1(i_\kappa, i_\nu) \neq 0 \wedge \gamma_2(j_\kappa, j_\nu) \neq 0}} \eta_1(i_\kappa, i_\nu).$$

Sind γ_1 und γ_2 mit β_1 und β_2 beschriftet, so muß zusätzlich für alle $\kappa \in \underline{k}$ gelten: $\beta_1(i_\kappa) = \beta_2(j_\kappa)$.

Beweis: Klar.

Es ist noch zu bemerken, daß die Definition von \bar{d}_κ aufgrund der Isomorphie von η_1 und η_2 unabhängig davon ist, über welches η summiert wird. □

Falls nicht Voll- sondern Substrukturen verschmolzen werden sollen, die über freie Valenzen verfügen (s. 4.1.11.ii), dann müssen die Wasserstoffatome zu den Graden hinzugezählt werden, da diese dann nicht zur freien Disposition stehen.

Wir haben bei unserer bisherigen Strategie hauptsächlich Wert auf die Ecken gelegt; daher kann es vorkommen, daß sich Zuordnungen ergeben, bei denen Bindungen im ersten Ausgangsgraphen vorhanden sind, im zweiten aber nicht (oder umgekehrt). Wir müssen folglich prüfen, ob gilt:

$$(5.2.3) \quad \gamma_1(i_\kappa, i_\nu) = \gamma_2(j_\kappa, j_\nu) \quad \forall \kappa, \nu \in \underline{k}, \kappa \neq \nu$$

Ein weiterer Aspekt in diesem Zusammenhang ist die Anzahl der Bindungen, die bei einer Verschmelzung u.U. hinzugefügt werden müssen. Um dem Anwender die Möglichkeit zu geben, eine obere Schranke für diesen Wert vorzuschreiben, bestimmen wir ihn auf folgende Weise:

$$(5.2.4) \quad b(\eta_1, \eta_2) = \sum_{k=1}^k \sum_{\substack{v=1, \dots, k+1 \\ \gamma_1(i_k, v) = 0 \\ \gamma_2(i_k, v) > 0}} 1$$

Wir wollen nämlich die Verschmelzung als das Überlagern des ersten durch den zweiten Ausgangsgraphen auffassen. Eine „neue“ Bindung verstehen wir dann in dem Sinne, daß eine solche im ersten, also dem zu überlagernden, noch nicht existiert; sie muß zu diesem hinzugefügt werden.

Für die Formulierung des Algorithmus verwenden wir die Bezeichnungen

- s : $\underline{q} \rightarrow \{p+1, \dots, p+q-k\}$, die Umnummerierung der Ecken von γ_2 , durch die dieser an γ_1 angehängt wird. Dabei können die Ecken aus T_2 auf einen beliebigen Wert abgebildet werden.
- σ : $T_1 \rightarrow T_2$, die aktuelle Zuordnung zwischen den Eckenmengen.
- k die Länge des aktuellen Teilgraphen: $k = |T_1| = |T_2|$.
- $\eta_2^{(\sigma)}$ der Teilgraph, der aus der Eckenpermutation entsteht, die durch die Zuordnung σ im Vergleich zur Referenzzuordnung σ_0 induziert wird.

Damit können wir den Verschmelzungsalgorithmus als Ganzes formulieren.

5.2.5 Algorithmus (Verschmelzung zweier Graphen). Gegeben seien zwei Graphen $\gamma_1 \in m^{\underline{p}^{[2]}}$ und $\gamma_2 \in m^{\underline{q}^{[2]}}$.

- i) Berechne mit 4.2.3 alle gemeinsamen Teilgraphen, u.U. unter Berücksichtigung gegebener Restriktionen.
- ii) Für jeden gemeinsamen Teilgraphen (η_1, η_2) berechne mit 5.1.4 alle Zuordnungen σ und führe jeweils durch:
 - (a) Überprüfe die Kriterien 5.2.1 bzw. 5.2.2 sowie 5.2.3 für $\eta_1, \eta_2^{(\sigma)}$. Ist eines davon nicht erfüllt, fahre mit dem nächsten Element in Schritt ii fort.
 - (b) Überprüfe, falls vorgegeben, die Einhaltung der Schranke für $b(\eta_1, \eta_2^{(\sigma)})$ (vgl. 5.2.4). Ist dies nicht erfüllt, fahre mit dem nächsten Element in Schritt ii fort.
 - (c) Bestimme den resultierenden Gesamtgraphen $\gamma \in m^{\underline{p+q-k}^{[2]}}$; Initialisiere ihn zunächst mit $\gamma(i, j) = 0 \forall i, j \in \underline{p+q-k}$ und setze dann
 - $\gamma(i, j) = \gamma_1(i, j)$ für $i, j \in \underline{p}$.
 - $\gamma(s(i), s(j)) = \gamma_2(i, j)$ für $i, j \in \underline{q} \setminus T_2$.
 - $\gamma(\sigma^{-1}(i), s(j)) = \gamma_2(i, j)$ für $i \in T_2, j \in \underline{q} \setminus T_2$.
 - $\gamma(\sigma^{-1}(i), \sigma^{-1}(j)) = \gamma_2(i, j)$ für $i, j \in T_2$, falls in γ_1 nicht vorhanden, d.h. $\gamma_1(\sigma^{-1}(i), \sigma^{-1}(j)) = 0$ (Hinzufügen einer Bindung).

▽

5.2.6 Beispiel. Wir greifen Beispiel 5.1.5 wieder auf und betrachten die Zuordnungen:

- i) $(1, 2, 4, 5) \mapsto (1, 5, 3, 7)$: Hierbei ist Kriterium 5.2.2 verletzt, denn für die Ecke 4 im ersten Graphen gilt:

$$val_4 = 4, \text{ aber } d_4^{(1)} = 3, d_4^{(2)} = 3, \overline{d}_3 = 1, \text{ also } d_4^{(1)} + d_4^{(2)} - \overline{d}_3 = 5 > 4$$

- ii) $(1, 2, 4, 5) \mapsto (1, 5, 7, 3)$: Diese Zuordnung erfüllt alle Kriterien. Wir können daher den Verschmelzungsgraphen γ bestimmen; zum Vergleich zeigen wir die Adjazenzmatrizen von γ_1, γ_2 und γ :

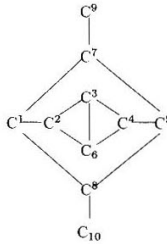
$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

γ_1 γ_2

$$\left(\begin{array}{cccccc|cccc} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array} \right)$$

γ

Hierbei ist das Gerüst von γ_1 noch deutlich zu erkennen. Der zugehörige Graph hat die Form:



iii) $(1, 2, 4, 5) \mapsto (5, 1, 3, 7)$: Auch hierbei ist Kriterium 5.2.2 verletzt, denn für die Ecke 2 im ersten Graphen gilt:

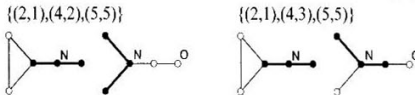
$$val_2 = 4, \text{ aber } d_2^{(1)} = 3, d_1^{(2)} = 3, \bar{d}_2 = 1, \text{ also } d_2^{(1)} + d_1^{(2)} - \bar{d}_2 = 5 > 4$$

Insgesamt gibt es zu diesen beiden Strukturen 37 Verschmelzungsgraphen, diese in Abbildung 5.1 gezeigt werden. Wie bei 3.4.10 wurde die Platzierung der Graphen automatisch mit dem entsprechenden Modul aus **MOLGEN** erstellt (s. etwa [14, 112]).

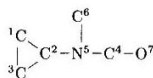
◇

Wir haben sowohl bei der Teilgraphenkonstruktion in 4.2.3 als auch bei der Berechnung der Zuordnungen in 5.1.4 stets darauf geachtet, nur nichtisomorphe Strukturen zu erzeugen. Trotzdem führt Algorithmus 5.2.5 zuweilen zu doppelten Graphen. Es liegt nämlich eine ähnliche Situation wie in Abschnitt 3.4 vor. Auch hier haben wir uns auf kleinere Gruppen beschränkt, obwohl eine größere für das Isomorphieproblem relevant wäre. Wieder erhalten wir nur *semikanonische* Lösungen, deren Kanonizität einer weiteren Überprüfung bedarf.

5.2.7 Beispiel. Wir kommen auf die beiden Strukturen **VII** und **VIII** aus 4.1.10.i zurück, deren gemeinsame Teilgraphen wir im letzten Kapitel berechnet haben (s. Abb. 4.2, S. 69). Zwei dieser Teilgraphen waren:



Beide führen bei der Verschmelzung zu



◇

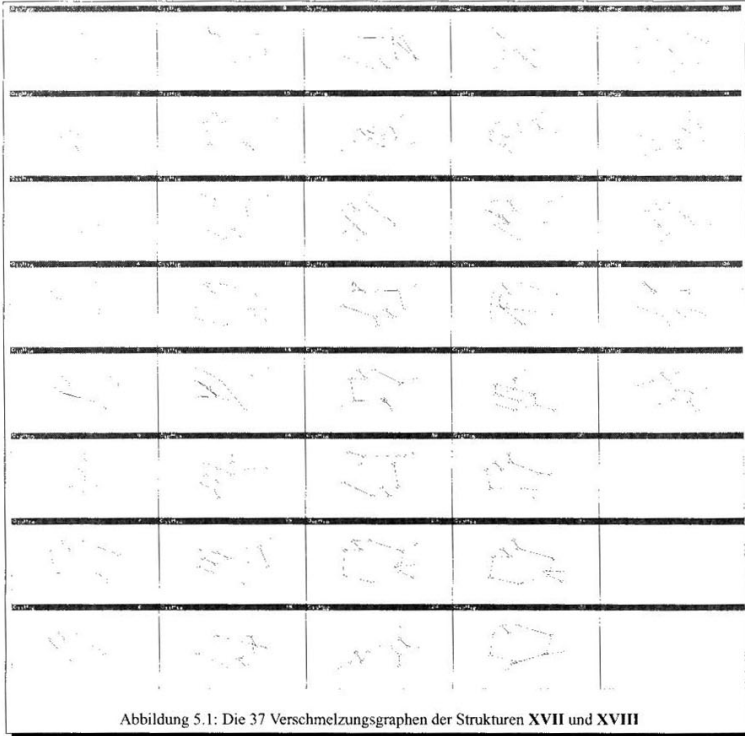


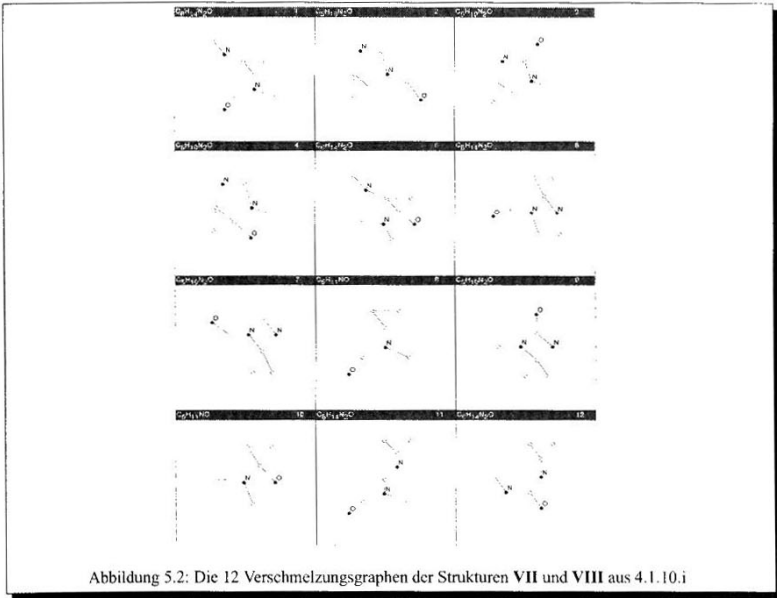
Abbildung 5.1: Die 37 Verschmelzungsgraphen der Strukturen XVII und XVIII

Wir haben wieder zwei Möglichkeiten, diesen Mangel zu beheben: einen Kanonizitätstest oder eine Hash-Tabelle. Da die Verhältnisse hier aufgrund der Kombination von Teilgraphensuche und Doppelnebenklassenberechnung deutlich komplizierter sind als bei 3.4.4, wollen wir auf den Einsatz (und die Beschreibung) eines Kanonizitätstests verzichten. Stattdessen verwenden wir wieder eine Hash-Funktion wie 3.4.7 und führen die Erzeugung kanonischer Verschmelzungen analog zu 3.4.9 durch.

5.2.8 Beispiel. Es zeigt sich, daß der in 5.2.7 beschriebene Fall der einzige bei den Strukturen VII und VIII ist. In Abb. 5.2 sind alle Lösungen dargestellt.

◇

Auch bei anderen Beispielen sind doppelte Lösungen nicht allzu häufig.



5.3 Experimentelle Resultate

In diesem Abschnitt wollen wir die Ergebnisse einiger Computer-Berechnungen vorstellen, die mit dem Verschmelzungsverfahren 5.2.5 durchgeführt wurden.

Wie in den Abschnitten 3.5 und 4.4.2 haben wir auch hier zur Sicherstellung eines möglichst breiten Spektrums von Graphen alle 5.338 Isomere der Formel C_iH_j mit $1 \leq i \leq 7$ und $2 \leq j \leq 16$ herangezogen. Diese wurden wieder mit Hilfe von **MOLGEN** erzeugt und anschließend mit mehreren Teststrukturen verschmolzen.

Als Vergleichsgraphen wurden diesmal folgende Strukturen untersucht:



Wir zeigen in den Tabellen 5.1 und 5.2 wieder die arithmetischen Mittelwerte; die Tabellenspalten bedeuten:

- \bar{c} Arithmetisches Mittel der Anzahlen von gemeinsamen Teilgraphen
- \bar{n} Arithmetisches Mittel der Anzahlen von Verschmelzungsgraphen
- n_{\max} Höchste Anzahl von Verschmelzungsgraphen beim Verschmelzen von der Vergleichsstruktur mit allen Isomeren
- n_{\min} Kleinste Anzahl von Verschmelzungsgraphen
- σ Standardabweichung von n
- \bar{t} Mittlere CPU-Zeit für die Erzeugung aller Verschmelzungsgraphen in Sekunden
- $\frac{\bar{t}}{n}$ Mittlere CPU-Zeit pro Repräsentant in Sekunden

Die Resultate wurden auf einer DEC 3000/600 AXP Workstation mit 333 MHz ermittelt.

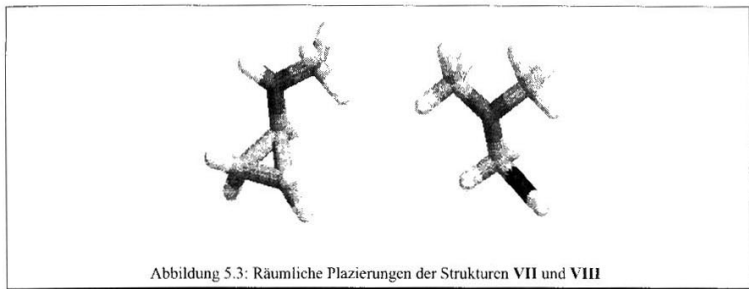
Ein Vergleich mit anderen Ansätzen ist aufgrund der Neuartigkeit der hier vorgestellten Problemlösung leider nicht möglich. Den Aufwand für die beiden Teile unseres Verfahrens, die Teilgraphensuche und die Ordnungstreu- Erzeugung der Zuordnungen, haben wir bereits in den Abschnitten 4.4.1 und 3.5 diskutiert.

Formel	Isomere	\bar{v}	\bar{w}	n_{\max}	n_{\min}	σ	\bar{t}	$\frac{\bar{t}}{n}$
C ₁ H ₄	1	0,00	0,00	0	0	0,00	0,00	0,00000
C ₂ H ₂	1	0,00	0,00	0	0	0,00	0,00	0,00000
C ₂ H ₄	1	0,00	0,00	0	0	0,00	0,00	0,00000
C ₂ H ₆	1	3,00	2,00	2	2	0,00	0,00	0,00000
C ₃ H ₂	2	7,00	3,00	3	3	0,00	0,00	0,00167
C ₃ H ₄	3	8,67	5,33	6	4	0,94	0,01	0,00111
C ₃ H ₆	2	8,50	7,00	10	4	3,00	0,00	0,00050
C ₃ H ₈	1	9,00	8,00	8	8	0,00	0,01	0,00125
C ₄ H ₂	7	10,43	2,57	3	2	0,49	0,01	0,00500
C ₄ H ₄	11	12,45	6,55	13	3	2,71	0,02	0,00226
C ₄ H ₆	9	15,11	11,44	18	6	3,20	0,02	0,00182
C ₄ H ₈	5	14,40	13,60	23	7	5,50	0,02	0,00193
C ₄ H ₁₀	2	12,50	11,00	14	8	3,00	0,02	0,00232
C ₅ H ₂	21	19,67	4,29	8	2	1,69	0,03	0,00753
C ₅ H ₄	40	23,32	9,65	19	4	3,47	0,04	0,00416
C ₅ H ₆	40	25,73	15,93	26	6	4,97	0,05	0,00303
C ₅ H ₈	26	26,73	21,12	33	8	6,75	0,05	0,00256
C ₅ H ₁₀	10	26,80	25,00	39	8	8,97	0,05	0,00226
C ₅ H ₁₂	3	21,33	19,67	28	6	9,74	0,05	0,00274
C ₆ H ₂	85	31,68	4,68	8	2	1,83	0,05	0,01227
C ₆ H ₄	185	37,08	11,37	22	4	4,09	0,07	0,00631
C ₆ H ₆	217	40,25	19,05	34	4	6,38	0,08	0,00432
C ₆ H ₈	159	41,64	26,50	48	6	8,48	0,09	0,00344
C ₆ H ₁₀	77	41,82	32,55	53	10	10,36	0,09	0,00301
C ₆ H ₁₂	25	38,84	34,68	60	9	13,91	0,09	0,00280
C ₆ H ₁₄	5	32,80	30,00	43	16	9,98	0,09	0,00303
C ₇ H ₂	356	52,23	5,47	8	2	1,94	0,10	0,01982
C ₇ H ₄	920	58,67	13,35	23	4	4,27	0,12	0,00945
C ₇ H ₆	1230	62,59	22,67	41	6	6,62	0,13	0,00621
C ₇ H ₈	1031	63,70	32,05	57	7	9,18	0,15	0,00474
C ₇ H ₁₀	575	63,80	40,80	69	12	11,73	0,16	0,00395
C ₇ H ₁₂	222	61,99	47,14	75	9	14,51	0,16	0,00356
C ₇ H ₁₄	56	58,91	50,68	84	11	17,26	0,17	0,00344
C ₇ H ₁₆	9	48,33	43,00	79	22	18,80	0,15	0,00365

Tabelle 5.1: Rechenergebnisse bei der Erzeugung von Verschmelzungsgraphen von Butan (VI) und den Isomeren zu C_iH_j mit $1 \leq i \leq 7$ und $2 \leq j \leq 16$

Formel	Isomere	\bar{c}	\bar{n}	n_{\max}	n_{\min}	σ	\bar{z}	$\frac{I}{n}$
C ₁ H ₄	1	0,00	0,00	0	0	0,00	0,00	0,00000
C ₂ H ₂	1	0,00	0,00	0	0	0,00	0,00	0,00000
C ₂ H ₄	1	0,00	0,00	0	0	0,00	0,00	0,00000
C ₂ H ₆	1	3,00	1,00	1	1	0,00	0,00	0,00000
C ₃ H ₂	2	6,00	1,00	1	1	0,00	0,00	0,00500
C ₃ H ₄	3	7,00	2,67	4	2	0,94	0,01	0,00333
C ₃ H ₆	2	6,50	3,00	5	1	2,00	0,01	0,00600
C ₃ H ₈	1	7,00	4,00	4	4	0,00	0,01	0,00250
C ₄ H ₂	7	8,00	1,29	2	1	0,45	0,01	0,00929
C ₄ H ₄	11	9,64	2,91	5	1	1,16	0,02	0,00582
C ₄ H ₆	9	10,89	4,56	7	3	1,42	0,02	0,00400
C ₄ H ₈	5	10,60	5,60	9	3	1,96	0,02	0,00404
C ₄ H ₁₀	2	9,50	5,50	6	5	0,50	0,02	0,00467
C ₅ H ₂	21	13,95	1,71	3	1	0,76	0,03	0,02087
C ₅ H ₄	40	16,27	3,92	7	1	1,40	0,04	0,01144
C ₅ H ₆	40	17,77	6,35	10	3	2,13	0,04	0,00764
C ₅ H ₈	26	18,38	8,65	14	2	2,70	0,05	0,00624
C ₅ H ₁₀	10	18,50	10,50	15	3	3,83	0,06	0,00564
C ₅ H ₁₂	3	16,33	10,00	14	5	3,74	0,06	0,00718
C ₆ H ₂	85	20,98	1,86	3	1	0,78	0,06	0,03527
C ₆ H ₄	185	24,36	4,62	9	1	1,74	0,07	0,01701
C ₆ H ₆	217	26,22	7,83	15	2	2,74	0,08	0,01122
C ₆ H ₈	159	27,36	11,21	20	2	3,50	0,09	0,00857
C ₆ H ₁₀	77	27,83	14,26	22	4	4,23	0,10	0,00731
C ₆ H ₁₂	25	26,64	16,08	25	5	5,43	0,10	0,00663
C ₆ H ₁₄	5	24,00	15,60	22	10	4,08	0,11	0,00689
C ₇ H ₂	356	33,52	2,19	4	1	0,86	0,12	0,06022
C ₇ H ₄	920	37,52	5,58	12	1	2,10	0,14	0,02708
C ₇ H ₆	1230	40,04	9,81	21	3	3,41	0,15	0,01697
C ₇ H ₈	1031	41,45	14,49	29	2	4,64	0,17	0,01230
C ₇ H ₁₀	575	42,37	19,30	34	5	5,80	0,18	0,00983
C ₇ H ₁₂	222	42,32	23,47	38	6	6,82	0,19	0,00851
C ₇ H ₁₄	56	41,70	26,68	40	5	8,02	0,20	0,00803
C ₇ H ₁₆	9	36,78	25,22	42	16	8,90	0,20	0,00826

Tabelle 5.2: Rechenergebnisse bei der Erzeugung von Verschmelzungsgraphen von Dimethylpropan (XV) und den Isomeren zu C_iH_j mit 1 ≤ i ≤ 7 und 0 ≤ j ≤ 16



5.4 Dreidimensionale Verschmelzungen

Bisher haben wir Graphen stets nur in ihrer topologischen Beschreibung betrachtet, d.h. nur die Bindungsverhältnisse. Gerade in der Chemie ist es aber von Bedeutung, die tatsächliche Gestalt eines Moleküls im Raum zu betrachten. Begrifflich macht man dabei folgende Unterscheidungen (nach [82]):

5.4.1 Definition.

- Die Konstitution gibt bei gegebener Summenformel die Art der Bindungen und die gegenseitige Verknüpfung der Atome eines Moleküls an.
- Die Konfiguration gibt die räumliche Anordnung der Atome wieder. Moleküle, die durch intramolekulare Rotationen ineinander übergehen, werden nicht unterschieden.
- Die Konformation stellt die räumliche Anordnung aller Atome eines Moleküls gegebener Konfiguration dar, die durch Rotationen um Einfachbindungen erzeugt werden und nicht miteinander zur Deckung gebracht werden können.

Es gibt mehrere Methoden, um eine dreidimensionale Platzierung eines Moleküls zu erhalten. Neben den experimentellen basieren die wichtigsten mathematischen auf Distanzgeometrie (nach [34]), auf Konformationsanalyse (etwa [62]), auf empirischen Regeln (z.B. [102, 109]) oder auf Energiebetrachtungen. Der letzte Weg wurde auch hier gewählt. Das Programmsystem **MOLGEN**, mit dem die dreidimensionalen Platzierungen berechnet wurden, verwendet ein vereinfachtes *MM2*-Energiemodell nach [1]. Es gehen jeweils die Abweichungen von Tabellenwerten in den Bereichen Bindungsabstand, Bindungswinkel, Torsionswinkel und van-der-Waalskräfte ein. Zunächst werden die Atome zufällig platziert. Mit Hilfe des Verfahrens der konjugierten Gradienten (nach [38]) wird dann ein (i.a. lokales) Minimum der Energiefunktion ermittelt. Die so gewonnenen dreidimensionalen Platzierungen sind insbesondere für organische Substanzen hinreichend genau.

