

Ein wissensbasiertes Verfahren zur Komposition von frühklassischen Menuetten

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik der
Universität Stuttgart zur Erlangung der Würde eines Doktors der
Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung

vorgelegt von
Mathis Löthe
aus Berlin

Hauptberichter	Prof. Dr. Egbert Lehmann
Mitberichter	Prof. Dr. Thomas Ertl
Tag der Einreichung	18. Februar 2002
Tag der mündlichen Prüfung	12. Juni 2003

Institut für Intelligente Systeme (IIS)
der Universität Stuttgart

2003

CR-Klassifikation: J.5.5, I.2.8

Inhaltsverzeichnis

Abbildungsverzeichnis	6
Abstract	8
Zusammenfassung	9
1 Einleitung	10
1.1 Ziele der Arbeit	10
1.2 Eigenschaften der Aufgabenstellung	13
1.3 Überblick über die Arbeit	17
2 Grundlagen	18
2.1 Quellen für musikalisches Wissen	18
2.1.1 Zeitgenössische Literatur	19
2.1.2 Systematische Literatur	20
2.1.3 Befragung von Experten	22
2.1.4 Manuelle Analyse von Kompositionen	22
2.1.5 Computerunterstützte Analyse von Menuetten	23
2.1.6 Computerexperimente	24
2.1.7 Kognitive Introspektive nach Komposition	25
2.2 Musikalische Grundlagen	25
2.2.1 Hintergründe von Stil und Gattung	25
2.2.2 Tonsystem und Bezeichnungen	26
2.2.3 Aufbau eines Menuetts	28
2.2.4 Analyse eines Menuetts	30
2.3 Verwandte Arbeiten	33
2.3.1 Kriterien zum Vergleich von Kompositionsaufgaben	33
2.3.2 Vierstimmiger Satz (Ebcioğlu und andere)	34
2.3.3 Kontrapunkt im Palestrinastil (Schottstaedt)	36
2.3.4 Komposition klassischer Klaviersonaten (Berggren)	36
2.3.5 Patternbasierte Stilkopie (Cope)	37

2.3.6	Harmonische Analyse (Maxwell)	38
3	Kompositionsverfahren	41
3.1	Aufbau des Verfahrens	41
3.2	Partiturokarte für musikalische Grundelemente	46
3.2.1	Anforderungen an die Repräsentation	46
3.2.2	Karte aus Zeitelementen	48
3.2.3	Tonhöhen, Intervalle und Stufen	49
3.2.4	Tonart und Stufen	50
3.2.5	Harmonien	52
3.3	Kompositionssteuerung und Strukturbeschreibung	53
3.3.1	Kompositionspläne	53
3.3.2	Regelsätze und Parameterverwaltung	54
3.3.3	Makrostruktur	55
3.3.4	Mikrostruktur	56
3.4	Suche und Zustandsverwaltung	57
3.4.1	Formulierung vom Problemen als Suchproblem	58
3.4.2	Zustände	59
3.4.3	Einfluß der Eigenschaften des Suchraums auf das Verfahren	60
3.4.4	Suchverfahren	61
3.4.5	Heuristische Suche mittels Gewichtsfaktoren	64
3.5	Stimmengenerierung mit Alternativenregeln	64
3.5.1	Darstellung der Stimmengenerierung als Suchproblem	65
3.5.2	Alternativenauswahl durch Regeln	66
3.5.3	Bottom-Up-Strukturelemente in Alternativen	67
3.5.4	Darstellung von Korrespondenzen	68
3.5.5	Aufbau der Alternativenregeln	70
3.6	Regeln für die Stimmengenerierung	73
3.6.1	Parameterwerte	73
3.6.2	Regeln für die Startnoten eines Satzes	75
3.6.3	Allgemeine Intervallregeln	76
3.6.4	Leittonregeln	78
3.6.5	Dreiklangsschrittregeln	79
3.6.6	Regeln zu Betonung und Dauer	81
3.6.7	Harmonie und Zweistimmigkeit	82
3.6.8	Präferenzregeln (Weiche Regeln)	84
3.6.9	Regeln für Bottom-up-Strukturelemente	85
3.6.10	Regeln zur Beschleunigung von Endformeln	89
3.6.11	Aufruf anderer Komponenten	89
3.6.12	Rhythmus und Dauernvergabe	90

3.6.13	Beispiele für Regelfolgen	92
3.7	Stimmführung	93
3.7.1	Untersuchung von Stimmführungsquadrupeln	93
3.7.2	Definitionssprache für Stimmführungsregeln	97
3.7.3	Reichweite von Stimmführungsregeln	98
3.7.4	Stimmführungsregeln	101
3.8	Muster und Patternmatching	104
3.8.1	Grundlagen von Patternmatcher und Patternsprache	104
3.8.2	Muster für Endformeln	104
4	Implementierung	107
4.1	Prinzipien der Implementierung	107
4.1.1	Wahl der Programmiersprache und der Sprachmittel	107
4.1.2	Programmiertechnische Festlegungen	109
4.1.3	Überblick über den Systemaufbau auf technischer Ebene	112
4.1.4	Bedeutung der Effizienz	114
4.2	Das Datenformat COMF	116
4.2.1	Partiturkarte und Dauerelemente	117
4.2.2	Tonale Elemente	118
4.2.3	Verwendung der definierten Klassen	119
4.2.4	Ausgabe der Noten mit Lilypond	120
4.3	Kompositionssteuerung	120
4.3.1	Aufbau eines Regelsatzes	121
4.3.2	Aufbau der Strukturbeschreibung	121
4.3.3	Abarbeitung von Kompositionsplänen	122
4.3.4	Initialisierung der Stimmengenerierung	124
4.4	Suchstrategien und Zustandsverwaltung	125
4.4.1	Basisdefinitionen zur Suche	125
4.4.2	Suchstrategien	126
4.4.3	Problembeschreibung für die Stimmengenerierung	126
4.4.4	Benutzerinformation während der Suchvorgänge	127
4.4.5	Statistikkomponente	128
4.5	Regeln und Stimmengenerierung	129
4.5.1	Gemeinsame Elemente aller Regeln	129
4.5.2	Regeln für die Stimmengenerierung	130
4.5.3	Fortsetzungselemente	131
4.5.4	Strukturelemente	132
4.6	Stimmführung	135
4.6.1	Stimmführungsquadrupel und Reichweiten	136
4.6.2	Definition von Stimmführungsregeln	138
4.7	Endmuster und Pattern matching	140

4.7.1	Definition von Endformeln	141
4.7.2	Matching-Verfahren	141
4.7.3	Einsatz des Patternmatchers für das Matching von Endformeln	142
5	Ergebnisse	144
5.1	Menuette mit strukturellen Beziehungen	144
5.2	Untersuchung einer Lösungsmenge	150
5.3	Variation des Musikwissens	154
5.3.1	Menuettmelodien ohne Korrespondenzen	154
5.3.2	Baßstimme	157
5.3.3	Spezielle Melodien	159
5.4	Baß zu einer zeitgenössischen Melodie	161
5.5	Statistiken zum Suchvorgang	163
6	Ausblick	169
6.1	Erweiterungen und Verbesserungen	169
6.1.1	Erweiterungen der musikalischen Mittel	169
6.1.2	Verbesserung von Verfahren und Strategien	174
6.1.3	Verbesserung der Vorbereitung des Wissens	175
6.2	Perspektiven	176
6.3	Fazit	178
A	Weitere Kompositionsergebnisse	179
B	English Summary	186
B.1	Introduction	186
B.2	Knowledge Based Minuet Composition	188
B.2.1	Voice Generation	189
B.2.2	Structure	190
B.2.3	Voice Leading Rules	191
B.2.4	Pattern Matching of End Formulas	191
B.2.5	Technical Aspects	192
B.3	Results	192
B.4	Summary	194
	Literatur	195
	Lebenslauf	199

Abbildungsverzeichnis

1.1	Grobklassifizierung intelligenter Systeme	13
1.2	Verhältnis von erzeugten und möglichen Menuetten	14
2.1	Akkorde auf den Stufen der Tonleiter	27
2.2	Menuett F-Dur KV 2 von W.A.Mozart (1762)	31
3.1	Überblick über das Verfahren	42
3.2	Quintenzirkel	51
3.3	Beispiel für Kompositionsplan und Lösungszahl	54
3.4	Regelfolge für die Startnoten der Melodie	92
3.5	Regelfolge für die Startnoten des Basses	92
3.6	Regelfolge für die Folgeintervalle der Melodie	94
3.7	Regelfolge für die Folgeintervalle des Basses	95
3.8	Grundbegriffe des Stimmführungsquadrupels	96
3.9	Beispiele für die Reichweite der Oktavparallele aus [Koc82]	99
3.10	Ein Endformelmuster für eine Kadenz	106
4.1	Systemkomponenten auf physischer Ebene	113
4.2	Die Klassen für COMF	117
4.3	Datenstruktur der Partiturnote im Box-Pointer-Diagramm	118
4.4	Klassenhierarchie der Elemente des Kompositionsplans	122
4.5	Klassenhierarchie der Zustände	125
4.6	Klassen für die verschiedenen Regeln	129
4.7	Klassen für Fortsetzungselemente	131
4.8	Klassen für Strukturelemente	132
4.9	Klasse für Stimmführungsquadrupel	136
	Musikbeispiel 1 Dauerngleichheit	146
	Musikbeispiel 2 Dauerngleichheit mit Monte	147
	Musikbeispiel 3 Intervallähnlichkeiten	147
	Musikbeispiel 4 Menuett mit Umkehrung	148
	Musikbeispiel 5 Menuett mit Viertelrepetitionstakt	149

Musikbeispiel 6 Menuett mit externem Rhythmusmodell	150
Musikbeispiele 7 Ergebnismenge mit Strukturvorgabe	151
Musikbeispiel 8 Dreiklangsmelodie	155
Musikbeispiel 9 Skalenbasierte Melodie	156
Musikbeispiel 10 Mischmelodie	156
Musikbeispiel 11 Baß ohne Linienverbesserung	157
Musikbeispiel 12 Baß ohne Zusammenklangsregeln	158
Musikbeispiel 13 Baß aus Akkordgrundtönen	159
Musikbeispiel 14 * Baß ohne Stimmführung	159
Musikbeispiel 15 Menuett aus Viertelnoten	160
Musikbeispiel 16 Menuett mit Vierteln und Halben	160
Musikbeispiel 17 * Verzicht auf Endformeln	161
Musikbeispiel 18 Baß zu gegebener Melodie	162
Musikbeispiel 19 Baß zu gegeb. Melodie, unvollk. Konson. auf 1 . .	162
5.1 Statistiken zur Suche bei der Komposition	166
Musikbeispiele 20 Ergebnismenge mit Richtungsgleichheiten	179
Musikbeispiele 21 Ergebnismenge mit Gegenrichtung	183
Musikbeispiele 22 Ergebnismenge mit Strukturvorgabe	185
Musikbeispiel 23 (=1) Minuet with Rhythmical Model	193
Musikbeispiel 24 (=4) Minuet with Melodic Relations	193
Musikbeispiel 25 (=19) Bass to a given melody	194

Abstract

This thesis presents a knowledge-based system for composition of piano minuets in early classical style (e.g. young W.A. Mozart). The problem takes an intermediate position between restricted and free problems of computerized composition of music.

The composition process is split into 3 subtasks: structure planning, melody composition and bass composition. The structure plan consists of two levels. The macrostructure describes, how the minuet consists of segments called sentences which are ended by an end formula. The microstructure defines further properties for those segments. Melody and bass composition are both tasks of voice generation. Voice generation composes note by note using rules to determine the set of possible alternatives and using search algorithms to search the state space of these alternatives. The end formula of each sentence is taken into account by using pattern matching. On top of this, bass composition considers voice leading rules.

The knowledge based approach allows experiments with different combinations of musical knowledge. This includes composition of minuets from scratch with different rules and parameter settings, composition of a bass to a given melody and experiments about the degradation of the result when rules are removed.

An extended summary in english can be found in appendix B on page 186.

Zusammenfassung

Diese Arbeit beschreibt ein wissensbasiertes System, das Klaviermenuette im frühklassischen Stil (z.B. Jugendwerke von W.A. Mozart) komponiert. Die Aufgabenstellung nimmt eine Mittelstellung zwischen stark festgelegten und freien Aufgabenstellungen der Computerkomposition ein.

Der Kompositionsprozeß besteht aus drei Teilaufgaben: Strukturkomposition, Melodiekomposition und Baßkomposition. Die Strukturkomposition erzeugt den Strukturplan des Stücks in zwei Stufen. Zuerst entsteht die Makrostruktur, die das Menuett in sogenannte Sätze gliedert. Die Mikrostruktur fügt dann Details innerhalb der Sätze hinzu. Die Komposition von Melodie und Baß sind beides Stimmengenerierungsaufgaben. Bei der Stimmengenerierung wird Note für Note komponiert, indem mittels Regeln die Menge der zulässigen Fortsetzungen bestimmt und dann mit einem Suchverfahren der Zustandsraum durchsucht wird. Zusätzlich wird mit einem Pattern-Matching-Verfahren die Einhaltung der Endformeln der Sätze geprüft. Bei der Baßkomposition untersucht das Verfahren darüberhinaus noch die Einhaltung der Stimmführungsregeln.

Das wissensbasierte Vorgehen ermöglicht Experimente mit unterschiedlich zusammengestelltem Wissen. Es wurden Menuette mit unterschiedlichen Regeln und Parametereinstellungen komponiert, ein Baß zu einer gegebenen Melodie gesetzt und der Qualitätsverlust der Ergebnisse beim Weglassen von Regeln untersucht.

Eine erweiterte Zusammenfassung (auf Englisch) steht in Anhang B auf S. 186.

Kapitel 1

Einleitung

Die vorliegende Arbeit beschreibt ein Verfahren, mit dem ein Computerprogramm Menuette für Klavier im frühklassischen Stil komponiert, seine theoretischen Voraussetzungen aus moderner und zeitgenössischer (historischer) Literatur, die Realisierung in Common Lisp/CLOS und die Ergebnisse. Im folgenden werden die weiteren Ziele der Arbeit mit ihren Hintergründen und die besonderen Aspekte dieser Aufgabenstellung erläutert.

1.1 Ziele der Arbeit

Das Menuettkompositionssystem ist an den folgenden Zielen ausgerichtet.

- **Musikalische Korrektheit und Qualität:** Die Ergebnisse des Kompositionssystems sollen von einem Experten als typischer Vertreter von Stil und Gattung akzeptiert werden. Die Aufgabe, Musik in einem vorgegebenen historischen Kompositionsstil zu komponieren, nennt man *Stilkopie*.

Es ist *nicht* Ziel des Systems, „künstlerisch wertvolle“ oder neuartige Musik zu komponieren. Diese Arbeit befaßt sich nicht mit Fragen der Ästhetik und des künstlerischen Anspruchs, sondern mit Bereichen, bei denen ein Experte von klaren Satzfehlern, Stilbrüchen oder zumindest Unschönheiten spricht.

- **Entwicklung von Techniken,** die verschiedene Aspekte der Komposition, wie Melodik, Struktur, Harmonie und Stimmführung miteinander verbinden.

Die bisherigen Arbeiten im Bereich der Stilkopie durch einen Computer konzentrieren sich ausdrücklich auf einen Teilaspekt, um eine gute

Qualität der musikalischen Ergebnisse zu erhalten. Die Integration verschiedener Aspekte ist daher eine wichtige Neuerung in dieser Arbeit.

- Formalisierung und Untersuchung von Wissen: Das System arbeitet mit explizit repräsentiertem Wissen. Damit schafft es die Möglichkeit, die Auswirkung einzelner musikalischer Wissensselemente auf das Gesamtergebnis zu untersuchen.

Dieses Vorgehen ist vergleichbar mit dem Ansatz der experimentellen Archäologie, bei dem Theorien durch anschauliches Nachbauen und Nachspielen historischer Situationen überprüft werden. Ein Trend hin zu einer stärker formalen Darstellung von Wissen ist in den Geisteswissenschaften, beispielsweise in der Linguistik, in den letzten Jahrzehnten allgemein zu verzeichnen gewesen. Diese Arbeit soll helfen, auch für die Musikwissenschaft die Ausgangsbasis für experimentelles Arbeiten zu schaffen.

- Verwendung zeitgenössischer Literatur: (Literatur aus der Zeit des betreffenden Stils) Es ist interessant, zu untersuchen, inwiefern zeitgenössische Literatur für diese Arbeit verwertbar ist.

Kennzeichnend für die Stilkopie ist, daß einerseits Gesetzmäßigkeiten von Stil und Gattung die zulässigen Ergebnisse einschränken, andererseits aber noch ein gewisser Freiraum bleibt. Die Fähigkeit, einerseits derlei kreative Freiräume auszunutzen und dabei andererseits durch Regeln festgelegte Vorgaben einzuhalten, ist eine typische Stärke menschlicher Intelligenz, die sich nicht nur im Bereich der Komposition von Musik zeigt. Die Nachbildung solcher Fähigkeiten auf dem Rechner ist daher ein interessantes Forschungsthema.

Zielstil und Zielgattung dieser Arbeit sind Menuette im Stil der Frühklassik, das ist die Zeit des jungen Mozart und seiner Zeitgenossen. Folgende Gründe waren für die Wahl ausschlaggebend:

- Das Menuett ist eine kurze, übersichtliche und regelmäßig aufgebaute Gattung, so daß die Beherrschung der Gesetzmäßigkeiten bei brauchbarer musikalischer Qualität möglich ist.
- Trotz dieses einfachen Aufbaus enthält das Menuett die für die klassische Komposition wesentlichen Aspekte. Nicht ohne Grund war das Menuett bereits in der zeitgenössischen Kompositionsliteratur eine Modellgattung zur Demonstration von Kompositionstechniken und -regeln.

- Die frühklassischen Menuette sind musikwissenschaftlich gründlich erforscht, so daß Quellen für musikwissenschaftliches Wissen und Kriterien zur Beurteilung der Ergebnisse des Systems zur Verfügung stehen.

Diese Arbeit grenzt sich damit deutlich von künstlerisch orientierten computermusikalischen Arbeiten ab, die nach einer vom Autor selbst festgelegten Ästhetik beurteilt werden.

- Für den Stil und die Gattung ist zeitgenössische Literatur verfügbar.

Wissen über verschiedene Teilaspekte der Komposition von Menuetten ist in der Literatur verbal beschrieben. Es gibt jedoch bislang keine geschlossene, algorithmisch gefaßte Theorie oder deterministische Vorgehensmodelle für den Kompositionprozeß, sondern die Teilaspekte sind einzeln beschrieben. Bringt man ein so strukturiertes Problem auf den Rechner, kann man auf den Gewinn von neuen informatischen Erkenntnissen im Bereich der Bearbeitungsstrategien und der Modellierung von Wissen hoffen.

Wenn ein Mensch tonale, stilgebundene Musik komponiert, muß er den Forderungen der o.g. verschiedenen Teilaspekte genügen. Durch Üben und Erfahrung erwirbt er die Fähigkeit, bei der Komposition Regeln, Ge- und Verbote und andere Formen des Wissens aus den einzelnen Teilbereichen zu berücksichtigen. Dies erfordert nach üblicher Auffassung Intelligenz.

Aufgaben auf den Rechner zu bringen, die als Intelligenz erforderlich gelten, wenn sie ein Mensch erledigt, fällt in das Arbeitsgebiet der *Künstlichen Intelligenz*(KI).

Abbildung 1.1 zeigt, wie man die verschiedenen Definitionen nach [RN95] durch Kombination von zwei Merkmalen in vier Gruppen einteilen kann. Die dieser Arbeit zugrunde liegende Auffassung von künstlicher Intelligenz ist im Klassifikationsystem ein *System, das rational denkt*. Es ist also die Arbeitsweise des Systems und das modellierte Wissen interessant, während das Nachbilden menschlicher Denkweisen nicht gefordert wird. Dies ist kein Widerspruch zum eingangs genannten Ziel, das modellierte Musikwissen zu untersuchen.

Diese Arbeit hat das Ziel, den Kompositionsstil so gut wie möglich anzunähern. Viele der kognitiv motivierten Arbeiten – solche, die das menschliche Denken nachvollziehen wollen, indem sie es algorithmisch nachbauen – machen Abstriche bei der Qualität der Ergebnisse. Wenn man aber beurteilen möchte, ob das Musikwissen angemessen modelliert und brauchbar ist, ist die Qualität der Ergebnisse wichtig. Angemessenheit des modellierten Wissens wichtig. Deswegen wird auf die Modellierung des menschlichen Denkens verzichtet.

Hauptinteresse liegt auf:	Den Menschen nachbildend	
	Ja	Nein
Informations- verarbeitung	Systeme, die wie Menschen denken	Systeme, die rational denken
Intelligentem Handeln	Systeme, die wie Menschen handeln	Systeme, Systeme, die rational handeln

Abbildung 1.1: Grobklassifizierung intelligenter Systeme nach [RN95]

Die internen Denkvorgänge in einem Komponisten sind ohnehin schwer beobachtbar. Trotzdem kann auch bei dieser Arbeit in Einzelfällen die Arbeitsweise des Menschen als Anregung und Vorbild zur Gewinnung von Wissen benutzt werden.

Diese Arbeit muß mit explizit repräsentiertem Wissen arbeiten, um die Untersuchung des modellierten Wissens zu ermöglichen. Sie basiert daher auf der *physical symbol system*-Hypothese [RK91], die besagt, daß sich Intelligenz durch Rechnen auf zusammengesetzten symbolischen Ausdrücken, denen man eine Bedeutung zuordnet, erklären und simulieren läßt¹.

1.2 Eigenschaften der Aufgabenstellung

Ergebnis der Komposition sind Klaviermenuette in Form von Noten bzw. einer internen Darstellung durch Dauern und Tonhöhen. Es handelt sich damit ungefähr um die gleiche Information, die die Komponisten der Frühklassik damals aufschrieben. Sie schrieben zu den Noten manchmal noch dynamische und artikulatorische Anweisungen dazu, dies wird hier nicht berücksichtigt. Die Interpretation, d.h. die Umsetzung der notierten Darstellung in Klang wird hier ebenfalls nicht behandelt.

Der frühklassische Stil und die Gattung der Menuette soll so gut wie möglich getroffen werden, d.h. ein Experte sollte das Stück als ein typisches Exemplar eines frühklassischen Klaviermenuetts akzeptieren. Solche Menuette werden im folgenden als *stilistisch korrekt* oder kurz – in Anlehnung an die Begriffe aus der Linguistik² – als *stilistisch* schlechthin bezeichnet. Das Gegenteil, ein stilistisch nicht korrektes Stück, wird als *unstilistisch* bezeichnet. Im konkreten Fall kann man noch unterscheiden, ob das Stück wegen

¹Im Gegensatz dazu steht der *konnektionistische Ansatz*, bei dem man auf eine Repräsentation von explizitem Wissen verzichtet.

²In der Linguistik bezeichnet man syntaktisch korrekte Sätze als *grammatisch* und Sätze, die nicht korrekt sind als *ungrammatisch*.

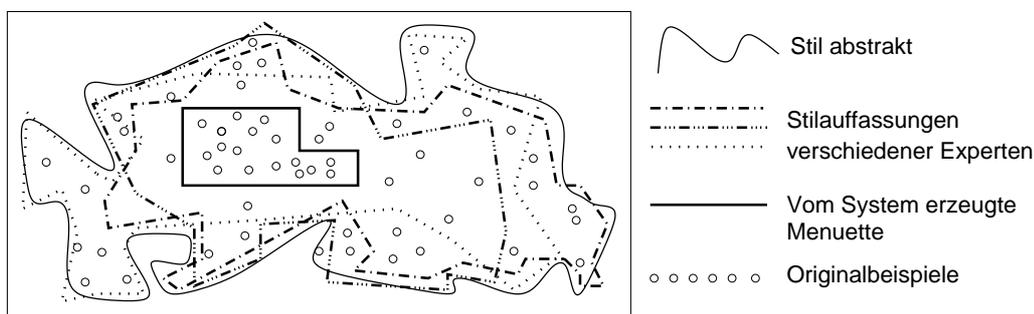


Abbildung 1.2: Verhältnis von erzeugten und möglichen Menuetten

eines Verstoßes gegen den frühklassischen Stil, die Gattung Menuett oder die Anforderungen der Besetzung (hier eine Klavierkomposition) unstilistisch ist. Bei den folgenden allgemeinen Betrachtungen wird dies zunächst vernachlässigt.

Ein Experte, der ein Stück beurteilt, entwickelt seine Vorstellung vom Stil eines frühklassischen Klaviermenuetts durch eine Kombination von explizitem und implizitem Wissen. Einerseits kennt er explizite Kompositionsregeln und verbale Beschreibungen des Stils aus der Literatur. Andererseits betrachtet er Beispiele von Originalmenuetten und beurteilt neue Stücke durch Analogieschlüsse beim Betrachten. Auch wenn die Vorstellungen verschiedener Experten im Bereich des impliziten Wissens nie vollständig übereinstimmen, wird im Normalfall ein gemeinsamer unumstrittener Bereich vorhanden sein.

Ein generierendes System soll korrekt sein, muß aber nicht vollständig sein. In unserem Fall heißt das, daß das System nur stilistische Menuette erzeugen darf, es aber ganze Klassen von Menuetten geben kann, die das System bestimmt nie komponiert, obwohl sie eigentlich stilistisch wären. Konkret bedeutet das, daß das System gewisse, eigentlich zulässige Kompositionselemente (z.B. bestimmte Akkordverbindungen, melodische Intervalle etc.) nicht kennt, nicht vorsieht oder nicht zuläßt. Der Hintergrund für diese Festlegung ist, daß manche musikalischen Elemente schwer beherrschbar sind, d.h. die Bedingungen, in welchem Zusammenhang ihre Verwendung stilistisch korrekt ist und in welchem nicht, ist schwer formal zu definieren. Abbildung 1.2 zeigt das Verhältnis zwischen den vom System erzeugten und den stilistischen Menuetten.

Versucht man nun, Stil und Gattung der frühklassischen Klaviermenuette zu beschreiben, so erhält man unterschiedlich streng bindende Anforderungen und Regeln:

- Es gibt Kompositionsregeln, die in einem korrekten Menuett unbedingt eingehalten werden müssen. Typische Beispiele sind die in diesem Zu-

sammenhang vielzitierten Satzregeln (Stimmführungsregeln), die Verwendung konsonanter Harmonien und deren korrekte Abfolge.

- Ebenso gibt es sogenannte „weiche“ Regeln, deren Verletzung nicht zu einem stilistisch inkorrekten Menuett führt, allenfalls zu einem etwas unschöneren. Die zu häufige Verletzung weicher Regeln führt am Ende aber auch zu einem unstilistischen Werk.
- Noch „weicher“ sind Regeln, bei denen man sagt, daß ein Stück durch sie schöner, interessanter oder für Stil und Gattung typischer wird, ohne daß sie einen Einfluß darauf haben, ob ein Stück stilistisch ist oder nicht.

Der Begriff „Kompositionsregel“ läßt einen an eine Formulierung durch Verbote und Warnungen denken. Versucht man aber, den Stil zu beschreiben, dann wird man außer Verboten auch Gebote nennen, d.h. man beschreibt Phänomene, die man in einem stilistischen Menuett fordert, zuläßt oder gerne sieht.

Betrachtet man eine Komposition in üblicher Notendarstellung, so erkennt man 2 Richtungen, die horizontale (Zeitachse) und die vertikale (synchrone).