5.4.2 Beispiel. Abb. 5.3 zeigt räumliche Platzierungen der Strukturen VII und VIII aus Beispiel 4.1.10.i. (Wir legen hier keinen Wert auf Konfigurationsisomerie und verwenden daher ein beliebiges Isomer. Genaueres siehe [127, 128, 129]). ◇

Die dreidimensionale Gestalt der Ausgangsmoleküle läßt sich bei unserem Verschmelzungsalgorithmus 5.2.5 am besten durch ein zusätzliches Kriterium in Schritt ii-a berücksichtigen, da bei allen anderen Schritten nur topologische Informationen eingehen. Denn es muß z.B. ein dreidimensionaler gemeinsamer Teilgraph auch ein topologischer gemeinsamer Teilgraph sein.

Wir verwenden die Bezeichnungen aus Abschnitt 5.2 weiter. Damit kann ein Kriterium lauten:

5.4.3 Kriterium. Zwei molekulare Graphen (γ_1, β_1) und (γ_2, β_2) mit dreidimensionalen Koordinaten $x_i^{(1)} \in \mathbb{R}^3$ bei $i \in \mathbf{p}$ und $x_j^{(2)} \in \mathbb{R}^3$, $j \in \mathbf{q}$, können über den gemeinsamen k -Teilgraphen (η_1, η_2) mit $k > 1$ sowie $T_1 = \{i_1, \dots, i_k\}$ und $T_2 = \{j_1, \dots, j_k\}$ unter der Zuordnung $\sigma(i_\kappa) = j_\kappa$ verschmolzen werden, wenn folgende Bedingungen erfüllt sind:

i) Für alle Paare (i_κ, i_λ) mit $\eta_1(i_\kappa, i_\lambda) \neq 0$ und (j_κ, j_λ) mit $\eta_2(j_\kappa, j_\lambda) \neq 0$ gilt:

$$\|x_{i_\kappa}^{(1)} - x_{i_\lambda}^{(1)}\| \approx \|x_{j_\kappa}^{(2)} - x_{j_\lambda}^{(2)}\|$$

ii) Für alle Pfade $(i_\kappa, i_\lambda, i_\mu)$ und $(j_\kappa, j_\lambda, j_\mu)$ in η_1 bzw. η_2 (vgl. 2.1.3) gilt:

$$\angle(x_{i_\kappa}^{(1)}, x_{i_\lambda}^{(1)}, x_{i_\mu}^{(1)}) \approx \angle(x_{j_\kappa}^{(2)}, x_{j_\lambda}^{(2)}, x_{j_\mu}^{(2)})$$

iii) Für alle Pfade $(i_\kappa, i_\lambda, i_\mu, i_\nu)$ und $(j_\kappa, j_\lambda, j_\mu, j_\nu)$ in η_1 bzw. η_2 gilt:

$$\text{died}(x_{i_\kappa}^{(1)}, x_{i_\lambda}^{(1)}, x_{i_\mu}^{(1)}, x_{i_\nu}^{(1)}) \approx \text{died}(x_{j_\kappa}^{(2)}, x_{j_\lambda}^{(2)}, x_{j_\mu}^{(2)}, x_{j_\nu}^{(2)})$$

wobei died den Diederwinkel bezeichne.

Dies ist nur ein mögliches Kriterium. Insbesondere können von den Bedingungen i-iii auch eine oder mehrere außer acht gelassen werden. Zudem ist nicht erklärt worden, was unter der näherungsweise Gleichheit „ \approx “ zu verstehen ist; in der Implementation in NEXTOR etwa wird dabei eine bis zu 15%-ige Abweichung akzeptiert. Bei der Anpassung an spezielle Anwendungen können diese Aspekte genauer spezifiziert werden.

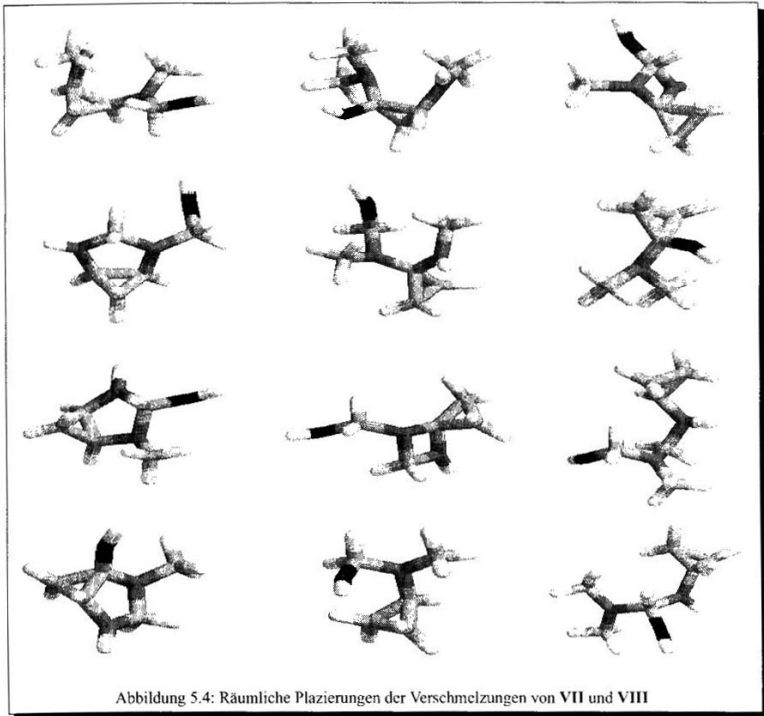
Die Topologie der Verschmelzung wurde in Algorithmus 5.2.5 eingehend erläutert. Für die 3D-Koordinaten verwenden wir ein Überlagerungsverfahren nach [60, 66]. Dabei werden Translationen und Rotationen auf die Koordinaten von γ_2 angewendet und so der Abstand

$$\sum_{\kappa=1}^k \|x_{i_\kappa}^{(1)} - x_{j_\kappa}^{(2)}\|$$

minimiert. Dies ist nur bei $k > 1$ möglich. Im Fall $k = 1$ ($T_1 = \{i\}$, $T_2 = \{j\}$) wenden wir nur eine Translation an, durch die $\|x_i^{(1)} - x_j^{(2)}\| = 0$ wird.

In jedem Fall kann die räumliche Platzierung der Verschmelzungsstruktur von realistischen Koordinaten abweichen. Daher gibt es die Möglichkeit, im Anschluß an die Verschmelzung die Energieoptimierung nochmals anzuwenden, um eine naturgetreue Anordnung zu erhalten.

5.4.4 Beispiel. Als Beispiel zeigen wir in Abb. 5.4 räumliche Anordnungen der Verschmelzungen der Strukturen VII und VIII aus Beispiel 4.1.10.i. Aufgrund des 3-Rings sind nicht alle Platzierungen, die sich bei bloßer Überlagerung ergaben, realistisch, so daß eine Nachoptimierung vorgenommen wurde. ◇



Kapitel 6 Regelbasierte Verbindung von molekularen Graphen

6.1 Allgemeine Verbindungssyntax

In den letzten Kapiteln haben wir gesehen, wie Graphen auf verschiedene Weise algorithmisch miteinander verbunden werden können. Wir haben dabei Wert auf mathematisch genaue und vollständige Verfahren gelegt. Bei molekularen Graphen gibt es aber auch den Fall, daß nicht alle möglichen Verbindungen eine Rolle spielen, sondern nur einige wenige, genau festgelegte. Solche Verbindungen ergeben sich z.B. im Verlauf einer chemischen Reaktion. In diesem Kapitel wollen wir als weiteres Verfahren zur Graphenverbindung solche Reaktionsverbindungen über Teilstrukturen untersuchen.

Zunächst wollen wir eine Syntax vorstellen, mit deren Hilfe eine Zweikomponentensynthese, etwa



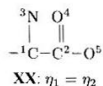
formal beschrieben werden kann. Verantwortlich für eine Reaktion ist selten das gesamte Molekül. Vielmehr sind es i.a. *Teilstrukturen*, also Teilgraphen, die den Ablauf bestimmen.

6.1.1 Definition. Seien (η_1, β_1) und (η_2, β_2) mit $\eta_1 \in m^{\mathbb{Z}^{|\beta_1|}}$ und $\eta_2 \in m^{\mathbb{Z}^{|\beta_2|}}$ molekulare Graphen. Unter einem Reaktionsschema verstehen wir ein Tripel $((\eta_1, \beta_1), (\eta_2, \beta_2), \rho)$ (bzw. (η_1, η_2, ρ) , wenn die Beschriftung klar ist), wobei $\rho: \underline{\mathbf{r}} \times \underline{\mathbf{s}} \rightarrow \mathbb{Z} \cup \{-\infty\}$ eine Abbildung sei mit

$$\rho(i, j) = \begin{cases} k & i \text{ und } j \text{ werden durch eine } k\text{-fache Kante verbunden} \\ 0 & i \text{ und } j \text{ bleiben unverbunden} \\ -\infty & \text{eines der Atome } i \text{ oder } j \text{ fällt weg} \end{cases}$$

Mit Hilfe dieser Definition* lassen sich viele Zweikomponentenreaktionen gut beschreiben. Was uns an einer solchen Reaktion interessiert, ist ja hauptsächlich die Veränderung der Graphen, weniger die experimentellen Aspekte (wie Reaktionsbedingungen, Zusatzstoffe, Katalysatoren oder Gleichgewichte). Die Matrix ρ stellt dabei nur eine formale Repräsentation der Vorgänge dar und soll nicht etwa als Operator für die Adjazenzmatrizen verstanden werden.

6.1.2 Beispiel. Peptide sind aus mindestens zwei Aminosäuren aufgebaute Eiweißmoleküle [82], die in der Biochemie eine zentrale Rollen spielen. Die Bindung zwischen den einzelnen Aminosäuren erfolgt durch Kondensation (Wasserabspaltung) der Säuregruppe (COOH) und der Amino-Gruppe (NH₂). Die entscheidende Reaktionsstruktur ist also die α -Aminosäure-Gruppe, die bei beiden Reaktionspartnern vorhanden sein muß:



*Diese Definition stellt die Situation stark vereinfacht dar und soll lediglich der Formalisierung der späteren Konstruktionsaufgabe dienen. Für weitergehende Zwecke sind umfassendere Beschreibungen wie etwa die Algebra der *be-kr*-Matrizen nach [35] nötig.

Die Kondensationsreaktion der Aminosäuren entspricht der Matrix

$$\rho = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\infty & -\infty & -\infty & -\infty & -\infty \end{pmatrix}$$

◇

Damit können wir einen Algorithmus formulieren:

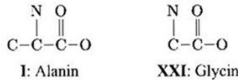
6.1.3 Algorithmus (Verbindung zweier Graphen nach einem Reaktionsschema). Gegeben seien ein Reaktionsschema $((\eta_1, \chi_1), (\eta_2, \chi_2), \rho)$ sowie zwei molekulare Graphen (γ_1, β_1) und (γ_2, β_2) mit $\gamma_1 \in m^{\mathbb{P}^{[1]}}$ bzw. $\gamma_2 \in m^{\mathbb{P}^{[2]}}$.

- i) Bestimme alle nichtisomorphen molekularen Teilgraphen der Form $\zeta_1^{(1)}, \dots, \zeta_k^{(1)} \subseteq \gamma_1$ und $\zeta_1^{(2)}, \dots, \zeta_l^{(2)} \subseteq \gamma_2$ mit Eckenmengen $T_i^{(1)} \subseteq \mathbb{P}$ bzw. $T_j^{(2)} \subseteq \mathbb{Q}$ für $i \in \underline{k}, j \in \underline{l}$, für die gilt: Für alle $i \in \underline{k}$ und $j \in \underline{l}$ existieren kantenerhaltende Isomorphismen $\sigma_i : \zeta_i^{(1)} \rightarrow \eta_1$ und $\tau_j : \zeta_j^{(2)} \rightarrow \eta_2$ mit $\chi_1(\sigma_i(x)) = \beta_1(x)$ für $x \in T_i^{(1)}, i \in \underline{k}$ und $\chi_2(\tau_j(y)) = \beta_2(y)$ für $y \in T_j^{(2)}, j \in \underline{l}$.
- ii) Bestimme für alle Paare $(\zeta_i^{(1)}, \zeta_j^{(2)})$ einen resultierenden Gesamtgraphen γ mittels ρ , indem γ_1 und γ_2 zusammengefaßt, die wegfallenden Ecken eliminiert und die in ρ festgelegten Kanten hinzugefügt werden.
- iii) Speichere jedes γ in einer Hash-Tabelle analog zu 3.4.9.

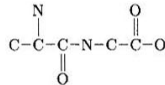
▽

Zur Teilgraphensuche können wir einen rekursiven Suchalgorithmus (s. z.B. [126]) oder ein auf iterierter Klassifizierung basierendes Verfahren (etwa aus [11]) einsetzen. Die Isomorphie läßt sich anhand der Automorphismengruppen entweder durch Anwendung aller $\pi \in \text{Aut}(\gamma_i)$ auf die bisherigen Teilgraphen, durch einen Kanonizitätstest oder im Laufe der iterierten Klassifizierung überprüfen. Auf weitere Details wollen wir hier verzichten.

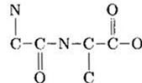
6.1.4 Beispiel. Wir betrachten die Aminosäuren Alanin (Struktur I aus 2.4.6) und Glycin:



Diese Strukturen wollen wir nach dem Schema für Peptidbrücken aus Beispiel 6.1.2 verbinden. Beide enthalten offensichtlich den Teilgraphen η_1 aus XX. Trotz der Gleichheit der Teilstrukturen ist die Reihenfolge der beiden Ausgangsgraphen (Edukte) wesentlich. Nimmt man Alanin als ersten, so entsteht mit Algorithmus 6.1.3:



Bei Glycin als γ_1 erhält man:



(In der Laborsynthese schaltet man daher Reaktionen vor, die die Säure- oder die Amid-Gruppe blockieren, um nur eine der beiden Möglichkeiten zu erhalten.)

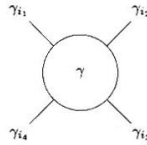
◇

6.2 Mehrfachanlagerungen an eine Kernstruktur

Einen besonderen Typ eines Reaktionsschemas haben wir vorliegen, wenn eine Kernstruktur mit mehreren Reaktionszentren und eine Reihe von Ligandenverbindungen gegeben ist.

Sei $((\eta_1, \chi_1), (\eta_2, \chi_2), \rho)$ ein Reaktionsschema, (γ, β) ein molekularer Graph, der k Teilstrukturen isomorph zu (η_1, χ_1) mit $k > 1$ enthält, und $(\gamma_1, \beta_1), \dots, (\gamma_n, \beta_n)$ eine Reihe von molekularen Graphen; von diesen wollen wir der Einfachheit halber voraussetzen, daß jeder genau eine Teilstruktur isomorph zu (η_2, χ_2) enthält. (Es ist auch der Fall mit mehreren Teilstrukturen denkbar. Allerdings verkompliziert er nur die Beschreibung, ohne neue Erkenntnisse zu liefern.)

Die erste Aufgabe ist es nun, alle Zuordnungen der Liganden auf die Plätze des Kerns zu bestimmen, wobei die Plätze durch die Teilstrukturen aus dem Schema gegeben sind. Bei $k = 4$ haben wir dann etwa:



Die Austauschbarkeit der Plätze ist durch $Aut(\gamma, \beta)$ gegeben, wobei sich die entsprechende Gruppe $P_\gamma \leq S_k$ analog zu 2.3.2 als induzierte Teilgraphenautomorphismengruppe bei Betrachtung etwa des Teilgraphen ergibt, der sich aus der jeweils ersten Ecke der ζ_i zusammensetzt (vgl. 6.1.3).

Somit operiert P_γ auf den Plätzen \underline{k} , die mit n verschiedenen Liganden belegt werden sollen. Zusammengefaßt bedeutet dies:

6.2.1 Lemma. *Die wesentlich verschiedenen Möglichkeiten der Belegung einer Kernstruktur (γ, β) mit k verschiedenen Reaktionsplätzen (gemäß einem vorgegebenen Reaktionsschema) mit n Ligandenstrukturen, die jeweils den entsprechenden Reaktionsteilgraphen genau einmal enthalten, entspricht*

$$\mathcal{T}(\underline{n}^k // P_\gamma).$$

Damit können wir ein Verfahren formulieren:

6.2.2 Algorithmus (Belegung einer Kernstruktur mit Liganden).

- i) Bestimme alle Teilgraphen $\zeta_1, \dots, \zeta_k \subseteq \gamma$, die isomorph zu η_1 sind. (Eine Isomorphie der ζ_i unter $Aut(\gamma)$ sei ausdrücklich erlaubt.) Ermittle ferner die Gruppe P_γ .
- ii) Berechne für $i \in \underline{k}$ die Teilgraphen $\zeta^{(i)} \subseteq \gamma_i$, die isomorph zu η_2 sind.
- iii) Erzeuge ordnungstreu den nächsten Repräsentanten $f \in \underline{n}^k$ unter der Operation von P_γ .
- iv) Bestimme den Gesamtgraphen, der sich durch Anbindung der Liganden $\gamma_{f(1)}, \dots, \gamma_{f(k)}$ an γ ergibt, analog zu 6.1.3.ii.
- v) Gibt es noch weitere Repräsentanten, so gehe zu Schritt iii).

▽

Durch die Ordnungstreue Erzeugung in Schritt iii) und die Eindeutigkeit der Teilgraphen in den Liganden erhalten wir nur nichtisomorphe Lösungen, so daß ein Hashing, wie wir es bei 6.1.3 eingesetzt haben, hier nicht nötig ist.

Beispiele dazu werden wir in einem größeren Anwendungszusammenhang in Kapitel 8 vorstellen.

6.2.3 Bemerkung. Das Problem, mehrere Liganden auf alle verschiedenen Weisen an ein Gerüst zu binden, wurde bereits in der anorganischen Chemie unter dem Begriff *Permutationsisomerie* untersucht [20, 21, 75, 100, 108, 127, 129]. Dort ist das Gerüst ein höhervalentes Atom, an das andere Atome als Liganden gebunden sind. Die Situation der Permutationsisomerie unterscheidet sich von der hier diskutierten dadurch, daß dort die Partitionen, d.h. die Häufigkeiten der einzelnen Ligandentypen, nicht frei sind, sondern meist beschränkt bzw. fest vorgegeben. Dadurch führt die Permutationsisomerie i.a. zu Doppelnebenklassenproblemen [108, 127, 129], während hier „nur“ Repräsentanten von Symmetrieklassen zu berechnen sind.

Ein weiterer Unterschied besteht noch darin, daß bei der Permutationsisomerie die Liganden an das Gerüst nur über Kanten verbunden werden, wogegen wir es hier mit Reaktionsschemata zu tun haben.

Teil III

Anwendungen in der Chemie

Kapitel 7 Zielgerichtete stufenweise Konstruktion im Moleküldesign

7.1 Topologische Indices

In diesem und den folgenden Kapiteln wollen wir Möglichkeiten untersuchen, wie die oben vorgestellten Konstruktionsverfahren angewendet werden können. Zunächst soll es um eine iterative Erzeugung gehen, die durch Vorgaben gesteuert wird.

Eine Art von Vorgaben werden strukturelle Ähnlichkeiten zu einem gegebenen Zielgraphen sein. Da wir diese mit Hilfe von topologischen Indizes beschreiben wollen, dient dieser Abschnitt der Vorstellung der verwendeten Werte.

Es besteht in der Chemie ein großes Interesse, Zusammenhänge zwischen der Molekülstruktur einer Substanz und ihrer physikalischen oder biologischen Wirkung zu finden. Da empirische Daten nicht für alle denkbaren Verbindungen verfügbar sind und Experimente oder quantenchemische Rechnungen, insbesondere bei einer größeren Anzahl von Strukturen, mit einem beträchtlichen Aufwand verbunden sind, bieten sich sog. *topologische Indizes* (oder *Grapheninvarianten*) [5, 18, 72, 104, 107, 126, 130] als Unterscheidungskriterien und Vorhersagewerkzeuge an.

Ein topologischer Index ist ein numerischer Wert, der nur aus dem molekularen Graphen berechnet wird und der unabhängig von der Numerierung des Graphen ist. (Er bezieht sich somit nur auf die Isomorphieklasse.) Es wurde in der Literatur eine schier unübersichtliche Vielzahl solcher Indizes vorgestellt, stets angepaßt an die individuellen Bedürfnisse der jeweiligen Studie. Wir werden uns hier auf einige wenige beschränken, die heute als „klassisch“ gelten. Wir gehen dazu von einem molekularen Graphen (γ, β) mit $\gamma \in m^{\mathbb{P}^{(2)}}$ aus.

Es ist in vielen Fällen sinnvoll, die Zahl der Atome und die Zahl der Bindungen heranzuziehen:

$$(7.1.1) \quad A = p$$

$$(7.1.2) \quad B = |\{(i, j) \in \mathbb{P} \times \mathbb{P} \mid i < j \wedge \gamma(\{i, j\}) \neq 0\}|$$

Die Anzahl der Ringe wird bestimmt durch die cyclomatische Zahl

$$(7.1.3) \quad C = B - A + 1.$$

Das Molekulargewicht M berechnet sich als die Summe der mittleren Atomgewichte in kg/mol. Die Anzahlen von Stereozentren SZ und Stereoisomeren SI lassen sich mit den Verfahren aus [127, 128] ermitteln und als topologische Indizes verwenden.

Eine Reihe wichtiger Werte basiert auf der Adjazenzmatrix und der Inzidenzmatrix. Mit letzterem Begriff wollen wir die Matrix A' mit

$$a'_{i,j} = \begin{cases} 1 & \text{falls } \gamma(\{i, j\}) \neq 0 \\ 0 & \text{sonst} \end{cases}$$

bezeichnen. Neben dem Grad einer Ecke (s. 2.4.1), auf den wir uns hier der besseren Unterscheidbarkeit halber mit „deg.“ beziehen wollen, betrachten wir noch die Zahl der Bindungen an einer Ecke als

$$\delta_i := \sum_{j=1}^p a'_{i,j}.$$

Die *Zusammenhangsindizes* k_χ und k_χ^b für $k = 0, 1, 2$ nach [72] bilden jeweils die Summe über alle Pfade der Länge k im Graphen, wobei einmal die Zeilensummen der Inzidenz- und einmal die der Adjazenzmatrix verwendet werden:

$$(7.1.4) \quad {}^0\chi = \sum_{i=1}^p (\delta_i)^{-\frac{1}{2}}$$

$$(7.1.5) \quad {}^0\chi^b = \sum_{i=1}^p (\text{deg}_i)^{-\frac{1}{2}}$$

$$(7.1.6) \quad {}^1\chi = \sum_{\text{Kanten } i,j} (\delta_i \delta_j)^{-\frac{1}{2}}$$

$$(7.1.7) \quad {}^1\chi^b = \sum_{\text{Kanten } i,j} (\text{deg}_i \text{deg}_j)^{-\frac{1}{2}}$$

$$(7.1.8) \quad {}^2\chi = \sum_{\text{Pfade } i,j,k} (\delta_i \delta_j \delta_k)^{-\frac{1}{2}}$$

$$(7.1.9) \quad {}^2\chi^b = \sum_{\text{Pfade } i,j,k} (\text{deg}_i \text{deg}_j \text{deg}_k)^{-\frac{1}{2}}$$

Diese Werte zusammen ergeben eine gute Charakterisierung der Struktur, insbesondere hinsichtlich Form, Volumen und Oberfläche, und wurden daher für eine Vielzahl von Korrelationen verwendet [6, 72, 88].