In der horizontalen Richtung kann man verschiedene strukturelle Ebenen unterscheiden. Die *Mikrostruktur* befaßt sich mit einzelnen Tönen und ihren Nachbarn und beschreibt Elemente, wie z.B. Dreiklangsbrechungen, Durchgänge oder Vorhalte. Die *Makrostruktur* betrachtet das Verhältnis größerer Teile miteinander, d.h. die Gliederung des Stücks und Beziehungen zwischen diesen Teilen. Diese *Form* des Stücks wird in Kap. 2.2 beschrieben.

In der Vertikalen betrachtet man Zusammenklänge von mehreren gleichzeitigen Tönen, deren Anzahl (Stimmenzahl) meist beschränkt ist. Diese Zusammenklänge werden durch die Harmonien beschrieben.

Die Stimmführungsregeln kombinieren die mikrostrukturell horizontale und die vertikale Ebene; ein Beispiel ist das Verbot von Quintparallelen.

Ein Kennzeichen des klassischen Stil sind vielerlei strukturelle Beziehungen innerhalb der Komposition. Um diese aufzubauen, muß das System auch Wissen über die Komposition, das nicht direkt zu den Notenabfolge gehört, berücksichtigen. Ein einfaches Beispiel für solches Wissen ist die oben beschriebene Abfolge der Harmonien. Die Darstellung des Stücks im Rechner muß also außer der Repräsentation der Noten selbst auch die Repräsentation des strukturellen Wissens ermöglichen.

Die Komposition eines Menuetts kann in folgende Teilaufgaben aufgespalten werden:

- Erstellen der Strukturbeschreibung. Diese Teilaufgabe kann selbst wiederum in einzelne Detaillierungsschritte unterteilt werden.
- Komposition der Melodie innerhalb der Vorgaben durch die Strukturbeschreibung.
- Komposition der Begleitstimmen (Baß, evtl. Mittelstimmen).

Kompositionsliteratur, musikalische Analysen und Übformen aus dem Kompositionsunterricht betrachten diese Teilaufgaben oft separat. Man könnte sich unter Umständen auch vorstellen, daß ein Mensch Menuette komponiert, indem er in dieser Reihenfolge vorgeht. Normalerweise folgt ein Komponist jedoch dieser Ordnung nur ungefähr und integriert die Teilaufgaben. Ein großer Teil der menschlichen Leistungsfähigkeit bei der Komposition³ kommt von der virtuellen Beherrschung von Strategien, die mehrere Phasen in Wechselwirkung bearbeiten.

- Eine Möglichkeit, Teilaufgaben zu integrieren, wäre z.B., ein viertaktiges Melodistück zu komponieren (mit dem Wissen, daß dies üblicherweise in einer Makrostruktur eines Menuetts gut verwendbar ist) und erst dann zu überlegen, wie das Melodistück und seine Variationen in der Makrostruktur eingesetzt werden.
- Eine andere vom Menschen verwendete Strategie ist, nichtmonoton vorzugehen, d.h. frühere Entscheidungen zu korrigieren. Bei der Komposition durch einen Menschen läßt sich beobachten, daß manchmal, wenn während der Bearbeitung einer späteren Teilaufgabe Probleme auftauchen, kleinere Korrekturen an Ergebnissen der früheren Phase vorgenommen werden. Bei diesen Korrekturen versucht der Mensch, die dazwischen getroffenen Entscheidungen soweit beizubehalten, wie dies ohne Inkonsistenzen möglich ist.

Diese Strategie steht damit im Gegensatz zu beliebten Strategien der künstlichen Intelligenz wie dem chronologischen Backtracking, bei dem an dem Punkt, an dem eine Entscheidung zurückgenommen wird, alle davon abhängigen späteren Entscheidungen auch zurückgenommen werden und die Arbeit von dort ab ganz neu beginnt.

³Dies gilt auch für andere Aufgabengebiete, bei denen verschiedene Strukturebenen berücksichtigt werden müssen.

- Eine andere Strategie des Menschen ist, die Verwendbarkeit von musikalischem Material in späteren Phasen mitzubedenken. Er läßt sich dabei von einer groben Vorstellung von anderen Phasen leiten, ohne diese im Detail durchgeplant zu haben.

Extrembeispiele für diese Fähigkeit des Mitberücksichtigens sind musikalische Kunststückchen, die speziell für den frühklassischen Stil in der Literatur belegt sind. In [Rie52] findet man auf S. 35 ein zweistimmiges Palindrommenuett⁴, bei dem die zweite Hälfte die rückwärts gespielte erste Hälfte ist. Auf S. 236 findet man ein dreistimmiges Menuett (zwei Violinen und Baß), bei dem eine Violine und der Baß notiert sind. Der Baß ist symmetrisch zur Mitte, die Vl.1 spielt die Oberstimme von vorne und Vl.2 dieselbe Stimme von hinten. Noch kunstvoller ist das zweite Menuett auf dieser Seite. Dieses zweistimmige Stück ist einstimmig notiert, die Baßstimme erhält man, indem man das Notenblatt auf den Kopf stellt (und natürlich als Baßschlüssel interpretiert).

1.3 Überblick über die Arbeit

Das folgende Kapitel beschreibt den Rahmen für diese Arbeit: die verschiedenen möglichen Wissensquellen, die wesentlichen musikalischen Eigenschaften frühklassischer Menuette und verwandte Arbeiten. Kapitel 3 gibt einen Überblick über das System und beschreibt die Teilaufgaben, die Bestandteile des Systems, und das in ihnen modellierte Musikwissen.

Den technischen Aufbau des Programms, die Grundprinzipien der Implementierung und die Funktionsweise des Systems beschreibt Kapitel 4. Dieses Kapitel ist für einen Leser interessant, der System für eigene Experimente einsetzen, es erweitern oder Teile benutzen will. In Kapitel 5 werden die Ergebnisse untersucht. Dort sind verschiedene, vom System komponierte Menuette abgedruckt, die mit unterschiedlichen Parametereinstellungen gewonnen wurden.

Kapitel 6 zeigt Erweiterungsmöglichkeiten für das Verfahren und Perspektiven für zukünftige Arbeiten auf.

⁴Original: *minuetto cancrino*

Kapitel 2

Grundlagen

Dieses Kapitel beschreibt die Ausgangspunkte aus den Bereichen Musiktheorie und Informatik und faßt die direkten Vorarbeiten kurz zusammen.

2.1 Quellen für musikalisches Wissen

Diese Arbeit geht, wie eingangs beschrieben, wissensbasiert vor. Ein wissensbasiertes System trennt zwischen dem Abarbeitungsmechanismus, der *Inferenzmaschine*, und dem Wissen selbst. Im Gegensatz zu trainierenden Verfahren (wie z.B. neuronalen Netzen) ist in einem wissensbasierten System Wissen in expliziter Form repräsentiert (d.h. in geeigneten Datenformaten) codiert. Für die Erstellung des Systems muß dieses Wissen beschafft und in eine geeignete Form überführt werden.

Da Wissen im Prinzip Teil des Programms ist, muß es ebenso wie das Verfahren selbst gewartet¹ werden. Dazu ist es hilfreich, die Quellen für das Wissen zu dokumentieren, da man dann bei anstehenden Änderungen anhand der Quelle entscheiden kann, wie die Änderung im Einzelnen durchzuführen ist.

Die Wissensquellen für diese Arbeit werden in folgende Gruppen eingeteilt:

WQ1 Zeitgenössische Literatur

WQ2 Systematische Literatur

WQ3 Befragung von Experten

¹Im Gegensatz zur Wartung von technischen Geräten, bei der Verschleißerscheinungen behoben werden, geht es bei der Softwarewartung darum, Anpassungen und Erweiterungen durchzuführen, die durch Fortentwicklungen anderer Programmteile oder Änderungen im Anwendungsbereich notwendig werden.

WQ4 Manuelle Analyse

WQ5 Computerisierte Analyse

WQ6 Experimente mit dem Verfahren

WQ7 Kognitive Introspektive

Während das Wissen in WQ1 und WQ2 bereits in expliziter Form vorliegt, sollen WQ3-WQ7 Zugang zu Wissen geben, das für stilistische Kompositionen benötigt wird, aber nicht in der Literatur zu finden ist. [Ebc86] beschreibt dessen Existenz.

Der Rest dieses Abschnitts beschreibt die einzelnen Wissensquellen genauer.

2.1.1 Zeitgenössische Literatur

Zeitgenössische Literatur² ist eine wichtige Wissensquelle, weil sie eine authentische Sicht auf den Kompositionsstil gibt. Man kann in den meisten Fällen davon ausgehen, daß Lehrbücher einerseits die zuvor informell üblichen Konventionen codifizieren, jedoch andererseits auch in manchen Fällen eine Rückwirkung auf die Kompositionspraxis haben, d.h. das Begriffssystem von Kompositionsschülern geformt haben.

Die folgenden Bücher wurden in dieser Arbeit herangezogen:

- Das Werk von Riepel [Rie52] ist stark praktisch orientiert. Es ist ausführlich, (manchmal etwas geschwätzig) und im Stil eines Lehrer-Schüler-Dialogs abgefaßt. Besonders wertvoll sind die zahlreichen positiven und negativen Beispiele für die Melodiebildung und -gliederung. Da das Werk verhältnismäßig früh entstanden ist (der erste Teil erschien 1752) und praxisnahe Beispiele enthält, ist die o.g. Rückwirkung auf die Kompositionspraxis der Zeit denkbar.
- Das Buch von Koch [Koc82] ist streng in Paragraphen gegliedert und stärker theoretisch durchformalisiert. Diese Strenge ist bei der Beschreibung der Stimmführungsregeln besonders nützlich. Die 3 Bände des Werks entstanden zwischen 1782 und 1793, d.h. nach der Stilepoche.
- Außerdem wurden weitere, etwas ältere Werke als Hintergrundwissen herangezogen um Lücken im Musikwissen aufzufüllen. Im Bereich der Stimmführungsregeln sind dafür J.A. Scheibes *Compendium musices* (abgedruckt in [Ben60]) und Mattheson [Mat39] interessant.

²In der Musikwissenschaft bezeichnet man mit dem Begriff *zeitgenössische Literatur* Literatur, die zur selben Zeit entstanden ist, wie der Stil bzw. die Epoche, die sie beschreibt.

- Eine ausführliche Theorie zur Idee der Variation von Motiven findet man bei Lobe [Lob44] Anfang des 19 Jhds. Er sieht die klassische Epoche allerdings bereits aus der Rückschau. Es ist also ein Grenzfall, ob Lobe als zeitgenössische oder als systematische Literatur anzusehen ist.

Außer der zeitgenössischen Literatur selbst kann auch moderne sekundäre Literatur, die sich die Auswertung der zeitgenössischen Kompositionsliteratur zum Ziel gesetzt hat, als Quelle authentischen Musikwissens herangezogen werden.

- [Bud83] hat die zeitgenössische Kompositionsliteratur der Klassik, insbesondere [Rie52] und [Koc82] ausgewertet und Information wie Gliederungsschemata und Kadenzmuster z.T. in tabellarischer Form zusammengetragen.

Außerdem findet man bei Budday Analysen von Werken.

- [Ben60] betrachtet und vergleicht die unterschiedlichen Autoren des gesamten 18. Jahrhunderts, d.h. außer der Klassik auch die Spätphase des Barock.

Er beschreibt die Unterschiede im vermittelten Musikwissen, an denen man den Wandel des Kompositionsstils vom späten Barock zur Klassik verfolgen kann. Ebenso beobachtet er die sich wandelnde Herangehensweise an die Wissensvermittlung. Zu Beginn des 18. Jahrhunderts hatten ästhetische Theorien und auch astrologische und zahlensymbolische Betrachtungen durchaus noch einen Platz in einem Kompositionslehrbuch, während die oben erwähnten Autoren der klassischen Zeit sich auf praktisch verwertbares Wissen konzentrierten.

[Ben60] beschreibt unterschiedlichste Herangehensweisen an die Vermittlung der Kompositionspraxis.

- [Rit69] stellt Wissen verschiedener Autoren des 18. Jahrhunderts zu verschiedenen Themen vergleichend gegenüber. Sein Hauptthema ist zwar die Entwicklung der Sonatenform. Die zugehörigen Betrachtungen über die Tonordnung sind aber auch für diese Arbeit interessant.

2.1.2 Systematische Literatur

Systematische Literatur beschreibt musikalische Gesetzmäßigkeiten in einem durchformalisierten Begriffssystem. Sie sind von einem konkreten Stil unabhängig und werden zumeist auf induktivem Wege erhalten, d.h. der Autor stellt die Theorie auf und belegt deren Sinn durch Beispiele, deren Phänomene er durch seine Theorie erklären kann.

Beispiele für solche Theorien sind:

- Die übliche Harmonielehre [KD84], die die Harmonien als Stufen innerhalb einer Tonart notiert und Funktionen von Akkorden in der Tonart angibt.
- Die Schenkersche Ursatztheorie, die Musik als schrittweise Verfeinerung einer Urlinie beschreibt. Sie wurde beispielsweise von [Ebc86] verwendet.
- Beschreibung von Musik durch generative Grammatiken, wie sie in der Linguistik benutzt werden [LJ83].

Der Vorteil systematischer Theorien für die Verwendung in einem Computerprogramm ist, daß sie abstrakter und stärker formalisiert sind und daher einfacher und direkter zu implementieren sind. Diese Theorien werden meist durch Induktion aus Beispielen über verschiedene Stilepochen hinweg gewonnen. Daraus resultiert oft der Nachteil, daß die Theorie nicht der Denkweise und den Begrifflichkeiten der Komponisten der Stilepoche entspricht.

Nach der dieser Arbeit zugrunde liegenden Definition (vgl. Kap. 1.1) von künstlicher Intelligenz als *Systeme, die rational denken*, sind gute, stilistische Ergebnisse und die Nachvollziehbarkeit der Arbeitsweise gefordert, nicht jedoch die Ähnlichkeit mit menschlichen Vorgehensweisen. Systematische Theorien können (müssen aber nicht) nützliche Ergebnisse liefern, obwohl die Theorie nicht die korrekten, der Sache angemessenen Begriffe verwendet.

Bevor man eine systematische Theorie einsetzt, sind zwei mögliche Probleme zu untersuchen:

- Die Theorie kann für einen konkreten Stil unangemessen sein, d.h. die Ergebnisse sind unstilistisch.
- Die Theorie kann für die zu implementierende Aufgabe nutzlos sein.

Die in heutigen Harmonielehrebüchern zu findende Theorie beschreibt eine Art gemeinsamen Bereich der abendländischen tonalen Kompositionsstile. Dieser Bereich wird im Englischen manchmal als *common practice* bezeichnet. Die darin eingeschlossenen Stile umfassen ungefähr die Zeit vom Barock bis zum Ende der Romantik (ca. 1650-1900)[KD84]. Viele Phänomene eines konkreten Stils aus diesem Zeitraum kann man mit den Begriffen des *common practice* einigermaßen sinnvoll beschreiben.

2.1.3 Befragung von Experten

Experten für Stilkopie sind zumeist Kompositions- oder Tonsatzlehrer. Deren Befragung kann weiteres explizit formuliertes Wissen ergeben, das nicht in der Literatur beschrieben wurde. Es handelt sich dabei zumeist um Regeln, die nicht im Vordergrund des Interesses stehen, da ein Kompositionsschüler selten gegen sie verstößt. Die meisten Kompositionsschüler lernen sie nicht durch explizites Wissen oder Versuch und Irrtum, sondern durch Analogiebeobachtungen und ihr sich entwickelndes Stilgefühl.

Dennoch werden solche, dem Musikexperten absurd erscheinenden Stilfehler in seltenen Einzelfällen doch gemacht. Bei der Analyse und Korrektur solcher Fehler muß der Kompositionslehrer das Problem beschreiben und erwirbt selbst damit zusätzliches explizites Wissen.

2.1.4 Manuelle Analyse von Kompositionen

Die Analyse von Werken ist eine der wichtigsten Arbeitsmethoden der musikwissenschaftlichen Forschung. Die Analyse hat die Partitur (notierte Information) als Eingabe, sucht Metainformation über die Musik und beschreibt diese verbal oder formal. Beispiele für das Ergebnis von Analysen sind die Abfolge der Harmonien, die Form des Stücks (Gliederung in Teile), das Auftreten von Endformeln und Verwandtschaften zwischen Teilen.

Selbst bei einem kurzen Menuett kann eine solche Analyse nicht erschöpfend sein. Der Analysierende kann von den durch Analyse gewonnenen Erkenntnissen ausgehend immer weitere Abstraktionen und neue Interpretationen bilden. Er kann diese auch in seiner Analyse aufschreiben, da er immer neue Begriffe definieren kann, wenn die Terminologie, die er zur Beschreibung verwenden darf, nicht beschränkt³ ist. In der Musikwissenschaft gibt es eine sehr große Zahl möglicher Beschreibungskriterien und Aspekte.

So gibt es zwei grundlegende Vorgehensweisen:

Frei beschreibende Analyse Einerseits kann man ohne festgesetzte Zielrichtung die Stücke und die entdeckten Aufbauprinzipien beschreiben und versuchen, davon ausgehend Gesetzmäßigkeiten zu formulieren. Der Musikwissenschaftler interessiert sich dabei besonders für Strukturen und Phänomene, die ihm neu sind oder in diesem Zusammenhang ungewöhnlich erscheinen. Das Vokabular ist prinzipiell frei und die Form der gesuchten Gesetzmäßigkeiten ist nicht vorgegeben.

³Aus der Sicht der theoretischen Informatik ist die Terminologie beschränkt, da das Alphabet endlich ist und von Menschen geschriebene Texte aus praktischen Gründen nicht unendlich lang werden können. Man ist jedoch beim Verfassen natürlichsprachlicher Texte von dieser theoretischen Grenze sehr weit entfernt.

Gezielte Suche Andererseits kann man auch gezielt nach bestimmten Phänomenen suchen, um eine Hypothese durch Beispiele zu stützen oder durch Gegenbeispiele zu widerlegen.

Ein Beispiel für eine solche Hypothese wäre: *Ein Stück, das volltaktig beginnt, beginnt in der Melodie auf der 1. Stufe.*

Man kann nun Stücke gezielt daraufhin untersuchen und dabei evtl. auch statistische Verfahren verwenden.

Für die Anfangsphase der Arbeit war die erste Vorgehensweise, die frei beschreibende Analyse wichtig, um eine Vorstellung von den Teilaufgaben und den musikalischen Zusammenhängen zu erhalten. Während der Erstellung der Regeln war dagegen die gezielte Analyse nach konkreten Hypothesen hilfreich.

2.1.5 Computerunterstützte Analyse von Menuetten

Die frei beschreibende Analyse ist für den Rechner nicht möglich, da ein Rechner nicht mit freiem Vokabular arbeiten kann, sondern ein definiertes Zieldatenformat benötigt. Denkbar wäre es höchstens, zur Unterstützung eines Wissenschaftlers fest umrissene Teilaufgaben zu automatisieren. Für die zweite Form der Analyse, die gezielte Suche nach Phänomenen, ist der Computer besser geeignet. Dies gilt insbesondere dann, wenn die gesuchten Phänomene auf einfache Art formal beschreibbar sind.

Hier kann man sich die unterschiedlichen Stärken des Menschen und des Rechners zunutze machen. Beispielsweise ist es für einen Musikwissenschaftler einfach, eine schwache Kadenz (Absatz) zu finden, da er durch seine Erfahrung einen *“Blick”* dafür hat. Sie ist von ihrer musikalischen Bedeutung her ein Gliederungsmittel und ist für den Musikwissenschaftler *“auffällig”*.

Während man einige formale Eigenschaften einer schwachen Kadenz aufzählen kann, ist es schwierig, dafür eine rechneraugliche Beschreibung aufzustellen, die so vollständig ist, daß der Rechner weder falsche Positive liefert (schwache Kadenz erkennt, die keine sind), noch falsche Negative bestimmt (tatsächlich vorhandene Vorkommen übersieht).

Auf der anderen Seite gibt es Phänomene, die für den Menschen unauffällig sind (was dazu führt, daß ein Teil der Vorkommen übersehen wird), jedoch auf einfache Art formal beschreibbar sind, so daß der Rechner sie finden kann.

Die Verwendung des Rechners ist in diesen Fällen noch aus einem besonderen Grund vorteilhaft: Bei der Formulierung einer Anfrage hat der Mensch einen musikalischen Grund danach zu suchen und eine gewisse Vorstellung

von dem musikalischen Kontext, in dem ihn das Phänomen interessiert. Bei einer manuellen Analyse sucht der Mensch das Phänomen unbewußt in erster Linie in dem betreffenden musikalischen Kontext. Die Neigung des Menschen zu falschen Negativen, d.h. dazu, korrekte Auftreten des Phänomens zu übersehen ist wegen dieser Kontextabhängigkeit der Wahrnehmung kein zufälliger Fehler (den man durch Verwenden einer größeren Menge von Material beheben könnte), sondern ein systematischer.

Ein Beispiel für eine Anfrage, bei der ein Mensch leicht Vorkommen übersieht, ist: *Drei aufeinanderfolgende Noten mit Terzen oder Quarten dazwischen, wobei die mittlere Note stärker betont ist, als ihre Nachbarn.*

Wenn der Mensch nun die Anfrage formuliert und die Analyse vom Rechner durchführen läßt, dann erhält er Stellen geliefert, die korrekt erkannt wurden, von ihm aber nicht gemeint waren. In Bezug auf seine Intention sind sie falsche Positive, in Bezug auf die Anfrage sind sie korrekt. Betrachtet er nun das Ergebnis genauer, so stellt er die Diskrepanz zwischen seiner Intention und der wirklich gestellten Anfrage fest und kann letztere erweitern, d.h. seine Vorstellung vom musikalischen Kontext explizit angeben. So erhält man Wissen, das durch den Computer verifiziert ist, was für den Einsatz im Kompositionssystem besonders nützlich ist.

Zur Unterstützung dieser Arbeit wurde die automatisierte Analyse eingesetzt. Ein System mit einer Patternsprache zur automatischen Suche und die dafür nötigen Hilfsmittel zur Erfassung von Orginalkompositionen im Partitурkartenformat wurden im Rahmen der Diplomarbeit von Ina Becker [Bec00] geschaffen. Da viele interessante Suchmuster harmonische Informationen benötigen, wurde von Nicola Höhle [Hö00] die harmonische Analyse nach Maxwell [Max84] reimplementiert. Eine nähere Beschreibung der Arbeit von Maxwell findet man in Kap. 2.3.6.

2.1.6 Computereperimente

Ein weiterer Schritt bei der Gewinnung von Gesetzmäßigkeiten ist die systematische Untersuchung, welche Auswirkungen die Verwendung einer bestimmten Regel auf das Kompositionsergebnis des Systems hat. Die einfachste Untersuchung ist ein Vergleich der Ergebnisse, wenn das Kompositionsprogramm mit und ohne eine bestimmte Regel und denselben übrigen Parametern und Einstellungen komponiert. Diesen Vergleich kann man auch von einem Experten vornehmen lassen.

Wichtig ist insbesondere, die genaue Art und Ursache der beim Vergleich entdeckten Probleme zu berücksichtigen. Damit kann man die Regel verbessern, z.B. zusätzliche Voraussetzungen für deren Anwendung anbringen oder den Gültigkeitsbereich einschränken.

Eine Suchstrategie jedoch, die Regeln hinzunimmt, wenn sie das Ergebnis verbessern und sie wegläßt wenn nicht, ist problematisch, da es starke Wechselwirkungen gibt. So könnte im Falle zweier Regeln die Hinzunahme von nur einer Regel das Ergebnis verbessern, während sich bei gleichzeitiger Hinzunahme beider Regeln das Ergebnis verschlechtert.

2.1.7 Kognitive Introspektive nach Komposition

Bei der kognitiven Introspektive versucht man, Gesetzmäßigkeiten zu gewinnen, indem man jemanden, der ein Menuett komponiert hat (eventuell sich selbst), fragt, wie er dabei vorgegangen ist.

Besonders interessant ist die Betrachtung der Vorgehensweise und der Abhängigkeiten, d.h. in welcher Reihenfolge die Teilaufgaben angegangen wurden und welche Teile der bestehenden Komposition für eine bestimmte Entscheidung berücksichtigt wurden.

2.2 Musikalische Grundlagen

Zum Verständnis des Kompositionsverfahrens ist es nötig, vorab einige relevante Eigenschaften von Stil und Gattung und die Grundbegriffe des Tonsystems zu betrachten. Anschließend wird die Makrostruktur (Form) des Menuetts beschrieben und die Begriffe werden durch die Analyse eines Menuetts am Beispiel veranschaulicht. Die darüber hinausgehenden Einzelheiten des verwendeten Musikwissens werden dagegen erst zusammen mit ihrer Modellierung in Kapitel 3 beschrieben.

2.2.1 Hintergründe von Stil und Gattung

Die Zeit der Frühklassik ist von einem starken Wunsch nach Veränderung des Stils gegenüber dem Althergebrachten – der Epoche des Barocks – geprägt. Hintergrund des ästhetischen Wandels sind allgemeine geistesgeschichtliche Strömungen, nämlich einerseits die vernunftbetonten Ideale der Aufklärung und andererseits die Ziele der Empfindsamkeit, die den Ausdruck von Emotionen fordert.

Daraus folgt eine Reihe typischer Eigenschaften [Blu97]:

- Grundprinzip der Einfachheit und Klarheit der Gestaltung.
- Regelmäßiger formaler Aufbau des Stücks.
- Große Bedeutung der tonartlichen Anlage des Stücks.

- Homophones Satzideal, Melodie mit Begleitung.

Benary [Ben60] beschreibt diesen Wandel im Satzideal: *Der Generalbaß, der wie gesagt durch Ausführung der vokalen Baßstimme vom Kontrapunkt herkommt, wird nun der Oberstimmen-Periodik unterworfen und damit zum vier- oder achttaktig gegliederten Harmonieträger.*

Das Menuett ist ein mäßig schneller Tanz im Dreiertakt – im Normalfall im 3/4-Takt, in Einzelfällen auch im 3/8-Takt. Es war ab ca. 1650 am französischen Hof, später an allen westeuropäischen Höfen als ausgesprochen offizieller, formeller Tanz in Gebrauch. Die mit dieser Rolle im Hofzeremoniell verbundenen komplizierten Figuren und Schrittfolgen fordern von der Tanzmusik einen gut erkennbaren, regelmäßigen Aufbau. Die übliche Form der für den Tanz vorgesehenen Menuette besteht nach [uGM87] aus zwei achttaktigen Teilen, die jeweils in zwei viertaktige Teile gegliedert sind. Jeder der beiden achttaktigen Teile wird normalerweise wiederholt.

Aus zwei Menuetten in dieser Form wird das *Menuett mit Trio* aufgebaut. Es beginnt mit dem Hauptmenuett, dem ein zweites Menuett, das Trio, folgt. Das Trio wird gerne durch zurückhaltendere Gestaltung gegen das Hauptmenuett abgesetzt, beispielsweise bei Orchesterstücken kleiner besetzt (ursprünglich: Trio, weil es mit drei Spielern besetzt war). Manchmal setzt man das Trio in eine vom Hauptmenuett entferntere Grundtonart, gerne in eine im Quintenzirkel tiefer stehende. Dem Trio folgt wieder das Hauptmenuett, wobei beim zweiten Mal die Wiederholungen weggelassen werden.

Bei der Entwicklung der Kunstmusik im 17. und 18. Jhd. wurden immer wieder Ideen und Konzepte aus der Tanzmusik übernommen. Ein Beispiel ist im 17. Jhd. die Verwendung eines Takts mit Akzentstufungen⁴, welcher für die Musik ein immer wiederkehrendes System verschieden starker Schwerpunkte festlegt. Ebenso hat die für die Klassik wichtige Idee, ein Stück durch Zusammenfügen von gleichlangen, thematisch verwandten Teilen aufzubauen, Vorbilder in der Tanzmusik. Die Tanzmusik war somit Ideengeber für den gesamten Stil. Das Menuett, das eine wichtige Tanzgattung war, wurde zur Üb- und Demonstrationsgattung für Musikbeispiele in der Kompositionsliteratur der Klassik.

2.2.2 Tonsystem und Bezeichnungen

Die Bezeichnungen des Tonsystems und der Tonarten hatten sich zur Zeit der Klassik den heutigen Begriffen bereits weitgehend angenähert. Daher werden für die einfachen Grundelemente der Musik, wie Noten, Tonart und Akkorde

⁴Dies ist der heute übliche (common-practice) Begriff von Takt

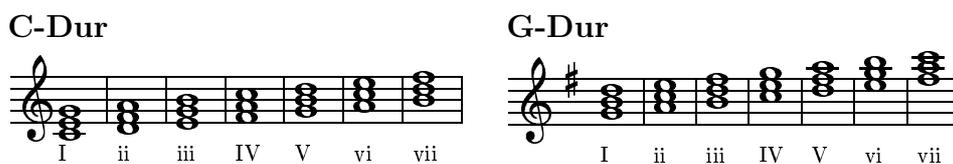


Abbildung 2.1: Akkorde auf den Stufen der Tonleiter

die modernen Bezeichnungen (common practice) verwendet. Sie werden hier nur überblicksweise beschrieben — Details findet man in den verschiedensten Harmonielehrebüchern z.B. [KD84]. Die der zeitgenössischen Literatur entnommenen Begriffe - vor allem im Bereich der Form - werden dagegen in dieser Arbeit etwas ausführlicher beschrieben.

Die Musik des klassischen Stils ist stark auf ihre Tonart, das tonale Zentrum bezogen. Das bedeutet, daß das Stück relativ zur Tonart gedacht ist und ein Stück damit im Prinzip in jede Tonart transponiert werden kann, sofern nicht der Tonumfang oder andere Beschränkungen der Ausführung auf einem Instrument dem entgegenstehen. Der größte Teil der Stücke steht trotzdem in Grundtonarten mit wenigen Vorzeichen, um dem Ideal der Einfachheit zu genügen [Blu97].

Noten und Akkorde werden mit Stufennummern relativ zur Tonart bezeichnet. Die Bezeichnung der Akkorde mit Stufennummern ist in der Klassik noch nicht üblich aber durchaus angemessen, da ein zu Abb 2.1 ähnliches Beispiel mit Akkorden auf Tonleiterstufen (ohne die 7. Stufe) bei Riepel [Rie52] zu finden ist, nur daß die Stufen bei Riepel keine Nummern tragen.

Für die klassische Epoche sind nur noch die Tongeschlechter Dur und harmonisches Moll relevant, der Einfluß der Kirchentonarten ist (jedenfalls außerhalb von bewußt historisierender Kirchenmusik) in der Klassik nicht mehr zu spüren, so daß die Repräsentation sich auf Dur und Moll beschränken kann.

Im Verlauf eines Stücks unterscheidet man (mindestens) zwei Ebenen der Tonart. Die *Grundtonart* gilt für das ganze Stück. Die Grundtonart wird durch die Vorzeichen zu Beginn des Stücks angegeben. Normalerweise beginnt und endet das Stück in der Grundtonart.

Die Stufe in der Tonart legt die Verwendungsmöglichkeiten für die Töne und Akkorde Verwendungsmöglichkeiten in der Tonart fest. Für die Akkorde haben sich sowohl in der zeitgenössischen als auch der systematischen Literatur Bezeichnungen *Tonika* für die erste, *Dominante* für die fünfte und *Subdominante* für die vierte Stufe eingebürgert.

Im Verlauf des Stücks kann zeitweise eine andere Tonart zur *aktuellen* Tonart werden. Wird diese andere Tonart durch eine Kadenz befestigt, so

spricht man von einer förmlichen Ausweichung, sonst von einer durchgehenden [Koc82]⁵. Solange diese andere Tonart als tonales Zentrum gültig ist, sind alle Töne als Stufen in dieser aktuellen Tonart zu interpretieren. Die Ausweichung erkennt man an den Abweichungen des Tonvorrats der aktuellen Tonart von dem der Grundtonart (hierbei werden zusätzliche Vorzeichen benötigt), insbesondere an der Einführung des Leittons (7. Stufe) der aktuellen Tonart.

Im klassischen Stil sind Verwandtschaftsverhältnisse zwischen der Grundtonart und den Tonarten, die für Ausweichungen verwendet werden, festgelegt. Da die Tonvorräte verwandter Tonarten viele gemeinsame Töne haben sind sowohl Beginn als auch Ende einer Ausweichung oft nicht exakt bestimmbar. Die Bedeutung der Rollen der verwandten Tonarten beschreibt Riepel [Rie52] durch Analogie mit den Aufgaben beim Gesinde eines großen Bauernhofs. Die Gewichtung dieser Rollen gibt an, wie häufig und an welchen strukturell wichtigen Stellen welche Tonarten verwendet werden. Die Tonarten werden durch die Stufe in der Grundtonart bezeichnet. Die wichtigste Ausweichung ist die zur 5. Stufe.

2.2.3 Aufbau eines Menuetts

In der klassischen Epoche spielt die Form, also die Makrostruktur und die Beziehungen zwischen den Formteilen eine wichtige Rolle. Die zeitgenössischen Literatur beschreibt die Form durch zwei Ordnungen:

- Die Taktordnung befaßt sich mit der Gliederung der Stücke in Abschnitte. Sie gibt die Anzahl der Teile in einem Takt und damit die Längenverhältnisse der Teile an.
- Die Tonordnung beschreibt die tonartlichen Verhältnisse innerhalb und zwischen den Abschnitten.

Ein abgeschlossener Abschnitt heißt in der zeitgenössischen Literatur *Satz*⁶. Die normale Länge des Satzes ist nach [Rie52] vier Takte. Der viertaktige Satz wird im folgenden (wie auch bei [Rie52]) kurz als *Vierer* bezeichnet. Sätze anderer Längen entstehen meist aus einem Vierer durch Einschub von Motiven oder Entfernen von Teilen.

⁵Der common-practice-Begriff ist *Modulation*.

⁶Achtung: Der Begriff *Satz* hat in der Musik außerdem noch weitere Bedeutungen, z.B. Satz einer Symphonie (selbständiger Teil eines zusammengesetzten Werks), Lied- oder Choralsatz (Unterstimmen zu einer Melodie), Saxophonsatz in der Bigband (Untergruppe eines Orchesters) uvm. Dafür findet man in der Literatur für die hier gemeinte Bedeutung gelegentlich andere Begriffe.

Kennzeichnend für einen Satz ist, daß er durch eine Schlußformel beendet wird, die auf den Schlussakkord des Satzes hinführt. Für die Tonordnung ist der Schlußakkord und die Stärke der Schlußformel wichtig. Die Schlußformeln teilt man grob in zwei Stärkegrade ein (Begriffe nach [Bud83]):

- Die *Kadenz* (zur Präzisierung auch volle oder starke Kadenz genannt) verlangt als vorletzten Akkord die Dominante (5. Stufe) in Grundstellung. Die Dominante wird gerne durch eine Subdominante bzw. deren Stellvertreter oder einen $\frac{6}{4}$ -Vorhalt auf der Dominante vorbereitet, was die Schlußwirkung der Kadenz noch verstärkt. Die Kadenz führt die Tonart ihres Schlußakkords ein. Weicht diese Tonart von der Grundtonart oder der bisher herrschenden Momentantonart ab, so ist mit der Kadenz die Tonart der Ausweichung befestigt.
- Der *Absatz* ist eine schwache Form der Kadenz. Die vorkommenden Akkorde müssen nicht in Grundstellung stehen.

Die Schlußformeln werden durch eine Kombination von 4 Merkmalen festgelegt: Melodie, Harmonik, Baßlinie und Betonungsverhältnisse der Akkorde. Üblicherweise werden Kadenzformen als prototypische Muster, *Kadenzformeln*, aufgeschrieben. Beispiele findet man in [Bud83].

Betrachtet man nun die Tonordnung, so interessiert man sich für die Tonarten der Schlüsse von Sätzen relativ zur Grundtonart. Kadenzen in der 1. Stufe der Grundtonart heißen *Grundkadenz* (GK), Absätze auf der 1. Stufe *Grundabsatz* (GA). Kadenzen auf der 5. Stufe (Dominante) heißen *Quintkadenz* (QK), Absätze auf der 5. Stufe *Quintabsatz* (QA). Die Kadenzformeln werden relativ zur Zieltonart angegeben.

Für die Menuette der Klassik gibt es nach [Bud83] eine prototypische⁷ Form, die eine Detailierung der vorhin beschriebenen Tanzmenuettform ist.

Die Form wird durch Längen und Schlußformeln der Sätze beschrieben. Die Abfolge der Sätze sieht folgendermaßen aus, wobei die beiden Hälften des Stücks (Satz 1+2 und Satz 3+4) jeweils wiederholt werden:

Satz 1, Takte 1-4 Der erste Satz beginnt in der Grundtonart und wird durch einen Grundabsatz geschlossen.

Satz 2, Takte 5-8 Der zweite Satz schließt mit einer Quintkadenz.

⁷Prototypisch heißt, daß diese Form ein Grundmuster für Menuette ist und die Form eines konkreten Menuetts meist durch Ersetzung von einzelnen Elementen durch verwandte Elemente entsteht. Prototypisch heißt auch, daß diese Form nicht nur eine theoretische Abstraktion aller Formen ist, sondern durchaus direkt als Form eines Menuetts vorkommen kann.

Satz 3, Takte 9-12 Aufgabe des dritten Satzes ist es, zurückzumodulieren. Dieser Satz bringt gerne musikalisch neues Material und harmonisch interessantere Teile.

Zwei Grundmuster sind nach [Rie52] gebräuchlicher Standard:

Fonte-Sequenz⁸ Ein (meist zweitaktiges) Motiv mit Dominante - Tonika-Beziehung beginnt so, daß es auf eine 2. Stufe hinführt, und wird eine Sekunde nach unten sequenziert. Damit endet der Satz auf einem Grundabsatz.

Monte-Sequenz⁹ Das Motiv mit Dominante-Tonika-Beziehung beginnt so, daß es auf eine 4. Stufe hinführt, und wird eine Sekunde nach oben sequenziert. Damit endet der Satz auf einem Quintabsatz.

Satz 4, Takte 13-16 Am Schluß des Stücks muß eine Grundkadenz stehen.

Häufig verwendet man hier Material aus dem 1. Satz (wenn man den Absatz zur Kadenz stärken kann) oder aus dem 2. Satz (wenn man den 2. Satz in die Grundtonart transponieren kann).

Von dieser Grundform sind nun verschiedene Abwandlungen möglich. Beispielsweise trifft man manchmal starke Kadenzen statt der Absätze an oder umgekehrt Absätze anstelle der Kadenzen. Sätze können durch verschiedene Mittel, z.B. „Anhang“ nach der Endformel, Wiederholung von Takten oder „Einschiebsel“¹⁰ fremden Materials verlängert werden[Bud83]. Es können auch Sätze hinzukommen, z.B. durch variierte Wiederholung oder nach einer ausgeflohenen Kadenz¹¹.

2.2.4 Analyse eines Menuetts

Die bisher beschriebenen theoretischen Begriffe werden nun an der Analyse von Mozarts Menuett KV2 [Moz91] erläutert.

Dieses Menuett ist 1762 entstanden, als der Komponist sechs Jahre alt war. Es steht – wie bei Menuetten üblich – im 3/4-Takt und beginnt volltaktig. Die Grundtonart ist F-Dur. Das Menuett umfaßt 24 Takte, wobei die ersten 8 Takte wiederholt werden. Berücksichtigt man die Wiederholung, dann hat man zwei Hälften mit zusammen 32 Takten, so daß die Gesamtlänge der prototypischen Form entspricht.

⁸*Fonte* ital./lat. In den Brunnen hinunter

⁹*Monte* ital./lat. Den Berg hinauf

¹⁰Bezeichnung von Riepel [Rie52]

¹¹common-practice Begriff *Trugschluß*



Urtext nach [Moz91] (ohne Fingersatz)

Abbildung 2.2: Menuett F-Dur KV 2 von W.A.Mozart (1762)

Das Stück ist folgendermaßen aus Vierern (viertaktigen Sätzen) aufgebaut:

T1-4 Auffällig ist eine eintaktige Dreiklangsbrechungsfigur mit einem Rhythmus von zwei Achteln und zwei Vierteln, die die Melodie des gesamten Stücks durchzieht. Sie steht im ersten Vierer dreimal absteigend, beginnend in der Tonika (F-Dur) mit f' abwärts über Subdominante und Dominante mit Quartsextvorhalt. Betrachtet man ganze Takte (d.h. Vorkommen der Grundfigur), dann zieht sich über den Vierer eine fallende Gesamtlinie, der im Baß (von der Kadenzformel abgesehen) eine steigende entgegensteht. Der Vierer wird durch eine Grundkadenz mit Septimvorhalt $e' - f'$ auf dem Tonikaakkord beschlossen.

T5-8 Der zweite Vierer ist eine Art Umkehrung sowohl in der Figur, bei der die beiden Viertelnoten im Gegensatz zum ersten Vierer höher sind als die Achtel, als auch in der Gesamtlinie, da die Figur über die Takte ansteigt.

Der Vierer wird durch einen Quintabsatz geschlossen. Während melodisch eine deutliche Schlußwirkung erreicht wird - durch die Triole statt der beiden Achtel der Grundfigur und den fallenden Vorhalt - - wird nicht nach C-Dur moduliert (d.h. kein Leitton h , der C-Dur befestigen würde, sondern b in Takt 7). Statt des üblichen Schlusses

5. Stufe - 1. Stufe endet der Vierer auf 4. Stufe - 1. Stufe (bezogen auf Schlussakkord C-Dur).

T9-12 Im dritten Vierer werden die fallende Grundfigur und die fallende Gesamtlinie des ersten Vierers wieder aufgegriffen. Neu ist die Harmonik: Ein verminderter Septakkord führt als Zwischendominante nach g-moll. Die Kadenz, deren Form der Kadenz des ersten Vierers ähnelt, befestigt das g-moll deutlich.

T13-16 In einer Fontesequenz wird der dritte Vierer eine Sekunde nach unten sequenziert. Auch hier startet der Vierer mit einem verkürzten Septakkord und moduliert so sofort wieder zur Grundtonart F-Dur zurück.

T17-20 Der fünfte Vierer wiederholt den ersten Vierer und folgt damit einer Gewohnheit der prototypischen Form. Die Baßlinie ist gegenüber dem Anfang etwas vereinfacht. Dieser Vierer endet in einer ausgeflohenen Kadenz (6. Stufe statt der 1. Stufe).

T21-24 Der sechste Vierer entspricht dem fünften, endet aber in einer Grundkadenz auf der Tonika.

Die Melodie des Stücks ist einheitlich auf Grundlage von Dreiklangsbrechungen und unter Verwendung der oben genannten eintaktigen rhythmisch-melodischen Grundfigur gestaltet. Die Schlüsse der Vierer tragen alle einen Vorhalt mit Halbe-Viertel-Rhythmus. Bei den Kadenzen (alle Vierer außer dem zweiten) ist es immer ein Septimvorhalt, bei dem der Leitton zur Oktave aufgelöst wird. Die Vorhalte sind alle vorbereitet.

Der Baß ist recht ruhig gestaltet. Die Harmonien werden taktweise verwendet. Ausnahme sind die dritten Takte aller Sätze, wo zur Vorbereitung des der Kadenz bzw. des Absatzes zwei Harmonien im Takt sind. An diesen Stellen findet man teilweise Viertelbewegung im Baß. Auf den Schlußtakt der Sätze steht die bei Menuetten verbreitete fallende Baßendformel Oktave-Quinte-Grundton. Vorne in den Vierern herrscht dagegen der Halbe-Viertel-Rhythmus oder punktierte Halbe vor. Der Baß enthält meist Grundtöne, an einigen Stellen Terzen.

Die Form dieses Menuetts ist gegenüber der prototypischen Menuettform in der zweiten Hälfte erweitert. Statt einer Wiederholung der zweiten Hälfte sind hier die beiden Teile der zweiten Hälfte jeweils variiert verdoppelt: Die Rolle des modulierenden Vierers der prototypischen Form übernimmt hier die Fonte-Sequenz (Takte 8-15) zweier Vierer. Durch den Trugschluß beim ersten Vorkommen wird auch die Wiederholung des schließenden Vierers sinnvoll motiviert.

Ungewöhnlich sind bei diesem Stück die Stärkeverhältnisse der Schlußformeln, denn alle Vierer bis auf den zweiten schließen mit einer Kadenz. Damit ist der Schluß des zweiten Vierers schwächer als der des ersten. Nach der prototypischen Form würde man hier dagegen einen stärkeren Schluß erwarten. Der Effekt ist, daß der Gesamteindruck besonders durch die Reihung der Vierer geprägt wird, die durch ihre starken Endformeln und die Wiederholungen besonders hörbar wird. Vom dritten Vierer abgesehen stehen alle vollen Kadenzen in der Grundtonart. Diese einfache tonartliche Disposition stärkt das Gewicht der Grundtonart.

2.3 Verwandte Arbeiten

Die Komposition von Musik durch Computer umfaßt ein weites Arbeitsgebiet. Die folgende Beschreibung konzentriert sich daher auf Aufgabenstellungen, die mit der Menuettkomposition soweit verwandt sind, daß sie durch Transfer von Resultaten oder Vorgehensweisen für diese Arbeit von Nutzen sein können.

Maxwells System zur harmonischen Analyse ist analysierend, die anderen Systeme generieren eine Stilkopie. Alle Aufgaben arbeiten auf Notenebene, d.h. sie generieren bzw. analysieren Noten (Partitursynthese, bzw. -analyse), keine Klänge (Klangsynthese bzw. -analyse). In einigen Abschnitten sind kleinere, weniger bekannte Arbeiten mit erwähnt, die die gleiche Aufgabenstellung bearbeiten oder ähnliche Techniken benutzen.

Einen Überblick über den gesamten Bereich der Computerkomposition inklusive der formalen Verfahren aus der Zeit vor dem Computer und den frühen Computerverfahren (z.B. [HI59]) findet man in [Loy89].

2.3.1 Kriterien zum Vergleich von Kompositionsaufgaben

Die folgenden Beschreibungen der verwandten Arbeiten betrachten die Eigenschaften der Aufgabenstellung und besondere Merkmale des Verfahrens. Hier wird u.a. auf folgende Eigenschaften geachtet:

- Der *Freiheitsgrad* des Stils ist unterschiedlich. Bei stark beschränkten Aufgaben ist die Anzahl der korrekten Möglichkeiten eingeschränkt.

Außer der reinen Anzahl der Möglichkeiten ist außerdem zu berücksichtigen, inwiefern die unterschiedlichen Möglichkeiten als musikalisch verschieden empfunden werden. Beispielsweise sind beim Aussetzen ei-

nes Generalbasses oft mehrere Lagen¹² eines Akkords möglich. Da sich diese jedoch ähnlich anhören, wird man die verschiedenen Lösungen als gleich oder ähnlich einstufen und die Aufgabe dann trotzdem als eher beschränkt ansehen.

- Aufgaben können strukturelle Aspekte berücksichtigen oder nur einzelne Noten betrachten.
- Dem System kann musikalisches Material vorgegeben sein oder nicht.
- Das Verfahren kann musikalisches Wissen explizit modellieren oder implizit enthalten.
- Verfahren, die Wissen verwenden, greifen dazu auf Erkenntnisse der Musiktheorie zurück. Hier kann zeitgenössische oder moderne systematische Literatur verwendet werden.
- Neben dem übergeordneten Ziel, ein möglichst gutes System zu erstellen, können unterschiedliche zusätzliche Ziele verfolgt werden.

2.3.2 Vierstimmiger Satz (Ebcioğlu und andere)

Das System CHORAL von Ebcioğlu [Ebc86] setzt eine gegebene Choralmelodie vierstimmig im Stil von J.S. Bach aus (d.h. es komponiert die drei unteren Stimmen). Obwohl Ebcioğlu eine Zahl von ca. 10^{300} möglichen Harmonisierungen für eine Choral nennt, ist die Aufgabe eher als stark beschränkt einzustufen, da vielfach Lösungen als nicht sehr verschieden empfunden werden. Beim Choralsatz sind Kriterien für stilistisch und satztechnisch korrekte Lösungen vorhanden und gut dokumentiert. Trotzdem ist die Aufgabe nicht auf einfache Art mechanisch durchzuführen, da unterschiedliche, getrennt beschriebene Gesetzmäßigkeiten zu berücksichtigen sind.

Ebcioğlu repräsentiert das musikalische Wissen in Regeln, die in der von ihm entworfenen und implementierten backtracking-Sprache BSL formuliert sind. Der BSL-Compiler – ein Lisp-Programm – erzeugt aus den Regeln C-Code, so daß auch auf der damaligen Hardware eine einigermaßen effiziente Ausführung möglich war. Um über die satztechnischen und harmonischen Regeln hinaus den Stil von Bach gut anzunähern, sind einige, sich über mehrere Töne erstreckende Verwendungsmuster (Klauseln) zu benutzen. Die Regeln bestehen aus Kompositionsregeln, die Noten hinzufügen, Beschränkungen (Constraints), und Heuristiken, die angeben, was vorzuziehen ist, jedoch verletzt werden können, falls es keine andere Lösung gibt.

¹²Verteilung der Akkordtöne auf die Stimmen, insbesondere Wahl des Soprantons

Das musikalische Wissen in den Regeln ist in vier Sichten auf die Musik gruppiert: die Akkordskelettsicht, die den Choral als Folge von Harmonien betrachtet, die Figurenkontrollansicht (*fill-in-view*), die für die korrekte Verwendung von Figuren (z.B. Stimmführung, Auflösung von Dissonanzen, Klauseln ...) sorgt, die melodische Sicht, die auf die Melodik der einzelnen Stimmen achtet und die Schenkersche-Analyse-Sicht¹³.

Der vierstimmige Choralsatz ist eine der Grundaufgaben im Kompositionsunterricht, da dabei die Beachtung verschiedener Randbedingungen erlernt wird, während strukturelle Fragen durch Vorgabe der Choralmelodie vorerst ausgeklammert bleiben. Der Vorteil der Aufgabenstellungen ist außerdem, daß sie musikwissenschaftlich gut erforscht ist. Nach Ebcioğlu bahnbrechender Arbeit wurde der Choralsatz daher auch zu einem Experimentierfeld für Folgearbeiten mit verschiedenen Programmstrategien.

Ein kleines System zur Choralharmonisierung im Kantionalsatz des 17. Jahrhunderts unter Verwendung von (an die Linguistik angelehnten) generativen Theorien [LJ83] entstand in Karlsruhe [Lan98]. Diese Arbeitsgruppe hat auch Arbeiten auf der Basis von neuronalen Netzen durchgeführt.

Nachdem der Bereich des vierstimmigen Choralsatzes gründlich untersucht ist, dient er auch als Experimentierfeld, um die Eignung verschiedener Strategien aus der künstlichen Intelligenz für die tonale Komposition zu untersuchen. Wiggins [WPA99] vergleicht die Leistungsfähigkeit eines regelbasierten Verfahrens und eines genetischen Algorithmus für den vierstimmigen Satz von Melodien, wobei er sich nicht auf Choräle beschränkt. Der genetische Algorithmus kombiniert die bisher besten Hypothesen mechanisch, ohne Musikwissen zu verwenden, zu neuen Hypothesen. Die neuen Hypothesen bewertet der Algorithmus mit einer Bewertungsfunktion, die das Musikwissen des Systems enthält.

Dieses Verfahren findet in stark festgelegten Bereichen des Stücks, z.B. im Bereich der Kadenzten, keine gute Lösung, obwohl die Bewertungsfunktion dort korrekt bewertet (gute Lösungen hoch, schlechte niedrig), also das Musikwissen in Ordnung ist. Diese Arbeit zeigt die Bedeutung der Suchstrategie für die Komposition.

In [Wö95] beschreibt G. Wötzer ein System, das einen Generalbaß vierstimmig aussetzt. Diese Aufgabe ist von den zu beachtenden Problemen her (Satztechnik, vorgegebenes musikalisches Material) mit dem Choralsatz verwandt. Da die Harmonie durch die Bezifferung ebenfalls vorgegeben ist, ist die Aufgabe noch stärker eingeschränkt als der Choralsatz. Als wichtigste Wissensquelle wird dabei die zeitgenössische Literatur des frühen 18. Jahrhunderts (z.B. [Mat39]) herangezogen.

¹³Aufteilung in Sichten hier nach dem ersten Kapitel von [Ebc86]. Die Auflistung der Regeln im Anhang unterteilt die Sichten noch etwas feiner.

Obwohl das Wissen in der Beschreibung des Systems explizit, regelmäßig formuliert ist, findet im Programm keine Trennung zwischen Inferenzmaschine und Wissen statt. Stattdessen sind diese Regeln in C-Funktionen ausprogrammiert.

2.3.3 Kontrapunkt im Palestrinastil (Schottstaedt)

Schottstaedt [Sch89] arbeitet ebenfalls wissensbasiert mit Regeln. Die Kompositionsregeln bewerten die möglichen Alternativen mit Strafpunkten. Im Verlauf der Komposition werden die Strafpunkte aufsummiert. Das System sucht nach einer Lösung mit möglichst wenig Strafpunkten.

Als Wissensquelle für den Palestrinastil (Kontrapunkt des 16. Jahrhunderts) verwendet Schottstaedt zeitgenössische Literatur, den *doctor gradus ad parnassum* von J.J.Fux (1725)¹⁴.

Die Aufgabe hat einen größeren Freiheitsgrad als der vierstimmige Choralatz. Es können Kontrapunkte verschiedener Grundtypen (1 gegen 2, 1 gegen 4, frei ...) zu einem cantus firmus erstellt werden. Ebenso scheint¹⁵ es möglich den cantus firmus vom System erstellen zu lassen, so daß das System von Schottstaedt in diesem Fall ohne musikalische Eingaben auskommt. Unter den Kompositionsaufgaben ohne musikalische Eingabe ist der Kontrapunkt die am meisten beschränkte Aufgabe.

Es gibt beim Kontrapunkt nur die Sicht auf melodische Fortschreitungen in einer Stimme und von zwei Stimmen. Strukturelle Beziehungen oder Form spielen beim Palestrinastil keine Rolle. Ebensowenig gibt es den Begriff der Harmonik. Die Phänomene, die man nach heutiger Auffassung durch Harmonik beschreibt, werden in der zeitgenössischen Kompositionsliteratur ebenfalls in Stimmführungsregeln ausgedrückt.

2.3.4 Komposition klassischer Klaviersonaten (Berggren)

Die bisher beschriebenen Arbeiten haben bewußt Aufgaben mit geringem Freiheitsgrad ohne Berücksichtigung struktureller Fragen gewählt. Den umgekehrten Ansatz verfolgt die Arbeit von Berggren¹⁶ [Ber95, Lar92]. Sein

¹⁴Streng genommen ist Fux nicht mehr zeitgenössische Literatur, da er erst zu Beginn des 18. Jahrhunderts entstanden ist und bei genauer Betrachtung Einflüsse des harmonischen Denkens der Generalbaßzeit zu erkennen sind. Trotzdem hatte Fux die Absicht, den Stil von Palestrina originalgetreu darzustellen.