Eine weitere wichtige Klasse von topologischen Indizes basiert auf der Abstandsmatrix D . Jeder Eintrag d_{ij} bezeichnet die Länge, d.h. die Zahl der Kanten, des kürzesten Pfades im Graphen von der Ecke i zur Ecke j (vgl. 2.1.3). Die Abstandsmatrix kann etwa mit Hilfe des Algorithmus von Floyd [39] berechnet werden. Die erste Anwendung von topologischen Indizes überhaupt wurde von H. WIENER [132] entwickelt. Sein Index lautet

$$(7.1.10) \quad W = \frac{1}{2} \sum_{i,j} d_{ij} = \sum_{i>j} d_{ij}$$

Er ist insbesondere ein Maß für die Verzweigkeit eines Moleküls, da er sein Maximum bei Ketten und sein Minimum bei stark verzweigten Strukturen annimmt. Daher wurde er auch von Wiener zur Modellierung des Siedepunktes von Alkanen eingesetzt.

Ein anderer Pionier dieses Fachgebiets ist A. BALABAN. In [3] stellte er einen Wert vor, der heute allgemein als *Balaban-Index* bezeichnet wird. Er wurde entwickelt, um besser zwischen Bindungsisomeren unterscheiden zu können, und ist die Umformulierung von 7.1.6 auf Abstandsmatrizen mit zusätzlicher Gewichtung bei Ringen. Er lautet (mit $d_i := \sum_j d_{ij}$ und den Bezeichnungen aus 7.1.1-7.1.3):

$$(7.1.11) \quad J = \frac{A}{C+1} \sum_{\text{Kanten } i,j} (d_i d_j)^{-\frac{1}{2}}$$

Von Balaban und Motoc stammt der *Index des mittleren quadratischen Abstands (mean square distance index)* [107]

$$(7.1.12) \quad MSD = \left(\frac{\sum_{i=1}^{d_{\max}} i^2 g_i}{\sum_{i=1}^{d_{\max}} g_i} \right)^{\frac{1}{2}}$$

wobei g_i die Anzahl der Eckenpaare in γ sei, die den Abstand i haben, und d_{\max} der größte Eintrag in D .

Von J. GALVEZ und Mitarbeitern wurde eine weitere Klasse von Indices eingeführt [43, 44], um Ladungsverteilungen besser modellieren zu können. Dazu leiten sie aus der Abstandsmatrix eine weitere Matrix D^* ab mit

$$d_{ij}^* = \begin{cases} \frac{1}{d_{ij}^\gamma} & \text{für } d_{ij} \neq 0 \\ 0 & \text{sonst} \end{cases}$$

und setzen $M := A \cdot D^*$ mit der Adjazenzmatrix A . Daraus ergeben sich die *Ladungsindizes* für $k = 1, 2, \dots, d_{\max}$ als

$$(7.1.13) \quad G_k = \sum_{i=1}^{p-1} \sum_{j=i+1}^p |m_{ij} - m_{ji}| \delta(k, d_{ij})$$

wobei δ wieder die Kroneckersche Deltafunktion sei (vgl. S. 88).

Schließlich verwenden wir auch noch *informationstheoretische Indizes* [9, 18]. Dabei wird eine Menge mit n Elementen in h Klassen aufgeteilt, wobei der relative Anteil der i -ten Klasse p_i sei. Nach der Formel von Shannon ergibt sich dann der mittlere Informationsgehalt als $-\sum_{i=1}^h p_i \log_2 p_i$. Eine der ersten topologischen Indizes dieser Art stammt von D. BONCHEV und N. TRINAJSTIĆ [19] und beschreibt den mittleren Informationsgehalt der Abstände in einem Graphen:

$$(7.1.14) \quad I_D^W = W \log_2 W - \sum_{i=1}^{d_{\max}} g_i \cdot i \log_2 i$$

Eine natürliche Aufteilung der Ecken ergibt sich durch die Bahnen der Automorphismengruppe. Sei daher p_i die Länge der i -ten Bahn (bei etwa h Bahnen) dividiert durch die Anzahl p der Ecken. Dann lassen sich folgende Indizes definieren [6, 9]:

$$(7.1.15) \quad IC = - \sum_{i=1}^h p_i \log_2 p_i$$

$$(7.1.16) \quad SIC = \frac{IC}{\log_2 p}$$

$$(7.1.17) \quad CIC = \log_2 p - IC$$

Während IC den Informationsgehalt repräsentiert, kommt SIC der Redundanz der Information näher; CIC stellt den Unterschied zwischen der maximalen Komplexität und der tatsächlichen topologischen Information heraus.

Es gibt viele Arten der Anwendung topologischer Indizes. Bei der Suche nach quantitativen Beziehungen zwischen Strukturen und ihren physico-chemischen Aktivitäten (meist als *QSAR* abgekürzt) verwendet man sie oft, um experimentelle Daten zu korrelieren und so Vorhersagen zu ermöglichen (vgl. etwa [6, 7, 8, 72, 124, 130] und S. 143). Allerdings ist davor zu warnen, sie als „Allheilmittel“ anzusehen. Sie sollten stets nur als vorübergehendes Werkzeug verwendet werden, bis die Effekte, die sie statistisch simulieren, verstanden sind und in einem theoriebezogenen Modell berechnet werden können.

7.2 Iterative Konstruktion

7.2.1 Allgemeiner Algorithmus

In diesem Abschnitt wollen wir die Grundlagen für die oben angekündigte Möglichkeit zur Konstruktion vorstellen. Die Idee ist, mit einer Reihe von Bausteinen zu beginnen und diese gemäß der vorgestellten Algorithmen untereinander zu verknüpfen bzw. zu verschmelzen. Die so entstandene Menge von Strukturen kann dann wieder mit allen Bausteinen verbunden werden usw. Auf diese Weise entstehen iterativ Mengen von immer größeren Graphen.

Da die Anzahlen der Elemente dieser Mengen aufgrund der kombinatorischen Vielfalt aber schnell über alle Grenzen wachsen, wollen wir die Konstruktion mit einem Ziel versehen. Das heißt, es soll über eine Wertungsfunktion jeder Struktur eine reelle Zahl zugewiesen werden; für die nächste Konstruktionsstufe wird dann nur noch eine Teilmenge von Graphen mit den besten Wertungen verwendet.

Bei der nachfolgenden Beschreibung des Algorithmus verwenden wir die Bezeichnungen:

S	eine Wertungsfunktion mit $S(\gamma) \in [0; \infty]$; ein gegebener Graph γ ist umso günstiger, je kleiner $S(\gamma)$ ist
η_1, \dots, η_t	eine Menge von Bausteinen
\mathcal{G}_k	die Menge der in Stufe $k = 1, 2, \dots$ erzeugten Graphen
\mathcal{G}_0	die Menge der Graphen, mit der die Konstruktion startet, z.B. $\mathcal{G}_0 = \{\eta_1\}$
r	Anteil (in %) der Elemente von \mathcal{G}_k , die im Schritt $k+1$ als neue Basisstrukturen verwendet werden sollen (z.B. $r = 0, 25$)
i, k	Laufvariablen

7.2.1 Algorithmus (Iterative bewertete Generierung).

- i) Initialisiere $k \leftarrow 1$.
- ii) Setze $\mathcal{G}_k \leftarrow \emptyset$. Durchlaufe alle $\gamma \in \mathcal{G}_{k-1}$ und führe jeweils durch:
 - (a) Setze $i \leftarrow 1$.
 - (b) Berechne die Menge \mathcal{M} aller Verknüpfungen mit 3.4.9 oder aller Verschmelzungen mit 5.2.5 der Graphen γ und η_i .
 - (c) Füge \mathcal{M} zu \mathcal{G}_k hinzu: $\mathcal{G}_k \leftarrow \mathcal{G}_k \cup \mathcal{M}$.
 - (d) Falls $i < t$, so setze $i \leftarrow i + 1$ und gehe zu Schritt a.
- iii) Ordne die Elemente $\gamma^{(k)} \in \mathcal{G}_k$ nach ihren Wertungen $S(\gamma^{(k)})$ monoton aufsteigend an.
- iv) Wähle eine Schranke ε so, daß für r % der Elemente $\gamma^{(k)} \in \mathcal{G}_k$ gilt: $S(\gamma^{(k)}) < \varepsilon$. Setze

$$\mathcal{G}_k \leftarrow \{\gamma^{(k)} \in \mathcal{G}_k \mid S(\gamma^{(k)}) < \varepsilon\}$$
- v) Falls noch eine weitere Stufe gewünscht wird, setze $k \leftarrow k + 1$ und gehe zu Schritt ii.

▽

7.2.2 Strukturaufklärung mittels Spektreninterpretation

Eine Anwendung dieses Verfahrens ist in der Strukturaufklärung möglich und wurde in vereinfachter Form in [133] vorgeschlagen. Die dortigen Autoren haben ein System *SpecSolv* entwickelt, mit dem in folgenden Schritten vorgegangen werden kann:

- i) Messung eines ^{13}C -NMR Spektrums und Extraktion der chemischen Verschiebungen, Intensitäten und Vielfachheiten
- ii) Teilstruktursuche in einer eigens erzeugten Datenbank mit Korrelationen zwischen Teilspektren und Teilstrukturen
- iii) Zusammensetzung der Substrukturen und Verifikation der Zwischenprodukte

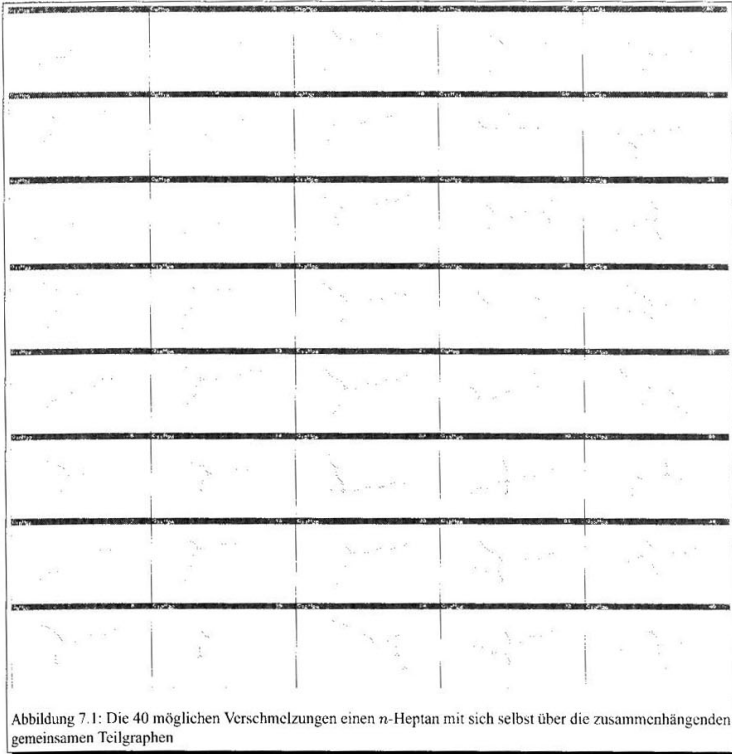


Abbildung 7.1: Die 40 möglichen Verschmelzungen eines n -Heptan mit sich selbst über die zusammenhängenden gemeinsamen Teilgraphen

Die ersten beiden Schritte können von uns aufgrund fehlender Apparaturen und Daten leider nicht nachvollzogen werden. Der dritte Schritt ist allerdings in oben beschriebenem Algorithmus enthalten. Als Wertungen werden die Teilstruktur-Teilspektren-Korrelationen herangezogen – je näher das Spektrum der aktuellen $\gamma^{(k)}$ am aufzuklärenden Spektrum liegt, desto besser ist die Wertung. Das Verfahren bricht ab, wenn das gesamte Spektrum erklärt ist.

In [133] wird für die Zusammensetzung der Strukturen ein Verfahren beschrieben, das eine starke Vereinfachung der von uns genannten Verschmelzung ist. Dort werden nur kettenförmige Endstücke überlappt, wobei die Isomorphie getrennt getestet wird. Für das kettenförmige n -Heptan mit sieben Kohlenstoffatomen geben die Autoren sechs Möglichkeiten an. Unser Algorithmus 5.2.5 liefert 40 Lösungen (s. Abb. 7.1), falls man die Konstruktion auf zusammenhängende gemeinsame Teilgraphen beschränkt, ansonsten 435 Lösungen.

Es ist zu vermuten, daß durch die Kombination unseres Verfahrens mit den Spektrgorithmen von Will et al. eine deutliche Steigerung der Erklärungsrate von derzeit 80 % möglich ist. Denn viele Möglichkeiten bleiben vom dort verwendeten Zusammensetzungsverfahren unberücksichtigt. Erst die Vollständigkeit, die unserem Ansatz innewohnt, kann diese Beschränkung überwinden helfen.

7.2.3 Inverses Problem der QSAR-Forschung

Wir haben bereits am Ende von Abschnitt 7.1 die Suche nach quantitativen Struktur-Aktivitätsbeziehungen (QSAR) kurz angesprochen, die Aussagen über die physiko-chemischen und insbesondere pharmakophorischen Eigenschaften von Verbindungen anstrebt. Auf diesem Gebiet geht man oft so vor [7, 8, 130], daß man einen Zusammenhang zwischen verschiedenen topologischen Indizes und der biologischen Wirkung durch eine Testreihe und aufbauende statistische Analysen herstellt. Eine wichtige Fragestellung ist in diesem Zusammenhang das sog. *Inverse Problem der QSAR-Forschung*: Finde die Menge von Strukturen, die in einem gegebenen Indexbereich liegt. Dazu gibt es eine Reihe von Arbeiten [48, 59, 73, 74, 115], die auf verschiedene Weisen das Problem angehen, im Generierungsaspekt aber nicht immer effektiv sind. Wir wollen an einem Beispiel zeigen, daß obiger Algorithmus 7.2.1 auch gut zur Untersuchung dieses Problem geeignet ist.

In [73] bezogen sich die Autoren auf Daten über die Wirksamkeit von Isonarkotika (vorwiegend Alkohole und Ester) aus [76, 91]. Eine Regressionsanalyse lieferte folgende Beziehungen zwischen der Aktivität pC und den Indizes ${}^1\chi$ und ${}^2\chi$ aus 7.1.6 bzw. 7.1.8:

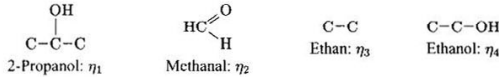
$$(7.2.2) \quad pC = 0,396 ({}^1\chi + {}^2\chi) - 0,431 ({}^1\chi - {}^2\chi) - 0,659$$

$$(7.2.3) \quad pC = 0,801 {}^1\chi - 0,658$$

Zur Illustration wurde nun ein enger Bereich der Aktivität vorgegeben, der etwa die Breite der Standardabweichung hat. Daraus ergeben sich folgende Grenzen für die Zusammenhangsindizes:

$$(7.2.4) \quad pC \in [2,2; 2,4] \Rightarrow {}^1\chi \in [3,568; 3,818], \quad {}^2\chi \in [2,621; 2,814]$$

Für Algorithmus 7.2.1 verwenden wir folgende Bausteine:



Wir verwenden in Schritt 7.2.1.iv nicht eine prozentuale, sondern eine feste Schranke von 100 Strukturen. Wir starten mit $\mathcal{G}_0 = \{\eta_1\}$.

Die Wertungsfunktion trennen wir nach den beiden Indizes als $S = S_1 + S_2$ und geben ihr den Wert 0 innerhalb und 1 außerhalb des angestrebten Intervalls. Um nicht einen zu krassen Gegensatz hervorzurufen, schwächen wir die Wertungsfunktionen in einem Bereich der Breite 0.5 um die Intervallgrenzen linear ab. Damit ergibt sich:

$$S_1 = \begin{cases} 1 & \text{falls } {}^1\chi < 3,068 \\ -2 {}^1\chi + 7,136 & \text{falls } 3,068 \leq {}^1\chi < 3,568 \\ 0 & \text{falls } 3,568 \leq {}^1\chi \leq 3,818 \\ 2 {}^1\chi - 7,636 & \text{falls } 3,818 < {}^1\chi \leq 4,318 \\ 1 & \text{falls } {}^1\chi > 4,318 \end{cases}$$

Analog für S_2 .

Die folgende Tabelle gibt einen Überblick über die erzeugten Strukturen. Dabei bezeichne \mathcal{G}'_k die Menge der Graphen, die nach der Reduktion in Schritt iv des Algorithmus 7.2.1 verbleiben und die als Basisstrukturen für die Stufe $k + 1$ verwendet werden. Zudem sei n die Zahl der Strukturen mit Wertung 0, d.h. deren Indizes ganz in den vorgeschriebenen Intervallen liegen.

Stufe k	$ \mathcal{G}_k $	$ \mathcal{G}'_k $	n
1	15	15	0
2	181	100	2
3	2468	100	15
4	4174	100	0

Abbildung 7.2 zeigt die 17 Graphen, die in den Stufen 2 und 3 erzeugt wurden. (Die Autoren von [73] konnten dort nur 8 Verbindungen angeben.)

Die nächste Tabelle listet schließlich die Index-Werte diese Strukturen zusammen mit der nach Gl. 7.2.2 berechneten Wirksamkeit auf:

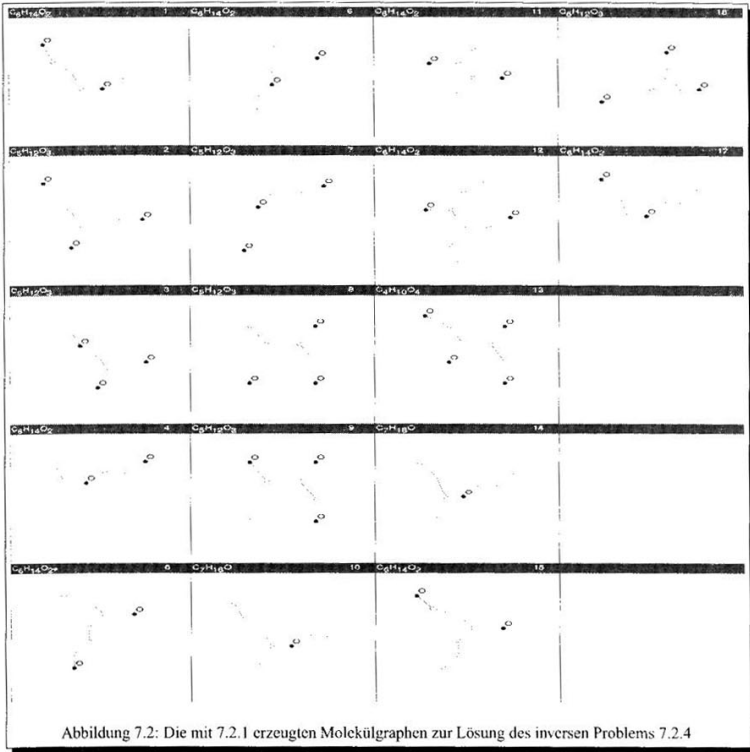


Abbildung 7.2: Die mit 7.2.1 erzeugten Molekülgraphen zur Lösung des inversen Problems 7.2.4

Nummer	χ_1	χ_2	pC
1	3,808	2,683	2,396
2	3,808	2,683	2,396
3	3,808	2,656	2,397
4	3,808	2,656	2,397
5	3,808	2,656	2,397
6	3,808	2,656	2,397
7	3,808	2,656	2,397
8	3,719	2,771	2,319
9	3,719	2,771	2,319
10	3,808	2,683	2,396
11	3,719	2,771	2,319
12	3,719	2,771	2,319
13	3,719	2,771	2,319
14	3,808	2,656	2,397
15	3,808	2,683	2,396
16	3,808	2,656	2,397
17	3,808	2,656	2,397

7.3 Design von Struktur- und Wirkstoffanaloga

Im Bereich der Entwicklung von Arzneimitteln werden zur Zeit viele neue computergestützte Verfahren vorgestellt [29, 46, 78, 79, 87, 90]. Diese basieren zumeist auf dem Modell, daß ein Wirkstoff in Wechselwirkung mit einem Protein tritt und dadurch seine Aktivität entfaltet (vgl. S. 143). Viele biologische Vorgänge sind aber noch nicht so weit aufgeklärt, daß man die Struktur des Proteins genau kennt. Daher bleibt die Aufgabe bestehen, zu einem bekannten Wirkstoff analoge Verbindungen zu suchen, die etwa dieselbe Haupt-, aber andere Nebenwirkungen haben, leichter herzustellen oder zu verabreichen sind. Auch in den Werkstoffwissenschaften stellt sich dieses Problem.

Wir wollen in diesem Abschnitt untersuchen, wie sich unser iteratives Konstruktionsverfahren 7.2.1 hierzu einsetzen läßt. Dabei verwenden wir verschiedene Wertungsfunktionen, die einmal auf Struktur- und einmal auf Wirkungseigenschaften der Verbindungen abzielen.

7.3.1 Strukturanaloga über topologische Ähnlichkeiten

Um strukturelle Ähnlichkeiten auszunutzen, geben wir eine Zielstruktur vor und vergleichen verschiedene topologische Indizes (vgl. Abschnitt 7.1) von dieser und den erzeugten Graphen aus \mathcal{G}_k . Sei γ eine solche Zielstruktur und $\gamma^{(k)} \in \mathcal{G}_k$ ein in Stufe k erzeugter Graph. Dann läßt sich die Wertungsfunktion S formulieren als:

$$S(\gamma^{(k)}) = \frac{|A(\gamma) - A(\gamma^{(k)})|}{A(\gamma)} + \frac{|B(\gamma) - B(\gamma^{(k)})|}{B(\gamma)} + (C(\gamma) - C(\gamma^{(k)})) +$$

$$\frac{|M(\gamma) - M(\gamma^{(k)})|}{M(\gamma)} + \frac{|SZ(\gamma) - SZ(\gamma^{(k)})|}{SZ(\gamma)} +$$

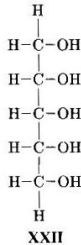
$$\frac{|{}^0\chi(\gamma) - {}^0\chi(\gamma^{(k)})|}{{}^0\chi(\gamma)} + \frac{|{}^1\chi(\gamma) - {}^1\chi(\gamma^{(k)})|}{{}^1\chi(\gamma)} + \frac{|{}^2\chi(\gamma) - {}^2\chi(\gamma^{(k)})|}{{}^2\chi(\gamma)} +$$

$$\frac{|W(\gamma) - W(\gamma^{(k)})|}{W(\gamma)} + \frac{|IC(\gamma) - IC(\gamma^{(k)})|}{IC(\gamma)} +$$

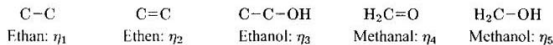
$$\frac{|SIC(\gamma) - SIC(\gamma^{(k)})|}{SIC(\gamma)} + \frac{|CIC(\gamma) - CIC(\gamma^{(k)})|}{CIC(\gamma)}$$

Im allgemeinen wird also die relative Abweichung vom Wert der Zielstruktur als Wertung verwendet. Nur bei der Zahl der Ringe ziehen wir den absoluten Fehler heran, da bei einer Abweichung hier kaum noch von „Analogon“ gesprochen werden kann.

Wir wollen die Technik der Berechnung von Strukturanaloga am Beispiel des Pentahydroxypentans



veranschaulichen. Um ein breites Spektrum möglicher Analogstrukturen abzudecken, verwenden wir als Bausteine:



Als Schranke setzen wir $\epsilon = 0, 25$, aber höchstens 200. Bei weniger als 10 verwenden wir alle erzeugten Verbindungen. Wir starten mit $\mathcal{G}_0 = \{\eta_1\}$.

Die folgende Tabelle gibt einen Überblick über die erzeugten Strukturen. Dabei bezeichne \mathcal{G}_k die Menge der Graphen, die nach der Reduktion in Schritt iv des Algorithmus 7.2.1 verbleiben und die als Basisstrukturen für die Stufe $k + 1$ verwendet werden.

Stufe k	$ \mathcal{G}_k $	$ \mathcal{G}'_k $
1	3	3
2	24	6
3	84	21
4	472	118
5	3165	200

Wir zeigen in Abb. 7.3 und 7.3 jeweils die besten zwölf (bzw. alle) Verbindungen der einzelnen Stufen. Die Wertung der einzelnen Graphen ist mit angegeben.

Aus der Abbildung wird deutlich, in welcher Weise unser Verfahren ähnliche Strukturen zu einer gegebenen Verbindung produziert. Auch die Ausgangsverbindung selbst erscheint in Stufe 5 in der Liste der Produkte.