¹⁵Es ist aus [Sch89] nicht klar zu erkennen.

¹⁶alias Ulf Larnestam

System komponiert klassische Sonatensätze für Klavier. Sein Hauptinteresse gilt der Form (Makrostruktur).

Er teilt den Kompositionsprozeß in die folgenden drei Hauptdetaillierungsphasen auf, die jeweils wieder in Unterphasen aufgeteilt sind.

- Erzeugung des Zielplans: Hier wird die Gliederung des Stücks bis auf die Ebene der sogenannten Bausteine¹⁷ – deren Länge beträgt normalerweise ein oder zwei Takte – erzeugt. Für die Bausteine wird die Länge, ihre Aufgabe in der Gesamtform des Stücks und die Textur festgelegt.
- Erzeugung des Detailplans: Hier werden für jeden Baustein die Harmonik und Parameter für die Melodie und den Rhythmus festgelegt.
- Komposition des Stücks, d.h. Erzeugung der Noten.

Die Ergebnisse jeder Hauptphase werden in einer Datei gespeichert.

Das Ziel ist, die Form klassischer Sonatensätze zu beherrschen. Der Schwerpunkt der Arbeit ist deswegen die Makrostruktur. An den Nahtstellen zwischen den Bausteinen entdeckt man deswegen verschiedene deutlich hörbare Seltsamkeiten und Ungeschicklichkeiten.

Der Freiheitsgrad von Berggrens Arbeit ist größer als bei den bisher beschriebenen Arbeiten, da sowohl verschiedene Aufbauschemata, harmonische Verläufe und unterschiedliche Texturen für Begleitung und Melodie möglich sind.

Berggren verwendet sowohl moderne, systematische Theorien, z.B. die Harmonielehre und die Begriffe der Sonatenhauptsatzform als auch die zeitgenössische Literatur der klassischen Epoche. Neben dem Ziel, eine möglichst gute Stilkopie zu erhalten und Wissen über klassische Formen zu vertiefen und experimentell zu präzisieren, interessiert Berggren sich auch für kognitive Aspekte der Komposition, d.h. dafür, wie ein Komponist bei der Komposition eines Sonatensatzes vorgeht.

2.3.5 Patternbasierte Stilkopie (Cope)

Das System von Cope [Cop91, Cop95] komponiert Stücke in beliebigem Stil, nachdem es aus einigen ähnlichen Originalkompositionen Intervall- und Rhythmusmuster entnommen hat. Das System verwendet eine sehr einfache Repräsentation des Datenmaterials als Liste von Tonhöhen und Liste von Dauern. Das Trainingsdatenmaterial wird vom Autor speziell ausgewählt. Bei der Komposition einer Bachinvention wurden beispielsweise Stücke mit

¹⁷Original: *building blocks*

gleicher Taktart und Machart (Sechzehnteltonleitern, kontrapunktisch) ausgewählt und die Musterlängen nach typischen Einheiten der musikalischen Struktur des Ausgangsmaterials festgesetzt. Es ist hier nicht einmal nötig, die Musterlänge durch die Gesamtdauer des Motivs (beispielsweise eine halbe Note) anzugeben, sondern es genügt, nach der Anzahl der Töne vorzugehen. Das o.g. Beispiel mit den Bachinventionen arbeitet mit Motivlänge 4, wobei sowohl 4 Sechzehntel als auch 4 Achtel eine sinnvolle Sinneinheit ergeben.

Die Auswahl des Materials, die Einstellung der Parameter und die Auswahl der Ergebnisse, die Cope bei seinen Experimenten selbst vornimmt, sind für die Qualität der Ergebnisse ein entscheidender Faktor.

Die vom System komponierten Stücke hören sich streckenweise ziemlich original an, da Originalmaterial verwendet wurde. Da das System nur wenig musikalisches Wissen verwendet, sind an einzelnen Stellen sehr deutliche Fehler zu hören. Als einziges musikalisches Wissen hat Cope einfaches heuristisches Wissen über Harmonik eingebaut, da einerseits harmonische Zusammenhänge beim Zerkleinern des Ausgangsmaterials verloren gehen, andererseits Fehler im harmonischen Bereich bei tonaler Musik besonders auffällig sind. Das System arbeitet also imitierend (verwandt mit trainierenden Techniken, z.B. neuronalen Netzen), kombiniert diese aber an einigen Schlüsselstellen mit wissensbasierten Techniken. Diese Kombination ist mit dem Lernprozeß eines menschlichen Komponisten verwandt, der Stilkopien teils durch Studium von Beispielen und Analogieschlüssen, teils durch Verwendung theoretischer Regeln komponiert.

Cope, der von seinem Werdegang her Komponist ist, möchte mit mäßigem Programmieraufwand Stücke vom Computer komponieren lassen, die er nachher einer Art öffentlichem Turingtest¹⁸ stellt. Er führt sie bei öffentlichen Aufführungen zusammen mit Originalkompositionen auf und fragt Konzertbesucher, welches Stück vom Computer und welches von Komponisten der entsprechenden Stilepoche geschrieben wurde. Cope hat nicht das Ziel, musikwissenschaftliche, kognitive oder informatische Forschung zu betreiben.

2.3.6 Harmonische Analse (Maxwell)

Die harmonische Analyse tonaler Musik ist im Gegensatz zu den bisher beschriebenen Arbeiten keine Aufgabe der stilkopierenden Komposition, sondern ein System, das bestehende Musik analysiert. Maxwells System zur harmonischen Analyse tonaler Musik [Max84]¹⁹ wird hier betrachtet, weil es Re-

¹⁸Beim Turingtest führt ein Betrachter an einem Terminal einen Dialog und muß danach entscheiden, ob er sich mit einem Computerprogramm oder einem Menschen unterhalten hat.

¹⁹Kurzbeschreibung davon: [Max92]

präsentationstechniken verwendet und harmonische Probleme behandelt, die auch für diese Arbeit relevant sind.

Maxwells System faßt ein Stück als Folge von Zusammenklängen auf. Ein Zusammenklang kann entweder ein selbständiger Akkord sein, d.h. eine harmonische Funktion im Stück haben, oder als Nebennote (z.B. Durchgang) einem Akkord untergeordnet sein. Nach der Untersuchung des Tonvorrats eines Zusammenklangs wird zuerst diese Zuordnung der Zusammenklänge zu Akkorden vorgenommen. Anschließend werden die Akkorde nach ihrem Dissonanzgrad eingestuft und Beziehungen zu den Nachbarakkorden bestimmt. Damit kann die Abfolge der augenblicklichen Tonarten (die sich durch Modulationen ändern können) und die Funktionen der einzelnen Akkorde in der augenblicklichen Tonart bestimmt werden.

Maxwell verwendet als Repräsentation des Stücks eine Partiturliste, in der die Noten in Stimmen angeordnet sind. In jeder Stimme sind die Noten jeweils mit ihren Vorgängern und Nachfolgern verbunden, so daß eine doppelt verzweigte Liste entsteht. Außerdem sind über das Zusammenklangsobjekt die gleichzeitig erklingenden Noten miteinander verbunden. Die von Maxwell verwendete Repräsentation des Stücks in einer Partiturliste und die Darstellung von Harmonien ist ein wichtiges Vorbild für die in dieser Arbeit verwendete Partiturliste.

Maxwell möchte eine möglichst angemessene harmonische Analyse erhalten. Er möchte allgemeine tonale Musik (im Sinne der *common practice*) analysieren und sich auf keinen speziellen Stil festlegen. Deswegen verwendet er die Stufenbezeichnungen der systematischen Harmonielehre. Funktionen behandelt er nicht explizit.

Maxwells System, das in Franz-Lisp implementiert ist, wurde im Rahmen einer Studienarbeit [Hö00] in Common Lisp reimplementiert. Die musikalische Repräsentation wurde dabei auf das in dieser Arbeit verwendete Format COMF²⁰ umgestellt. Maxwells Analysealgorithmus wurde dann mit Musik verschiedener Stile innerhalb des *common-practice*-Bereichs getestet. Es stellte sich heraus, daß Maxwells Analyseregeln auf spätbarocke²¹ Harmonik ausgerichtet sind und diese auch in schwierigen Fällen gut analysieren können. Bei anderen Stilen wird gelegentlich ein Wechsel der augenblicklichen Tonart nicht richtig erkannt, was der Analyse insgesamt schadet.

Eine weitere Nachfolgearbeit zu Maxwell ist die *Music Theory Workbench* von Heinrich Taube²² [Tau99]. Die Analyse hat hier das Ziel, Fehler in studentischen Übungskompositionen zu entdecken und zu beschreiben, so daß

²⁰vgl. Kap 3.2

²¹Bach, Händel und Zeitgenossen

²²Nicht zu verwechseln mit Taubes bekanntem System *Common Music* [Tau94], einem System zur algorithmischen Komposition.

im Gegensatz zu Maxwell das Gewicht nicht auf der möglichst perfekten Analyse von Kompositionen, die u.U. harmonisch komplex sind, liegt, sondern auf einem möglichst zuverlässigen Funktionieren bei unvollständigen Kompositionen, ein angemessener Umgang mit Fehlern sowie eine Erläuterung der Fehler für Studenten. Deshalb ist bei Taubes System die graphische Darstellung der Noten und der Analyseergebnisse wichtig, z.B. soll das System Satzfehler in dem vom Studenten eingegebenen Notenbild markieren.

Kapitel 3

Das Verfahren zur Komposition von Menuetten

Dieses Kapitel gibt zuerst einen Überblick über das Kompositionsverfahren und erläutert die Grundideen, die alle Teile des Verfahrens betreffen. Anschließend wird für die einzelnen Komponenten des Verfahrens die Arbeitsweise und das enthaltene Musikwissen beschrieben. Einblicke in die programmiertechnische Umsetzung des hier vorgestellten Verfahrens gibt dann Kapitel 4.

3.1 Aufbau des Verfahrens

Das Verfahren läßt sich grob in drei Schichten einteilen (siehe Abbildung 3.1). Weitere Details über den Systemaufbau findet man in Abbildung 4.1 in Kap. 4.1.3.

- Die oberste Ebene besteht aus der Abfolge der drei Teilaufgaben der Menuettkomposition, nämlich Erstellung der Strukturbeschreibung, Melodiekomposition und Baßkomposition (vgl. Kapitel 1.2). Sie kombinieren die verschiedenen Wissenskomponenten des Verfahrens.

Die sequentielle Folge von Melodie- und Baßkomposition ist von [Rie52] gedeckt (WQ1). Er hält Änderungen an der Melodie zugunsten einer Verbesserung der Baßlinie nicht für angemessen. Daher kann man so vorgehen, daß für einen Abschnitt des Stücks zuerst die Melodie und anschließend der Baß erstellt wird.

- Die mittlere Schicht stellt den o.g. Teilaufgaben Mechanismen zur Bearbeitung der verschiedenen musikalischen Aufgaben zur Verfügung.

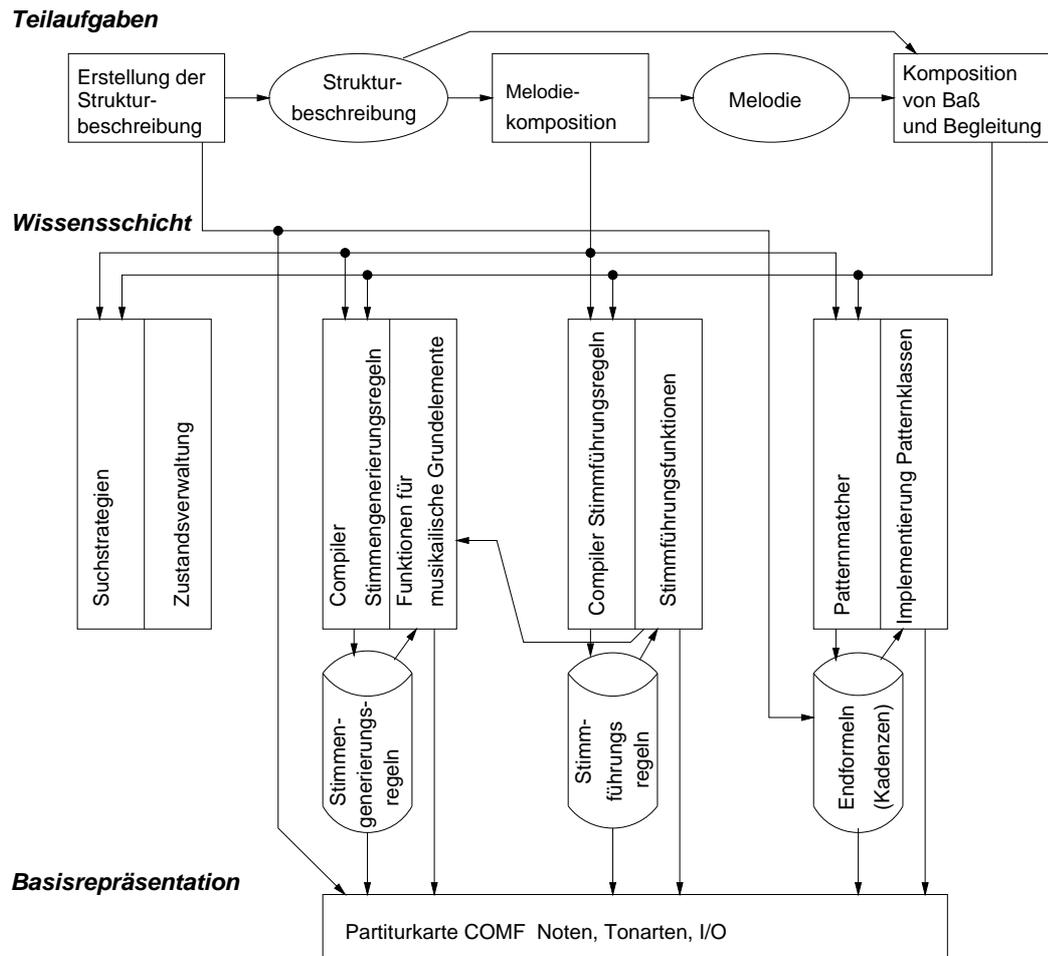


Abbildung 3.1: Überblick über das Verfahren

Die Komponenten dieser Schicht setzen das musikalische Wissen ein. Sie bestehen aus jeweils einem Interpreter, der die Repräsentationssprache definiert, den Funktionen, die die Grundelemente implementieren und der Wissensbasis.

- Die Repräsentation und die Darstellung der musikalischen Grundbegriffe und der Noten in der Partiturnote werden von allen Komponenten des Kompositionsverfahrens verwendet.

Die Auswirkungen von einzelnen Wissensselementen und die wechselseitigen Abhängigkeiten zwischen Entscheidungen in den einzelnen Teilaufgaben sind im Voraus nicht zu überblicken. Deswegen ist es nicht möglich, vor Beginn der Implementierung die Funktionsweise der einzelnen Systemkomponenten vollständig zu spezifizieren. Es ist vielmehr nötig, mit dem lauffähigen Programm zu experimentieren und dann auf das Ergebnis reagierend Änderungen vorzunehmen (Wissenquelle WQ6). Diese Technik wird auch als *exploratives Programmieren* bezeichnet. Bei diesen Änderungen kann es sich um kleine Anpassungen oder Erweiterungen handeln, die ohne Modifikation grundlegender Strukturen möglich sind. Es kann sich aber auch um tiefgreifende Änderungen handeln, wie z.B. eine vollkommen andere Such- oder Bearbeitungsstrategie. Außerdem ist zu berücksichtigen, daß eine Menge von kleinen Änderungen in der Summe einen starken Effekt haben kann (z.B. eine deutlich verringerte Dichte der Lösungen in einem Suchraum), so daß aus praktischen (z.B. Rechenzeit-) Gründen eine andere Grundstrategie notwendig ist.

Der Aufbau des Systems muß also soviel Flexibilität anbieten, daß auch gravierende Änderungen der Arbeitsweise des Verfahrens möglich sind, ohne daß man die Arbeit an der Modellierung der musikalischen Inhalte dafür lange unterbrechen muß.

Folgende Vorgehensweisen wurden eingesetzt, um die nötige Flexibilität zu erzielen:

- Einführen von expliziten Repräsentationen und Modellen für Abläufe, d.h. Auftrennen des Verfahrens in deklaratives Wissen und Inferenzmaschine. Als deklarative Formalismen werden bei dieser Arbeit Regeln und Muster eingesetzt.

Dieses Vorgehen erlaubt es, mit demselben Grundverfahren (Inferenzmaschine) und unterschiedlich zusammengestelltem Wissen zu experimentieren.

- Verwendung von Repräsentationen, die musikalisch angemessene Begriffe verwenden und gut lesbar sind. Wenn die Repräsentationen die in

der Musiktheorie üblichen Grundbegriffe angemessen nachbilden, kann man die Systemkomponenten in unterschiedlichem Zusammenhang verwenden, weil die Bedeutung und damit die Verwendungsmöglichkeiten klar definiert sind. Somit gewinnt man die geforderte Flexibilität.

Der Vorteil von gut lesbaren Darstellungen ist außerdem, daß es für den Menschen einfach ist, die Ergebnisse des Verfahrens zu untersuchen. Dies erleichtert die Fehlersuche.

Ausgangspunkt für die Repräsentationen ist die Terminologie in der zeitgenössischen und der systematischen Literatur (WQ1 und WQ2).

- Die Speicherung von Zwischenschritten der Berechnungsprozesse in expliziten Datenstrukturen erlaubt es, Berechnungsprozesse nachzuvollziehen und erleichtert somit ebenfalls die Fehlersuche.
- Für die Inferenzmaschine kommt ein Interpreter in Frage, der die Darstellung des Wissens bei jeder Ausführung durchwandert oder ein Compiler, der zu Beginn aus dem Wissen Lisp-Code erzeugt, welcher dann ausgeführt wird. Ein System mit Compiler ist schneller als eines mit Interpreter, ersterer ist aber auch aufwendiger zu erstellen.

Für die meisten Teile des Menuettkompositionssystems ist Effizienz zweitrangig. Nur für einige besonders häufig ausgeführte Programmteile wird ganz gezielt auf Effizienz geachtet.

Die Umsetzung dieser Vorgehensweisen in Prinzipien für den technischen Aufbau wird in Kapitel 4.1 beschrieben.

Die Verwendung expliziter Repräsentationen erlaubt es auch, wie eingangs gefordert, die Auswirkungen von einzelnen Wissens-elementen auf die Komposition zu beobachten.

Die Komponenten des Verfahrens sind - wie man in Abbildung 3.1 sieht - in drei Schichten angeordnet. Es folgt nun ein kurzer Überblick über die Aufgaben der Komponenten und ihre Beziehungen gegeben. Der Rest des Kapitels beschreibt die einzelnen Komponenten dann detaillierter.

Kompositionssteuerung Der Kompositionsprozeß ist in eine Folge von Einzelschritten unterteilt. Die Kompositionssteuerung arbeitet den vorgegeben Kompositionsplan ab, indem sie für die einzelnen Schritte die entsprechenden Funktionen der mittleren Schicht aufruft. Bei der Stimmengenerierung ist ein Kompositionsschritt die Komposition einer Stimme für einen Satz. Details siehe Kap. 3.3.1.

Die von außen vorgegebenen Parameter für den Kompositionslauf sind im *Regelsatz* festgelegt.

Erstellung der Strukturbeschreibung Bei der Erstellung der Strukturbeschreibung werden strukturelle Vorgaben für die folgenden Kompositionsschritte gemacht. Diese Angaben werden dann bei der Erzeugung der Stimmen berücksichtigt. Details siehe Kap. 3.3.3f.

Melodiekomposition Die Melodiekomposition ist eine Stimmengenerierungsaufgabe. Sie greift auf die Angaben aus dem Regelsatz und aus der Strukturbeschreibung zu. Das Verfahren benutzt die Suche im Zustandsraum (Kap. 3.4), die Regeln zur Stimmengenerierung (Kap. 3.6) und den Mustervergleich für die Endmustersuche (Kap. 3.8).

Ergebnis ist die Melodie, zu der noch strukturelle Information tritt, die bei der Melodiekomposition entstanden ist (Kap. 3.5.3).

Baßkomposition Die Baßkomposition ist ebenso wie die Melodiekomposition eine Stimmengenerierungsaufgabe. Eingabe der Baßkomposition sind die Melodie und die Strukturbeschreibung.

Es werden alle Komponenten der Melodiekomposition verwendet. Für den Baß nimmt das Verfahren eine andere Auswahl von Stimmengenerierungsregeln und andere Endmuster als für die Melodie.

Zusätzlich wird noch die Stimmführungskomponente gerufen, die prüft, ob die Stimmführung korrekt ist (Kap. 3.7).

Suche und Zustandsverwaltung Die Einzelstimmengenerierung erfolgt notenweise. Ein Zustand enthält den aktuellen Stand der Komposition (Noten) und zusätzliche strukturelle Information. Der Zustandsraum wird durch das Datenformat für Zustände, die Nachfolgerfunktion, welche die möglichen Fortsetzungen der Komposition bestimmt, und Start und Ziel beschrieben. Der so beschriebene Zustandsraum wird mit den Suchstrategien durchsucht. Die Definitionen von Zustand und Zustandsraum sowie die Suchstrategien werden in Kapitel 3.4 beschrieben.

Stimmengenerierung mittels Alternativeregeln Die Regeln für die Stimmengenerierung beschreiben die Möglichkeiten (Alternativen), eine Stimme (sei es Melodie oder Baß) fortzusetzen. Im Regelsatz ist die Folge der Stimmengenerierungsregeln für diese Stimme festgelegt.

Die Regelsprache mit den Grundelementen für die Definition von Regeln wird in Kapitel 3.5 beschrieben. Kapitel 3.6 erläutert die Regeln.

Stimmführung Bei der Komposition von Baß und eventuell weiteren Begleitstimmen sind Stimmführungsregeln zu berücksichtigen. Diese Regeln betrachten ein *Stimmführungsquadrupel*, nämlich in zwei Stimmen

je zwei aufeinanderfolgende Noten. Die Stimmführungsregeln werden von einem Stimmführungsregelcompiler zur einer Funktion kombiniert, die entscheidet, ob die Stimmführung für dieses Stimmführungsquadrupel korrekt ist.

Die Stimmführungsfunktionen implementieren die Grundelemente für die Regelvoraussetzungen und berechnen den Zugriffsbereich, aus dem die Stimmführungsquadrupel erstellt werden.

Patternmatching Der Patternmatcher führt einen Ähnlichkeitsvergleich zwischen kleinen Ausschnitten der Komposition und einem Muster durch. Dieses Verfahren wird für die Einhaltung der Endformeln eingesetzt.

Partiturrepräsentation COMF Hier werden die Darstellungen für die Grundelemente, das sind Noten aus Tonhöhe und Dauer, Intervalle, Tonarten, Harmonien, die Speicherung dieser Elemente in einer Partiturokarte und die Zugriffswege und Konversionsfunktionen definiert. Die Partiturokarte wird in Abschnitt 3.2 genauer beschrieben.

3.2 Repräsentation der musikalischen Grundelemente in einer Partiturokarte (Chart)

3.2.1 Anforderungen an die Repräsentation

Ein wichtiger Schritt bei der Lösung eines Problems ist eine angemessene Repräsentation der Objekte des Anwendungsbereichs.

Wiggins und Smaill [WS97] nennen zwei Eigenschaften einer musikalischen Repräsentation: Bei der *expressive completeness* (Vollständigkeit der Ausdrucksfähigkeit) geht es darum, wie vollständig die musikalischen Ausdrucksmöglichkeiten für die Interpretation des Stücks abgelegt werden können. Dazu gehört u.a. Information über Lautstärke, Klangfarbe, Artikulation, Lautstärkenverläufe usw. Die *structural generality* (Strukturfähigkeit) bezeichnet, wieviel Zusatzinformation über den strukturellen Aufbau des Stücks repräsentiert werden kann.

Bei der Komposition von Menuetten ist die Strukturfähigkeit wichtig, weil der klassische Stil vielfältige Beziehungen zwischen den Einzelelementen der Komposition erfordert. Da keine Klangsynthese betrieben wird, ist die Vollständigkeit der Ausdrucksfähigkeit für diese Arbeit nicht wichtig.

Es wird nur die grundsätzlich in der Notenschrift enthaltene logische Information dargestellt, wie sie frühklassische Komponisten niedergeschrieben

haben¹, d.h. Tonart und Metrum des Stücks und die Noten bestehend aus Tonhöhe und Dauer bzw. Pausen. Gegenüber der Notenschrift fällt die typographische Information weg.

Rich [RK91] nennt folgende Aufgaben, die das Repräsentationsverfahren eines wissensbasierten Systems zu erfüllen hat:

- *Die Repräsentation muß es ermöglichen, die benötigte Information darzustellen.*

Um diese Forderung zu erfüllen, werden Repräsentationselemente für die in der Notation aufgeschriebenen Elemente wie Noten, Pausen, Tonart und Metrum definiert. Außerdem gibt es Elemente für die grundlegendste Metainformation, nämlich Zeitpunkte, Zeitbereiche, Betonungsstufen und Akkorde und deren Umkehrung.

- *Die Information muß gespeichert werden.*
- *Die Repräsentation muß es ermöglichen, auf die dargestellte Information zuzugreifen, wenn Entscheidungen anstehen.*

Für die üblichen Entscheidungen bei der klassischen Komposition sind folgende Zugriffswege wichtig:

- Benachbarte Noten eines Tons für Melodik und Stimmführung.
- Gleichzeitige Noten für Harmonik.
- Zugriff über den Zeitpunkt, d.h. Suche von Noten an einem Zeitpunkt und Suche des Zeitpunktes einer gegebenen Note im Stück, für Bestimmung von metrischer Position und Betonung.

- *Die Repräsentation muß effizient sein.*

Die Forderung nach Effizienz ist bei dieser Aufgabe gegenüber den anderen Forderungen nachrangig, da die Kardinalitäten nicht besonders groß sind. Sofern kein Konflikt mit anderen Forderungen entsteht, wird aber auf Effizienz geachtet.

Die Grundelemente der Repräsentation werden nicht nur als Teile der Partiturnote verwendet, sondern werden als Datentypen auch den verschiedenen Funktionen der höheren Schichten des Programms übergeben. Bei der Implementierung der verschiedenen musikalischen Gesetzmäßigkeiten werden

¹Dynamische Bezeichnungen und Artikulationsanweisungen wurden zwar von frühklassischen Komponisten gelegentlich dazugeschrieben. Sie werden hier aber trotzdem nicht repräsentiert.

unterschiedliche Aspekte eines musikalischen Elements benötigt, beispielsweise kann bei einem Ton die absolute Tonhöhe oder die Stufe in der Tonart oder die Stufe innerhalb des aktuellen Akkords von Interesse sein. Die Repräsentation realisiert die nötigen Konversionen und Zugriffe.

Von diesen Anforderungen ausgehend wurde das Datenformat COMF entwickelt, das die Grundelemente der Partitur und strukturelle Elemente in einem objektorientierten Format realisiert. Die Begriffe stammen aus der systematischen (modernen) Literatur (WQ2) [KD84]. Für die von COMF abgedeckten Bereiche weichen systematische und zeitgenössische Literatur (WQ1) allerdings kaum noch voneinander ab (siehe Kapitel 2.2). Der Aufbau der Datenstrukturen ist in Kapitel 4.2 beschrieben. Das Datenformat COMF ist für verschiedene Aufgaben der tonalen Musik geeignet und wurde bereits als Repräsentationsformat für die Suche nach Mustern [Bec00] eingesetzt. Außerdem dient es als Grundlage für die Reimplementierung des Systems von Maxwell [Max84] zur harmonischen Analyse im Rahmen von [Hö00].