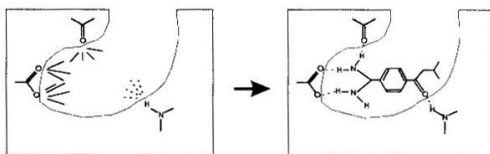
7.3.2 Wirkungsanaloga

Die Wechselwirkung von Ligand, d.h. Wirkstoff, und Rezeptor wird weniger von der genauen topologischen Struktur des Liganden bestimmt; hauptsächlich sind es *Wasserstoffbrückenbindungen*, die für die Rezeptoranlagerung verantwortlich sind. Dabei bindet sich ein Wasserstoffatom, das an einem elektronegativen Element X kovalent gebunden ist, mit dem freien Elektronenpaar eines anderen elektronegativen Atoms Y. Symbolisch bedeutet das:



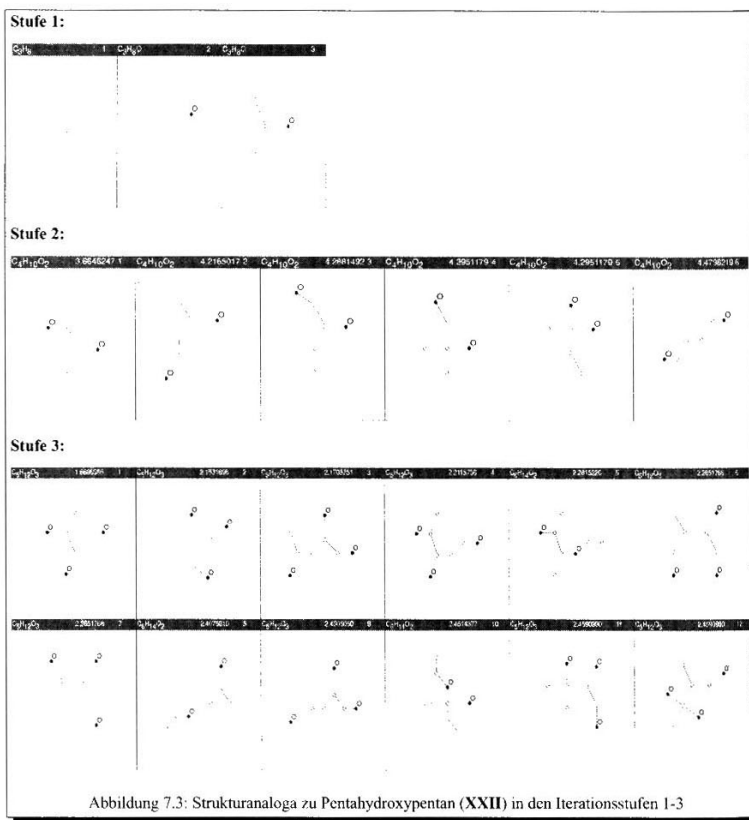
Ausschlaggebend ist also die Ladungverschiebung aufgrund der Elektronegativitätsunterschiede. Man bezeichnet X dabei auch als *Protonendonator* und Y als *Protonenakzeptor*.

In der Bindetasche des Rezeptors bestimmen die Positionen für Wasserstoffbrücken die möglichen Liganden, wie etwa in (nach [79])



wo die HN-Gruppe rechts als Akzeptor und die drei Sauerstoffe links und oben als Donatoren wirken. Ein Ligand, der zu einem gegebenen ähnlich sein soll, muß demzufolge also dieselben Möglichkeiten für Wasserstoffbrücken aufweisen.

Um dieser Problematik bei unserem Verfahren Rechnung zu tragen, haben wir neue topologische Indices entwickelt. Sie sind abgeleitet vom Wiener-Index 7.1.10 und berücksichtigen Anzahl und Verteilung von H-Brückenbindungspartnern. Dabei verwenden wir

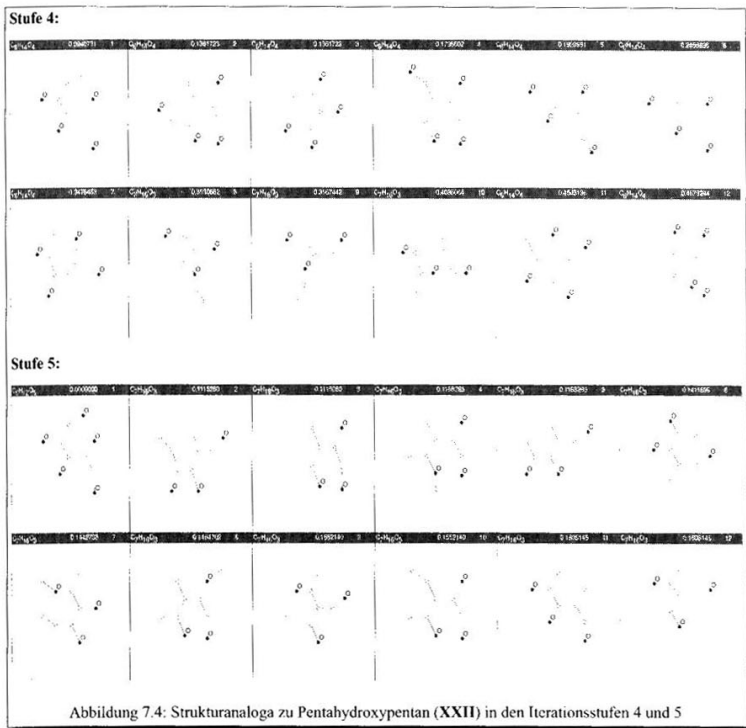


als Akzeptoren:

- Halogene (F, Cl, Br, I),
- Sauerstoff (sp^2 -hybridisiert),
- Stickstoff (sp -hybridisiert oder sp^2 -hybridisiert ohne Wasserstoffe)

als Donatoren:

- Sauerstoff (sp^3 -hybridisiert mit einem H) und
- Stickstoff (sp^3 -hybridisiert mit mindestens einem H)



Sei wieder p die Gesamtzahl der Nichtwasserstoffatome und D die Abstandsmatrix. Dann lauten unsere Indizes:

$$(7.3.1) \quad HB_A = \# \text{ Akzeptoren} + \sum_{i=1}^p \sum_{\substack{j=i+1 \\ \text{Akzeptor}}}^p d_{i,j}$$

$$(7.3.2) \quad HB_D = \# \text{ Donatoren} + \sum_{i=1}^p \sum_{\substack{j=i+1 \\ \text{Donator}}}^p d_{i,j}$$

Diese wollen wir als Teil einer Wertungsfunktion bei der iterativen Konstruktion einsetzen. Daneben verwenden wir noch

- die Anzahl C der Ringe (vgl. 7.1.3). Dieser Wert wird hinzugenommen, weil Ringe starke Auswirkungen auf die konformationelle Freiheit eines Liganden haben können.
- R – die Zahl der frei drehbaren Bindungen. Diese definieren wir als die Anzahl aller $C-C$ -Einfachbindungen, die nicht Teil eines Rings sind.

- die Ladungsindizes G_1 , G_2 und G_3 aus 7.1.13. Diese können nach [43, 44] gut dazu verwendet werden, die Elektronenverteilung innerhalb des Moleküls zu modellieren.

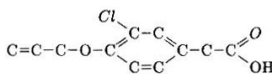
Unsere Wertungsfunktionen lautet schließlich (wobei nur die Summanden auch tatsächlich verwendet werden, deren Nenner ungleich 0 ist):

$$S(\gamma^{(k)}) = \frac{|G_1(\gamma) - G_1(\gamma^{(k)})|}{G_1(\gamma)} + \frac{|G_2(\gamma) - G_2(\gamma^{(k)})|}{G_2(\gamma)} + \frac{|G_3(\gamma) - G_3(\gamma^{(k)})|}{G_3(\gamma)} +$$

$$\frac{|HB_A(\gamma) - HB_A(\gamma^{(k)})|}{HB_A(\gamma)} + \frac{|HB_D(\gamma) - HB_D(\gamma^{(k)})|}{HB_D(\gamma)} +$$

$$\frac{|R(\gamma) - R(\gamma^{(k)})|}{R(\gamma)} + |C(\gamma) - C(\gamma^{(k)})|$$

Als Ausgangsverbindung nehmen wir



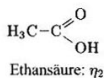
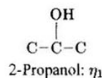
XXIII: Alclofenac

Diese hat die Referenzwerte

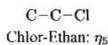
G_1	G_2	G_3	HB_A	HB_D	R	C
5	12,3333	2,375	8	1	5	1

Die kursiv markierten Atome Cl und O sind die beiden Akzeptoren; diese haben Abstand 6. Der einzige Donator ist die OH-Gruppe des Carboxylrestes auf der rechten Seite.

Da wir wieder ein breites Spektrum möglicher Analogstrukturen anstreben, verwenden wir als Bausteine:

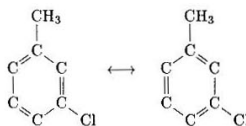


Benzol: η_4



Wir setzen als Schranke $\varepsilon = 0,25$, aber höchstens 200, und starten mit $\mathcal{G}_0 = \{\eta_1\}$.

Bei der Erzeugung tritt eine der größten Unzulänglichkeiten des bislang von uns verwendeten graphentheoretischen Molekülmodells aus 2.4.4 zutage: Die Einfach- und Doppelbindungen im Benzol-Ring sind lediglich eine Hilfszeichnung; in Wirklichkeit sind alle sechs Bindungen gleichwertig. Diesen besonders von Benzol und seinen Derivaten bekannten Effekt bezeichnet man als *aromatische Mesomerie*. Es existieren dabei zwei oder mehr graphentheoretisch verschiedene Molekülmodelle, die aber chemisch identisch sind, z.B. *meta*-Chlor-Toluol:



Daher wurde ein Verfahren entwickelt, das diese Fälle herausfiltert (vgl. auch [127, 128]):

7.3.3 Algorithmus (Filterung aromatischer Mesomere).

- i) Markiere alle aromatischen Bindungen.
- ii) Versch die Atome mit einer kanonischen Numerierung, die mittels iterierter K-Klassifizierung (etwa nach [11]) berechnet wird.
- iii) Speichere die so numerierten Verbindungen mittels eines Hash-Verfahrens (vgl. 3.4.9). Tritt dabei eine Kollision auf und ist das zu testende Molekül tatsächlich mit einem bereits gespeicherten identisch, so verwirf es.

▽

Dieses Verfahren können wir dazu verwenden, aromatische Doubletten aus den Ergebnismengen der einzelnen Stufen zu entfernen.

Die folgende Tabelle gibt einen Überblick über die erzeugten Strukturen. Dabei bezeichne \mathcal{G}_k die Menge der Graphen, die nach der Reduktion in Schritt iv des Algorithmus 7.2.1 verbleiben und die als Basisstrukturen für die Stufe $k + 1$ verwendet werden. Weiterhin sei \mathcal{G}_k'' die Menge aller nicht aromatisch-mesomeren Verbindungen aus \mathcal{G}_k .

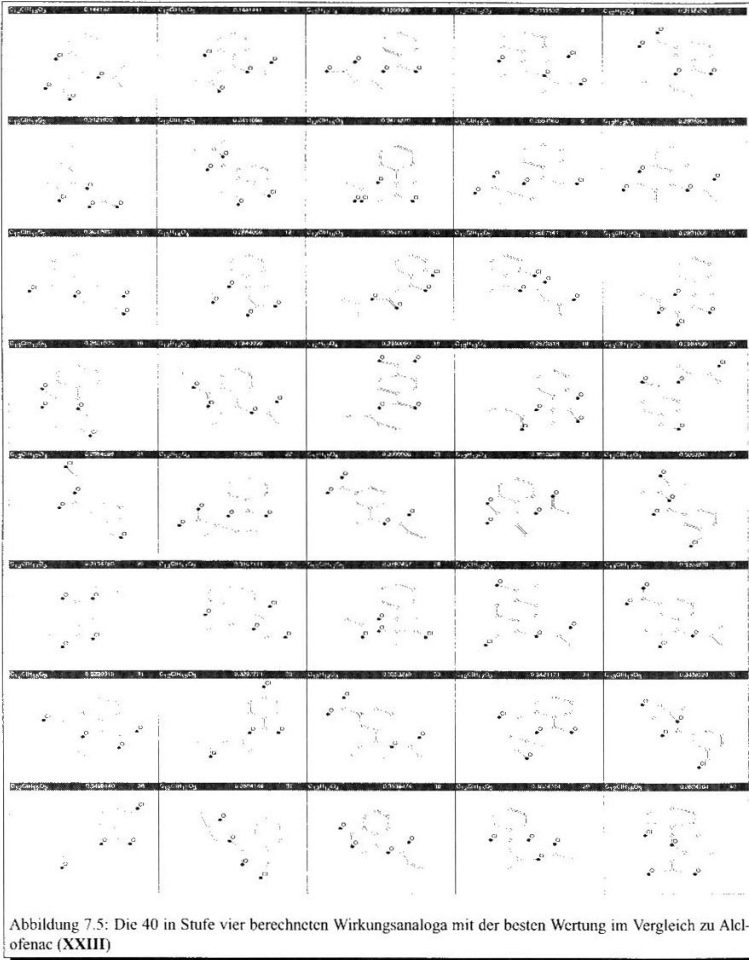
Stufe k	$ \mathcal{G}_k $	$ \mathcal{G}_k' $	$ \mathcal{G}_k'' $
1	22	5	5
2	198	49	45
3	2625	200	159
4	16718	200	122
5	23078	200	127
6	22916	200	119
7	21734	200	120
8	21096	200	121
Ges.:	108387		

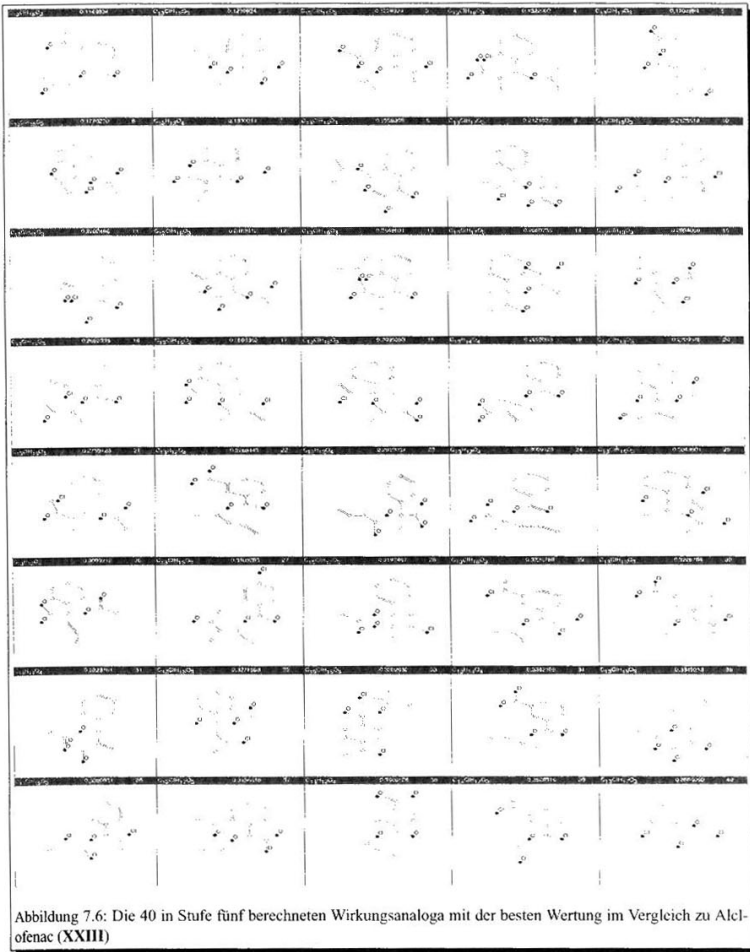
Wir zeigen in Abb. 7.5 - 7.7 jeweils die besten 40 Verbindungen der Stufen vier bis sechs. Die Wertung der einzelnen Graphen ist mit angegeben. Dabei zeigt sich, trotz der strukturellen Unterschiede, die Nähe der Ergebnisse zur Verteilung der Wirkzentren in **XXIII** deutlich – ein Argument für die Wasserstoffbrücken-Indizes 7.3.1 und 7.3.2. Um nun aus allen Stufen die besten Analoga herauszufinden, geben wir eine Schranke ρ für die Wertung vor. In Abhängigkeit von dieser ergeben sich die Anzahlen $|\{\gamma \in \mathcal{G}_k'' \mid S(\gamma) \leq \rho\}|$ als:

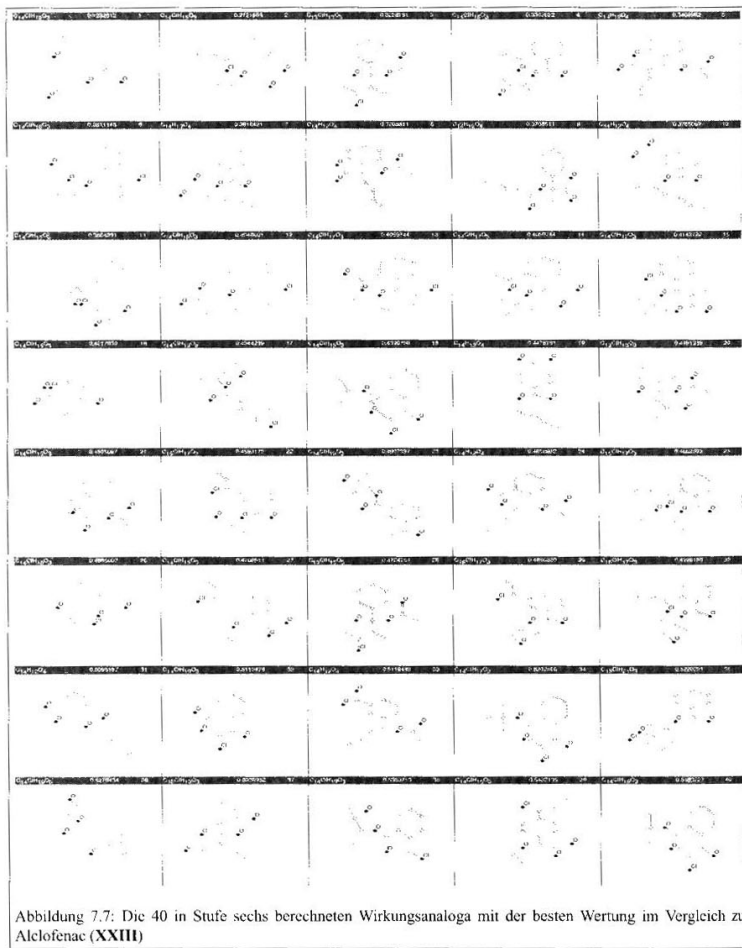
ρ	Stufe 1	Stufe 2	Stufe 3	Stufe 4	Stufe 5	Stufe 6	Stufe 7	Stufe 8	Summe
0,10	0	0	0	0	0	0	0	0	0
0,15	0	0	0	2	4	1	0	0	7
0,20	0	0	0	3	8	1	0	0	12
0,25	0	0	0	8	12	1	0	0	21
0,30	0	0	0	22	23	2	1	1	49
0,35	0	0	0	36	37	5	2	2	82
0,40	0	0	0	47	52	11	2	2	114
0,45	0	0	0	71	69	20	7	7	174
0,50	0	0	1	98	93	30	8	8	238

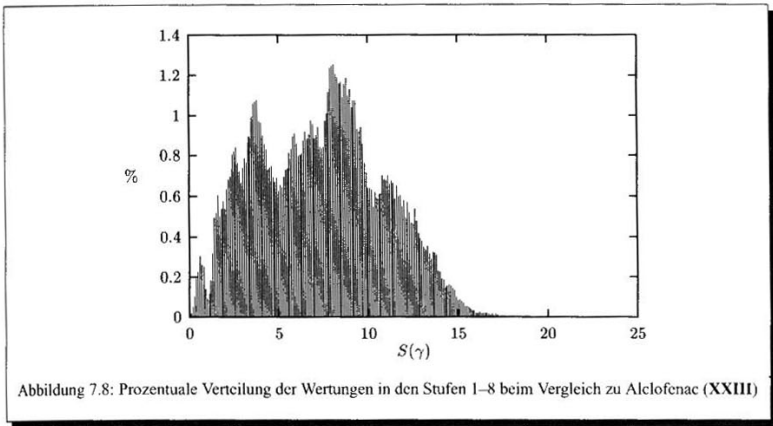
Die besten Analoga werden also in den Stufen vier bis sechs berechnet, die auch aus diesem Grunde hier abgedruckt sind. Die absolut niedrigste Wertung haben dabei die Spitzenreiter aus Stufe vier.

In Abb. 7.8 sind schließlich die relativen Häufigkeiten der Wertungen aller Strukturen aufgetragen, die in den Stufen 1-8 generiert wurden. Da die meisten Verbindungen relativ große Wertungen haben, erweist sich unsere rigide Schranke von 200 als gerechtfertigt.









Kapitel 8 **Simulationen in der kombinatorischen Chemie**

8.1 Hintergrund und Techniken

8.1.1 Einleitung

Der herkömmliche Weg, einen neuen pharmazeutischen Wirkstoff zu entwickeln, ist es, einen Naturstoff aus Bakterien, Pflanzen oder Tieren zu extrahieren, das vorhandene Potential in Labor und Datenbank der Firma heranzuziehen oder gezieltes Wirkstoffdesign auf der Grundlage von Mechanismus oder Struktur einzusetzen. Durch neue Analyseroboter ist es möglich geworden, mehrere Tausend Substanzen pro Tag auf ihre Wirksamkeit zu untersuchen (zu *screenen*). Zusammen mit der Notwendigkeit einer Kostenreduktion in der industriellen Forschung hat sich aus dieser Situation der Wunsch ergeben, neue Moleküle in sehr großer Zahl verfügbar zu haben. Hier hielt die industrielle Revolution auch Einzug in die Synthesechemie, was die Ablösung von Handarbeit durch schnelle und effiziente Maschinen bedeutete.

Erst seit wenigen Jahren setzt man dazu eine neue Technik ein, die nicht das klassische Ziel verfolgt, *einen* Stoff möglichst rein zu synthetisieren, sondern bewußt die Vielfalt ausnutzt, um gleichzeitig eine große Zahl von Verbindungen zu gewinnen. Diese Technik nennt man *kombinatorische Chemie* [24, 40, 42, 49, 119, 125]. Meistens verwendet man dabei einen Satz von *Bausteinen* (*building-blocks*) und setzt diese systematisch auf alle kombinatorisch möglichen Weisen mit einem *Grundgerüst* (*core*) zusammen, wobei die eigentlichen Reaktionen chemische, biologische oder biosynthetische Verfahren nutzen. Die Menge aller resultierenden Moleküle wird als *kombinatorische Bibliothek* bezeichnet.

Neben der hohen Effektivität dieser Technik gibt es noch einen weiteren Aspekt, der sie interessant und elegant erscheinen läßt. Dieser Zugang entspricht nämlich den Vorgängen in der Natur selbst, als am Ursprung des Lebens vor Jahrmillionen durch die Kombination von einfachen Kohlenwasserstoffen und Aminosäuren die ersten Proteine und Kohlenhydrate entstanden.

Ein ganz wesentlicher Gesichtspunkt bei der kombinatorischen Chemie ist die *Diversität* (*diversity*, Unähnlichkeit) [65, 88, 101]. Eine große kombinatorische Bibliothek erfüllt zwar die Forderung nach der Bereitstellung vieler Verbindungen; zur effizienten Untersuchung sollte aber erreicht werden, daß nicht die einzelnen Elemente einer Bibliothek zu ähnlich sind und daher eine pharmakologische Klasse immer wieder getestet wird. Das Ziel ist es, daß die Elemente einer Bibliothek möglichst unterschiedlich sind, so daß durch das Screening ein breites Spektrum abgedeckt wird, ohne daß zu viele Einzelstoffe notwendig sind.

Daher ist der Begriff der *Ähnlichkeit* im Zusammenhang mit der kombinatorischen Chemie von zentralem Interesse [9, 64, 65, 72, 88]. Die Möglichkeiten, diese zu definieren und zu bestimmen, sind sehr vielfältig. Es hat sich gezeigt, daß Ähnlichkeit nicht – wie etwa die Isomorphie – allgemein formuliert werden kann. Vielmehr hängt es sowohl von der Struktur als auch der untersuchten Wirkung ab, was als ähnlich anzusehen ist und was nicht. Oftmals wird der Begriff „Ähnlichkeit“ aber auch im Sinne von „struktureller Ähnlichkeit“ verwendet; im folgenden werden wir ebenfalls darauf zurückgreifen.