Im folgenden werden die Bestandteile von COMF beschrieben. Für die Beispiele wurde das externe Format von COMF benutzt. In diesem Format werden Elemente auf Bildschirm oder Datei ausgegeben und können auch wieder eingelesen werden.

3.2.2 Karte aus Zeitelementen

Musik des klassischen Stils ist metrisch gebundene Musik, d.h. die Grundelemente haben Dauern, die als Bruchteil einer ganzen Note ausgedrückt werden. Meistens sind die Nenner der Noten Zweierpotenzen, z.B. halbe Noten ($\frac{1}{2}$), Viertelnoten ($\frac{1}{4}$), Achtelnoten ($\frac{1}{8}$) usw. Punktierte Noten stellt man durch eine 3 im Zähler dar (z.B. $\frac{3}{8}$ für die punktierte Viertelnote). Bei einer triolischen Teilung hat man einen Nenner, der eine 3 als Primfaktor hat, z.B. Achteltriolen entsprechen $\frac{1}{12}$.

Die Grundelemente der Musik, die Noten und Pausen, sind solche Dauerelemente. Damit die benötigten Zugriffswege auf die Elemente implementiert werden können, werden die musikalischen Elemente der mehrstimmigen Musik in einer sogenannten *Partiturskarte* organisiert. Die einzelnen Stimmen stehen in Spuren (tracks). Ein Dauerelement ist mit seinem Vorgänger, seinem Nachfolger und der Karte verbunden. Die gesamte Karte besteht aus einer Menge von solchen Spuren und speichert außerdem als globale, für das ganze Stück geltende Information noch Metrum und Grundtonart des Stücks.

Der Zeitpunkt, zu dem ein Kartenelement in die gesamte Karte eingeordnet ist, kann über die Nachbarschaftsbeziehungen bestimmt werden. Ein Zeitpunkt wird durch Takt, Schlag und Schlagteil (angegeben als Dauer seit

Beginn des aktuellen Schlags) beschrieben. Zur Interpretation dieser Angaben muß das Metrum bekannt sein. Über die Karte und den Zeitpunkt lassen sich die gleichzeitigen Elemente finden.

Außer den Noten selbst ist noch die Strukturinformation abzulegen. Beispiele dafür sind Harmonieverlauf und Beziehungen zwischen Teilen. Sofern diese Information Zeiträume betrifft, kann sie ebenfalls durch Dauerelemente dargestellt werden. In vielen Fällen überdecken Folgen solcher Strukturelemente den Zeitraum des Stücks und werden analog zu den Stimmen des Stücks auch als Spur in der Karte dargestellt. Zur Unterscheidung erhalten Spuren von Strukturelementen den Typ `:struct` während die Spuren mit den Noten selbst den Spurtyp `:notes` haben.

Für strukturelle Angaben gibt es ein vielseitiges Element, das die Gültigkeit eines Tags (beliebige Nutzdaten) für eine bestimmte Dauer beschreibt.

Beispiel

```
$(track (:struct :my-boolean-seq)
  $(tagel 2/4 t)$ (tagel 1/4 nil)$ (tagel 3/4 t))
```

Dieser Strukturtrack mit Namen `:my-boolean-seq` repräsentiert eine Folge von Booleschen Werten: Zeitraum einer halbe Note „wahr“, Zeitraum einer Viertelnote „falsch“, Zeitraum einer punktierten Halben „wahr“.

3.2.3 Tonhöhen, Intervalle und Stufen

Diese Elemente beschreiben die Tonhöhe eines Tons entweder absolut (Tonhöhe) oder relativ zu einem anderen Ton (Intervalle) oder relativ zu einer Grundtonart (Stufen). Zur Verwendung in der Partiturnote können diese Elemente noch mit einer Dauer verbunden werden. Die Attribute aller tonalen Elemente sind intern durch Integerzahlen dargestellt, extern werden sie in einer an die übliche Terminologie angelehnten Notation ausgegeben.

Ausgangspunkt ist die absolute Darstellung der Tonhöhe durch den Tonnamen, das Vorzeichen und die Oktave. Die Verbindung mit der Dauer heißt Note. Diese drei Komponenten werden getrennt benötigt, denn man muß beispielsweise für tonale Vorgänge zwischen `fis` und `ges` unterscheiden können. Oktave und Tonhöhe selbst werden getrennt gehalten, da die Oktave bei einigen Berechnungen unwichtig ist. Im externen Format werden für die Tonhöhe und Vorzeichen die deutschen Bezeichnungen, für die Oktave die Midi-Oktave benutzt.

Beispiel: Viertelnote `gis'` `$(note vi gis 4)`

Je nachdem, welche Information bei einer Note gefragt ist, gibt es verschiedene Vergleichsfunktionen:

Notenvergleich: Alle Attribute, d.h. Notename, Vorzeichen und Oktave müssen gleich sein.

enharmonischer Vergleich Die beiden Noten müssen enharmonisch gleich sein (gleiche Midi-Nummer, gleiche Klaviertaste).

Tonhöhenvergleich Hier wird die Dauer vernachlässigt.

Tonhöhenklassenvergleich Hier werden Dauer und Oktave vernachlässigt.

Der Abstand zwischen zwei Tönen wird durch Intervalle angegeben. Intervalle kann man bei tonaler Musik entweder exakt oder relativ zur Grundtonart angeben. Das exakte Intervall wird durch die Zahl der diatonischen Tonschritte² und den Modifikator des Intervalls angegeben, z.B. *kleine Terz* oder *verminderte Quinte*. Eine positive Schrittzahl steht für Intervalle nach oben, eine negative für Intervalle nach unten. Modifikatoren sind vermindert, klein, groß, rein, übermäßig etc.

Häufig befaßt man sich mit tonalen Intervallen (*dintv*). Bei diesen spricht man von einer Terz aufwärts innerhalb der Tonart, wobei man nicht angibt, ob es sich um eine kleine oder eine große Terz handelt. Dies wird dann durch die Position des Anfangstons in der Tonart festgelegt. Ein solches diatonisches Intervall wird durch die diatonische Schrittzahl und die Alteration gegen den von der Tonart vorgesehenen Ton beschrieben.

Ein Intervall kombiniert mit einer Dauer ergibt einen *Relativton* (*drel*). Man kann Melodiestücke durch einen Anfangston und eine Folge von Relativtönen beschreiben.

Beispiel

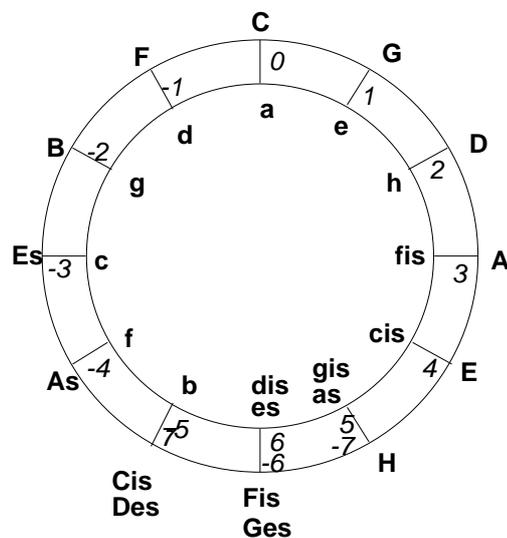
kleine Sexte aufwärts (exakt)	<code>\$(cintv :kl sexte :auf)</code>
Sexte ab (tonal)	<code>\$(dintv sexte :ab)</code>
Sexte ab,	
Zielton gegen Tonart erniedrigt (tonal)	<code>\$(dintv sexte :ab :tief)</code>
Relativton Sexte, Halbe (tonal)	<code>\$(drel ha sexte :ab)</code>

Auch bei der Bezeichnung von Tönen und Intervallen stimmen WQ1 und WQ2 überein.

3.2.4 Tonart und Stufen

Die Übersetzung von diatonischen Intervallen und Stufen auf den genauen Ton geschieht über die Skala der Tonart. Eine Tonart wird durch ihren Grundton und das Geschlecht (Dur bzw. Moll) beschrieben.

²Der Klassenname heißt etwas ungeschickt *cintv* von *chromatisches Intervall*, weil die chromatische Schrittzahl eindeutig beschrieben ist.



Großbuchstaben Dur, Kleinbuchstaben moll

Abbildung 3.2: Quintenzirkel

Zu einer Tonart kann man dann aus der Position des Grundtons im Quintenzirkel (siehe Abb. 3.2) und dem Geschlecht die Zahl der Vorzeichen berechnen (Kreuze werden durch eine positive Zahl, Bes durch eine negative Zahl dargestellt). Die Molltonarten entstehen, indem man die Vorzeichenzahl um drei reduziert.

In der Harmonielehre der systematischen Literatur werden die Tonarten im Quintenzirkel angeordnet, so daß bei benachbarten Tonarten die Grundtöne eine Quinte voneinander entfernt sind und sie sich um ein Vorzeichen unterscheiden, d.h. ihr Tonvorrat sich in einem Ton unterscheidet. Die gleichnamigen Molltonarten erhält man, indem man die Vorzeichenzahl um drei reduziert. In der klassischen Epoche, wo stark harmonisch gedacht wird, ist die Verwendung der harmonischen Molltonleiter der Regelfall. Bei der Bestimmung des Tonvorrats für harmonisches Moll ist also die Erhöhung der 7. Stufe gegen die natürliche Tonleiter zu berücksichtigen.

Dieses Skalenmodell mit Grundton, Vorzeichenzahl und Alterierung von Stufen kann auf einfache Art auf beliebige 7-stufigen Skalen innerhalb der 12-tönigen Skala erweitert werden.

In der Musiktherie der klassischen Epoche hat sich die Auffassung von 12 gleichberechtigten Tonarten bereits durchgesetzt, so daß ein Stück einfach relativ zu seiner Grundtonart betrachtet werden kann. Ein Ton, relativ zum Grundton der Tonart aufgefaßt, ist eine *Stufe*. Eine Stufe wird durch ihre diatonische Stufe (eine Zahl) in der Skala, ihre Alteration gegen den Skalenton

in Halbtonschritten und ihren harmonischen Bezug, die Tonart, angegeben. Zwei Stufen sind gleich, wenn sie in Stufe und Alteration übereinstimmen. Durch die Verbindung einer Stufe mit einer Dauer entsteht ein *Stufenton*.

Beispiel:

Tonart D-Dur \$(tonkey D :dur)
 erhöhte 5. Stufe \$(degr 5 1)
 Viertelnote, 3. Stufe \$(degnote 1/4 3 0)

Die Grundtonart eines Stücks wird in der Partiturnote abgelegt. Die Grundtonart spielt in der Kompositionslehre der Klassik eine wichtige Rolle. Im Regelfall beginnt und endet das Stück in seiner Grundtonart. Dazwischen kann das Stück die Tonart wechseln, in der die aktuellen Noten und Akkorde zu interpretieren sind. Diese *augenblickliche Tonart* wird durch die Stufe (relativ zur Grundtonart) und das Geschlecht dargestellt. Wenn im folgenden von „Tonart“ gesprochen wird, ist die augenblickliche Tonart gemeint.

3.2.5 Harmonien

Eine Harmonie wird durch ihren Grundton, angegeben als Stufe in der Tonart, und die Akkordtöne als Stufen relativ zur Stufe des Harmoniegrundtons bezeichnet. Die Stufen werden innerhalb der Tonart interpretiert, wobei man Erhöhungen und Erniedrigungen angeben kann. Die einfachsten Harmonien sind Dreiklangsharmonien, d.h. sie enthalten relativ zu ihrem Grundton die Stufen 1 3 und 5. Zu einer Dreiklangsharmonie kann man noch zusätzliche Akkordtöne angeben. Für Nichtdreiklangsharmonien wird die gesamte Stufenmenge explizit angegeben. Eine Harmonie mit Dauer wird als *Harmonieelement* bezeichnet.

Beispiel:

Leitereigener Akkord, erste Stufe, Dauer: Halbe \$(harmel 1/2 1)
 Dominantseptakkord, Dauer: Viertel \$(harmel 1/4 5 7)

Die Harmonien werden mit den üblichen Stufennummern [KD84] (gehört zu WQ2) bezeichnet. Diese Theorie ist einfach und effizient zu realisieren.

Die Partiturnote verwendet die stilunabhängigen Begriffe aus dem Common-Practice-Bereich. Deswegen enthält das Format COMF keine Repräsentationen für weiterführende Begriffe, die stilabhängig sind, wie z.B. die Funktion eines Akkords oder die Tatsache, ob ein Akkord konsonant oder dissonant ist. Die stilunabhängigen Begriffe werden von den Komponenten definiert, die sie benötigen.

3.3 Kompositionssteuerung und Strukturbeschreibung

Aufgabe der Kompositionssteuerung ist es, die einzelnen Kompositionskomponenten der mittleren (Wissens-)ebene zu rufen, mit den richtigen Parametern zu versorgen und die Ergebnisse zu verwalten.

3.3.1 Kompositionspläne

Der Ablauf der gesamten Komposition ist in Schritte aufgeteilt. Die Folge dieser Schritte ist in einem Kompositionsplan gegeben. Ein Interpret arbeitet diese Kompositionsschritte der Reihe nach ab und aktualisiert nach jedem Schritt die Lösungsmenge. Zu Beginn enthält die Lösungsmenge eine leere Chart ohne Noten oder Strukturangaben.

Folgende Arten von Kompositionsschritten werden zur Komposition benutzt:

Strukturbeschreibung Dieser Schritt erstellt die Strukturbeschreibung und trägt sie in die Chart ein.

Stimmengenerierung Die Schrittbeschreibung enthält Stimme, Name des zu komponierenden Satzes und die gewünschte Anzahl der Lösungen.

Dieser Schritt erzeugt zu jeder Chart aus der Lösungsmenge die vorgegebene Anzahl von fortsetzenden Lösungen für den angegebenen Satz und die angegebene Stimme.

Dafür werden aus Regelsatz und Strukturbeschreibung die für den Satz und die Stimme gültigen Angaben und Regeln herausgesucht. Aus den Regeln wird die Alternativenfunktion (vgl. Kapitel 3.5) erzeugt und mit der Suche im Zustandsraum (vgl. Kapitel 3.4) die gewünschte Zahl von Lösungen bestimmt.

Reduktion Reduktionsschritte suchen von der aktuellen Lösungsmenge einen Teil aus.

Dazu treten noch weitere Schritttypen für die Fehlersuche. Eine vollständige Beschreibung findet man in Kap. 4.3.3.

Abbildung 3.3 zeigt die Entwicklung der Lösungsmenge bei der Abarbeitung eines Kompositionsplans. Das Beispiel in der Abbildung geht vom Idealfall aus, nämlich daß bei den Kompositionsprozessen auch die vom Planelement geforderte Anzahl von Lösungen entsteht. Es kann auch vorkommen, daß ein Kompositionsvorgang keine Lösungen oder weniger Lösungen als gefordert erzeugt.

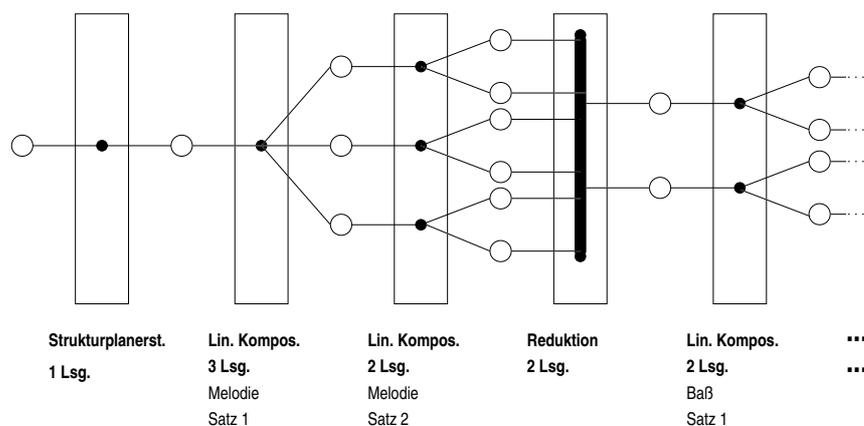


Abbildung 3.3: Beispiel für Kompositionsplan und Lösungszahl

3.3.2 Regelsätze und Parameterverwaltung

Die verschiedenen Kompositionsschritte verwenden gemeinsame Komponenten, aber setzen zum Teil unterschiedliches Wissen ein. Das Wissen, das von den verschiedenen Kompositionsschritten benötigt wird, ist im sogenannten *Regelsatz* in Form von Attribut-Wertpaaren abgelegt. Der Regelsatz enthält sowohl Angaben, die für den gesamten Kompositionsprozeß gültig sind, als auch solche, die nur für eine einzelne Stimme gültig sind.

Deswegen gibt es im Regelsatz mehrere solcher Listen. Die Angaben, die für das gesamte Stück gelten, werden im Regelsatz unter dem Bezeichner `:score` abgelegt, die für die einzelnen Stimmen unter dem Namen der jeweiligen Stimme.

Der Regelsatz legt für jede Stimme zwei Regelfolgen zur Stimmengenerierung fest. Die erste Regelfolge sind die Regeln zur Bestimmung der ersten Note eines Satzes, die andere wird zur Berechnung der Intervalle für die folgenden Noten benutzt. Für die Baßstimme ist außerdem noch eine Folge von Stimmführungsregeln angegeben. Die Untersuchung der Stimmführungsregeln wird von einer speziellen Stimmgenerierungsregel angestoßen.

Weitere Angaben für die Stimmengenerierung sind in der Strukturbeschreibung abgelegt, die als erster Kompositionsschritt erstellt wird. Die Strukturbeschreibung ist ebenso wie der Regelsatz in Angaben für die gesamte Partitur und Angaben für die einzelnen Stimmen aufgeteilt. Die Strukturbeschreibung wird in zwei Stufen erstellt: Zuerst die Makrostruktur, die das Stück in Sätze gliedert und dann die Mikrostruktur. Die Makrostruktur, in der die Abfolge der Sätze definiert wird, enthält noch weitere Parameter, die die einzelnen Sätze betreffen.

Ein wichtiges Element bei der Komposition von Menuetten sind Verwandtschaftsbeziehungen zwischen Teilen. Im Regelsatz kann man ein externes Modell angeben, aus dem die Mikrostrukturkomposition die Folge der Korrespondenzelemente erstellt. Außerdem kann man eine Folge von Korrespondenzelementen direkt angeben.

3.3.3 Makrostruktur

Nach der zeitgenössischen Theorie wird ein Stück in Sätze gegliedert³. Die Makrostruktur ordnet nun den einzelnen Sätzen Angaben zu:

Die Taktordnung gibt die Anzahl der Sätze und deren Längen an. Es wird eine Anlage von 4 Sätzen zu je 4 Takten erzeugt.

Jeder Satz wird durch eine Endformel, das ist entweder eine (volle) Kadenz oder ein Absatz, geschlossen, welche bei kurzen Stücken wie dem Menuett in der Grundtonart oder in der Quinte⁴ stehen können.

Die Tonordnung ordnet jedem Satz die Kategorie der Endformel zu, legt aber nicht die Endformel selbst fest. Es wurden die beiden gebräuchlichsten Tonordnungen (Abfolgen von Endformeln) realisiert:

Grundabsatz - Quintkadenz - Quintabsatz - Grundkadenz
Grundabsatz - Quintkadenz - Grundabsatz - Grundkadenz

Die beiden Tonordnungen unterscheiden sich nur im dritten Vierer. Hinter den Festlegungen der Literatur stehen Prinzipien von Stärke und Tonartverhältnissen, die Riepel und Koch mit Reimschemata in der Dichtkunst vergleichen. Der erste Grundabsatz ist üblich, um zu Beginn die Grundtonart des Stücks zu befestigen.

Innerhalb des Stücks sollte eine Ausweichung⁵ in die Quinttonart stattfinden, wobei die Mitte des Stücks dafür ein guter Ort ist. Als Gliederung in der Mitte des Stücks sollte die Endformel des zweiten Satzes auch ihre Nachbarn an Stärke übertreffen. Deswegen ist eine Quintkadenz als Schluß des zweiten Satzes üblich. Der letzte (vierte) Satz muß in die Grundtonart des Stücks zurückkehren und seine Endformel muß eine volle Kadenz sein.

So bleibt Wahlfreiheit beim dritten Satz. Hier wird zwischen den beiden Absätzen gewählt, um in der Wirkung schwächer als die benachbarten Kadenzen zu sein.

³vgl. Kapitel 2.2

⁴Abhandlungen über die zu erreichenden Tonarten findet man bei [Rie52]. Die Verwendung weiterer Ausweichungstonarten ist für längere Stücke und für andere Genre („*fugenmäßige*“ vs. „*konzertmäßige*“ Ordnung) durchaus zugelassen

⁵Begriff der Ausweichung s. Kap. 2.2.2

In der Makrostrukturbeschreibung wird außerdem noch festgelegt, ob im dritten Satz eine Fonte- (im Falle eines Grundabsatzes) bzw. eine Montesequenz (im Falle des Quintabsatzes) oder keines von beiden durchgeführt werden soll. Obwohl dies strenggenommen nicht mehr die Makrostruktur sondern innere Eigenschaften eines Satzes betrifft, ist es für die Struktur und für die Bestimmung der Mikrostruktur bedeutsam. Riepel rät stark zum Einbau einer Fonte- oder Montesequenz.

Beispiel einer Makrostrukturbeschreibung

```
((:SEC1 ((:EF-TYPE :GRUNDABSATZ) (:SECBARS 4) (:DUR $(timevec 5 1 0))))
(:SEC2 ((:EF-TYPE :QUINTKADENZ) (:SECBARS 4) (:DUR $(timevec 5 1 0))))
(:SEC3 ((:EF-TYPE :QUINTABSATZ) (:SECBARS 4) (:DUR $(timevec 5 1 0))
(:FM-SEQ :MONTE)))
(:SEC4 ((:EF-TYPE :GRUNDKADENZ) (:SECBARS 4) (:DUR $(timevec 5 1 0))))
```

Die Angaben der Taktzahl und der Dauer durch ein `timevec`-Element sind redundant.

3.3.4 Mikrostruktur

Der nächste Schritt ist die Erzeugung der Mikrostruktur. Sie besteht aus verschiedenen Folgen von Tagelementen (Strukturelementen), die die Länge des Stücks abdecken und so für die Teile des Stücks Angaben festlegen. Auch hier gibt es Information, die das gesamte Stück betrifft und solche, die einzelne Stimmen betrifft.

Folgende Sequenzen gehören zur Mikrostruktur:

Menuetttaktformen Speziell für die Melodie von Menuetten beschreibt Riepel[Rie52] besondere Anforderungen an den Rhythmus. Von einem Taktpaar muß einer der beiden rhythmisch ein *vollkommener* Menuetttakt sein während der andere *unvollkommen* sein darf. Die genaue Bedeutung wird mit den Regeln zur Dauernvergabe in Kapitel 3.6.12 beschrieben.

Die Schlußtakte der Sätze und bei den Kadenzten auch die vorletzten Takte sind von einer Endformel regiert und daher von der Regelung ausgenommen.⁶

Korrespondenzen Korrespondenzen beschreiben Ähnlichkeiten zwischen Teilen des Stücks.

⁶Riepel schlägt außerdem noch zur Vorbereitung der Schlußkadenz einen drittletzten Takt mit Achteltonleiterkette vor, was hier aber nicht realisiert wurde.

Die Korrespondenz ist ein Tagelement, das den Korrespondenzbereich überdeckt. Die Beschreibung einer Korrespondenz besteht aus dem Modellbereich (dem Bereich, der dem Korrespondenzbereich als Vorbild dient), dem Vergleichsprädikat für die Ähnlichkeit der ersten Note und dem Vergleichsprädikat für die Ähnlichkeit der Intervalle (Einzelheiten siehe Kap. 3.5.4).

Für die Montesequenz wird für die Melodie die Aufwärtssequenzierung eines zweitaktigen Melodiestücks in die Korrespondenzfolge eingetragen, für die Fontesequenz die Abwärtssequenzierung.

Wird ein externes Modell vorgegeben, so wird die Verwandtschaft mit diesem Modell für alle Teile des Stücks, die nicht zu einer Endformel gehören, gefordert. Direkt vorgegebene Korrespondenzelemente werden unverändert übernommen.

Endformelanzeiger Einige der Stimmengenerierungsregeln benötigen die Information, ob man sich in einer Endformel befindet, oder nicht. Dafür wird eine Folge von Tagelementen mit booleschen Werten erzeugt.

Harmonische Basis Für die harmonische Basis sind Bausteine definiert, für die die erste Harmonie und die Kategorie der Endformel angegeben sind. Es werden nun Bausteine ausgesucht, die zu den Festlegungen der Makrostruktur passen, und zur Harmoniefolge zusammengesetzt.

3.4 Suche und Zustandsverwaltung

Ein Arbeitsschritt der Stimmengenerierung ist das Hinzufügen einer Note zum bisherigen Zustand der Komposition. Die Komposition wird unter Berücksichtigung des Regelsatzes und der bisherigen Komposition um eine Note verlängert.

Das Anwendungswissen über einen Zustand ist eher lokal gegeben, d.h. es sind die Möglichkeiten des nächsten Arbeitsschritts beschrieben. Es gibt aber kein formales Verfahren für den Gesamtzusammenhang. So können folgende problematischen Fälle auftreten:

- An einem Zustand sind aus lokaler Sicht mehrere Schritte möglich und es gibt keine Möglichkeit, zu entscheiden, welcher der beste ist.
- Von einem Zustand aus kommt man nicht mehr weiter. Ein möglicher Grund ist, daß man in eine Sackgasse gelaufen ist, d.h. es gibt keinen Folgezustand, obwohl die Aufgabe noch nicht beendet ist. Die andere Möglichkeit ist, daß der Zustand gescheitert ist (engl. *fail*), d.h. man

erkennt, daß der Zustand bestimmt nicht mehr zu einem korrekten Zustand vervollständigt werden kann.

Es gibt nun Strategien, um Aufgaben mit solchen Eigenschaften zu beherrschen. Damit die Bearbeitung des Problems mit unterschiedlichen Strategien möglich ist, trennt man das Verfahren in eine anwendungsunabhängige Suchstrategie und ein Anwendungsmodell auf.

3.4.1 Formulierung vom Problemen als Suchproblem

Diese Auftrennung erreicht man, indem man das Problem als Suchproblem nach [RN95] formuliert.⁷ Dafür werden folgende Komponenten identifiziert:

Zustand Das Datenformat des Zustands wird definiert.

Startzustand Die Suche beginnt mit dem Startzustand.

Nachfolgerfunktion Die Nachfolgerfunktion erhält einen Zustand als Eingabe und gibt die Menge der möglichen Folgezustände aus. Diese Menge kann mehrere Elemente enthalten, aber auch einelementig oder leer sein.

Zieltest Der Zieltest erhält einen Zustand als Eingabe und gibt an, ob mit dem Zustand das Ziel der Suche erreicht ist. In diesem Fall wird die Suche mit Erfolg beendet.

Fail-Test Der Fail-Test (Fehlschlagstest) erhält einen Zustand und gibt an, ob die Suche fehlgeschlagen ist, d.h. bekannt ist (oder angenommen wird), daß von diesem Zustand aus das Ziel nicht mehr zu erreichen ist.