Die Vorgehensweise in der kombinatorischen Chemie besteht also aus folgenden Schritten:

- i) *Auswahl der Bausteine*
- ii) *Aufbau der Bibliothek*
- iii) *Screening der Bibliothekselemente auf die gewünschte Wirkung*

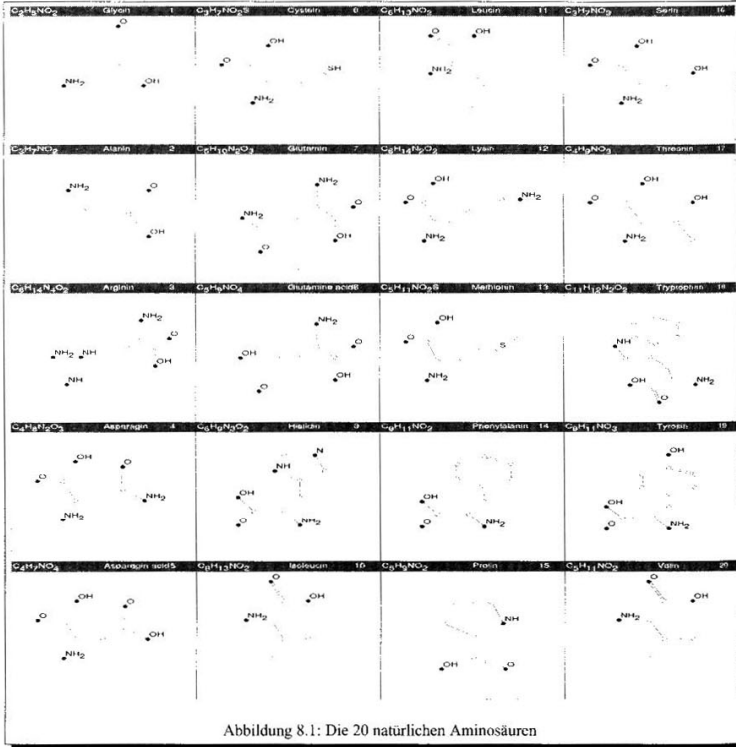


Abbildung 8.1: Die 20 natürlichen Aminosäuren

Ziel einer Computersimulation ist es natürlich, diese Schritte möglichst vollständig im Rechner nachzubilden. Im folgenden wollen die Möglichkeiten und Grenzen einer solchen Nachbildung untersuchen.

8.1.2 Auswahl der Bausteine

Bei der Auswahl der Bausteine gibt es eine Reihe von Möglichkeiten [88, 101, 113]. Ihr Einsatz hängt stark davon ab, welches Ziel mit der kombinatorischen Bibliothek verfolgt werden soll. Wir wollen hier zwei Techniken vorstellen, die graphentheoretische Berechnungen mit statistischer Auswertung kombinieren.

Als Beispiel dient uns die Menge der zwanzig natürlichen Aminosäuren (s. Abb. 8.1).

i) Topologische Indizes

Wie bereits bei der Vorstellung in Abschnitt 7.1 erwähnt, eignen sich topologische Indizes in begrenztem Umfang zur Modellierung von Moleküleigenschaften wie Flexibilität, Oberfläche, Verzweigkeit usw. Wir wollen sie daher verwenden, um die Ähnlichkeit einer Menge von Bausteinen zu untersuchen.

Tabelle 8.1 zeigt eine Reihe von Werten topologischer Indizes von Aminosäuren. Eine solche Vielzahl an Werten ist für die weitere Behandlung unpraktisch. Ein häufig verwendetes Verfahren zur Datenreduktion ist

	Atome	Bond	Ringe	SuZent	Strom	Wiener	Gewicht	σ_x	σ_x^2	1_x	1_x^2	2_x	2_x^2	3_x	3_x^2	I_{P^2}	MSD	Balab	IC	SIC	CIC
gly	5	4	0	0	1	18	80,010	4,284	3,914	2,270	1,914	1,802	1,311	57,250	1,732	1,917	2,922	0,880	0,400		
ala	6	5	0	1	2	29	96,010	5,155	4,784	2,643	2,297	2,488	2,001	109,900	2,000	2,405	2,931	0,792	0,369		
arg	12	11	0	4	8	247	188,000	9,560	8,820	5,537	4,835	4,900	3,950	143,000	4,243	3,032	3,194	0,680	1,506		
asn	9	8	0	1	2	96	140,000	7,439	6,699	4,037	3,335	3,851	2,909	479,700	2,828	3,047	3,102	0,759	0,986		
asp	9	8	0	1	2	96	140,000	7,439	6,699	4,037	3,335	3,851	2,909	479,700	2,828	3,047	3,078	0,770	0,922		
cys	7	6	0	1	2	46	128,100	5,862	5,492	3,181	2,835	2,620	2,156	195,600	2,226	2,666	3,325	0,873	0,483		
gln	10	9	0	1	2	136	156,000	8,146	7,406	4,537	3,835	4,192	3,243	719,400	3,317	3,047	3,041	0,704	1,280		
glu	10	9	0	1	2	136	156,000	8,146	7,406	4,537	3,835	4,192	3,243	719,400	3,317	3,047	3,005	0,708	1,243		
his	11	11	1	6	16	165	164,000	8,268	7,431	5,198	4,199	4,607	3,409	921,000	3,317	2,029	3,504	0,811	0,818		
ile	9	8	0	2	4	92	144,000	7,439	7,069	4,091	3,746	3,489	3,021	461,400	2,646	3,207	2,768	0,621	1,692		
leu	9	8	0	1	2	96	144,000	7,439	7,069	4,037	3,691	3,851	3,377	479,700	2,828	3,047	2,768	0,621	1,692		
lys	10	9	0	1	2	143	160,000	7,983	7,613	4,681	4,335	3,717	3,243	753,700	3,606	2,899	2,781	0,607	1,804		
met	9	8	0	1	2	102	160,100	7,276	6,906	4,181	3,835	3,164	2,890	507,100	3,162	2,865	3,022	0,699	1,200		
phe	12	12	1	1	2	212	176,000	8,975	7,878	5,698	4,392	4,961	3,437	128,000	3,696	2,066	2,888	0,639	1,635		
pro	8	8	1	2	4	62	124,000	5,983	5,613	3,805	3,459	3,289	2,828	288,600	2,226	1,961	2,814	0,688	1,274		
ser	7	6	0	1	2	46	112,000	5,862	5,492	3,181	2,835	2,620	2,156	195,600	2,226	2,666	3,039	0,798	0,768		
thr	8	7	0	2	4	65	128,000	6,732	6,362	3,553	3,208	3,347	2,879	303,400	2,449	3,026	3,052	0,747	1,036		
trp	15	16	2	4	8	369	216,000	10,840	9,885	7,182	5,609	6,503	4,542	240,000	3,873	1,794	3,320	0,698	1,435		
tyr	13	13	1	1	2	268	192,000	9,845	8,801	6,092	4,803	5,583	3,937	1625,000	3,873	2,106	3,183	0,694	1,402		
val	8	7	0	1	2	65	128,000	6,732	6,362	3,553	3,208	3,347	2,879	303,400	2,449	3,026	2,774	0,653	1,474		
Mittelwert:	9,350	8,650	0,300	1,650	3,550	124,450	146,611	7,470	6,870	4,302	3,677	3,830	3,016	683,358	2,939	2,645	3,026	0,722	1,196		
Standardabh.	2,412	2,834	0,571	1,424	3,502	89,753	33,552	1,614	1,386	1,193	0,882	1,110	0,732	589,951	0,697	0,485	0,296	0,080	0,407		

Tabelle 8.1: Topologische Indizes der 20 natürlichen Aminosäuren aus Abb. 8.1

die *Hauptkomponentenanalyse*. Dabei werden implizite Korrelationen zwischen den Werten ausgenutzt und über die Bestimmung der Eigenvektoren einige wenige, neue Variablen, sog. Faktoren ermittelt, die einen Großteil der Varianz der ursprünglichen Daten erklären.

Für die Daten aus Tab. 8.1 haben wir eine Hauptkomponentenanalyse durchgeführt.* Dabei ergaben sich drei Faktoren, die zusammen 93,2 % der Varianz erklären. Deren Regressionswerte zeigt folgende Tabelle:

	f_1	f_2	f_3		f_1	f_2	f_3
gly	-1,83387	1,13563	-1,41418	leu	-0,07249	-1,36088	0,09030
ala	-1,33258	0,49452	-0,50175	lys	0,32192	-1,47271	-0,02419
arg	1,25580	-0,08585	1,85784	met	-0,10939	-0,50501	0,30335
asn	-0,27493	-0,05936	0,56538	phe	0,94399	-0,54371	-1,75631
asp	-0,29056	-0,01905	0,54880	pro	-0,42065	0,24750	-1,54562
cys	-1,00464	1,17446	0,44508	ser	-1,01410	0,44544	-0,00660
gln	0,17849	-0,57497	0,49897	thr	-0,51357	0,09556	0,86045
glu	0,16735	-0,58538	0,46081	trp	2,38110	1,17295	-1,23013
his	0,84679	2,60881	1,48386	tyr	1,38372	0,00789	-1,27212
ile	-0,10004	-1,16748	0,60622	val	-0,51236	-1,00835	0,02984

Man erkennt bereits hieran, daß Strukturen, die sich nur wenig unterscheiden, auch ähnliche Werte aufweisen, wie etwa Asparagin (asn) und Asparaginsäure (asp) – eine Bestätigung unserer Ausgangsannahme, daß topologische Indizes zur Ähnlichkeitsbestimmung geeignet sind.

Bestimmt man daraus noch die euklidischen Distanzen, also

$$\sqrt{(f_1^{(i)} - f_1^{(j)})^2 + (f_2^{(i)} - f_2^{(j)})^2 + (f_3^{(i)} - f_3^{(j)})^2},$$

so ergibt sich die Abstandsmatrix in Tabelle 8.2. Diese zeigt noch stärker, wie groß die Unterschiede zwischen den einzelnen Molekülen quantitativ sind.

Um diese Ergebnisse nun zur Bausteinauswahl zu verwenden, kann man eine Gruppeneinteilung bestimmen. Dabei sollen alle die Strukturen in eine Gruppe kommen, deren Faktoren einen euklidischen Abstand unter einem Schwellwert ε haben. Die Größe dieses Schwellwertes und damit die Feinheit der Aufteilung wird durch die Anforderungen des Experiments bestimmt. Bei $\varepsilon = 1.0$ etwa ergibt sich folgende Aufteilung:

{gly}, {ala, ser}, {arg}, {asn, asp, gln, glu, met, thr},
 {cys}, {his}, {ile, leu, lys, val}, {phe, tyr}, {pro}, {trp}

In einem ersten Schritt kann man sich dann darauf beschränken, nur je einen Repräsentanten jeder Gruppe als Baustein zu nehmen, um dann nur die Gruppen genauer zu untersuchen, deren Vertreter die gewünschte Wirkung gezeigt hat.

ii) Binäre Eigenschaftsvektoren

Zur Charakterisierung können auch Listen mit binären Eigenschaften betrachtet werden. Wir haben eine Teilmenge von 120 Deskriptoren aus den Strukturcodes des Massenspektreninformationssystems *MassLib* [61] verwendet, wie sie auch in K. Varmuzas Programm *ToSIM* [110, 122, 123] eingesetzt wird. Dabei werden folgende Typen von Struktureigenschaften berücksichtigt:

- Aromatische Verbindungen (z.B. Teilstruktur Phenyl)
- Verzweigungen in Ketten und Ringen (z.B. Kohlenstoff mit drei C-Nachbarn, die selbst nicht Teil eines Rings sind)
- Cyclische Verbindungen (z.B. Sechsringe)
- Doppel- und Dreifachbindungen in Ketten und Ringen (z.B. Doppelbindung in einem Ring)
- Elemente (z.B. Heteroatome)
- Funktionelle Gruppen (z.B. Aldehydgruppe)
- Spezielle Klassen von Verbindungen (z.B. Methylester)

*Für die Rechnungen wurde das Programm *SI/SS* in der Version 6.0.1 für Windows verwendet.

	gly	ala	arg	asn	asp	cys	gln	glu	his	ile	leu	lys	met	phe	pro	ser	thr	trp	tyr	val
gly	0	1,22	4,66	2,79	2,75	2,04	3,26	3,24	4,21	3,52	3,41	3,66	2,84	3,26	1,67	1,77	2,83	4,22	3,41	2,9
ala	1,22	0	3,55	1,6	1,57	1,21	2,1	2,08	3,63	2,25	2,22	2,61	1,77	2,8	1,41	0,591	1,64	3,84	3,87	1,79
arg	4,66	3,55	0	2	2,03	2,95	1,8	1,84	2,75	2,14	2,55	2,52	2,11	3,66	3,81	2,89	2,04	3,52	3,13	2,71
asn	2,79	1,6	2	0	0,0463	1,44	0,69	0,695	3,04	1,12	1,4	1,64	0,343	2,67	2,14	1,06	0,41	3,43	2,48	1,12
asp	2,75	1,57	2,03	0,0463	0	1,39	0,729	0,734	3,01	1,17	1,43	1,68	0,374	2,67	2,12	1,02	0,4	3,42	2,47	1,14
cys	2,04	1,21	2,95	1,44	1,39	0	2,11	2,11	2,56	2,52	2,72	3	1,91	3,41	2,27	0,858	1,26	3,78	3,16	2,28
gln	3,26	2,1	1,8	0,69	0,729	2,11	0	0,0411	3,4	0,663	0,921	1,05	0,355	2,38	2,28	1,65	1,03	3,3	2,22	0,941
glu	3,24	2,08	1,84	0,695	0,734	2,11	0,0411	0	3,42	0,657	0,892	1,02	0,328	2,35	2,25	1,64	1,04	3,29	2,2	0,909
his	4,21	3,63	2,75	3,04	3,01	2,56	3,4	3,42	0	3,99	4,31	4,38	3,46	4,52	4,04	3,22	2,93	3,43	3,83	4,13
ile	3,52	2,85	2,14	1,12	1,17	2,52	0,663	0,657	3,99	0	0,552	0,818	0,728	2,66	2,6	1,95	1,35	3,87	2,67	0,726
leu	3,41	2,32	2,55	1,4	1,43	2,72	0,921	0,892	4,31	0,552	0	0,426	0,883	2,26	2,32	2,04	1,71	3,77	2,42	0,567
lys	3,66	2,61	2,52	1,64	1,68	3	1,05	1,02	4,38	0,818	0,426	0	1,11	2,06	2,41	2,34	1,98	3,56	2,21	0,956
met	2,84	1,77	2,11	0,343	0,574	1,91	0,355	0,328	3,46	0,728	0,883	1,11	0	2,31	2,02	1,35	0,913	3,37	2,23	0,7
phe	3,26	2,8	3,66	2,67	2,67	3,41	2,38	2,35	4,52	2,66	2,26	2,06	2,31	0	1,59	2,81	3,06	2,3	0,856	2,35
pro	1,67	1,41	3,81	2,14	2,12	2,27	2,28	2,25	4,04	2,6	2,32	2,41	2,02	1,59	0	1,66	2,41	2,97	1,84	2,02
ser	1,77	0,91	2,99	1,06	1,02	0,838	1,65	1,64	3,22	1,95	2,04	2,34	1,35	2,81	1,66	0	1,06	3,68	2,75	1,54
thr	2,83	1,64	2,04	0,41	0,4	1,26	1,03	1,04	2,93	1,35	1,71	1,98	0,913	3,06	2,41	1,06	0	3,73	2,86	1,38
trp	4,22	3,84	3,52	3,43	3,42	3,78	3,3	3,29	3,43	3,87	3,77	3,56	3,37	2,3	2,97	3,68	3,73	0	1,53	3,84
tyr	3,41	2,87	3,13	2,48	2,47	3,16	2,22	2,2	3,83	2,67	2,42	2,21	2,23	0,856	1,84	2,75	2,86	1,53	0	2,51
val	2,9	1,79	2,71	1,12	1,14	2,28	0,941	0,909	4,13	0,726	0,567	0,956	0,7	2,35	2,02	1,54	1,38	3,84	2,51	0

Tabella 8.2: Abstandsmatrix der Hauptkomponenten der topologischen Indizes aus Tab. 8.1

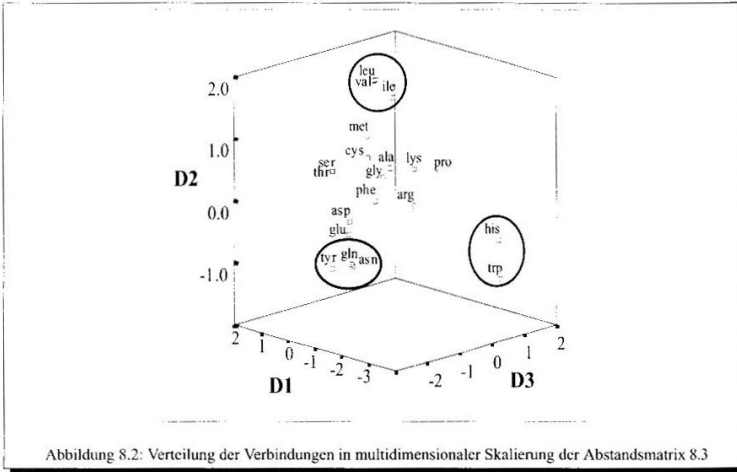


Abbildung 8.2: Verteilung der Verbindungen in multidimensionaler Skalierung der Abstandsmatrix 8.3

Viele dieser Eigenschaften sind über Teilstrukturen beschreibbar, so daß einer der wichtigsten Algorithmen zu deren Bestimmung die Teilstruktursuche ist. Hierbei wird ein Verfahren aus [11] verwendet. Andere Eigenschaften werden nach [14, 71, 131] berechnet.

Zur Auswertung verwenden wir den *Tanimoto*-Koeffizienten [134], der die Ähnlichkeit zwischen Bitstrings mißt:

$$T_{ij} = \frac{2C_{ij}}{E_i + E_j}$$

wobei C_{ij} die Anzahl der Eigenschaften ist, die die i -te und die j -te Struktur gemeinsam haben, und E_i die Gesamtanzahl der Eigenschaften der i -ten Verbindung ist. Damit ist $T \in [0, 1]$, mithin 1 bei völliger Übereinstimmung und 0 bei völliger Unähnlichkeit.

Wir bestimmen daraus wieder eine Abstandsmatrix D mit $d_{ij} = 1 - T_{ij}$. Diese ist in 8.3 dargestellt. Aus dieser Matrix wurden mit Hilfe einer *Multidimensionalen Skalierung* Anordnungen im Raum berechnet. Die geometrischen Abstände zwischen den Strukturen werden dabei so gewählt, daß sie zu den paarweisen Abständen eine so enge Beziehung wie möglich aufweisen. Abbildung 8.2 zeigt die Anordnungen der Strukturen im Raum. Auch hier können dann wieder, ähnlich wie bei den Hauptkomponenten der topologischen Indizes, eng benachbarte Verbindungen zu Gruppen zusammengefaßt werden (wie durch die Ellipsen in der Abbildung teilweise angedeutet).

Mit der Konstruktion der Bibliotheken werden wir uns im nächsten Abschnitt beschäftigen.

	gly	ala	arg	asn	asp	cys	gln	glu	his	ile	leu	lys	met	phe	pro	ser	thr	trp	tyr	val
gly	0	0	0.2	0.333	0.2	0.0476	0.333	0.2	0.391	0.1	0.143	0.1	0.0476	0.238	0.182	0.143	0.143	0.538	0.333	0.143
ala	0	0	0.2	0.333	0.2	0.0476	0.333	0.2	0.391	0.1	0.143	0.1	0.0476	0.238	0.182	0.143	0.143	0.538	0.333	0.143
arg	0.2	0.2	0	0.238	0.2	0.238	0.238	0.2	0.304	0.3	0.333	0.2	0.238	0.333	0.364	0.333	0.333	0.538	0.429	0.333
asn	0.333	0.333	0.238	0	0.238	0.364	0	0.238	0.5	0.429	0.455	0.238	0.364	0.455	0.478	0.273	0.273	0.556	0.364	0.455
asp	0.2	0.2	0.2	0.238	0	0.238	0.238	0	0.478	0.3	0.333	0.3	0.238	0.333	0.364	0.238	0.238	0.556	0.364	0.182
cys	0.0476	0.0476	0.238	0.364	0.238	0	0.364	0.238	0.417	0.143	0.182	0.143	0	0.273	0.217	0.182	0.182	0.556	0.364	0.455
gln	0.333	0.333	0.238	0	0.238	0.364	0	0.238	0.5	0.429	0.455	0.238	0.364	0.455	0.478	0.273	0.273	0.556	0.364	0.455
glu	0.2	0.2	0.2	0.238	0	0.238	0.238	0	0.478	0.3	0.333	0.3	0.238	0.333	0.364	0.238	0.238	0.556	0.333	0.333
his	0.391	0.391	0.304	0.5	0.478	0.417	0.5	0.478	0	0.478	0.5	0.391	0.417	0.417	0.2	0.5	0.5	0.31	0.5	0.5
ile	0.1	0.1	0.3	0.429	0.3	0.143	0.429	0.3	0.478	0	0.0476	0.2	0.143	0.333	0.273	0.238	0.238	0.538	0.429	0.0476
leu	0.143	0.143	0.333	0.455	0.333	0.182	0.455	0.333	0.5	0.0476	0	0.238	0.182	0.364	0.304	0.273	0.273	0.556	0.455	0
lys	0.1	0.1	0.2	0.238	0.3	0.143	0.238	0.3	0.391	0.2	0.238	0	0.143	0.333	0.273	0.238	0.238	0.462	0.429	0.238
met	0.0476	0.0476	0.238	0.364	0.238	0	0.364	0.238	0.417	0.143	0.182	0.143	0	0.273	0.217	0.182	0.182	0.556	0.364	0.182
phe	0.238	0.238	0.333	0.455	0.333	0.273	0.455	0.333	0.417	0.333	0.364	0.333	0.273	0	0.304	0.364	0.364	0.407	0.0909	0.364
pro	0.182	0.182	0.364	0.478	0.364	0.217	0.478	0.364	0.2	0.273	0.304	0.273	0.217	0.304	0	0.304	0.304	0.357	0.391	0.304
ser	0.143	0.143	0.333	0.455	0.333	0.182	0.455	0.333	0.5	0.0476	0.2	0.238	0.182	0.364	0.304	0	0.0909	0.63	0.723	0.273
thr	0.143	0.143	0.333	0.455	0.333	0.182	0.455	0.333	0.5	0.0476	0.2	0.238	0.182	0.364	0.304	0.0909	0	0.63	0.723	0.273
trp	0.538	0.538	0.538	0.556	0.615	0.556	0.556	0.615	0.31	0.538	0.556	0.462	0.556	0.407	0.357	0.63	0.63	0	0.481	0.556
tyr	0.333	0.333	0.429	0.364	0.333	0.364	0.364	0.333	0.5	0.429	0.455	0.429	0.364	0.0909	0.391	0.273	0.273	0.481	0	0.455
val	0.143	0.143	0.333	0.455	0.333	0.182	0.455	0.333	0.5	0.0476	0	0.238	0.182	0.364	0.304	0.273	0.273	0.556	0.455	0

Tabelle 8.3: Abstandsmatrix der Tanimotokoeffizienten der binären Eigenschaftsvektoren

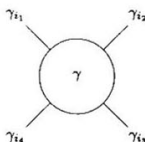
8.2 Aufbau einer kombinatorischen Bibliothek

8.2.1 Einstufige Verfahren

Für den Aufbau einer kombinatorischen Bibliothek aus gegebenen Bausteinen haben wir mit 6.2.2 schon eine geeigneten Algorithmus bereitgestellt. Denn hierbei befinden wir uns gerade – graphentheoretisch gesprochen – in der Situation, daß folgende Daten gegeben sind:

- ein Reaktionsschema $((\eta_1, \chi_1), (\eta_2, \chi_2), \rho)$,
- ein Grundgerüst (γ, β) , das k Teilstrukturen isomorph zu (η_1, χ_1) mit $k > 1$ enthält,
- eine Reihe von Bausteinen $(\gamma_1, \beta_1), \dots, (\gamma_n, \beta_n)$, die jeweils genau eine Teilstruktur isomorph zu (η_2, χ_2) enthalten.

Es gilt, unter Berücksichtigung der Symmetrie des Grundgerüsts alle wesentlich verschiedenen Anlagerungen (vgl. 6.1.3) der Bausteine zu finden, etwa von der Form



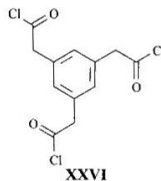
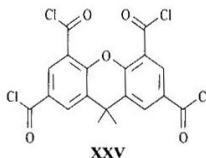
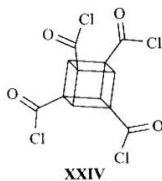
(Dabei setzen wir voraus, daß jeder Baustein für jeden Platz zulässig ist. Andernfalls müssen zusätzliche Regeln formuliert werden.)

Für die Praxis relevant ist noch, daß die Häufigkeit des Vorkommens eines bestimmten Bausteins restringiert werden kann, daß also etwa ein (γ_i, β_i) in allen Verbindungen der Bibliothek mindestens r und höchstens s mal vorkommt. Dies kann durch einen zusätzlichen Test in 6.2.2 zwischen Schritt iii und Schritt iv erreicht werden und ist auch in NEXTOR so implementiert.

Experimentell kann diese Restriktion durch eine geeignete Modifikation der Versuchsbedingungen eingehalten werden.

Als Beispiel wollen wir die kombinatorischen Bibliotheken aus [24] betrachten. Die Autoren verwendeten als Bausteine die zwanzig natürlichen Aminosäuren (siehe Abbildung 8.1 auf Seite 128) und als Grundgerüste verschiedene Säurechloride (s.u.). Durch die Reaktion der starren Gerüststruktur in Lösung mit einer Mischung kleinerer Moleküle an den Anknüpfungspunkten werden Bibliotheken erzeugt, in denen alle Verbindungen eine ähnliche räumliche Gestalt haben (bedingt durch die Form des Gerüsts), aber ein breites Spektrum von Wirkungen zeigen (aufgrund der funktionellen Gruppen der Bausteine).

Die Gerüstmoleküle waren



ein Cubanderivat (Struktur XXIV), Xanthen (XXV) und ein Benzoltrissäurechlorid (XXVI).

Das Reaktionsschema besteht aus den Teilstrukturen



und der Matrix

$$\rho = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\infty & -\infty & -\infty & -\infty & -\infty \end{pmatrix},$$

d.h. die Reaktion spaltet das Chlor ab und bindet den Kohlenstoff an den Stickstoff des Amins.

Das Cubanderivat **XXIV** ist die Struktur mit der größten Symmetrie, also der größten topologischen Automorphismengruppe (induziert von der Symmetriegruppe des Würfels ohne Inversionen). Diese hat die Ordnung 24. Da bei jedem Automorphismus auch die reagierenden Teilstrukturen aufeinander abgebildet werden, ist $|P_\gamma| = 24$. Da es sich nur um vier Plätze handelt, also $P_\gamma \leq S_4$, folgt $P_\gamma = S_4$ (vgl. 2.2.3).

Das Xanthen **XXV** hat eine Automorphismengruppe mit vier Elementen. Davon beinhalten zwei die Vertauschung der Methylgruppen am Kohlenstoffbrückenatom, so daß $|P_\gamma| = 2$. Neben der Identität ist dies die Spiegelung der Ringe an der senkrechten Symmetrieachse.

Das Benzoltrisäurechlorid **XXVI** hat eine cyclische Symmetrie. Daher ist P_γ gleich der cyclischen Gruppe C_3 und hat drei Elemente.

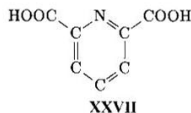
Auch wenn nur bei einem der drei Grundgerüste die Symmetrieverhältnisse etwas komplizierter sind, so zeigt sich doch daran der Vorteil des mathematischen Konzepts hinter 6.2.2. Durch den allgemeinen Ansatz mit einer beliebigen Permutationsgruppe und der bekanntermaßen effektiven Ordnungstreuen Erzeugung (vgl. 1.3.4, 3.3.2 und 5.1.4) können so die Bibliotheken in allen drei Fällen schnell generiert werden.

Angaben über die Größen der Bibliotheken geben wir in Abschnitt 8.3. Die Abbildung 8.3 zeigt jeweils acht Strukturen für die drei Fälle.[†]

8.2.2 Mehrstufige Verfahren

Es gibt auch die Möglichkeit, mehrere Reaktionen hintereinander ablaufen zu lassen und so die Produkte der einen als Grundgerüste der anderen Reaktion zu verwenden. Die mathematische Situation ist dabei jeweils wie im letzten Unterabschnitt; aufgrund der Verschiedenartigkeit der Grundgerüste bleibt auch die Redundanzfreiheit gewährleistet.

Als Beispiel wollen wir eine Siebenkomponentenreaktion aus [120] betrachten. Ausgangspunkt ist Pyridin-2,6-dicarbonsäure

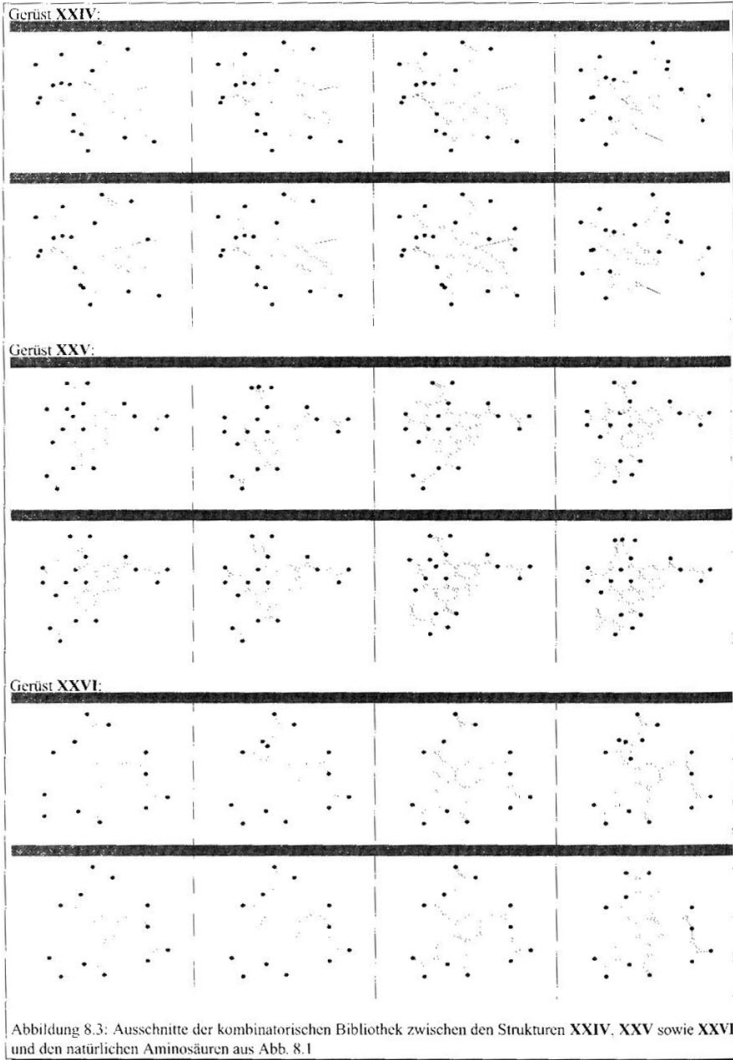


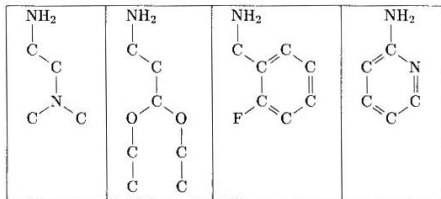
Hier tritt abermals die Unzulänglichkeit des graphentheoretischen Molekülmodells aus 2.4.4 zutage (vgl. Abschn. 7.3.2): Die Einfach- und Doppelbindungen im Pyridin-Ring sind lediglich eine Hilfszeichnung; in Wirklichkeit sind alle sechs Bindungen aufgrund der aromatischen Mesomerie gleichwertig. Wir können das Verfahren 7.3.3 dazu verwenden, aromatische Doubletten aus der berechneten kombinatorischen Bibliothek zu entfernen. Der Einfachheit halber wollen wir aber im folgenden von einer topologischen Äquivalenz der beiden COOH-Gruppen in **XXVII** ausgehen. Dann umfaßt P_γ zwei Elemente.

Die Reaktionsstufen sind im einzelnen:

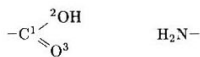
- i) In der ersten Stufe werden vier Amide hinzugefügt:

[†]Die Plazierungen wurden wieder mit **MOLGEN** automatisch erstellt. Hier zeigen sich die gegenwärtigen Unzulänglichkeiten bei der Verwendung dieses Plazierungsalgorithmus [14, 112] für kombinatorische Bibliotheken.





Das Reaktionsschema besteht aus den Teilstrukturen

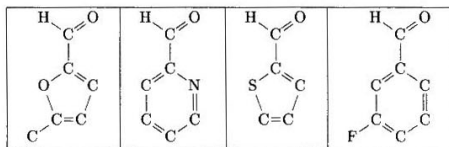


und der Matrix

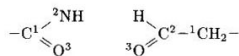
$$\rho = \begin{pmatrix} 1 \\ -\infty \\ 0 \end{pmatrix}.$$

Dafür gibt es 10 Möglichkeiten.

- ii) Die zweite Stufe nimmt vier aromatische Aldehyde als Bausteine:



Das zugehörige Reaktionsschema verwendet die Teilstrukturen

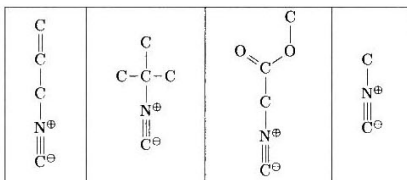


und die Matrix

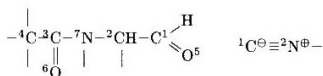
$$\rho = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Wählt man nacheinander alle 10 Verbindungen aus i) als Grundgerüst, so erhält man zusammen 136 Strukturen.

- iii) Die letzte Stufe besteht im Hinzufügen von vier Isocyaniden (wobei die Ladungen als zusätzliche Beschriftung aufzufassen sind, durch die die Valenzen entsprechend geändert werden):



Hierbei lauten die Teilstrukturen des Reaktionsschemas (die der Eindeutigkeit halber etwas größer gewählt werden)



und die Matrix

$$\rho = \begin{pmatrix} -\infty & 1 \\ -\infty & 0 \\ -\infty & 0 \\ -\infty & 0 \\ -\infty & 0 \\ -\infty & 0 \\ -\infty & 0 \end{pmatrix},$$

die auch die Neutralisierung des Stickstoffs mit dem Proton am Kohlenstoff I der zweiten Teilstruktur implizieren möge.

Bindet man diese Bausteine gemäß des Schemas auf alle wesentlich verschiedenen Weisen an die Gerüste, die in Stufe ii berechnet wurden, so erhält man eine Gesamtgröße der Bibliothek von 2080 konstitutionell verschiedenen Verbindungen. Berücksichtigt man noch die Stereoisomerie, die sich etwa mit [127, 128] berechnen läßt, so ergeben sich 8256 Produkte.

Einen (zufallserzeugten) Querschnitt von 15 Molekülen zeigt Abb. 8.4. Als Reaktion im Labor laufen diese Stufen gemeinsam ab; nach Aussagen in [120] ist davon auszugehen, daß ein Großteil der theoretisch (d.h. etwa mit obiger Vorgehensweise) erzeugten Strukturen auch tatsächlich als Produkt im Experiment gebildet wird.

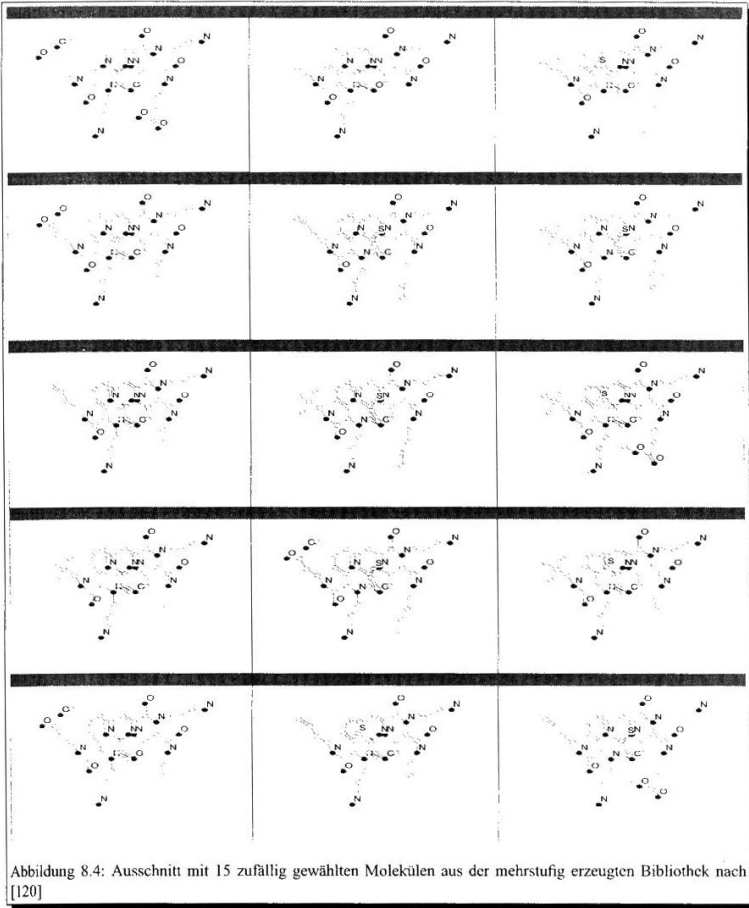


Abbildung 8.4: Ausschnitt mit 15 zufällig gewählten Molekülen aus der mehrstufig erzeugten Bibliothek nach [120]

8.3 Abzählen und Screenen einer Bibliothek

Wir haben in Lemma 6.2.1 festgestellt, daß die wesentlich verschiedenen Möglichkeiten der Belegung einer Gerüststruktur (γ, β) , die über k verschiedene Reaktionsplätzen verfügt, mit n Bausteinen einer Transversale der Bahnen von P_γ auf \underline{n}^k entsprechen. Daher können wir für die Größe dieser Transversalen folgendes Ergebnis festhalten:

8.3.1 Satz. Gegeben sei eine Gerüststruktur (γ, β) , die über k verschiedene Reaktionsplätze verfügt. $\text{Aut}(\gamma, \beta)$ induziere die Permutationsgruppe $P_\gamma \leq S_k$ auf den Plätzen.

Weiterhin sei eine Menge von n verschiedenen Bausteinen gegeben. Dann hat die kombinatorische Bibliothek, die sich mittels des entsprechenden Reaktionsschemas aus der Gerüststruktur und den Bausteinen bilden läßt,

$$|\underline{n}^k // P_\gamma| = \frac{1}{|P_\gamma|} \sum_{\pi \in P_\gamma} n^{c(\pi)}$$

Elemente.

Beweis: Folgt aus 6.2.1 und dem Lemma von Cauchy-Frobenius (s. etwa [67]). □

8.3.2 Beispiel. Wir betrachten wieder das Beispiel aus 8.2.1 (entnommen aus [24]) mit den Gerüsten **XXIV**, **XXV** und **XXVI** und den natürlichen Aminosäuren (siehe Abbildung 8.1 auf Seite 128) als Bausteinen.

- Die Gruppe P_γ für das Cubanderivat (**XXIV**) enthält 24 Elemente bei $k = 4$. Dann folgt aus 8.3.1:

$$|\underline{n}^4 // P_\gamma| = \frac{1}{24}(n^4 + 6n^3 + 11n^2 + 6n)$$

- Bei Xanthen (Struktur **XXV**) ist $k = 4$ und $P_\gamma = \{1, (12)(34)\}$. Wir erhalten mit obiger Formel:

$$|\underline{n}^4 // P_\gamma| = \frac{1}{2}(n^4 + n^2)$$

- Beim Benzoltrisäurechlorid (**XXVI**) ist $k = 3$ und die Gruppe $P_\gamma = \{1, (123), (132)\}$. Für die Anzahlen gilt:

$$|\underline{n}^3 // P_\gamma| = \frac{1}{3}(n^3 + 2n)$$

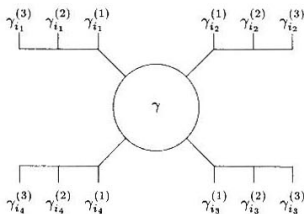
Folgende Tabelle gibt einen Überblick über die Größen der Bibliotheken in Abhängigkeit von der Anzahl der jeweils verwendeten Bausteine:

n	XXIV	XXV	XXVI
1	1	1	1
2	5	10	4
3	15	45	11
4	35	136	24
5	70	325	45
6	126	666	76
7	210	1225	119
8	330	2080	176
9	495	3321	249
10	715	5050	340
11	1001	7381	451
12	1365	10440	584
13	1820	14365	741
14	2380	19306	924
15	3060	25425	1135
16	3876	32896	1376
17	4845	41905	1649
18	5985	52650	1956
19	7315	65341	2299
20	8855	80200	2680

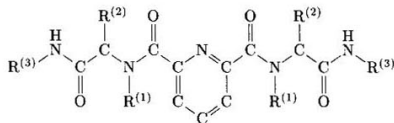
Trotz gleichem k ergeben sich bei **XXIV** aufgrund der hohen Symmetrie deutlich kleinere Bibliotheken als bei **XXV**. ◇

Etwas komplizierter ist die Situation bei mehrstufigen Verfahren wie in 8.2.2. Wir betrachten das Hinzufügen der Bausteine als Gesamtvorgang, so als ob alle Bausteine gleichzeitig angefügt werden. Für die Konstruktion einer Bibliothek ist diese Sichtweise weniger sinnvoll; zur Abzählung ist es indessen notwendig, um die Abhängigkeiten der einzelnen Anknüpfpositionen richtig bewerten zu können.

Damit ergibt sich folgende Skizze:



Bei unserem Pyridin-Gerüst aus [120] ist dies



Es sei wieder k die Anzahl der verschiedenen Anknüpfungspunkte. Ferner sei $\kappa = (\kappa_1, \dots, \kappa_m) \models k$ eine Partition von k und $k_i := \sum_{s=1}^i \kappa_s$ (mit $k_0 := 0$) die kumulierten Einträge der Partition. Die Anknüpfungspunkte seien entsprechend den Bahnen der Automorphismengruppe von γ partitioniert, d.h. wir setzen für P_γ voraus:

$$\forall \pi \in P_\gamma \leq S_k : k_i < r \leq k_{i+1} \Rightarrow k_i < r^\pi \leq k_{i+1} \text{ für } i = 0, \dots, m$$

Wir bezeichnen mit $c_i(\pi)$ die Anzahl der Zyklen von $\pi \in P_\gamma$ auf der Menge $\{k_i, \dots, k_{i+1} - 1\}$.

Die Bausteine seien Elemente der Mengen A_1, \dots, A_m mit $|A_i| =: n_i$. Damit ergibt sich für die Größen der Bibliotheken:

8.3.3 Satz. *Seien Grundgerüst und Bausteine wie oben gegeben. Dann existiert eine Bijektion von der kombinatorischen Bibliothek zu Bahnentransversalen der Operation von P_γ auf den Abbildungen*

$$\Phi := \underline{\mathbf{n}}_1^{\kappa_1} \times \dots \times \underline{\mathbf{n}}_m^{\kappa_m}.$$

Für die Anzahlen der Elemente der Bibliothek gilt:

$$|\Phi // P_\gamma| = \frac{1}{|P_\gamma|} \sum_{\pi \in P_\gamma} \prod_{i=1}^m n_i^{c_i(\pi)}$$

Beweis: Folgt wieder aus 6.2.1 und dem Lemma von Cauchy-Frobenius sowie obigen Definitionen. □

8.3.4 Beispiel. Bei der Bibliothek aus der Multikomponentenreaktion aus [120], die wir in 8.2.2 konstruiert haben, ist $k = 6$, $\kappa = (2, 2, 2)$ und $P_\gamma = \{1, (12)(34)(56)\}$. (Das heißt in obiger Molekülzeichnung, daß jeweils die Positionen für die Reste $R^{(1)}$, $R^{(2)}$ und $R^{(3)}$ untereinander vertauschbar seien.)

Damit gilt:

$$|\Phi // P_\gamma| = \frac{1}{2} (n_1^2 n_2^2 n_3^2 + n_1 n_2 n_3)$$

Die folgende Tabelle gibt Werte zu dieser Formel. Dabei ist n_1 vertikal und n_2 horizontal aufgetragen; der Wert $n_3 = 4$ ist fest. Der Eintrag für $(4, 4)$ gibt uns die bekannte Größe 2080 für jeweils vier Bausteine (s. 8.2.2).

$n_1 \setminus n_2$	1	2	3	4	5	6	7	8	9	10
1	10	36	78	136	210	300	406	528	666	820
2	36	136	300	528	820	1176	1596	2080	2628	3240
3	78	300	666	1176	1830	2628	3570	4656	5886	7260
4	136	528	1176	2080	3240	4656	6328	8256	10440	12880
5	210	820	1830	3240	5050	7260	9870	12880	16290	20100
6	300	1176	2628	4656	7260	10440	14196	18528	23436	28920
7	406	1596	3570	6328	9870	14196	19306	25200	31878	39340
8	528	2080	4656	8256	12880	18528	25200	32896	41616	51360
9	666	2628	5886	10440	16290	23436	31878	41616	52650	64980
10	820	3240	7260	12880	20100	28920	39340	51360	64980	80200

◇

Auch im Fall, daß man bei Einstufungsverfahren nicht jeweils alle Bausteine auf alle möglichen Weisen verteilen, sondern die Anzahlen der einzelnen einschränken will, läßt sich die Größe der Bibliotheken kombinatorisch abzählen. Das Werkzeug dazu haben wir mit der gewichteten Abzählung bereitgestellt.

8.3.5 Satz. Gegeben sei eine Gerüststruktur (γ, β) , die über k verschiedene Reaktionsplätze verfügt. Aut (γ, β) induziere die Permutationsgruppe $P_\gamma \leq S_k$ auf den Plätzen. Weiterhin sei eine Menge von n verschiedenen Bausteinen und eine Verteilung $f \in \mathbf{n}^k$ der Bausteine gegeben.

Dann ist die Anzahl der Elemente der Bibliothek, deren Verteilungen denselben Inhalt $c(f, \cdot)$ haben wie f , gleich dem Koeffizienten des Monoms $\prod_r y_r^{c(f, y_r)}$ im Polynom

$$\frac{1}{|P_\gamma|} \sum_{\pi \in P_\gamma} \prod_{i=1}^k \left(\sum_{r=1}^n y_r^i \right)^{\alpha_i(\pi)}$$

im Polynomring über den Unbestimmten y_1, \dots, y_n .

Beweis: Folgt aus 6.2.1 und der gewichteten Form des Lemmas von Cauchy-Frobenius (s. [67]).

□

8.3.6 Beispiel. Wir betrachten wie oben das Beispiel aus 8.2.1 (entnommen aus [24]), und zwar die Gerüste **XXV** und **XXVI** mit den natürlichen Aminosäuren als Bausteinen.