Der Fail-Test könnte aus der Problemformulierung eliminiert werden, indem die Überprüfungen in die Nachfolgerfunktion integriert werden. Die Verwendung eines zusätzlichen Fail-Tests vereinfacht die Struktur des Programms.

Das Ergebnis der Suche ist dann der Zielzustand.

Eine Erweiterung dieser Problemformulierung ist die heuristische Suche, bei der die Zustände mit einer Zahl bewertet werden. Für die Suche in Zustandsräumen mit und ohne Heuristik existieren verschiedene Algorithmen [RN95].

⁷Bei den Begriffen des Suchproblems gibt es kleine Freiheiten im Detail, kleine Unterschiede zwischen verschiedenen Literaturquellen und unterschiedliche Realisierungsmöglichkeiten für verschiedene Anwendungen. Der Übersichtlichkeit halber beschränkt sich die folgende Beschreibung darauf, die Begriffe so zu beschreiben, wie sie im System verwendet werden.

3.4.2 Zustände

Die Beschreibung eines Zustands enthält alle Information, die das Suchverfahren für seine Arbeit, d.h. für Nachfolgerfunktion, Zieltest und Fail-Test, benötigt. Ein Teil der Information bleibt für den gesamten laufenden Suchvorgang unverändert. Diese feste Information wird am Anfang im Startzustand abgelegt und von der Nachfolgerfunktion an die Folgezustände weiterpropagiert. Interessanter ist die variable Information im Zustand, die bei jeder Anwendung der Nachfolgerfunktion geändert oder erweitert wird.

Grundbaustein der Komposition von Melodie und Baß ist die Stimmengenerierung. Das Verfahren geht in Abschnitten in der Länge eines Satzes vor (vgl. Kapitel 2.2), da die Endformeln der Sätze bedeutende Einschnitte sind. Dabei erstellt je ein Suchvorgang für eine Stimme einen Abschnitt in der Länge eines Satzes. Die feste Information eines Suchvorgangs besteht hier aus der Strukturbeschreibung und aus den Ergebnissen der bisherigen Suchvorgänge, d.h. den vorangegangenen Sätzen und im Falle des Basses auch der Melodie.

Die im laufenden Suchvorgang erzeugte Notenfolge gehört zur variablen Information des Zustands. Jede Nachfolgerbildung verlängert dies um eine Note, d.h. der Anfang bleibt bei der Nachfolgerbildung unverändert.

Ein weiteres Element der variablen Information sind die bottom-up erzeugten Strukturelemente⁸. Strukturelle Elemente geben für einen metrischen Zeitbereich (beispielsweise für die Takte 3 und 4) besondere Bedingungen an, die die Nachfolgerfunktion berücksichtigen muß. Die meisten Strukturelemente werden durch die Strukturbeschreibung vorgegeben, d.h. top-down erstellt, und sind daher Teil der festen Information des Zustands. Es gibt aber auch den Fall, daß während der Komposition, also bottom-up, Strukturelemente erzeugt werden, die von den folgenden Schritten zu berücksichtigen sind. Beispielsweise ist die korrekte Fortführung von Durchgängen als Bottom-up-Strukturelement realisiert.

Die genaue Unterscheidung zwischen fester und variabler Information ist wichtig. Wenn der Kompositionsvorgang bei einem Zustand scheitert und die Suchstrategie in einem anderen Zustand weitersucht, dürfen keine Ergebnisse von Entscheidungen übrigbleiben, die während des Fehlversuchs getroffen wurden. Deswegen dürfen die während der Stimmengenerierung bottom-up erzeugten Strukturelemente nicht in die Strukturbeschreibung eingetragen werden.

Die Zustandsübergänge werden so realisiert, daß Datenstrukturen beim Zustandsübergang nur erweitert oder kopiert, nicht aber destruktiv oder mutierend bearbeitet werden. Jeder Zustand ist mit seinem Vorgängerzustand

⁸Einzelheiten siehe Kapitel 3.5.3.

verbunden, damit die Suchstrategie den bisher bearbeiteten Teil des Suchraums durchwandern kann.

3.4.3 Einfluß der Eigenschaften des Suchraums auf das Verfahren

Die Eignung der unterschiedlichen Suchverfahren hängt von den Eigenschaften des Suchraums ab, die sich u.a. durch folgende Faktoren beschreiben lassen:

- Die Regeln und Operatoren der Komposition beschreiben eine Menge von stilistischen (zulässigen) Stücken. Bei der Stimmengenerierung möchte man gerne eine oder auch mehrere Lösungen - eine Probe fester Größe - aus der Menge der stilistischen Lösungen erhalten.
- Um Erkenntnisse über die Natur des Suchraums und damit des verwendeten Wissens zu erhalten, kann man sich für alle Lösungen interessieren bzw. dafür, ob es überhaupt eine Lösung gibt. Dafür muß der Suchraum vollständig durchsucht werden, was natürlich nur für kleine Suchräume praktikabel ist.
- Die Tiefe des Suchraums ist die Anzahl der Zustandsübergänge bis der Zielzustand erreicht ist. Da bei jedem Zustandsübergang eine Note hinzugefügt wird und die Dauern der Noten nicht beliebig klein werden können, ist die Tiefe des Suchraums durch die Länge des Satzes (Vorgabe durch die Strukturbeschreibung) beschränkt. Da das Stück mit jedem Schritt wächst, ist der Zustandsraum zyklonfrei.

Bei Versuchen mit viertaktigen Melodieteilen mit gängigen Regelsätzen sind Längen von ca. 13-14 Noten (meist Achtel, Viertel und Halbe) am häufigsten.

- Der Verzweigungsgrad (Zahl der Folgezustände eines Zustands) variiert je nach Regelsatz und Zeitabschnitt.

Bei der Komposition von Melodieabschnitten, in denen die Strukturbeschreibung keine Korrespondenzbeziehungen fordert, werden bei gängigen Regelsätzen Verzweigungsgrade zwischen 3 und 6 beobachtet. In Zeitabschnitten, in denen durch strukturelle Vorgaben motivische Beziehungen gefordert sind, sinkt der Verzweigungsgrad auf 1-2.

Eine Untersuchung des Suchraums findet man in Kap. 5.5.

3.4.4 Suchverfahren

Die Aufgabe des Suchverfahrens ist es, zu entscheiden, welcher Zustand aus der Menge aller bisher bekannten Zustände als nächstes untersucht wird. Beim Untersuchen eines Zustands wird getestet, ob der Zustand ein Zielzustand ist und ob der Zustand gescheitert ist. Andernfalls werden die Folgezustände bestimmt. Die Folgezustände werden dann zur Menge der bekannten Zustände hinzugefügt.

Der Zustand besteht aus einem Anteil für die Problemdarstellung und einem problemunabhängigen Anteil für die Strategie. Letzterer stellt den Suchbaum, d.h. die Verkettung eines Zustands mit seinem Vorgängerzustand und die Markierung der bearbeiteten Alternativen dar. Dies wurde realisiert, indem die Klasse zur Beschreibung für den Stimmengenerierungszustand eine Subklasse eines allgemeinen Suchzustands ist (vgl. Kapitel 4.4). Somit hat man eine explizite Darstellung des Suchbaums, die verschiedene Strategien zur Auswahl des nächsten zu untersuchenden Knotens zuläßt.

Die Suchverfahren unterscheiden sich darin, welche Strategie bei der Auswahl angewandt wird. Daraus resultieren unterschiedliche Eigenschaften.

Tiefensuche

Eines der einfachsten Verfahren ist die aus der Literatur [RN95] bekannte Tiefensuche. Sie läßt sich zu einem Verfahren erweitern, das alle Lösungen liefert. Sie entspricht dem chronologischen Backtracking, denn es werden die zuletzt getroffenen Entscheidungen zuerst wieder zurückgenommen. Der Vorteil dieses Verfahrens ist, daß es nur wenig Speicher für Alternativen benötigt. Der Speicherbedarf wächst linear mit der Suchtiefe, in unserem Fall also der Länge der Stimme. Die Tiefensuche kann man zu einem Verfahren erweitern, das alle Lösungen sucht, indem man im Erfolgsfall die Lösung in die Ergebnismenge übernimmt, aber trotzdem weitersucht, bis alle Alternativen untersucht sind.

In manchen Abschnitten ist der Verzweigungsgrad groß und es gibt wenig Sackgassen, so daß auch die Lösungsmenge sehr groß ist. Eine einfache Tiefensuche ergibt bei jedem Aufruf des Verfahrens immer die selbe Lösung. Läßt man das Verfahren weitersuchen, bis die gewünschte Anzahl an Lösungen erreicht ist, erhält man Lösungen, die den gleichen Anfang haben und sich nur am Ende unterscheiden, da sie vom gleichen Zweig des Suchbaums stammen. Durch die Implementierung der Nachfolgerfunktion erhält man die Nachfolger in einer bestimmten Reihenfolge; beispielsweise erhält man z.B. Lösungen, die immer Schritte nach unten bevorzugen. Für die folgenden Kompositionsschritte ist aber eine Auswahl aus dem gesamten Bereich der Lösungen günstiger.

Ein anderes ernstes Problem bei der Tiefensuche ist, daß frühe Fehlentscheidungen erst spät als solche erkannt werden. Das Verfahren durchsucht diesen aussichtslosen Teilbaum dann vollständig. Bei einer Suchtiefe von 13 und einem Verzweigungsgrad von 3 - den vorhin genannten gängigen Zahlen - hätte ein solcher Zweig eine Tiefe von 12, d.h. es sind $3^{12} = 531\,441$ Knoten zu durchsuchen.

Zufallstiefensuche

Die Zufallstiefensuche ist eine Abwandlung der Tiefensuche. Sie wählt aus der Menge der Folgezustände per Zufall den zu untersuchenden Zustand aus. Die Zufallstiefensuche ist wie die gewöhnliche Tiefensuche ein vollständiges Verfahren, das jeden Zustand genau einmal untersucht. Ebenso wird chronologisches Backtracking betrieben, d.h. die zuletzt getroffene Entscheidung wird zurückgenommen. Der einzige Unterschied zur Tiefensuche ist die Verwendung des Zufalls bei der Auswahl des nächsten Zustands. Wenn es keine Folgezustände gibt oder ein Fehlschlag vorliegt, wird wie bei der Tiefensuche vorgegangen.

Daher löst auch die Tiefensuche mit Zufallssteuerung nicht das Problem, wenn (diesmal per Zufall) eine frühe Fehlentscheidung getroffen wird. Ebenso stammen die Lösungen der Lösungsmenge aus dem selben Zweig.

Der Vorteil gegenüber der gewöhnlichen Tiefensuche ist, daß eine Menge von Lösungen nicht wegen der Implementierung der Nachfolgerfunktion ähnlich aufgebaut sind, sondern eine etwas breiter gestreute Auswahl aus dem Lösungsraum bietet.

Zufallsschußverfahren

Um die Auswirkungen früher Fehlentscheidungen zu begrenzen, muß man im Falle eines Mißerfolgs bereit sein, frühe Entscheidungen zurücknehmen. Hier muß die Suchstrategie nicht nur bei der Alternativenauswahl sondern auch beim Fehlschlag einen Zustand auswählen. Die radikalste Lösung für dieses Problem ist das Zufallsschußverfahren. Nach einem Fehlschlag beginnt die Suche jedesmal beim Startzustand. Aus der Menge der Folgezustände wird ein Zustand per Zufall ausgewählt und weiterverfolgt. Die anderen werden nicht gespeichert. Im Falle des Erfolgs hat man eine Lösung. Im Falle des Mißerfolgs (Sackgasse oder Fehlschlag) gibt man den gesamten Versuch auf und startet (vom Startzustand aus) einen neuen Versuch. Diese Strategie führt direkt, ohne Seitenzweige zu verfolgen, entweder zu einem Erfolg oder einem Mißerfolg, daher die Bezeichnung Zufallsschuß.

Nachteil dieser Strategie ist, daß Berechnungen verschwendet werden, da man die weiteren Alternativen zu einem Zustand nicht berücksichtigt. Für Suchräume, bei denen Sackgassen ohnehin früh auffallen, oder Lösungen im Verhältnis zum Zustandsraum dünn gesät sind, ist die Zufallschußmethode ineffizient.

Das Zufallsschußverfahren führt über die durchgeführten Schüsse und die bereits besuchten Zustände nicht Buch, um den Aufwand (Speicherplatz und letztendlich Rechenzeit) für das Aufbewahren und Vergleichen so vieler Zwischenzustände einzusparen. Damit kann das Zufallsschußverfahren theoretisch Schüsse wiederholen und es kann sogar dieselbe Lösung in der Lösungsmenge mehrfach vorkommen. Bei einem großen Suchraum ist die Wahrscheinlichkeit allerdings eher gering. Im Unterschied zur Tiefensuche und zur Zufallstiefensuche kann das Zufallsschußverfahren nicht dazu verwendet werden, den gesamten Zustandsraum vollständig zu durchsuchen. Somit kann man mit dem Zufallsschußverfahren auch nicht feststellen, ob es keine Lösung gibt (die Schüsse könnten zufällig alle vorbeitreffen) und man kann auch nicht alle Lösungen bestimmen.

Trotzdem hat sich die Zufallsschußstrategie bei der gegebenen Aufgabenstellung als nützlich erwiesen, wenn im Suchraum sowohl Verzweigungsgrad als auch die Lösungsdichte groß ist, und frühe, aber spät erkannte Fehlentscheidungen zu befürchten sind. Hier werden dann Ergebnisse gefunden und durch den Verzicht auf Speicherung spart das Verfahren Rechenzeit.

Für die Melodiekomposition mit strukturellen Beziehungen hat sich die Zufallsschußstrategie bewährt. Im Suchraum sind durch die Endformeln und die Forderungen nach strukturellen Beziehungen frühe, aber spät erkannte Fehlentscheidungen recht häufig.

Zufallsgesteuerte Baumsuche

Wenn die Zwischenzustände der Suche wie in unserem Fall explizit repräsentiert und die Übergänge in der Datenstruktur abgelegt sind, kann man mit einer kombinierten Strategie auch die Eigenschaften der Tiefensuche mit der Zufallsschußstrategie kombinieren.

Die zufallsgesteuerte Baumsuche nimmt einen Zufallschuß vor, bewahrt aber die ununtersuchten Alternativzustände auf. Im Fall eines Fehlschlags wählt sie aus der Menge aller bekannten Zustände einen aus und startet von dort aus den nächsten Zufallschuß.

Man könnte die Zufallsauswahl bei jedem Schritt und nicht nur beim Fehlschlag auf die ganze Liste der bekannten Zustände anwenden. Dann hätte man eine Art zufallsgesteuerte Breitensuche. Die Breitensuche ist allerdings nachteilig, wenn wie hier alle Lösungen verhältnismäßig tief im Suchbaum

liegen. Der Vorteil der zufallsgesteuerten Baumsuche ist, daß das Verfahren sofort versucht, vollständige Lösungen zu erhalten, also ausreichend tief geht, bevor es weitere Zweige des Baums anlegt. Außerdem verschwendet es keine Berechnungen und ist zur vollständigen Suche geeignet.

Wenn der Suchraum viele Sackgassen enthält, dann hat aber diese Strategie das Problem, daß die Zahl der gespeicherten Zustände und damit der Speicherplatzbedarf zu groß wird.

3.4.5 Heuristische Suche mittels Gewichtungsfaktoren

In einigen Fällen hat man weiche Regeln, d.h. man bevorzugt bestimmte Fortführungen gegenüber anderen, möchte aber die anderen Fortführungen trotzdem zulassen. Letzteres ist vor allem für den Fall wichtig, daß die zuerst bevorzugte Fortführung später einen Fehlschlag verursacht.

Ein einfaches Verfahren zur Berücksichtigung solcher Bevorzugungen ist die Verwendung von Gewichtungsfaktoren (Zahlen). Um diese in den Suchvorgang zu integrieren, ist es einerseits nötig, daß die Nachfolgerfunktion Gewichtswerte erzeugt, andererseits muß die Suchstrategie die Gewichtswerte geeignet berücksichtigen.

Bei der Tiefensuche geschieht dies, indem nach der Alternative mit dem größten Gewicht zuerst weitergesucht wird. Bei den Verfahren, die Zufall verwenden, wird ein gewichteter Zufall benutzt, bei dem sich die Wahrscheinlichkeiten der Alternativen nach deren Gewicht verteilen.

3.5 Stimmengenerierung mittels Alternativenregeln

Sowohl bei der Komposition der Melodie als auch bei der Komposition des Basses geht es darum, eine Folge von Noten - die Stimme - unter Berücksichtigung der strukturellen Vorgaben zu erstellen. Dabei werden Note für Note die Alternativen erzeugt und der so entstehende Suchraum mit einem Suchverfahren durchsucht. Das Suchverfahren arbeitet satzweise (Begriff *Satz* vgl. Kapitel 2.2), d.h. es wird nacheinander für jeden Satz ein Suchvorgang durchgeführt. Im Vergleich zur Komposition des gesamten Stücks in einem einzigen Suchvorgang reduziert man so die Suchtiefe. Da die Enden der Sätze deutliche Einschnitte darstellen, ist dies auch musikalisch angemessen.

3.5.1 Darstellung der Stimmengenerierung als Suchproblem

Die notenweise Komposition einer Stimme wird als Suchproblem dargestellt. Die in Abschnitt 3.4 aufgezählten Begriffe werden folgendermaßen belegt:

Information im Zustand Der Zustand besteht aus der Information über den Suchraum, der variablen Probleminformation und der festen Probleminformation. Um das Durchwandern des Suchraums zu ermöglichen, ist in jedem Zustand ein Verweis auf den Vorgängerzustand abgelegt.

Am interessantesten ist die variable Zustandsinformation, die sich bei Anwendungen der Nachfolgerfunktion ändert:

- Die angefangene Stimme wird als Liste der Noten dargestellt. Sie wird bei jeder Nachfolgerbildung um eine Note erweitert.
- Ein Verweis auf den Vorgängerzustand
- Der Übergangoperator, d.h. die Darstellung der Alternative, mit der der Zustand vom Vorgängerzustand aus erreicht wurde.
- Aktuelle Bottom-up-Strukturelemente
- Aus Effizienzgründen wird Information, die eigentlich aus der Datenstruktur heraus bestimmt werden kann, aber sehr häufig benötigt wird, im Zustand gespeichert. Dazu zählt z.B. der aktuelle metrische Zeitpunkt, den man eigentlich durch Durchwanderung der bisherigen Zustände oder durch Aufsummieren der Dauern in der bisher komponierten Stimme bestimmen könnte.

Diese Datenstruktur genügt der in Kapitel 3.1 aufgestellten Forderung nach expliziten Repräsentationen.

Der feste Teil des Zustands, der als Hintergrundwissen bei der Komposition benötigt wird, enthält die Strukturbeschreibung und die bisher durch Stimmengenerierung erzeugten Teile der Komposition.

Startzustand Zu Beginn ist der komponierte Satz leer und es gibt keine Bottom-up-Strukturelemente. Die feste Information, nämlich die bisherige Chart und der Regelsatz werden eingetragen.

Nachfolgerfunktion Es gibt nun zwei Nachfolgerfunktionen:

- Eine für die Komposition des ersten Tons in einem leeren Satz. Dies ist der erste Zustandsübergang nach dem Startzustand.
- Eine zur Verlängerung eines nichtleeren Satzes um eine Note.

Die Nachfolgerfunktion wird durch Kombination von Regeln, die die Menge der zulässigen Alternativen bearbeiten, erstellt. Da die Auswahl der Regeln je nach Zusammenhang (Melodie, Baß) unterschiedlich ist, werden die Nachfolgerfunktionen (compilierte Lisp-Funktionen) für den ersten und die folgenden Zustandsübergänge im festen Teil des Zustands abgelegt.

Zieltest Die Erzeugung eines Satzes ist erfolgreich, wenn er die vom Strukturplan vorgesehene Länge erreicht hat und eine Endformel der durch die Makrostruktur vorgegebenen Art hat.

Fail-Test Die Komposition ist fehlgeschlagen, wenn sie die vorgegebene Länge erreicht hat, ohne mit einer den Vorgaben entsprechenden Endformel zu schließen oder wenn sie die vorgesehene Länge gar überschritten hat.

3.5.2 Alternativenauswahl durch Regeln

Die Stimmengenerierung beginnt mit einem Startzustand, bei dem die Notenfolge leer ist, und fügt dann Note um Note hinzu. Damit man mit unterschiedlichen Kombinationen von Gesetzmäßigkeiten experimentieren kann, wird die Menge der möglichen Fortsetzungen durch eine Folge von Stimmengenerierungsregeln bestimmt, die alle der Reihe nach angewandt werden. Die Fortsetzungsmöglichkeiten werden durch spezielle Datenobjekte, die *Alternativen*, repräsentiert.

Jede Stimmengenerierungsregel besteht aus Voraussetzung und Folgerung. Bei der Regelanwendung wird untersucht, ob die Voraussetzung erfüllt ist. In diesem Fall wird die Folgerung angewandt. Die Folgerung kann Alternativen zur aktuellen Alternativenmenge hinzufügen, löschen oder in den Alternativen Werte ändern. Die Alternativenmenge, die nach Anwendung der letzten Regel übrigbleibt, beschreibt die Folgezustände, die dann nach den Vorgaben der Alternative erzeugt werden.

Die Alternativen für die erste Note der Stimme werden absolut, durch Noten beschrieben. Für die folgenden Noten ist es bei vielen Gesetzmäßigkeiten der Stimmengenerierung einfacher, sie relativ, durch ein Intervall zum vorangehenden Ton zu formulieren, anstatt sie absolut durch den zu komponierenden Ton auszudrücken. Daher werden die Alternativen ab der zweiten Note durch Relativtöne - Kombinationen aus Intervall und Dauer - repräsentiert. Üblicherweise betrachtet man diatonische Intervalle innerhalb der augenblicklichen Tonart, d.h. man gibt an, daß eine Terz folgen soll, läßt aber offen, ob dies eine kleine oder eine große Terz sein soll. Dies ist innerhalb der Tonart durch den Startton des Intervalls festgelegt. Außerdem enthält die Al-

ternative die erzeugten (mitkomponierten) Bottom-up-Strukturelemente und einen Gewichtungsfaktor.

Der Regelsatz enthält für jede Stimme je zwei Regelfolgen, eine für den ersten Übergang (Startnote des Satzes) und eine für die folgenden Noten. Es werden unterschiedliche Regeln zur Bearbeitung der Alternativenmenge ausgewählt, die grundlegenden Mechanismen für Startnotenalternativen und fortsetzende Alternativen sind aber gleich.

3.5.3 Bottom-Up-Strukturelemente in Alternativen

Manche Alternativen sind nur zulässig, wenn sie in den folgenden Noten korrekt fortgeführt werden, d.h. die Komposition eine besondere Figur hat. Die Überprüfung solcher Gesetzmäßigkeiten ist Aufgabe von strukturellen Elementen. Im Gegensatz zu den Elementen, die durch die Strukturbeschreibung gegeben sind, sind die eben beschriebenen Bottom-up-Strukturelemente nicht vor Beginn dieses Suchvorgangs bekannt. Die Entscheidung für ein Bottom-up-Strukturelement fällt erst während des laufenden Suchvorgangs.

Eine Alternative, die eine solche Figur startet, enthält das entsprechende Bottom-up-Strukturelement. Wird die Alternative durch die Nachfolgerfunktion gewählt, dann wird dieses Strukturelement der aktuellen Komposition hinzugefügt, d.h. im Zustand abgelegt. Bei der Komposition der folgenden Noten berücksichtigen die Regeln nun die Existenz des Strukturelements.

Die Ausdehnung von Strukturelementen kann man auf unterschiedliche Art beschreiben, beispielsweise durch die Zahl der Noten, die zum Strukturelement zählt oder den metrischen Zeitbereich.

In der Strukturbeschreibung sind die Top-down-Strukturelemente durch den metrischen Zeitbereich begrenzt, da man vor dem Start der Stimmengenerierung nicht wissen kann, wieviele Noten an dieser Stelle in der Komposition stehen werden, jedoch festlegen möchte, für welchen Bereich der Komposition eine bestimmte Eigenschaft zutrifft.

Ein Bottom-up-Strukturelement wird während der Stimmengenerierung erzeugt und Note für Note verlängert. Dabei wird immer ein interner Zustand⁹ des Elements mitgeführt. Bei Erreichen des Endzustands des Elements ist das Element vollständig¹⁰. So kann man mittels Bottom-up-Strukturelementen auch komplizierte Figuren beschreiben.

In einfachen Fällen könnte man die Vollständigkeit auch durch die Anzahl der zum Element gehörenden Noten angeben. Eine Definition der Vollständigkeit eines Bottom-up-Strukturelements über den metrischen Zeitbereich ist im Prinzip auch denkbar - es wird hier aber nicht benötigt.

⁹Dieser wird *Elementzustand* genannt, im Gegensatz zum *Suchzustand*.

¹⁰Einzelheiten der Realisierung siehe Kapitel 4.5.4

Ein Anwendungsbeispiel für ein Bottom-up-Strukturelement ist der Durchgang. Die Durchgangsnote ist eine harmoniefremde Note, d.h. die Note ist ohne korrekte Fortführung nicht erlaubt. Deswegen enthält die Alternative, die die Durchgangsnote in die Komposition setzt¹¹ das Strukturelement. Wenn dann aus dieser Alternative der Folgezustand konstruiert wird, wird das Strukturelement in die Komposition eingetragen. Bei der Komposition der nächsten Note wird das Strukturelement durch eine spezielle Regel berücksichtigt. Diese entfernt für den Fall, daß ein Strukturelement für einen Durchgang offen ist, alle Alternativen, die diesen Durchgang nicht korrekt fortführen. Dieses komplizierte Vorgehen ist nötig, damit für die neue Note die anderen Regeln weiterhin gelten, z.B. muß die den Durchgang beendende Note wieder ein Harmonieton sein. Dies kann zur Folge haben, daß ein Zustand keine Folgezustände hat, d.h. der Durchgang an dieser Stelle nicht-gesetzt werden darf. Es ist also nicht möglich, das Problem zu lösen, indem man die durch den Durchgang festgelegte Fortführung einfach generiert, da sonst u.U. andere Regeln vernachlässigt würden.

Neben dem Durchgang sind als weitere Bottom-up-Strukturelemente der Viertelrepetitionstakt und der Quartvorhalt definiert.

3.5.4 Darstellung von Korrespondenzen

Ein wichtiges, für den klassischen Stil geradezu bezeichnendes Gestaltungselement sind Ähnlichkeiten zwischen Teilen des Stücks. Betrachtungen über solche Verwandtschaften und Variationen sind ein beliebtes Thema in der zeitgenössischen und systematischen Literatur.

Solche Beziehungen werden durch ein spezielles Strukturelement, die *Korrespondenz* repräsentiert. Das Korrespondenzelement beschreibt, wo sich das Vorbild, das sogenannte *Modell*, des korrespondierenden Bereichs befindet, und gibt Prädikate für die Ähnlichkeit von Korrespondenz und Modell an.

Es werden folgende Arten von Korrespondenz unterschieden:

Interne Korrespondenz Bei der internen Korrespondenz stammt das Modell aus den bereits komponierten Teilen des Stücks. Es wird angegeben, aus welcher Stimme das Modell stammt und wie groß der Zeitabstand zwischen Modell und Korrespondenz ist.

Externe Korrespondenz Die externe Korrespondenz gibt zusätzliches musikalisches Material als Modell vor.

Im Strukturplan werden Korrespondenzen durch eine Folge von Strukturelementen angegeben.

¹¹Genaugenommen: *setzen würde*, denn der Zustandübergang ist noch nicht erfolgt.

Die Art und Weise, wie die Korrespondenz dem Modell ähnelt, wird durch zwei Vergleichsprädikate beschrieben, einem für die erste Note der Korrespondenz und einem für den Rest der Korrespondenz.

Für den Vergleich zweier dauerbehafteter Intervalle sind folgende Vergleichsprädikate vordefiniert:

Rhythmusgleichheit Die Dauern sind bei Modell und Korrespondenz gleich.

Die im folgenden beschriebenen Prädikate für Intervallähnlichkeit schließen die Rhythmusgleichheit mit ein.

Intervallrichtungsgleichheit Es werden drei Kategorien von Intervalle unterschieden: aufwärts, abwärts und gleich.

Ähnlichkeit der Intervalle Die korrespondierenden Intervalle haben die gleiche Richtung. Die Schrittzahl unterscheidet sich höchstens um 1.

Diatonische Intervallgleichheit Die korrespondierenden Intervalle sind tonal gleich.

Dreiklangsschrittgleichheit Dreiklangsbasierte Melodien werden als ähnlich empfunden, wenn die korrespondierenden Intervalle die gleiche Anzahl von Dreiklangsschritten einschließen. So enthält beispielsweise das Intervall g'-d'' in G-Dur 2 Dreiklangsschritte und das Intervall g'-e'' in C-Dur ebenfalls zwei Dreiklangsschritte. Die beiden Intervalle sind damit dreiklangsschrittgleich.

Richtungsumkehrung Intervalle aufwärts korrespondieren mit Intervallen abwärts und Repetitionen korrespondieren miteinander.

Ähnlichkeitsumkehrung 2 Intervalle korrespondieren miteinander, wenn die Richtung umgekehrt wird und der Betrag der Schrittzahl sich nur um 1 unterscheidet.

Intervallumkehrung 2 Intervalle korrespondieren, wenn der Betrag der Schrittzahl gleich ist, die Richtung aber umgekehrt ist.

Das Korrespondenzelement zur Beschreibung der Sequenz abwärts enthält dann beispielsweise folgende Elemente:

- Die Korrespondenz folgt dem Modell unmittelbar, d.h. der Abstand zwischen Modell und Korrespondenz ist gleich der Länge des Strukturelements.
- Die erste Note der Korrespondenz ist eine Sekunde tiefer als die erste des Modells

- Die Intervalle in Korrespondenz und Modell sind gleich. Es wird mit diatonischen Intervallen gearbeitet.

Die Definition des Strukturelements sieht dann folgendermaßen aus:

```
$(sim-corr 6/4                                ;; Elementlaenge: 2 Takte
  '(:startfilter ,#'seq-starttest-f) ;; erste Note fuer Fonte
  '(:intfilter ,#'similar-int-p)      ;; Intervallaehnlichkeit
  '(:coffset $(timevec 3))))        ;; Abstand Korr.-Modell 2 Takte
```

Die Korrespondenzregel prüft, welche Alternativen den Korrespondenzen genügen, die die zu komponierende Note regieren. Die regierenden Korrespondenzen sind in drei Korrespondenzfolgen zu finden: In der Mikrostruktur stehen die von der Strukturkomposition erzeugten Korrespondenzelemente für Fonte und Monte, im Regelsatz sind interne Korrespondenzen und Korrespondenzen mit einem externen Modell angegeben.

3.5.5 Aufbau der Alternativenregeln

Die Syntax für die Definition von Regeln identifiziert die Bedingung und verschiedene Typen von Folgerungen der Regeln. Der Vorteil der Verwendung von Regeln anstelle einer ausprogrammierten Nachfolgerfunktion ist, daß die einzelnen Gesetzmäßigkeiten in unterschiedlichem Zusammenhang wiederverwendbar sind und man Regeln für verschiedene Experimente unterschiedlich zusammenstellen kann. Aufgabe des Regelcompilers ist es, die Codestücke aus den Regeln im richtigen Zusammenhang zu einer Funktion zusammenzubauen, die zu einem gegebenen Zustand die Menge der Alternativen generiert.

Die Regelsyntax behandelt nicht den Zugriff auf die COMF-Datenstrukturen. Stattdessen werden für Bedingung und Folgerung gewöhnliche Lisp-Ausdrücke verwendet, die auf einen festgelegten Satz von Variablennamen zugreifen können.

Die Zugriffe auf die Datenstrukturen sind im Lisp-Code bei geeigneter Definition der gerufenen Funktionen für die musikalischen Grundelemente ohnehin gut lesbar, so daß spezielle Definitionen für die Datenzugriffe in der Regelsprache - so wie es beispielsweise Regelsysteme wie OPS5 oder Regelsysteme für prozedurale Sprachen tun - keinen zusätzlichen Vorteil an begrifflicher Klarheit bringen.

Die Regeln für die Bearbeitung von Alternativen werden in folgendem Format definiert:

Syntax Stimmengenerierungsregeln:

```
(def-rule <name> [:start | :bi] Name und Gültigkeit
  [:cond <cond>]      Voraussetzung
  <folgtyp> <folge>) Folgerung
```

Die Regel wird unter ihrem Namen abgelegt. Das Gültigkeitsflag gibt an, ob die Regeln für die erste Note oder für die folgenden Noten verwendbar sind. Regeln mit dem Gültigkeitsflag `:start` sind nur für die Startnote des Satzes, solche ohne das `:start`-Flag sind nur für die folgenden, fortsetzenden Alternativen (Fortsetzungsintervalle) gedacht. Regeln mit dem Gültigkeitsflag `:bi` sind in beiden Fällen zulässig.

Die Regelbedingung oder Voraussetzung `:cond <cond>` ist ein Ausdruck, der einen booleschen Wert zurückgibt und bestimmt, ob die Folgerung ausgeführt wird oder nicht. In der Bedingung ist der Zustand über die Variable `st`, kurz für *state*) und die augenblickliche Alternativenmenge über die Variable `cs`, kurz für *continuation set* zugänglich. Läßt man die Bedingung weg, so wird die Folgerung immer angewandt (entspricht `<cond> = t`).

Aufgabe der Folgerungsangabe bestehend aus `<folgtyp>` und `<folge>` ist es, die gewünschte Änderung der Alternativenmenge zu spezifizieren. Die Folgerung besteht aus dem Folgerungstyp und dem Folgerungsausdruck. Je nach Folgerungstyp wird der Folgerungsausdruck entweder für die ganze Alternativenmenge oder für jede einzelne Alternative gerufen.

- In einem Ausdruck für die gesamte Alternativenmenge ist wie bei der Bedingung die Alternativenmenge und der Zustand zugänglich.
- Ein Ausdruck für eine einzelne Alternative wird für jede Alternative in der aktuellen Regelmenge gerufen. Der Ausdruck hat zusätzlich zu `st` und `cs` noch auf die Variable `ctel` (kurz für *continuation element*) Zugriff, die die aktuelle Alternative enthält.

Die Semantik der verschiedenen Folgerungstypen legt fest, wie mit dem Ergebnis des Folgerungsausdrucks weitergearbeitet wird. Für die Folgerungstypen, die in Kapitel 3.6 vorkommen, wird auch die Bezeichnung in der verbalen Beschreibung aufgeführt.

Folgerungstypen für Stimmengenerierungsregeln

:intset (Dann)

`<folge>` ist ein Ausdruck für die gesamte Menge. Das Ergebnis des Ausdrucks `<folge>` wird zur neuen Intervallmenge.

Gibt man beim Folgerungstyp `:intset` als Folgerungsausdruck die leere Liste an, so erhält man eine Regel, die prüft, ob der aktuelle Zustand zulässig ist. So kann man mit einer Regel ein Failprädikat simulieren.

:add (Füge hinzu)

`<folge>` ist ein Ausdruck für die gesamte Menge. Das Ergebnis des Ausdrucks `<folge>` wird zur Alternativenmenge hinzugefügt.

:filter (Verwende nur / Verwende keine)

`<folge>` ist ein boolescher Ausdruck, der für jede Alternative gerufen wird. Ist das Ergebnis `t`, wird die Alternative ins Ergebnis übernommen, sonst wird sie entfernt.

:map (Setze für alle)

`<folge>` wird für jede Alternative gerufen. In der Alternativenmenge wird die Alternative durch das Ergebnis des Ausdrucks ersetzt.

:show Druckt den Ausdruck `<folge>` auf den Bildschirm, ändert die Alternativenmenge aber nicht. Dieser Folgerungstyp dient zur Fehlersuche.

Außer den eben beschriebenen Alternativenregeln gibt es Hüllregeln. Steht eine Alternativenregel innerhalb einer Hüllregel, dann wird die Anwendung der eingehüllten Regel nur versucht, wenn die Bedingung der Hüllregel erfüllt ist.

Syntax der Hüllregel:

```
(def-rule <name> { :wrap-if | :in-section } <cond> <wrapped-rule> )
```

Es gibt also zwei Arten von Hüllregeln:

:wrap-if Die Bedingung ist ein Ausdruck abhängig von den Variablen `st` und `cs`.

:in-section Die Bedingung ist der Name eines Satzes. Die Regel wird nur berücksichtigt, wenn der aktuelle Suchvorgang im angegebenen Satz stattfindet. Mit solchen Hüllregeln kann man über die von der Makrostruktur vorgegebene Satzfolge die Regeln für einzelne Teile des Stücks zu- und abschalten.

Der Effekt von Hüllregeln ist der gleiche wie der einer zusätzlichen Voraussetzung. Hüllregeln ermöglichen es aber, Alternativenregeln und unterschiedliche Kontexte für ihren Einsatz getrennt zu implementieren. Beispielsweise kann man eine Regel implementieren und sie später durch Hüllregeln für Kontexte, gegeben durch strukturelle Elemente in der Strukturbeschreibung, aktivieren bzw. deaktivieren. Damit gewinnt man die zu Beginn dieses Kapitels geforderte Flexibilität.

3.6 Beschreibung der Alternativenregeln für die Stimmengenerierung

Mit dem eben beschriebenen Definitionsmechanismus werden nun Regeln definiert. Die Regeln sind in verbaler Form beschrieben und gegenüber der in Kapitel 3.5 beschriebenen Regelsprache in leicht vereinfachter Form abgedruckt. Es werden die dort eingeführten Begriffe benutzt. Zu beachten ist, daß es Regeln ohne Voraussetzung gibt. Ihre Folgerung wird auf jeden Fall ausgeführt. Die kleingedruckte Zeile unter dem Regelnamen gibt an, wie die Regeln üblicherweise in Regelfolgen eingesetzt werden.

Für Experimente werden aus der Gesamtmenge der im folgenden definierten Regeln Regelfolgen für Melodie und Baß gebildet. Für jede Stimme gibt es jeweils eine Regelfolge für die Startnote - die erste Note jedes Satzes - und für die Intervalle zwischen den Folgetönen.

In Menuetten der Klassik kann man zwei Grundtypen von Melodiebildungen beobachten:

Tonleiterbasiert In Tonleiterbasierten Melodieteilen sind die meisten Intervalle Sekunden.

Dreiklangsbrechungsbasiert Dreiklangsbasierte Melodieteile verwenden die Dreiklangstöne der aktuellen Harmonie.

Eine Melodie kann rein nach einem Grundtyp aufgebaut sein oder zwischen tonleiterbasierten und dreiklangsbasierten Abschnitten wechseln. Dies wird durch eine Folge von Strukturelementen (Tagelementen) im Regelsatz gesteuert. Die Regelfolge berücksichtigt den Melodietyp, indem sie Regeln, die für tonleiter- oder dreiklangsbasierte Melodiebildung spezifisch sind, nach dem Wert im aktuellen Tagelement der Melodietypstruktur mittels `:wrap-if-`Hüllregeln zu- und abschaltet.

3.6.1 Parameterwerte

Außer den Regelfolgen findet man im Regelsatz für ein Kompositionsexperiment noch Parameterwerte. Diese Werte können dann von den Stimmengenerierungsregeln in Voraussetzung und Folgerung verwendet werden. Die Parameter gelten wie in Kapitel 3.3 beschrieben, entweder für das ganze Stück oder für eine einzelne Stimme.

Parameter, die für alle Stimmen des Stücks gelten, sind Tonart, Taktart und die Parameter, die bei der Erstellung der Strukturbeschreibung berücksichtigt werden. Außerdem wird für das Menuett eine Viertelnote als maximale Dauer für eine Durchgangsnote angesetzt.

Für Melodie und Baß werden unter anderem¹² folgende stimmbezogene Parameter definiert:

Globale Tonumfangslimits Für die Melodie wird der Tonbereich g-e''' verwendet. Dies wurde durch Analysen festgestellt (WQ4). Der Bereich g-c' kommt dabei seltener vor.

Obwohl es sich um Klavierstücke handelt, halten frühklassische Menuette den typischen Ambitus der damaligen Melodieinstrumente ein.

Gelegentlich werden einzelne Ausreißer aus diesem Umfang beobachtet, sie sind aber besonders begründet (z.B. Oktavversetzung eines Motivs in Mozart KV 4[Moz91]).

Für den Baß wird der Bereich von C - f' verwendet.

Satztonumfangslimits Für die Melodie wird innerhalb eines Satzes der Umfang vom Ausgangston des Satzes ausgehend nochmals beschränkt. Der Bereich ist: eine Quarte nach unten, eine Duodezime (= Oktave + Quinte) nach oben.

Beim Baß gibt es hier keine Beschränkung.

Repetitionszahl Die maximal erlaubte Anzahl von Tönen der gleichen Tonhöhe, die direkt hintereinander kommen, kann hier beschränkt werden.

Für die Melodie wird hier der Wert 2, für den Baß der Wert 3 benutzt.

Startmenge für die Alternativenintervalle Bei der Abarbeitung der Regelfolge wird mit dieser Startmenge begonnen.

Bei der Melodie enthält die Startmenge Sekunden aufwärts, Sekunden abwärts und Primen, beim Baß alle Intervalle kleiner als eine Oktave ausgenommen die Septime abwärts.

Bei den Parametern handelt es sich um Werte, die während der Komposition eines Satzes für eine Stimme unverändert bleiben. Die Ergebnisse vorangegangener Kompositionsschritte (Strukturbeschreibung, fertig komponierte Stimmen) verhalten sich genauso und werden gleich behandelt.

¹²Es gibt noch weitere Parameter. Diese werden bei der Beschreibung der Regeln, in denen sie verwendet werden, mit erläutert.

3.6.2 Regeln für die Startnoten eines Satzes

Die Startregeln dienen zur Bestimmung der Anfangstöne eines Satzes. Zu diesen Regeln tritt noch die Korrespondenzregel, die für Startnoten und Folgeintervalle gilt.

Regel 1 Harmonietöne innerhalb Tonraum

Verwende nur Töne aus der aktuellen Harmonie innerhalb des Tonraums für Startnoten.

Die Grenzen des Tonraums für die erste Note sind für jede Stimme im Regelsatz definiert.

Regel 2 Übergang zwischen Sätzen

Verwende nur Töne, die die im Parametersatz angegebenen maximalen Übergangsintervalle einhalten

Regel 3 Startdauern

Dann verbinde jede Alternative mit jeder der angegebenen Startdauern.

Hier wird das kartesische Produkt der Alternativnoten und der Startdauern gebildet.

Regel 4 1. Stufe in erstem Akkord

Regel für den Baß

Bedingung Die Melodie enthält nicht die erste Stufe der aktuellen Harmonie

Verwende nur die erste Stufe der aktuellen Harmonie

Diese Regel fordert, daß der erste Akkord jedes Satzes entweder in Melodie oder Baß die erste Stufe enthält. Dies sorgt für eine klar hörbare Anfangstonart (WQ(2),3,4,6).

3.6.3 Allgemeine Intervallregeln

Die folgenden Regeln bearbeiten die Menge der zulässigen Folgeintervalle. Sie werden zum Aufbau der Regelfolgen für tonleiterbasierte Melodien, dreiklangsbasierte Melodien und für die Baßstimme benutzt.

Regel 5 Intervallmenge

Dann Verwende die Startintervallmenge aus Parametern.

Für die Melodie enthält die Startintervallmenge die Sekunde abwärts, die Repetition und die Sekunde aufwärts, für den Baß alle Intervalle aufwärts und abwärts bis einschließlich zur Oktave ohne die Septimen.

Regel 6 Terzen einstreuen

Regel für skalenbasierte Melodiebereiche

Bedingung Das vorangehende Intervall ist eine Sekunde oder kleiner.

Dann Füge zur Alternativenmenge die Terz abwärts und aufwärts hinzu.

Regel 7 Tritonusverbot

Verwende keinen Tritonus abwärts.

Eine strengere Variante verbietet den Tritonus überhaupt.

Regel 8 Repetitionslimit

Bedingung Die Zahl der vorangehenden gleichen Töne erreicht das Repetitionslimit aus den Parametern.

Verwende keine Primen.

Regel 9 Intervallsummenlimit

Regel für den Baß

Verwende nur Intervalle, deren Summe mit dem letzten Intervall unter einer Oktave bleibt.

Falls eines der beiden Intervalle eine Oktave ist, ist die Grenze eine Dezime.

Große Sprünge im Baß sind verbreitet, zwei große Sprünge in die gleiche Richtung machen den Baß aber zu unruhig und zu unzusammenhängend (WQ4,6).

Regel 10 Sequenzverbindung

Bedingung Ein Korrespondenzelement für eine Sequenz beginnt gerade.

Füge hinzu Intervalle aufwärts und abwärts bis zur Quinte

Bei einer Sequenz wird die Verbindung und damit die melodische Qualität durch die Verwandtschaft hergestellt. Sonst geschieht dies bei skalenbasierten Melodien durch die Verwendung kleiner Intervalle. Damit die möglichen sequenzierenden Motive nicht zu sehr eingeschränkt werden, erlaubt diese Regel größere Sprünge. Die Regel muß in der Regelfolge früh stehen, damit die Alternativen sämtlichen anderen Überprüfungen unterzogen werden (WQ4,6)

Regel 11 Ambitusprüfung

Verwende nur Noten innerhalb des durch die Parameter vorgegebenen Ambitusbereichs für Satz und Stück.

Der Ambitusbereich wird wie oben beschrieben durch die globalen Limits für die Stimme und bei der Melodie noch durch zusätzliche Limits für den laufenden Satz definiert. Globale Limits sind historisch durch die üblichen Tonumfänge der Melodieinstrumente gegeben, die in der Frühklassik auch für die Klaviermusik eingehalten wurden. Außerdem findet man bei [Rie52] Hinweise und Beispiele auf den üblichen Umfang von Melodie- und Baßstimme. Einschränkungen für den laufenden Satz wurden aus Beispielen von Riepel und durch die Beobachtung von Originalwerken und Kompositionsexperimenten gewonnen. (WQ2,4)

3.6.4 Leittonregeln

Regel 12 Leitton (stark)

Bedingung Der letzte Melodieton ist in der augenblicklichen Tonart die 7. Stufe.

Verwende nur Alternativen mit Intervall Sekunde aufwärts.

Die Fortführung des Leittons ist bei Koch beschrieben und gehört auch zu den im Common-Practice Bereich üblichen Regeln (WQ2).

Regel 13 Leitton akkordisch

Bedingung Der letzte Melodieton ist in der augenblicklichen Tonart die 7. Stufe und

Die Harmoniebasis für Vorgängernote und augenblickliche Note ist nicht die 5. Stufe.

Verwende nur Alternativen mit Intervall Sekunde aufwärts.

Die akkordische Leittonregel ist eine abgeschwächte Variante der starken Leittonregel. Die akkordische Leittonregel erlaubt innerhalb von Dreiklangsbrechungen der 5. Stufe (Dominante) auch andere Fortführungen der 7. Stufe, da sie die Rolle als Terz der Dominante gegenüber der Rolle als Leitton stärker gewichtet. (WQ4,6)

Regel 14 Leitton-Dominantsept stark

Regel für skalenbasierte Melodien

Bedingung Die Harmonie ist ein Dominantseptakkord und der letzte Melodieton ist die 4. Stufe der augenblicklichen Tonart (Septime des Dominantseptakkords),

Verwende nur Alternativen mit Intervall Sekunde abwärts.

Diese Regel ist beinahe wörtlich so bei [Koc82] beschrieben (WQ2).

Regel 15 Leitton-Dominantsept akkordisch

Regel für dreiklangsbasierte Melodien

Bedingung Die Harmonie ist ein Dominantseptakkord und

der letzte Melodieton die 4. Stufe der augenblicklichen Tonart (Septime des Dominantseptakkords) ist,

Verwende nur Alternativen mit Intervall abwärts.

Bei Harmoniewechseln nur eine Sekunde abwärts.

Diese abgeschwächte Variante der Septimauflösung wird in bei dreiklangsbasierten Melodieabschnitten eingesetzt. Die korrekte Auflösung wird nur bei Harmoniewechsel verlangt, nicht innerhalb einer Akkordbrechung (WQ4,6).

3.6.5 Dreiklangsschrittregeln**Regel 16 Dreiklangsschritt**

Regel für Dreiklangsmelodien

Bedingung Die Harmonie wechselt nicht.

Verwende nur Intervalle zum benachbarten Dreiklangston nach oben und nach unten.

Diese Regel erzeugt die Akkordbrechungen, während eine Harmonie steht. Varianten dieser Regel erlauben zusätzlich Repetitionen oder auch Sprünge von zwei Dreiklangsschritten. Die zu Beginn definierten Startintervalle werden dann nur zur Verbindung zwischen zwei Dreiklangabschnitten (durch Harmoniewechsel begrenzt) benutzt, innerhalb eines Dreiklangabschnitts gilt die gerade beschriebene Regel.

Varianten dieser Regel erlauben noch Repetition oder doppelte Dreiklangsschritte.

Regel 17 Terz in der Akkordbrechung*Regel für Dreiklangsmelodien.***Bedingung** Die Harmonie steht einen ganzen Takt lang.**Verwende** Eine den Takt vervollständigende Alternative nur dann, wenn im Takt die Terz des Akkords vorkommt.

Wenn eine Harmonie einen ganzen Takt steht, soll der Akkord durch die Melodie möglichst vollständig skizziert werden (WQ3,4,6).

Regel 18 Verbiete seltsame Dominantseptakkordbrechungen*Regel für dreiklangsbasierte Melodien***Bedingung** In der Brechung eines Dominantseptakkords treten hintereinander die Stufen 7-1-7 oder 5-7-1 auf.
(Stufen relativ zum **Akkord**grundton)**Dann** Lösche die Alternativenmenge.

Im Gegensatz zu den anderen Regeln betrachtet diese Regel mehrere vorangehende Töne. Das Löschen der Alternativenmenge entspricht einem Verbot, da der Zustand dann keine Nachfolger hat.¹³ Diese Regel muß in der Regelfolge nach allen Regeln, die Alternativen hinzufügen, stehen, um wirksam zu sein.

In Brechungen des Dominantseptakkords tritt die Septime zum Tonvorrat der Akkordbrechung dazu. Diese Septime hat also eher die Rolle eines gleichberechtigten Akkordtons als die einer dissonanten Zutat. Diese Auffassung ist für den klassischen Stil durch die zeitgenössische Literatur gedeckt. Die von dieser Regel verbotenen Stufenfolgen vermitteln nicht den typischen Klangeindruck des Dominantseptakkords. Die Regel wurde durch Untersuchung der Ergebnisse des Systems gefunden (WQ6).

¹³Abgesehen von dem Fall, daß Regeln, die später in der Regelfolge stehen, auch zu einer leeren Alternativenmenge etwas hinzufügen können.

3.6.6 Regeln zu Betonung und Dauer

Regel 19 Default-Dauerrestriktion

Setze für alle Alternativen die Dauerrestriktion auf den Defaultwert aus dem Parametersatz.

Für die Melodie wird als Defaultrestriktionsangabe `:min-mel-durs` benutzt, der für die Einhaltung der in Kapitel 3.6.12 beschriebenen Regelungen für Menuettrhythmen nach Riepel sorgt. Für den Baß werden mit der Restriktionsangabe `:no-sync` synkopenfreie Dauern gefordert.

Regel 20 Keine Achtelrepetition auf Schlag 1

Bedingung Der metrischen Zeitpunkt ist der erste Schlag, aber nicht Schlagteil 0.

Verwende keine Primen

Eine verschärfte Variante dieser Regel ist es, Achtelrepetitionen ganz zu verbieten. (WQ6)

Regel 21 Keine Schwach-Stark-Repetition

Bedingung Die Betonung der zu komponierenden Note ist stärker als die Betonung der Vornote.

Verwende keine Primen.

Obwohl diese Regel Wissen zu Dauern und zu Intervallen kombiniert, muß sie keine Dauernrestriktionen setzen, da nur das metrische Gewicht des Anfangszeitpunkts entscheidend ist und nicht die Dauer der zu komponierenden Note (WQ6).

Repetitionen allgemein (Verbot Regel 10) und Stark-Schwach-Repetitionen (Verbot durch Regel 11) führen zu seltsamen Gewichtsverteilungen und Betonungen in der Melodie und geben der Melodie den Charakter einer Mittelstimme. Die Regel wurde durch Beobachtung und Beurteilung von Kompositionsexperimenten gefunden.

3.6.7 Harmonie und Zweistimmigkeit

Regel 22 Harmonietöne

Verwende nur Intervalle, deren Zielton in der Harmonie liegt.

Die Regeln zum Einbau nichtharmonischer Töne (z.B. Durchgänge) definieren Ausnahmen davon. Dies geschieht, indem die Regel für den nichtharmonischen Ton in der Regelfolge nach dieser Regel steht. (WQ1,2)

Regel 23 Keine Terzverdopplung

Regel für den Baß

Bedingung Der aktuelle Melodieton ist die 3. Stufe des aktuellen Akkords.

Verwende keine 3. Stufe

Diese Regel berücksichtigt nur Melodietöne, die gemeinsam mit dem Baßton anschlagen. (WQ2,3)

Eine strikte Variante dieser Regel prüft auch Melodietöne, die nach dem Baßton anschlagen, aber gemeinsam mit dem Baßton erklingen, weil der Baßton liegen bleibt.

Regel 24 Stufen 1, 3 oder Umkehrung

Verwende nur die geforderte Stufe der Umkehrung

Falls keine Umkehrung angegeben verwende die Stufen 1 und 3

Dies sind die üblichen Stufen für den Baß. Die Regeln zum Einbau nichtharmonischer Töne (z.B. Durchgänge) definieren Ausnahmen davon. Dies geschieht, indem die Regel für den nichtharmonischen Ton in der Regelfolge nach dieser Regel steht.

Regel 25 Konsonanz mit Melodie

Bedingung Der gleichzeitige Melodieton ist nicht die Septime eines Dominantseptakkords.

Verwende nur Alternativen, die zu einem konsonanten Intervall zwischen Baß und Melodie führen.

Da diese Regel für den Anschlagszeitpunkt des Basses geprüft wird, sorgt sie Zusammen mit der Harmonietonregel dafür, daß der Baß während eines Durchgangs in der Melodie liegen bleibt und nicht zur Durchgangsnote der Melodie hin dissonant anschlägt (WQ1,2).

Regel 26 Keine gleichzeitigen langen Töne

Regel für den Baß

Bedingung Der Melodieton ist länger als eine Viertelnote.

Verwende nur Alternativen mit maximaler Dauer einer Viertelnote.