Dann liefert 8.3.5 folgende Formeln:

- Bei Xanthen (**XXV**) erhalten wir als Polynom

$$\frac{1}{2} ((y_1 + \dots + y_n)^4 + (y_1^2 + \dots + y_n^2)^2)$$

Für den Fall $n = 4$ gibt das die Summe

$$\begin{aligned} & 6 y_1 y_2^2 y_4 + 6 y_1 y_2 y_4^2 + 6 y_1 y_2^2 y_3 + y_1^4 + y_2^4 + y_3^4 + y_4^4 + 6 y_1 y_3^2 y_4 + \\ & 6 y_1^2 y_2 y_3 + 6 y_1^2 y_2 y_4 + 6 y_1^2 y_3 y_4 + 6 y_2^2 y_3 y_4 + 2 y_1 y_3^3 + 2 y_1 y_4^3 + \\ & 2 y_1 y_4^3 + 2 y_1^3 y_2 + 2 y_1^3 y_3 + 2 y_1^3 y_4 + 4 y_1^2 y_2^2 + 4 y_1^2 y_3^2 + \\ & 4 y_1^2 y_4^2 + 2 y_2 y_3^3 + 2 y_2 y_4^3 + 2 y_2^2 y_3 + 2 y_2^2 y_4 + 4 y_2 y_3^2 y_4 + \\ & 4 y_2 y_4^2 + 2 y_3 y_4^3 + 2 y_3^2 y_4 + 4 y_3^2 y_4^2 + 6 y_1 y_2 y_3^2 + 6 y_1 y_3 y_4^2 + \\ & 12 y_1 y_2 y_3 y_4 + 6 y_2 y_3^2 y_4 + 6 y_2 y_3 y_4^2 \end{aligned}$$

Will man beispielsweise nur die Bibliothekselemente, die den ersten Baustein genau einmal enthalten, so entspricht dies den Summanden

$$\begin{aligned} & 6 y_1 y_2^2 y_4 + 6 y_1 y_2 y_4^2 + 6 y_1 y_2^2 y_3 + 6 y_1 y_3^2 y_4 + 2 y_1 y_4^3 + \\ & 2 y_1 y_3^3 + 2 y_1 y_4^3 + 6 y_1 y_2 y_3^2 + 6 y_1 y_3 y_4^2 + 12 y_1 y_2 y_3 y_4. \end{aligned}$$

Es gibt also genau $6 + 6 + 6 + 6 + 2 + 2 + 2 + 6 + 6 + 12 = 54$ derartige Elemente.

- Beim Benzoltrisäurechlorid (**XXVI**) gelangen wir zum Polynom:

$$\frac{1}{3} ((y_1 + \dots + y_n)^3 + 2(y_1^3 + \dots + y_n^3))$$

Bei drei Bausteinen etwa lautet dies dann:

$$y_1^3 + y_1^2 y_2 + y_1^2 y_3 + y_1 y_2^2 + 2 y_1 y_2 y_3 + y_1 y_3^2 + y_2^3 + y_2^2 y_3 + y_2 y_3^2 + y_3^3$$

◇

Zum Abschluß unserer Betrachtungen zur Simulation kombinatorischer Bibliotheken wollen wir noch den dritten Schritt untersuchen, den wir in 8.1.1 zur Vorgehensweise in der kombinatorischen Chemie gezählt haben: *Das Screening der Bibliothekselemente auf die gewünschte Wirkung.*

Dieser Aspekt ist der schwierigste. Nach dem heutigen Stand der Forschung ist ein rein virtuelles Screening, d.h. nur im Computer und ohne Experiment, aufgrund der dazu notwendigen Rechenzeiten und Rechnerkapazitäten für die Praxis nicht relevant.

Viele biologische Mechanismen, die man kennt, beruhen auf der Ligand-Rezeptor-Wechselwirkung (s. etwa [78, 79] und 7.3.2). Diese Wechselwirkung hängt qualitativ und quantitativ entscheidend von der räumlichen Struktur der beiden Partner ab.

Eine mögliche Strategie ist, mit Hilfe quantitativer Struktur-Wirkungs-Beziehungen (QSAR) etwas Licht in das Dunkel zu bringen (s.a. 7.1). Dabei verwendet man i.a. eine Testreihe von Substanzen, um einen (oder mehrere) Wirkungsparameter zu bestimmen. Aus dieser Testreihe kann man dann statistisch eine Korrelation mit berechenbaren Struktureigenschaften herleiten, die man anschließend zur Extrapolation auf eine größere Menge von Verbindungen (wie etwa in einer kombinatorischen Bibliothek) verwenden kann (vgl. etwa [5, 6, 72, 130]). In unserem Fall heißt das, daß mittels QSAR die Größe der Bibliothek u. U. eingeschränkt werden kann und die Simulationen, die wir in diesem Kapitel beschrieben haben, eine akkuratere Vorbereitung für die Synthese ermöglichen.

Höher entwickelte Techniken verwenden *3D QSAR* [77, 79], darunter etwa die *vergleichende Molekülfeldanalyse (CoMFA)* [33, 77]. Dabei werden die Bibliothekselemente zunächst zu dreidimensionalen Strukturen konvertiert unter Verwendung eines entsprechenden Computerverfahrens [1, 34, 67, 102, 109] (vgl. Abschn. 5.4). Die entscheidende Anforderung, die ein solches Verfahren erfüllen muß, ist, daß die berechnete Konformation des Liganden zur Bindestelle des Rezeptors passen muß. Denn die üblichen Softwarepakete erzeugen meist nur Konformationen der Strukturen im Vakuum oder in Lösung; diese können sich aber stark von der Gestalt des an das Protein gebundenen Liganden unterscheiden. Anschließend wird bei der CoMFA eine Überlagerung des Liganden und der Bindestelle berechnet entsprechend den physikalischen Feldern (sterisch, elektrostatisch etc.) der Moleküle, um die Wirksamkeit des Liganden abschätzen zu können.

Somit bedeutet virtuelles Screening und damit *virtuelle kombinatorische Chemie* heute noch keine Konkurrenz zu Screening-Robotern, die einen Durchsatz von mehr als 10.000 Substanzen pro Tag haben. Angesichts der vielfältigen Entwicklungen kann es indessen schon morgen die Realität darstellen.

Kapitel 9 Verbesserung der Eingabe eines Isomerengenerators

9.1 Makroatome und Goodlist

In diesem Kapitel werden wir untersuchen, wie sich unser Verschmelzungsverfahren 5.2.5 im Zusammenhang mit einem klassischen Isomerengenerator einsetzen läßt.

Sowohl aus der Graphentheorie als auch aus der Chemie ist das Problem bekannt, zu einer Menge von Punkten mit gegebener Eckengradfolge, d.h. gegebener Valenz, alle Graphen bzw. Konstitutionsisomere (vgl. 5.4.1) zu konstruieren. Verschiedene Ansätze wurden dazu vorgestellt [25, 26, 36, 41, 50, 68, 89], doch erst mit ausgefeilteren mathematischen Methoden [53, 55, 96] konnten Softwaresysteme entwickelt werden, die eine schnelle, vollständige und redundanzfreie Lösung ermöglichen. Eines davon ist **MOLGEN**, vom dem oben (S. 59 und 75) bereits die Rede war.

MOLGEN erzeugt alle molekularen Graphen zu einer gegebenen Bruttoformel, vorgeschriebenen oder verbotenen Teilstrukturen und weiteren Nebenbedingungen.

Die verbotenen Teilstrukturen werden in der sog. *Badlist* zusammengefaßt, die wie ein Filter angewendet wird: Der Generator überprüft jede erzeugte Struktur, ob sie eine der verbotenen Teilstrukturen enthält, und verwirft sie gegebenenfalls.

Die vorgeschriebenen Teilstrukturen sind in zwei Gruppen aufgeteilt:

- *Makroatome*: Diese sind Teilstrukturen, die untereinander nicht überlappen dürfen. Sie werden vom Generator als fiktive Atome gehandhabt, was zu einer Verkleinerung der Bruttoformel und damit zu einer Effizienzsteigerung führt.
- *Goodlist*: Diese Teilstrukturen dürfen sich untereinander überlappen. Der Preis für diese Freiheit ist, daß ihr Vorkommen – ähnlich wie bei der Badlist – erst nach der Erzeugung jedes einzelnen Isomers überprüft werden kann.

Bei umfangreicheren Bruttoformeln kann es vorkommen, daß Millionen von Strukturen generiert werden müssen, aber nur einige wenige den Test auf die Goodlistbedingungen bestehen.

9.1.1 Bemerkung. Nach der Erzeugung von Graphen mit Makroatomen müssen diese auf ihre tatsächliche Größe expandiert werden. Dabei ist zusätzlich auf die Symmetrien unter den Anknüpfstellen und innerhalb der Makroatome zu achten. Es zeigt sich, daß dieser Vorgang eine weitere Anwendung des Homomorphieprinzips 1.3.1 ist. Da in den bisherigen Beschreibungen [14, 69, 131] dieser Aspekt weitgehend unbeachtet blieb, wollen wir kurz einen Blick darauf werfen.

Gegeben sei ein generierter Graph γ , der ein Makroatom und p weitere Ecken enthalte. Das Makroatom selbst bestehe aus k Ecken. Dann sind die zu betrachtenden Mengen

$$\Omega_1 = m^{\underline{\mathbf{p}+\mathbf{k}}^{[2]}} \quad \text{und} \quad \Omega_2 = m^{\underline{\mathbf{p}+1}^{[2]}}.$$

Die Abbildung

$$\phi: \underline{\mathbf{p}+\mathbf{k}} \rightarrow \underline{\mathbf{p}+1}, i \mapsto \begin{cases} i & \text{falls } i \leq p \\ p+1 & \text{sonst} \end{cases}$$

wird zur Definition des Epimorphismus zwischen beiden Mengen verwendet:

$$\sigma_\Omega: \Omega_1 \rightarrow \Omega_2, \gamma \mapsto \tilde{\gamma} \quad \text{mit} \quad \tilde{\gamma}(\{i, j\}) = \gamma(\{\phi(i), \phi(j)\}).$$

Da die Makrostruktur erhalten bleiben soll, müssen wir als operierende Gruppe auf der einen Seite $G_1 \simeq S_p \times Aut_M$ verwenden, wobei Aut_M die Automorphismengruppe der Makrostruktur sei. Die Gruppe G_1 ist demnach eine Untergruppe der Younggruppe $S(\mathbf{p} + \mathbf{k}^\lambda)$ mit $\lambda = (p, k)$.

Auf der anderen Seite operiert $G_2 \sim \overline{S_p} \times S_1$. Damit ergibt sich der Epimorphismus $\sigma_G : G_1 \rightarrow G_2$ als die natürliche Einschränkung, und auch die Verträglichkeitsbedingung ist klar.

Folglich liefert uns Satz 1.3.1, daß bei einem $\gamma_2 \in \Omega_2$ und zwei expandierten Strukturen $\gamma_1, \gamma'_1 \in \Omega_1$ mit $\gamma_1^{g_1} = \gamma'_1$ dieses $g_1 \in G_1$ nicht beliebig ist, sondern aus dem Urbild des Zentralisators von γ_2 stammt:

$$g_1 \in \sigma_G^{-1}(C_{G_2}(\gamma_2)) \simeq N_{Aut(\gamma_2)}(\gamma_2 \setminus \{p+1\}) / C_{Aut(\gamma_2)}(\gamma_2 \setminus \{p+1\}) \times Aut_M$$

Die Bestimmung des Urbildes $\sigma_G^{-1}(\gamma_2)$ erfordert dann die Berechnung der Doppelnebenklassen $A \setminus S_r / B$, wobei die Makrostruktur r freie Valenzen, also Anknüpfstellen nach außen, habe, A die Teilautomorphismengruppe dieser Stellen im erzeugten Graphen und B die Teilautomorphismengruppe der freien Valenzen im Makroatom seien.

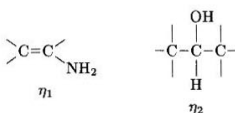
9.2 Vermeidung der Goodlist-Filterung

Wir wollen nun diskutieren, wie die in dieser Arbeit vorgestellten Algorithmen für die Verbesserung der Eingabe für den Isomergenerator genutzt werden können. Die Grundidee ist, mittels unseres Verfahrens 5.2.5 alle Überlappungen der verschiedenen Goodlist-Einträge zu konstruieren. Diese können dann jeweils als Makroatome verwendet werden, wodurch der Aufwand für das komplette Erzeugen mit nachträglicher Filterung wegfällt.

Da auf diese Weise nicht sichergestellt ist, daß die verschiedenen Generatorläufe nur nichtisomorphe Ergebnisse liefern, wird auf die Gesamtmenge der Resultate abschließend noch ein Isomorphietest zur Eliminierung doppelter Strukturen angewendet.

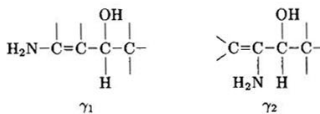
Diese Vorgehensweise wollen wir an einem einfachen Beispiel veranschaulichen:

9.2.1 Beispiel. Gegeben sei die Bruttoformel $C_6H_{11}NO$. Dazu erzeugt **MOLGEN** 13982 Isomere. Als Teilstrukturen seien vorgeschrieben (die leeren Plätze seien freie Valenzen):



Trägt man diese in die Goodlist ein, so ergeben sich 71 Lösungen.

Die Anwendung des Verschmelzungsalgorithmus 5.2.5 auf η_1 und η_2 unter Berücksichtigung der Tatsache, daß es sich um Teilstrukturen handelt (vgl. 4.1.11.ii und 5.2.2), liefert zwei Lösungen



Verwendet man diese sowie das Paar $\{\eta_1, \eta_2\}$ als Makroatome, so erhält man folgende Anzahlen von Lösungen:

Makroatome	Anz. Isomere
γ_1	22
γ_2	14
$\{\eta_1, \eta_2\}$	32

Unter diesen insgesamt 68 Strukturen sind 63 nichtisomorph. Dies steht in offensichtlichem Gegensatz zu den 71 Strukturen, die sich bei reiner Goodlist-Behandlung ergeben. ◇

Das Beispiel zeigt, daß die Ersetzung von Goodlist-Einträgen durch Makroatome nicht so einfach ist, wie wir oben annehmen. Das Problem sind Bindungen, die die zwei Teilstrukturen untereinander ausbilden können. Würden die beiden zu einem Makroatom verschmolzen, so sind nur noch Bindungen nach außen, nicht aber zwischen den Atomen der Makrostruktur zulässig, wie dies in unserem Fall nötig ist.

Wir brauchen also ein weiteres Verfahren, das redundanzfrei alle Verbindungen der freien Valenzen innerhalb eines Makroatoms erzeugt. Dieses Problem ist unserer Konstruktionsaufgabe 3.1.1 so ähnlich, daß eine leichte Modifikation des entsprechenden Algorithmus zu einer Lösung führt.

9.2.2 Konstruktionsproblem. Gegeben sei ein Multigraph $\gamma \in m^{\mathbb{P}^2}$ als Verschmelzung zweier Graphen $\eta_1 \in m^{\mathbb{P}^1}$ und $\eta_2 \in m^{\mathbb{P}^1}$. Zusätzlich habe jede Ecke i aus γ noch freie Valenzen, deren Anzahl als $FV(i) \in \{0, \dots, val_i - 1\}$ gegeben sei. Bestimme alle nichtisomorphen Graphen, die sich durch Hinzufügen einer Kantenmenge ϵ ergeben, für die gilt:

i) $\epsilon : \mathbb{P} \times \mathbb{P} \rightarrow \{0, 1, \dots, m\}$,

ii) $d_i + \epsilon(i, j) \leq val_i$ und $d_j + \epsilon(i, j) \leq val_j$ für alle $(i, j) \in \mathbb{P} \times \mathbb{P}$.

iii) $\epsilon(i, j) \leq \min(FV(i), FV(j))$,

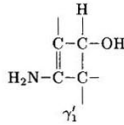
iv) $\epsilon(i, j) = 0$ falls $\gamma(\{i, j\}) \neq 0$ oder die Ecken i und j beide aus η_1 bzw. η_2 stammen.

Zur Berechnung wandeln wir die freien Valenzen von γ wieder wie in 3.1.2 in fiktive Ecken um. Dann können wir die Lösungen ordnungstreu erzeugen. Da wir hier nur ziemlich kleine Fälle betrachten, verzichten wir auf eine surjektive Auflösung nach dem Homomorphieprinzip wie in 3.2.1, halten aber ausdrücklich fest, daß eine solche Aufspaltung auch hier möglich ist.

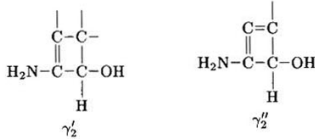
Das Verfahren zu Erzeugung aller Graphen gemäß 9.2.2 verläuft ganz analog zu 3.3.2, wobei als Gruppe die von $Aut(\tilde{\gamma})$ induzierte Teilautomorphismengruppe auf den Zeilen und Spalten von $\tilde{\epsilon}$ operiert. Der Test, ob an einer Stelle ein Eintrag gesetzt werden darf (vgl. 3.3.1), beinhaltet zusätzlich die Einschränkungen aus 9.2.2.i-iv. (Auf eine detaillierte Beschreibung wollen wir hier aus Platzgründen verzichten.)

9.2.3 Beispiel. Wir betrachten wieder die Strukturen aus Beispiel 9.2.1.

- i) Bei γ_1 ergibt sich nur eine Möglichkeit für eine zusätzliche Kante, da das zweite Kohlenstoffatom der Doppelbindung beiden Ausgangsgraphen gemeinsam ist.



- ii) Für γ_2 erhalten wir zwei Lösungen:



Für die Anzahlen an generierten Lösungen gilt dann:

Makroatom	Anz. Isomere
γ'_1	5
γ'_2	5
γ''_2	0

γ'' führt zu keinen Isomeren, da sich die restlichen Atome nicht mehr geeignet daran binden können. Insgesamt erhalten wir aus allen Makroatomen $\{\eta_1, \eta_2\}$, γ_1 , γ'_1 , γ_2 und γ'_2 eine Gesamtmenge von 78 Isomeren, unter denen genau 71 nichtisomorph sind. Ein genauer Vergleich zeigt, daß es sich dabei um dieselben Strukturen handelt, die man auch bei der Generierung mit zwei Goodlist-Einträgen erhält. In Abb. 9.1 sind die ersten 40 der 71 Ergebnisstrukturen abgedruckt.

Mit unserer Vorgehensweise müssen also statt 13982 nur 78 Moleküle generiert werden – ein Gegensatz, der die Vorteile dieses Ansatzes deutlich werden läßt.

◇

Prinzipiell ist auch eine Ausweitung dieser Methoden auf drei und mehr Goodlist-Einträge möglich. Eine genauere Diskussion würde aber den Rahmen wieder einmal übersteigen. Bereits durch zwei größere Strukturen läßt sich aber auch oft die Zahl der zu untersuchenden Isomere in überschaubare Größenordnungen bringen.

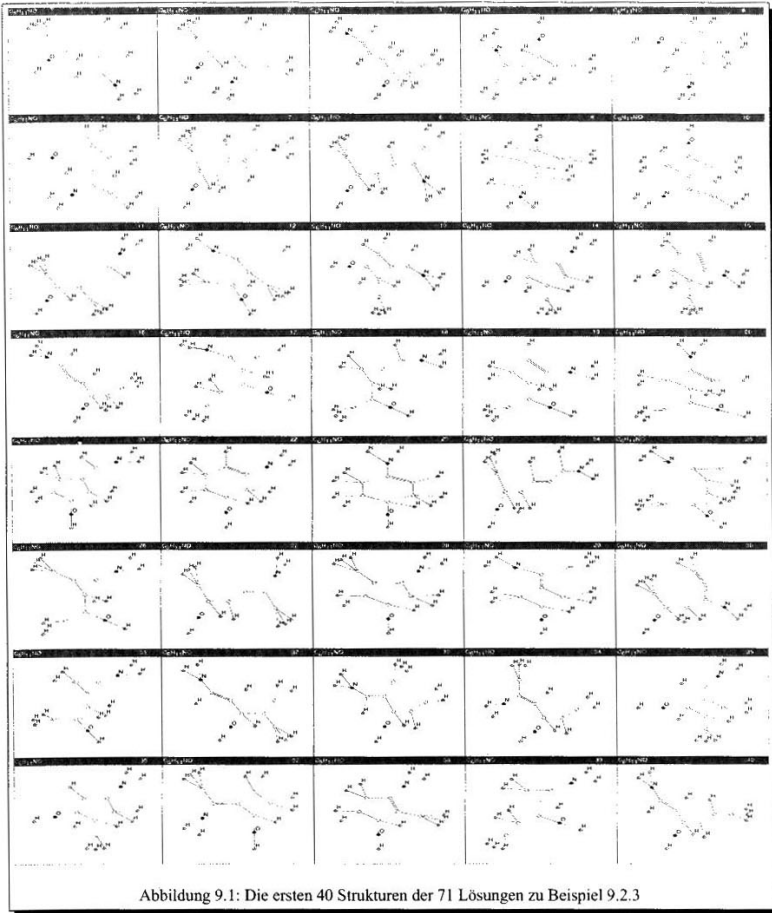


Abbildung 9.1: Die ersten 40 Strukturen der 71 Lösungen zu Beispiel 9.2.3

Literaturverzeichnis

- [1] ALLINGER, N.L. MM2. A Hydrocarbon Force Field Utilizing V_1 and V_2 Torsional Terms. *J. Am. Chem. Soc.*, **99**, S. 8127–8134, 1977.
- [2] AMBLER, A.P., H.G. BARROW, C.M. BROWN, R.M. BURSTALL UND R.J. POPPLESTONE. A versatile computer-controlled assembly system. In *Proc. of Third Intern. Joint-Conf. Artificial Intelligence*, S. 298–307, Stanford, CA, 1973.
- [3] BALABAN, A.T. Highly discriminating distance-based topological index. *Chem. Phys. Lett.*, **89**, S. 399–404, 1982.
- [4] BARROW, H.G. UND R.M. BURSTALL. Subgraph isomorphism, matching relational structures and maximal cliques. *Inf. Proc. Lett.*, **4**, S. 83–84, 1976.
- [5] BASAK, S.C. Use of molecular complexity indices in predictive pharmacology and toxicology: a QSAR approach. *Med. Sci. Res.*, **15**, S. 605–609, 1987.
- [6] BASAK, S.C., S. BERTELSEN UND G.D. GRUNWALD. Application of graph-theoretical parameters in quantifying molecular similarity and structure-activity studies. *J. Chem. Inf. Comput. Sci.*, **34**, S. 270–276, 1994.
- [7] BASAK, S.C., C.M. FRANE, M.E. ROSEN UND V.R. MAGNUSON. Molecular topology and acute toxicity: a QSAR study of monoketones. *Med. Sci. Res.*, **15**, S. 887–888, 1987.
- [8] BASAK, S.C. UND G.D. GRUNWALD. Predicting genotoxicity of chemicals using nonempirical parameters. In *Proceeding of the XVI. International Cancer Congress*, S. 413–416, 1995.
- [9] BASAK, S.C., V.R. MAGNUSON, G.J. NIEMI UND R.R. REGAL. Determining structural similarity of chemicals using graph-theoretic indices. *Discr. Appl. Math.*, **19**, S. 17–44, 1988.
- [10] BAYADA, D.M., R.W. SIMPSON, A.P. JOHNSON UND C. LAURENÇO. An algorithm for the multiple common subgraph problem. *J. Chem. Inf. Comput. Sci.*, **32**, S. 680–682, 1992.
- [11] BENECKE, C. *Objektorientierte Darstellung und Algorithmen zur Klassifizierung unscharfer endlicher Strukturen*. Dissertation, Universität Bayreuth, 1997. (Eingereicht).
- [12] BENECKE, C., R. GRUND, R. HOHBERGER, A. KERBER, R. LAUE UND T. WIELAND. Die Veranschaulichung von Molekülen im Chemieunterricht mit Hilfe des Programmsystems MOLGEN⁺. *MNU - Der mathem.-naturw. Unterr.*, **48**, S. 156–163, 1995.
- [13] BENECKE, C., R. GRUND, R. HOHBERGER, A. KERBER, R. LAUE UND T. WIELAND. MOLGEN, a Computer Algebra System for the Generation of Molecular Graphs. In FLEISCHER, J., J. GRABMEIER, F.W. HEHL UND W. KÜCHLIN, Hrsg., *Computer Algebra in Science and Engineering*, S. 260–272. World Scientific, Singapore, 1995.
- [14] BENECKE, C., R. GRUND, R. HOHBERGER, A. KERBER, R. LAUE UND T. WIELAND. MOLGEN+, a generator of connectivity isomers and stereoisomers for molecular structure elucidation. *Anal. Chim. Act.*, **314**, S. 141–147, 1995.
- [15] BENECKE, C., A. KERBER UND R. LAUE. Canonical Numbering of 3D-Structures. In 2. *Electronic Conference on Computational Chemistry (ECCC2)*, 1995.
- [16] BIEGHOLDT, J. *Computerunterstützte Berechnung von Multigraphen mittels Homomorphieprinzip*. Diplomarbeit, Universität Bayreuth, 1995.
- [17] BIGGS, N.L., K.E. LLOYD UND R.J. WILSON. *Graph theory 1736-1936*. Clarendon Press, 1977.
- [18] BONCHEV, D. *Information Theoretic Indices for Characterization of Chemical Structures*. Research Studies Press, Chichester, 1983.
- [19] BONCHEV, D. UND N. TRINAJSTIĆ. Information theory, distance matrix and molecular branching. *J. Chem. Phys.*, **67**, S. 4517–4533, 1977.
- [20] BROCAS, J. Double cosets and enumeration of permutational isomers of fixed symmetry. *J. Am. Chem. Soc.*, **108**, S. 1135–1145, 1986.
- [21] BROCAS, J., M. GIELEN UND R. WILLEM. *The permutational approach to dynamic stereochemistry*. McGraw-Hill, New York, NY, 1983.

- [22] BROWN, C.A., I. FINKELSTEIN UND P.W. PURDOM. A new Base Change Algorithm for Permutation Groups. *SIAM J. Comput.*, **18**, S. 1037–1047, 1989.
- [23] BROWN, H., L. HJELMELAND UND L. MASINTER. Constructive graph labeling using double cosets. *Discr. Math.*, **7**, S. 1–30, 1974.
- [24] CARELLI, T., E.A. WINTNER, A.J. SUTHERLAND, J. REBEK, JR., Y.M. DUNAYEVSKIY UND P. VOUROIS. New promise in combinatorial chemistry: synthesis, characterization, and screening of small-molecule libraries in solution. *Chem. & Biol.*, **2**, S. 171–183, 1995.
- [25] CARIHART, R.E., D.H. SMITH, H. BROWN UND C. DJERASSI. An Approach to Computer-Assisted Elucidation of Molecular Structure. *J. Am. Chem. Soc.*, **97**, S. 5755–5762, 1975.
- [26] CARIHART, R.E., D.H. SMITH, N.A. GRAY, J.G. NOURSE UND C. DJERASSI. GENOA: A Computer Program for Structure Elucidation Utilizing Overlapping and Alternative Substructures. *J. Org. Chem.*, **46**, S. 1708–1718, 1981.
- [27] CHEN, L. UND W. ROBIEN. MCSS: A new algorithm for perception of maximal common substructures and its application to NMR spectral studies. 1. The algorithm. *J. Chem. Inf. Comput. Sci.*, **32**, S. 501–506, 1992.
- [28] CHEN, L. UND W. ROBIEN. MCSS: A new algorithm for perception of maximal common substructures and its application to NMR spectral studies. 2. Applications. *J. Chem. Inf. Comput. Sci.*, **32**, S. 507–510, 1992.
- [29] CLARK, D.E., M.A. FIRTH UND C.W. MURRAY. MOLMAKER: De novo generation of 3D databases for use in drug design. *J. Chem. Inf. Comput. Sci.*, **36**, S. 137–145, 1996.
- [30] COLBOURN, C.J. UND R.C. READ. Orderly algorithms for generating restricted classes of graphs. *J. Graph Theo.*, **3**, S. 187–195, 1979.
- [31] CONE, M.M., R. VENKATARAGHAVAN UND F.W. MCLAFFERTY. Molecular structure comparison program for the identification of common substructures. *J. Am. Chem. Soc.*, **99**, S. 7668–7671, 1977.
- [32] COREY, E.J., W.T. WIPKE, R.D. CRAMER UND W.J. HOWE. Techniques for perception by a computer of synthetically significant structural features in complex molecules. *J. Am. Chem. Soc.*, **94**, S. 431–439, 1972.
- [33] CRAMER III, R.D., D.E. PATTERSON UND A. VITTORIA. *J. Amer. Chem. Soc.*, **110**, S. 5959–5967, 1988.
- [34] CRIPPEN, G.M. *Distance Geometry and Conformational Calculations*. J. Wiley, New York, NY, 1981.
- [35] DUGUNDJI, J. UND I.K. UGLI. An algebraic model of conformational chemistry as a basis for chemical computer programs. *Top. Curr. Chem.*, **39**, S. 19–64, 1973.
- [36] FARADZHEV, I.A. Generation of non-isomorphic graphs with given partition of vertex degrees. In FARADZHEV, I.A., Hrsg., *Algorithmic Studies in Combinatorics*, S. 11–19. Nauka, Moskau, 1978. (In Russisch).
- [37] FISCHER, G. UND R. SACHER. *Einführung in die Algebra*. Teubner, Stuttgart, 1983.
- [38] FLETCHER, R. UND C.M. REEVES. Function minimization by conjugate gradients. *Comp. J.*, **7**, S. 149–153, 1968.
- [39] FLOYD, R.W. Algorithm 97, Shortest path. *Comm. Assoc. Comp. Mach.*, **5**, S. 345, 1962.
- [40] FROBEL, K. UND T. KRÄMER. Trendbericht: Kombinatorische Chemie. *Nachr. Chem. Tech. Lab.*, **44**, S. 160–162, 1996.
- [41] FUNATSU, K., N. MIYABAYASHI UND S. SASAKI. Further Development of Structure Generation in the Automated Structure Elucidation System CHEMICS. *J. Chem. Inf. Comput. Sci.*, **28**, S. 18–28, 1988.
- [42] GALLOP, M.A., R.W. BARRETT, W.J. DOWER, S.P.A. FODOR UND E.M. GORDON. Applications of combinatorial technologies to drug discovery. 1. Background and peptide combinatorial libraries. *J. Med. Chem.*, **37**, S. 1233–1251, 1994.
- [43] GÁLVEZ, J., R. GARCÍA, T. SALABERT UND R. SOLER. Charge Indexes. New Topological Descriptors. *J. Chem. Inf. Comput. Sci.*, **34**, S. 520–525, 1994.
- [44] GÁLVEZ, J., R. GARCÍA-DOMENECH, J.V. DE JULIÁN-ORTIZ UND R. SOLER. Topological approach to drug design. *J. Chem. Inf. Comput. Sci.*, **35**, S. 272–284, 1995.
- [45] GAREY, M.R. UND D.S. JOHNSON. *Computers and intractability: A guide to the theory of NP-completeness*. Freeman & Co., New York, NY, 1979.
- [46] GLEN, R.C. UND A.W.R. PAYNE. A genetic algorithm for the automated generation of molecules within constraints. *J. Comp.-Aid. Mol. Des.*, **9**, S. 181–202, 1995.
- [47] GOLFENDER, V.E. UND A.B. ROZENBLIT. *Logical and combinatorial algorithms for drug design*. Research Studies Press, Letchworth, Herts., 1983.
- [48] GORDEEVA, E.V., M.S. MOLCHANOVA UND N.S. ZEFIROV. General methodology and computer programs for the exhaustive restoring of chemical structures by molecular connectivity indexes. Solution of the inverse problem in QSAR/QSPR. *Tetrahedron Comput. Meth.*, **3**(6B), S. 389–415, 1990.

- [49] GORDON, E.M., R.W. BARRETT, W.J. DOWER, S.P.A. FODOR UND M.A. GALLOP. Applications of combinatorial technologies to drug discovery. 2. Combinatorial organic synthesis, library screening strategies, and future directions. *J. Med. Chem.*, **37**, S. 1385–1401, 1994.
- [50] GRAY, N.A. *Computer-assisted Structure Elucidation*. Wiley-Interscience, New York, NY, 1986.
- [51] GRUND, R. *Computerunterstützte Berechnung von speziellen Doppelnebenklassentransversalen und deren Anwendung auf die konstruktive Kombinatorik*. Diplomarbeit, Universität Bayreuth, 1989.
- [52] GRUND, R. Konstruktion schlichter Graphen mit gegebener Gradpartition. *Bayreuther Mathem. Schr.*, **44**, S. 73–104, 1993.
- [53] GRUND, R. Konstruktion molekularer Graphen mit gegebenen Hybridisierungen und überlappungsfreien Fragmenten. *Bayreuther Mathem. Schr.*, **49**, S. 1–113, 1995.
- [54] GRUND, R., A. KERBER UND R. LAUF. MOLGEN, ein Computeralgebra-System für die Konstruktion molekularer Graphen. *MATCH*, **27**, S. 87–131, 1992.
- [55] GRUND, R., A. KERBER UND R. LAUE. Construction of discrete structures, especially isomers. *Discr. Appl. Math.*, **67**, S. 115–126, 1996.
- [56] GRÜNER, T. *Ein neuer Ansatz: zur rekursiven Erzeugung von schichten Graphen*. Diplomarbeit, Universität Bayreuth, 1995.
- [57] GRÜNER, T., R. LAUF UND M. MERINGER. Applications for group actions applied to graph generation. In FINKELSTEIN, L. UND C. KANTOR, Hrsg., *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Providence, RI, 1995.
- [58] HAGER, R., A. KERBER, R. LAUE, D. MOSER UND W. WEBER. Construction of orbit representatives. *Bayreuther Mathem. Schr.*, **35**, S. 157–169, 1991.
- [59] HALL, L.H., L.B. KIER UND J.W. FRAZER. Design of molecules from quantitative structure-activity relationship models. 2. Derivation and proof of information transfer relating equations. *J. Chem. Inf. Comput. Sci.*, **33**, S. 148–152, 1993.
- [60] HEISTERBERG, D.J. The FITEST program. (Unveröffentlichte Ergebnisse), 1990.
- [61] HENNEBERG, D. UND B. WEIMANN. *MassLib, Evaluation of low resolution mass spectra series*. Max-Planck-Institut für Kohlenforschung, Mülheim/Ruhr, 1992. Version 7.2.
- [62] HOFACK, J. UND P.J. DE CLERCQ. The SCA program: An easy way for the conformational evaluation of polycyclic molecules. *Tetrahedron*, **44**, S. 6667–6676, 1988.
- [63] JERRUM, M. A compact representation for permutation groups. *J. Alg.*, **7**, S. 60–78, 1986.
- [64] JOHNSON, M.A., NAIM, M., V. NICHOLSON UND C.C. TSAI. Unique mathematical features of the substructure metric approach to quantitative molecular similarity analysis. In KING, R.B. UND D.H. ROUVRAY, Hrsg., *Graph Theory and Topology in Chemistry*, S. 219–225. Elsevier Science Pub., Amsterdam, 1987.
- [65] JOHNSON, M.A. UND G.M. MAGGIORA, Hrsg. *Concepts and Applications of Molecular Similarity*, New York, NY, 1990. J. Wiley & Sons.
- [66] KEARSLEY, S.K. An algorithm for the simultaneous superposition of a structural series. *J. Comput. Chem.*, **11**, S. 1187–1192, 1990.
- [67] KERBER, A. *Algebraic Combinatorics Via Finite Group Actions*. BI-Wissenschaftsverlag, Mannheim, Wien, Zürich, 1991.
- [68] KERBER, A., Hrsg. *MATCH (Vol. 27): Generatoren für molekulare Graphen. Nomenklaturfragen*. 1992.
- [69] KERBER, A. UND R. LAUE. Group actions, double cosets, and homomorphisms: unifying concepts for the constructive theory of discrete structures. 1994.
- [70] KERBER, A., R. LAUE UND D. MOSER. Ein Strukturgenerator für molekulare Graphen. *Anal. Chim. Acta*, **235**, S. 221–228, 1990.
- [71] KERBER, A., R. LAUF UND T. WIFLAND. Erkennung, Beschreibung und Visualisierung molekularer Strukturen. In *Proceedings des Statusseminars der anwendungsorientierten Verbundprojekte auf dem Gebiet der Mathematik mit Förderung durch das BMBF*. Springer Verlag, Heidelberg, Berlin, 1996. (In Druck).
- [72] KIER, L.B. UND L.H. HALL. *Molecular Connectivity in Structure-Activity Analysis*. Research Studies Press, Chichester, 1986.
- [73] KIER, L.B. UND L.H. HALL. The generation of molecular structures from a graph-based QSAR equation. *Quant. Struct.-Act. Relat.*, **12**, S. 383–388, 1993.
- [74] KIER, L.B., L.H. HALL UND J.W. FRAZER. Design of molecules from quantitative structure-activity relationship models. 1. Information transfer between path and vertex degree counts. *J. Chem. Inf. Comput. Sci.*, **33**, S. 143–147, 1993.

- [75] KLEIN, D.J. UND A.H. COWLEY. Permutational isomerism. *J. Am. Chem. Soc.*, **97**, S. 1633–1640, 1975.
- [76] KUBINYI, H. *J. Med. Chem.*, **20**, S. 625, 1977.
- [77] KUBINYI, H., Hrsg. *3D QSAR in Drug Design. Theory, Methods and Applications*, Leiden, 1993. ESCOM Science Publishers.
- [78] KUBINYI, H. Der Schlüssel zum Schloß. I. Grundlagen der Arzneimittelwirkung. *Pharmazie in unserer Zeit*, **23**(3), S. 158–168, 1994.
- [79] KUBINYI, H. Der Schlüssel zum Schloß. II. Hansch-Analyse, 3D-QSAR und De novo-Design. *Pharmazie in unserer Zeit*, **23**(5), S. 281–290, 1994.
- [80] KVASNIČKA, V. UND J. POSPICAL. Maximal common subgraphs of molecular graphs. *Rep. Mol. Theo.*, **1**, S. 99–106, 1990.
- [81] LANG, E. *Datenstrukturen und Algorithmen für Permutationsgruppen*. Diplomarbeit, Universität Bayreuth, 1992.
- [82] LATSCHA, H.P. UND H.A. KLEIN. *Organische Chemie - Basiswissen II*. Springer Verlag, Berlin, Heidelberg, 2., völlig neu bearb. Aufl., 1990.
- [83] LAUE, R. Abbildungen und Algorithmen. In *Séminaire Lotharingien de Combinatoire*, S. 115–132, 14e Session 1986.
- [84] LAUF, R. Construction of combinatorial objects – a tutorial. *Bayreuther Mathem. Schr.*, **43**, S. 53–96, 1993.
- [85] LAUF, R. Construction of combinatorial objects. (Erweiterte Fassung von [84]), 1996.
- [86] LEVI, G. A note on the derivation of the maximal common subgraphs of two directed or undirected graphs. *Calcolo*, **9**, S. 341–352, 1972.
- [87] LEWIS, R.A. UND A.R. LEACH. Current methods for site-directed structure generation. *J. Comp.-Aid. Mol. Des.*, **8**, S. 467–475, 1994.
- [88] MARTIN, E.J., J.M. BLANEY, M.A. SIANI, D.C. SPELLMEYER, A.K. WONG UND W.H. MOOS. Measuring diversity: experimental design of combinatorial libraries for drug discovery. *J. Med. Chem.*, **38**, S. 1431–1436, 1995.
- [89] MASINTER, L.M., N.S. SRIDHARAN, J. LEDERBERG UND D.H. SMITH. Applications of Artificial Intelligence for Chemical Inference. XII. Exhaustive Generation of Cyclic and Acyclic Isomers. *J. Am. Chem. Soc.*, **96**, S. 7702–7714, 1974.
- [90] MATA, P., V.J. GILLET, A.P. JOHNSON, J. LAMPREIA, G.J. MYATT, S. SIKE UND A.L. STEBBINGS. SPROUT: 3D structure generation using templates. *J. Chem. Inf. Comput. Sci.*, **35**, S. 479–493, 1995.
- [91] MCGOWAN, J.C. *J. Appl. Chem.*, **2**, S. 323, 1952.
- [92] MCGREGOR, J.J. Backtrach search algorithms and the maximal common subgraph problem. *Software*, **12**, S. 23–34, 1982.
- [93] MCGREGOR, J.J. UND P. WILLET. Use of a maximal common subgraph algorithm in the automatic identification of the ostensible bond changes occurring in chemical reactions. *J. Chem. Inf. Comput. Sci.*, **21**, S. 137–140, 1981.
- [94] MCKAY, B.D. Computing automorphisms and canonical labelings of graphs. *Springer Lec. Not. Math.*, **686**, S. 223–232, 1977.
- [95] MELNIKOV, O., R. TYSHKEVICH, V. YEMELICHEV UND V. SARVANOV. *Lectures of Graph Theory*. BI-Wissenschaftsverlag, Mannheim, 1994.
- [96] MOLODTSOV, S.G. Computer-aided generation of molecular graphs. *MATCH*, **30**, S. 213–224, 1994.
- [97] MORGAN, H.L. The generation of a unique machine description for chemical structures. A technique developed at Chemical Abstract Service. *J. Chem. Doc.*, **5**, S. 107–113, 1965.
- [98] MOSER, D. *Die computerunterstützte Konstruktion molekularer Graphen*. Diplomarbeit, Universität Bayreuth, 1987.
- [99] NICHOLSON, V., C.C. TSAI, M. JOHNSON UND NAIM, M. A subgraph isomorphism theorem for molecular graphs. In KING, R.B. UND D.H. ROUVRAY, Hrsg., *Graph Theory and Topology in Chemistry*, S. 226–230. Elsevier Science Pub., Amsterdam, 1987.
- [100] NOURSE, J.G. Generalized Stereoisomerization Modes. *J. Am. Chem. Soc.*, **99**, S. 2063–2069, 1977.
- [101] PAVIA, M.R. The chemical generation of molecular diversity. *NetworkScience*, Aug. 1995.
- [102] PEARLMAN, R. Rapid generation of high quality approximate 3D molecular structures. *Chem. Design Autom. News*, (2), S. 5–6, 1987.
- [103] PÓLYA, G. Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und chemische Verbindungen. *Acta mathematica*, **68**, S. 145–253, 1937.
- [104] RANDIĆ, M. Design of molecules with desired properties. In JOHNSON, M.A. UND G.M. MAGGIORA, Hrsg., *Concepts and Applications of Molecular Similarity*, Kap. 5, S. 77–145. Wiley, New York, NY, 1990. (Ref. [65]).
- [105] READ, R.C. Every-one a winner. *Ann. Discr. Math.*, **2**, S. 107–120, 1978.

- [106] REDFIELD, H.J. The theory of group-reduced distributions. *Am. J. Math.*, **49**, S. 433–455, 1927.
- [107] ROUVRAY, D.H. Should we have designs on topological indices? In KING, R.B., Hrsg., *Chemical Applications of Topology and Graph Theory*, S. 159–177. Elsevier, Amsterdam, 1983.
- [108] RUCH, E., W. HÄSSELBARTH UND B. RICHTER. Doppelnebenklassen als Klassenbegriff und Nomenklaturprinzip für Isomere und ihre Abzählung. *Theor. Chim. Acta*, **19**, S. 288–300, 1970.
- [109] SADOWSKI, J. UND J. GASTFINGER. From atoms and bonds to three-dimensional atomic coordinates: Automatic model builders. *Chem. Rev.*, **93**, S. 2567–2581, 1993.
- [110] SCsIBRANY, H. UND K. VARMUZA. *Handbuch zu ToSIM (Software zur Untersuchung von topologischen Ähnlichkeiten in Molekülen)*. Technische Universität, Wien, 1994.
- [111] SEDGWICK, R. *Algorithmen in C++*. Addison-Wesley, Bonn, 1992.
- [112] SHELLEY, C.A. Heuristic Approach for Displaying Chemical Structures. *J. Chem. Inf. Comput. Sci.*, **23**, S. 61–65, 1983.
- [113] SHERIDAN, R.P. UND S.K. KEARSLEY. Using a genetic algorithm to suggest combinatorial libraries. *J. Chem. Inf. Comput. Sci.*, **35**, S. 310–320, 1995.
- [114] SIMS, C.C. Computation with permutation groups. In PETRICK, S.R., Hrsg., *Proc. of the Second Symposium on Symbolic and Algebraic Manipulation*, S. 23–28, New York, NY, 1971. Assoc. Comput. Mach.
- [115] SKVORTSOVA, M.I., I.I. BASKIN, O.L. SLOVOKHOTOVA, V.A. PLYULIN UND N.S. ZETIKOV. Inverse Problem in QSAR/QSPR studies for the case of topological indices characterizing molecular shape (Kier indices). *J. Chem. Inf. Comput. Sci.*, **33**, S. 630–634, 1993.
- [116] STUPER, A.J. UND P.C. JURs. Classification of psychotropic drugs as sedatives or tranquilizers using pattern recognition techniques. *J. Am. Chem. Soc.*, **97**, S. 182, 1975.
- [117] TINHOFER, G. *Methoden der angewandten Graphentheorie*. Springer-Verlag, Wien, 1976.
- [118] TSAI, C.C., M. JOHNSON, V. NICHOLSON UND NAIM, M. A topological approach to molecular similarity analysis and its application. In KING, R.B. UND D.H. ROUVRAY, Hrsg., *Graph Theory and Topology in Chemistry*, S. 231–236. Elsevier Science Pub., Amsterdam, 1987.
- [119] UGI, J. Fast and permanent changes in preparative and pharmaceutical chemistry through multicomponent reactions and their 'libraries'. *Proc. Eston. Acad. Sci. Chem.*, **44**, S. 237–273, 1995.
- [120] UGI, I., A. DÖMLING, B. GRUBER, M. HEILINGBRUNNER, C. HEISS UND W. HÖRL. Formale Unterstützung bei Multikomponentenreaktionen – Automatisierung der Synthesechemie. In MOLL, R., Hrsg., *Software-Entwicklung in der Chemie 9*, S. 114–128. GDCh, Frankfurt am Main, 1995.
- [121] VARKONY, T., Y. SHILOACH UND H. SMITH. Computer-assisted examination of chemical compounds for structural similarities. *J. Chem. Inf. Comput. Sci.*, **19**, S. 104–111, 1978.
- [122] VARMUZA, K. UND H. SCsIBRANY. Clusteranalyse isomerer chemischer Strukturen basierend auf binären Deskriptoren und der Hauptkomponentenanalyse. In *9. CiC-Workshop*, Bitterfeld, 1994. (Posterpräsentation).
- [123] VARMUZA, K., W. WERTHER, F. STANCL, A. KERBER UND R. LAUE. Computer-assisted structure elucidation of organic compounds, based on mass spectra classification and exhaustive isomer generation. In GASTEIGER, J., Hrsg., *Software-Entwicklung in der Chemie 10*, S. 303–314. GDCh, Frankfurt am Main, 1996.
- [124] VEITH, G.D., S.C. BASAK UND G.J. NIEMI. Predicting properties of molecules using graph invariants. *J. Math. Chem.*, **7**, S. 243–272, 1991.
- [125] WEBER, L., S. WALLBAUM, C. BROGER UND K. GUBERNATOR. Optimierung der biologischen Aktivität von kombinatorischen Verbindungsbibliotheken durch einen genetischen Algorithmus. *Angew. Chemie*, **107**, S. 2452–2453, 1995.
- [126] WIELAND, T. Object-oriented Discrete Algorithms for Chemistry and Graph Theory. (in Vorbereitung).
- [127] WIELAND, T. *Computerunterstützte Berechnung von Stereoisomeren*. Diplomarbeit, Universität Bayreuth, 1994.
- [128] WIELAND, T. Erzeugung, Abzählung und Konstruktion von Stereoisomeren. *MATCH*, **31**, S. 153–203, 1994.
- [129] WIELAND, T. Enumeration, generation, and construction of stereoisomers of high-valence stereocenters. *J. Chem. Inf. Comput. Sci.*, **35**, S. 220–225, 1995.
- [130] WIELAND, T. The use of structure generators in predictive pharmacology and toxicology. *Arzneim.-Forsch./Drug Res.*, **46** (I), S. 223–227, 1996.
- [131] WIELAND, T., A. KERBER UND R. LAUE. Principles of the generation of constitutional and configurational isomers. *J. Chem. Inf. Comput. Sci.*, **36**, S. 413–419, 1996.
- [132] WIENER, H. Structural determination of paraffin boiling point. *J. Amer. Chem. Soc.*, **69**, S. 17–20, 1947.
- [133] WILL, M., W. FACHINGER UND J.R. RICHERT. Fully automated structure elucidation — a spectroscopist's dream comes true. *J. Chem. Inf. Comput. Sci.*, **36**, S. 221–227, 1996.
- [134] WILLETT, P. *Similarity and clustering in chemical information systems*. J. Wiley & Sons, New York, NY, 1987.