Die zeitgenössische Literatur empfiehlt, daß nicht beide Stimmen gleichzeitig ruhen sollen, d.h. wenn die Melodie eine lange Note hat, soll der Baß sich bewegen und umgekehrt. (WQ2)

Regel 27 Unvollkommene Konsonanz auf Schlag 1

Regel für die Baßstimme

Bedingung Der Metrische Zeitpunkt ist Schlag 1, Schlagteil 0.

Verwende nur Alternativen, deren Intervall zur Melodie eine unvollkommene Konsonanz ist.

Diese Regel soll im zweistimmigen Satz dafür sorgen, daß die Harmonie so gut wie möglich skizziert wird, d.h. einigermaßen vollständig wirkt, um so dem klassischen Klangideal zu entsprechen. Leere Oktaven auf betonter Zeit sind zwar korrekt und legen die Harmonie sicher fest. Andererseits wirken sie klanglich „hohl“ und unvollständig. Für die erste Note eines Satzes und für den Bereich der Endformeln wird diese Regel daher durch eine Hüllregel deaktiviert. Die Endformeln fordern in den Außenstimmen an einigen Stellen Oktaven und Quinten fordern.

Anstoß zu dieser Regel gab die Untersuchung von Ergebnissen des Systems und die Introspektive bei eigenen Kompositionsversuchen (WQ3,6,7).

Regel 28 Unvollkommene Konsonanzen (strikt)

Diese Regel ist eine strengere Fassung von Regel RN27. Hier sind vollkommene Konsonanzen außerhalb der Endformeln durchweg verboten.

Regel 29 Verbot von Stimmkreuzungen

Verwende nur Alternativen, die nicht höher als der augenblickliche Melodieton sind.

(WQ2)

3.6.8 Präferenzregeln (Weiche Regeln)

Ein höheres Suchgewicht erhöht die Wahrscheinlichkeit, daß die Alternative ausgewählt wird. Die Alternativen mit geringem Gewicht sind aber immer noch möglich. Wichtig ist dies insbesondere, wenn die bevorzugten Alternativen später von einer anderen Regel aus der Alternativenmenge gestrichen werden.

Regel 30 Bevorzuge kleine Intervalle

Regel für die Baßstimme

Bedingung Der aktuelle Zeitpunkt gehört nicht zu einer Endformel.

Setze für alle Intervalle bis zur Quarte das Gewicht herauf.

(WQ2,3)

Regel 31 Sekunden in gleicher Richtung bevorzugen

Regel für skalenbasierte Melodien

Setze für alle Sekunden in der gleichen Richtung wie das Vorintervall das Suchgewicht herauf.

(WQ4,6)

Regel 32 Bevorzugung von Gegensprüngen*Regel für die Baßstimme***Wenn** das letzte Intervall eine Terz oder größer ist**Setze für alle** Alternativen in die entgegengesetzte Richtung das Suchgewicht herauf

(WQ3)

Regel 33 Bevorzugung der Gegenbewegung*Regel für den Baß***Setze für alle** Alternativen, bei denen zwischen Baß und Melodie eine Gegenbewegung entsteht, das Suchgewicht herauf.

(WQ1,3)

Regel 34 Prinzip kürzester Weg**Verwende nur** von zwei Intervallen \neq Oktave/Prime zum gleichen Ton in verschiedenen Oktaven das kürzere der beiden.

Beispiel: Vom Ton e ist sowohl eine Terz nach unten zum c als auch eine Sexte nach oben zum c' in der Alternativenmenge. Dann wird die Sexte gestrichen. Wäre die Terz nach unten aber nicht in der Alternativenmenge (weil sie durch eine andere Regel verboten wurde), so bleibt die Sexte als Alternative erhalten. Somit wird versucht, einen Zielton auf möglichst kurzen Weg zu erreichen.

Oktavsprünge dienen der Belebung der Baßlinie und werden den Repetitionen nicht vorgezogen. (WQ3,6)

3.6.9 Regeln für Bottom-up-Strukturelemente

Bisherige Regeln haben in der Strukturbeschreibung angegebene (top-down)-Strukturelemente berücksichtigt, indem in den Voraussetzungen einer Regel geprüft wird, ob ein Strukturelement mit bestimmten Parameterwerten vorhanden und zum augenblicklichen metrischen Zeitpunkt gültig ist.

Bottom-Up-Strukturelemente werden dagegen durch die Regeln eingeführt. Sie werden dann bei der Komposition der folgenden Noten von den Regeln

berücksichtigt und weitergeführt. Anfang und Ende der Bottom-up-Strukturelemente werden durch den Zustand des Strukturelements angegeben (vgl. Kapitel 3.5.3).

Die korrekte Behandlung des diatonischen Durchgangs wird durch zwei Regeln realisiert (WQ2,3):

Regel 35 Durchgang-Eröffnung

Bedingung Die Harmonie ändert sich nicht und es existiert nicht bereits ein unvollendetes Strukturelement und die Betonungsstufe ist nicht volltaktig.

Füge hinzu:

- Sekunde aufwärts mit zusätzlicher Dauernrestriktion für den Durchgang und neuem Strukturelement *Durchgang aufwärts*
- Sekunde abwärts mit zusätzlicher Dauerrestriktion für den Durchgang und neuem Strukturelement *Durchgang abwärts*

Die Bedingung der Durchgangseröffnung prüft, ob irgend ein Bottom-up-Strukturelement im Suchzustand existiert und verhindert damit, daß die Komposition versucht, gleichzeitig zwei Strukturelemente zu beginnen (die in den meisten Fällen unverträglich wären).

Die Eröffnung eines Durchgangs führt einen nichtharmonischen Ton, d.h einen Ton, der nicht im Tonvorrat der aktuellen Harmonie enthalten ist, ein. Daher muß die Durchgangseröffnungsregel in der Regelfolge nach der Regel zur Überprüfung der Harmonie stehen, damit letztere nicht die Alternativen für die Komposition des Durchgangs entfernt.

Die Dauerrestriktion für Durchgänge beschränkt die maximale Dauer der Durchgangsnote. Sie berücksichtigt das Limit aus dem Parametersatz (für Menuette ist es 1/4) und sorgt dafür, daß die Durchgangsnote kein stärkeres metrisches Gewicht als ihre beiden Nachbarnoten und keine längere Dauer als der Vorgängerton bekommt. Auch hier ist das recht komplizierte Vorgehen nötig, damit die üblichen anderen Einschränkungen bei der Dauernvergabe weiter gelten (siehe Kapitel 3.6.12).

Regel 36 Durchgang-Vollendung

Bedingung Der Suchzustand enthält ein Strukturelement vom Typ Durchgang, das noch nicht vervollständigt ist.

Dann Führe den Durchgang in der im Strukturelement angegebenen Richtung weiter.

Diese Regel läßt nur den verlängerten Durchgang in der Alternativenmenge und entfernt alle anderen Alternativen (siehe Definition Folgerungstyp `:intset/Dann` in Kapitel 3.5). Da die Zielnote des Durchgangs in der Harmonie sein muß, steht diese Regel vor der Harmonietonregel (RN22).

Regel 37 Melodieparalleler Durchgang

Regel für den Baß

Bedingung In der Melodie ist eine Durchgangnote.

Füge hinzu Note und Strukturelement für Durchgang in die gleiche Richtung.

Im Normalfall bleibt der Baß während eines Durchgangs in der Melodie liegen. Wenn zwischen Baß und Melodie eine unvollkommene Konsonanz ist, kann man auch in beiden Stimmen einen parallelen Durchgang setzen. Parallele Durchgänge von vollkommenen Konsonanzen werden von den Stimmführungsregeln abgewürgt, so daß diese Voraussetzung nicht extra geprüft werden muß.

Dieser Durchgang wird mit der normalen Regel zur Durchgangsvollendung RN36 vollendet.

Eine Ausnahme von den Repetitionsbeschränkungen ist eine menuetttypische Figur aus einem Takt, der drei gleiche Viertelnoten enthält. Die Regeln müssen vor der Harmonietonregel stehen (WQ2,4).

Regel 38 Vierteltakt-Beginn

Bedingung Der metrische Zeitpunkt ist Schlag 1 Schlagteil 0.

Füge hinzu Alternative mit Dauer Viertelnote und neuem Strukturelement für den Vierteltakt.

Regel 39 Vierteltakt-Fortsetzung

Bedingung Ein unvollständiges Strukturelement für einen Vierteltakt existiert.

Dann Setze eine Prime mit Dauer einer Viertelnote und führe das Strukturelement in den nächsten Elementzustand.

Auch hier ist nach Anwendung der Regel die Fortführung des Strukturelements die einzige Alternative in der Alternativenmenge.

Vorhalte sind nichtharmonische Töne auf betonter Zeit, die auf unbetonter Zeit aufgelöst werden. Die Dissonanzwirkung des Vorhalts soll das Stück interessanter machen. Im Gegensatz zum Durchgang, der von der musikalischen Betonung her unauffällig erzeugt der Vorhalt eine Art Schwerpunkt. Nach den strengen Satzregeln müssen Vorhalte vorbereitet und aufgelöst werden. Als Vorbereitung muß der Vorhaltsnote der gleiche Ton vorausgehen (bzw. eine Überbindung stehen). Aufgelöst wird durch einen Sekundschritt in eine Konsonanz hinein. Beim Quartvorhalt geht die Auflösung nach unten.

Regel 40 Quartvorhalt-Beginn

Regel wird nach Harmonie- und Repetitionsprüfung, aber vor Leittonprüfung eingesetzt

Bedingung Der metrische Zeitpunkt ist Schlag 1 Schlagteil 0 und Der vorangehende Ton ist die 4. Stufe der augenblicklichen Harmonie.

Dann Füge eine Prime hinzu mit einem neuen Strukturelement vom Typ Quartvorhalt.

Diese Regel erzeugt einen Quartvorhalt auf betonter Zeit, der nach den strengen satztechnischen Regeln durch die gleiche Note vorbereitet wird (WQ2,3).

In klassischen Kompositionen sind auch Quartvorhalte auf andere Zählzeiten und unvorbereitete Vorhalte zu finden. Koch [Koc82] unterscheidet bei den Dissonanzen zwischen strengem und freiem Satz. Die strenge Fassung der Regel erzeugt bei weitem nicht alle Quartvorhalte, die in klassischen Kompositionen zu finden sind, vermeidet aber unpassende Vorkommen und Verwendungsweisen weitgehend.

Regel 41 Quartvorhalt-Auflösung

Regel wird vor Harmonieprüfung eingesetzt

Bedingung Ein begonnener Quartvorhalt liegt vor.

Verwende nur Sekunden nach unten.

3.6.10 Regeln zur Beschleunigung von Endformeln

Die folgenden Regeln sind von der Theorie her nicht nötig, da für die Einhaltung der von ihnen festgelegten Bestimmungen ohnehin durch die Endformel gesorgt ist. Ihre Aufgabe ist es, die Zahl der Kompositionsschüsse, die an der Endformel scheitern, durch ein paar einfache Überprüfungen zu reduzieren und damit die Komposition zu beschleunigen. Beide Regeln werden für den Baß benutzt.

Regel 42 Endformel-Viertel-Hilfe

Bedingung Der aktuelle Zeitpunkt gehört zu einer Endformel.

Verwende nur Viertelnoten

Regel 43 Kadenz-Baß-Schlußtakt-Hilfe

Wenn Der aktuelle Zeitpunkt ist der letzte Takt einer Kadenz.

Verwende nur Intervalle abwärts und die Stufen 1 und 5

3.6.11 Aufruf anderer Komponenten

Regel 44 Dauernvergabe

Für Melodie und Baß, Details siehe Kapitel 3.6.12

Verbinde jede Alternative mit jeder Dauer, die durch die Dauernrestriktion der Alternative zugelassen ist.

Regel 45 Stimmführung

Regel für die Baßstimme, Details siehe Kapitel 3.7

Verwende nur Alternativen, die den Stimmführungsregeln genügen.

Zur Überprüfung der Stimmführungsregeln wird die entsprechende Funktion gerufen. Die Arbeitsweise wird in Kapitel 3.7 beschrieben.

Regel 46 Korrespondenzprüfung

Regel für Startnoten und Intervalle in allen Stimmen, Details siehe Kapitel 3.5.4

Bedingung Der metrische Zeitpunkt wird von (einer) Korrespondenz(en) regiert
(d.h. in Regelsatz oder Mikrostruktur sind für den aktuellen Zeitpunkt Korrespondenzen eingetragen.)

Verwende nur Alternativen, die den Forderungen aller dieser Korrespondenzen genügen.

3.6.12 Rhythmus und Dauernvergabe

Die Mehrzahl der Regeln befaßt sich entweder nur mit der Dauernvergabe oder nur mit den Intervallen. Dies ist dadurch gerechtfertigt, daß für die Kompositionsentscheidungen das metrische Gewicht, d.h. die Zeitposition des zu komponierenden Tons wichtig ist, während dessen Dauer eine geringere Rolle spielt. Die Dauer des zu komponierenden Tons bestimmt dann die Zeitposition des Folgetons, da die Stimme Note für Note komponiert wird. Bei den üblichen Regelsätzen werden zuerst die zulässigen Intervalle festgelegt und es werden anschließend die zulässigen Dauern vergeben.

Einige, wenige Regeln kombinieren Intervalle und Dauern. Die Alternativenelemente enthalten einen Slot für die Dauer, die das Element vom Relativton erbt. Die Stimmengenerierungsregeln können diesen Slot setzen. In den meisten Fällen wird der Dauernslot durch die Dauernvergaberegeln RN44 belegt. Ist dieser Dauernslot belegt, bedeutet dies, daß das Alternativenelement genau diese Dauer hat.

Anstelle einer festen Belegung des Dauernslots kann man in der Alternative die Menge der möglichen Dauern über eine *Dauernrestriktion* abstrakt angeben, und sie dann durch die Dauernvergaberegeln expandieren lassen. Dies hat u.A. folgende Vorteile:

- Zu Beginn der Alternativenbearbeitung muß nicht mit je einem Element für jede mögliche Dauer gearbeitet werden, sondern nur mit einem Element.
- Über die Dauernrestriktionen können abstrakte Begriffe eingeführt werden, die später relativ zum momentanen metrischen Zeitpunkt aufgelöst werden können. Dies erleichtert das Formulieren der Regeln.

Eine Regel kann eine bestehende Restriktion schrittweise erweitern, indem zur üblichen Dauernvergabe zusätzliche Einschränkungen hinzugefügt

werden. Die Dauernrestriktion ist ein Ausdruck, der die Menge der zulässigen Dauern beschreibt. Folgende Elemente sind in einer Dauernrestriktion möglich:

Bruchzahl Eine Bruchzahl erlaubt genau die angegebene Dauer.

Liste, erstes Element :and Diese Liste erlaubt nur Dauern, die alle Restriktionen in der Liste erfüllen. Die beschriebene Dauernmenge ist also die Schnittmenge der Dauernmengen der Elemente.

Sonstige Liste Die Liste beginnt nicht mit **:and** und besteht aus Dauernrestriktionen. Eine Dauer muß eine der Restriktionen aus der Liste erfüllen. Die beschriebene Dauernmenge ist die Vereinigungsmenge der Dauernmengen der Elemente.

Symbol :min-mel-durs Dies steht für die synkopenfreien Dauern nach den Menuetttaktformen von Riepel aus der im Regelsatz angegebenen Grundmenge.

Symbol :no-syn Dieses Symbol steht für beliebige synkopenfreie Dauern aus der im Regelsatz angegebenen Grundmenge.

Beispielsweise erlaubt der Dauernrestriktionsausdruck (**:and :no-syn (1/2 1/4 1/8)**) Halbe, Viertel-, und Achtelnoten, sofern sie keine Synkope erzeugen. Das Element **:and** verlangt, daß die Schnittmenge aus der Menge der synkopenfreien Dauern und (1/2 1/4 1/8) gebildet wird.

Die Dauernrestriktion **:min-mel-durs** steht für Dauern entsprechend den Menuetttaktformen nach Riepel, mit denen bei der Melodie der für diese Tanzgattung typische Rhythmus bestimmt wird. Hier sind folgende Arten von Menuetttakten definiert:

vollkommen Ein vollkommener Menuetttakt besteht aus 3 Vierteln, von denen bis zu zwei Viertel in Achtel geteilt werden dürfen. Takte die nur aus Achteln bestehen erlaubt Riepel als sogenannte *Lauffer* nur für Tonleiterketten, die melodisch die letzte Kadenz vorbereiten.

unvollkommen Ein unvollkommener Menuetttakt darf auch halbe Noten enthalten.

todt Ein *todter*¹⁴ Takt besteht aus einer einzigen Note (im 3/4-Takt einer punktierten Halben). Riepel erlaubt ihn im Menuett nur für die Schlußnoten der Sätze.

¹⁴historische Orthographie

<i>Nummer</i>	<i>Regelbezeichnung</i>
RN1	Harmonietöne innerhalb Tonraum
RN2	Satzübergang
RN3	Startdauern
RN46	Korrespondenzprüfung

Abbildung 3.4: Regelfolge für die Startnoten der Melodie

<i>Nummer</i>	<i>Regelbezeichnung</i>
RN1	Harmonietöne innerhalb Tonraum
RN4	1. Stufe in erstem Akkord
RN3	Startdauern
RN46	Korrespondenzprüfung

Abbildung 3.5: Regelfolge für die Startnoten des Basses

Die Begriffe stammen aus [Rie52]. Die Definitionen sind nicht explizit gegeben, sondern wurden aus den Beispielen erschlossen. Die Begriffe *vollkommen* und *unvollkommen* sind speziell für das Menuett eingeführt. Die Bezeichnung *todt* wird allgemein für einen Takt, in dem nur eine Note steht, verwendet.

Riepel ist hier verhältnismäßig streng, da er punktierte Viertel mit der Bemerkung „*taugt allenfalls für einen hinkenden Tanzmeister*“ verbietet. In klassischen Menuetten sind diese jedoch immer wieder anzutreffen und werden auch hier erlaubt. Auch die Achtelketten findet man in realen Kompositionen auch an Positionen vor, an denen Riepel sie nicht erlaubt.

3.6.13 Beispiele für Regelfolgen

Die Stimmengenerierungsregeln werden zum Aufbau von Regelsätzen benutzt. Für jede Stimme gibt es eine Folge von Stimmengenerierungsregeln für die Startnote und eine Folge für die weiteren Intervalle. Aus der Gesamtmenge der oben definierten Stimmengenerierungsregeln wird eine Auswahl getroffen.

Man kann mit unterschiedlichen Regelsätzen experimentieren. Hier werden einige typische Regelfolgen vorgestellt.

Die Regelfolge für die Bestimmung der Anfangsnoten ist verhältnismäßig kurz. Abbildung 3.4 zeigt ein Beispiel für die Anfangsnote eines Satzes der Melodie. Die in Abbildung 3.5 aufgeführte Regelfolge für die Anfangsnoten des Basses unterscheiden sich von der der Melodie nur in einer Regel.

Abbildung 3.6 zeigt eine typische Regelfolge für eine Melodie, die dreiklangs- und skalenbasierte Teile verbindet. Mittels Hüllregeln (vgl. Kapitel 3.5.5)

werden für Dreiklangsbrechungs-, Skalen und Endformelbereiche Regeln zu- bzw. abgeschaltet. Die Bereiche sind in der Strukturbeschreibung festgelegt. Die Endformeln gehören zum Skalenbereich.

Die Durchgangseröffnung (RN35) wird nach der Prüfung des Tonvorrats der Harmonie (RN22) vorgenommen, da der Durchgangston ein nichtharmonischer Ton ist, d.h. er ist vom Zwang, den Tonvorrat der Harmonie einzuhalten, ausgenommen. Die Vollendung des Durchgangs muß dagegen der Harmonieprüfung unterzogen werden, so daß RN36 vor RN22 steht.

Die in Abbildung 3.7 beschriebene Regelfolge für die Baßstimme enthält einen Teil der Intervallregeln für die Melodie und ergänzt sie durch baßspezifische Regeln. Im Vergleich zur Melodie sind beim Baß größere Sprünge zugelassen, bei denen jedoch dafür gesorgt werden muß, daß sie sinnvoll verwendet werden. Außerdem ist beim Baß der Zusammenklang (vertikale Regel) und die Stimmführung zu überprüfen.

3.7 Stimmführung

Die eben beschriebenen Stimmgenerierungsregeln behandeln Gesetzmäßigkeiten innerhalb der Stimme, d.h. in horizontaler Richtung. Die Einschränkungen des Tonvorrats durch die Harmonik betrachten Zusammenklänge, d.h. in vertikaler Richtung. Dies wird durch die Harmonietonregel (RN22) in die Stimmgenerierungsregeln integriert. Nun sind noch Gesetzmäßigkeiten zu untersuchen, die beide Richtungen kombinieren. Diese werden durch die *Stimmführungsregeln* beschrieben. Ab der Komposition der zweiten Stimme wird durch eine besondere Stimmgenerierungsregel (RN45) die Prüfung dieser Stimmführungsregeln angestoßen.

Stimmführungsregeln sind in der modernen wie auch der zeitgenössischen Literatur sorgfältig dokumentiert. Die im folgenden aufgestellten Begriffe und Gesetzmäßigkeiten folgen der Darstellung von Koch [Koc82] (WQ1) im 1. Band¹⁵ §§83-166. Das Wissen ist dort in einzelne Definitionen und Regeln aufgeteilt, die in verbaler Form gegeben sind. Die Regeln werden durch Anmerkungen und Notenbeispiele erläutert.

3.7.1 Untersuchung von Stimmführungsquadrupeln

Stimmführungsregeln betrachten je zwei aufeinanderfolgende Töne in zwei Stimmen. Um die Zugriffswege zu vereinheitlichen, wird diese Kombination von Noten im sogenannten *Stimmführungsquadrupel* abgelegt. Hierbei

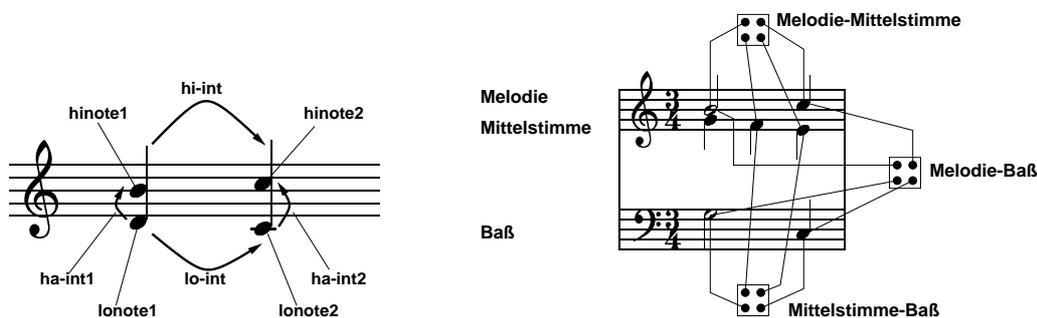
¹⁵Orig. Abtheilung

<i>Nummer</i>	<i>Regelbezeichnung</i>
RN5	Startintervalle
Hüll:	Hüllbed: Skalenbereich
RN6	Terzen einstreuen
RN10	Sequenzverbindung
Hüll:	Hüllbed: Dreiklangsbrechungsbereich
RN16	Dreiklangsschritt
RN19	Default-Dauernrestriktion
RN36	Durchgang Vollendung
RN22	Harmonietöne
Hüll:	Hüllbed: Skalenbereich
RN35	Durchgang Eröffnung
RN8	Repetitionslimit
RN21	Keine Schwach-Stark-Repetition
RN38	Vierteltakt-Beginn
RN39	Vierteltakt-Fortsetzung
RN11	Ambitusprüfung
Hüll:	Hüllbed: Skalenbereich
RN14	Leitton-Dominantsept stark
Hüll:	Hüllbed: Dreiklangsbereich
RN15	Leitton-Dominantsept akkordisch
Hüll:	Hüllbed: Skalenbereich
RN12	Leitton stark
Hüll:	Hüllbed: Dreiklangsbereich
RN13	Leitton akkordisch
RN44	Dauernvergabe
RN20	Keine Achtelrepetition auf Schlag 1
Hüll:	Hüllbed: Dreiklangsbereich, nicht Endformel
RN17	Terz in der Akkordbrechung
Hüll:	Hüllbed: Skalenbereich
RN31	Sekunden in gleicher Richtung bevorzugen
Hüll:	Hüllbed: Dreiklangsbereich
RN18	Verbiете seltsame Dominantseptakkordbrechungen
RN46	Korrespondenzprüfung

Abbildung 3.6: Regelfolge für die Folgeintervalle der Melodie

<i>Nummer</i>	<i>Regelbezeichnung</i>
RN5	Startintervalle
RN19	Default-Dauernrestriktion
RN11	Ambitusprüfung
RN25	Konsonanz mit Melodie
Hüll:	Hüllbed: Nicht Endformelbereich
RN24	Stufen 1, 3 oder Umkehrung
RN36	Durchgang Vollendung
RN22	Harmonietöne
Hüll:	Hüllbed: Nicht Endformelbereich
RN27	Unvollkommene Konsonanz auf Schlag 1
RN13	Leitton akkordisch
RN37	Melodieparalleler Durchgang
RN23	Keine Terzverdopplung
RN7	Kein Tritonus abwärts
RN30	Bevorzuge kleine Intervalle
RN9	Intervallsummenlimit
RN32	Bevorzugung von Gegensprüngen
RN33	Bevorzugung von Gegenbewegung
RN29	Verbot von Stimmkreuzungen
RN44	Dauernvergabe
RN26	Keine gleichzeitigen langen Töne
RN45	Stimmführungsprüfung
RN42	Endformel-Viertel-Hilfe
RN43	Kadenz-Baß-Schlußtakt-Hilfe
RN46	Korrespondenzprüfung

Abbildung 3.7: Regelfolge für die Folgeintervalle des Basses



Komponenten eines
Stimmführungsquadrupels

Alle Stimmführungsquadrupel der
Reichweitestufe :direct an Schlag 3

Abbildung 3.8: Grundbegriffe des Stimmführungsquadrupels

betrachtet man zwei aufeinanderfolgende Zweiklänge mit zusammen vier Tönen. Einen Überblick findet man in Abbildung 3.8.

Über das Stimmführungsquadrupel hat man nun Zugriff auf die melodischen Intervalle (innerhalb der Stimmen), die Zweiklänge (zusammenklingenden Intervalle) und das Verhältnis der melodischen Bewegungen. Zur Verwendung in den Regeldefinitionen werden Oberbegriffe für die zusammenklingenden Intervalle und die Bewegungsarten eingeführt, so daß grobe Klassen von Situationen entstehen, die gleich behandelt werden. Die Anzahl aller Kombinationen von vier Tonhöhen wäre sonst sehr groß.

Zusammenklingende Intervalle werden bei Koch folgendermaßen eingestuft:

vollkommene Konsonanz reine Oktaven und reine Quinten

unvollkommene Konsonanz große und kleine Terzen und große und kleine Sexten.

Quarte Die Quarte ist zum Baß dissonant, ansonsten konsonant und wird deswegen als eigene Klasse geführt.

Dissonanz alle übrigen Intervalle.

Dazu gehören die großen und kleinen Sekunden, die großen und kleinen Septimen und alle übermäßigen und verminderten Intervalle. Zu beachten ist, daß auch Intervalle, die enharmonisch verwechselt betrachtet konsonant wären, wie z.B. die übermäßige Sekunde, als dissonant zählen. Hier sind tonartliche Effekte wichtig.

Ebenso führt Koch Begriffe für das Verhältnis der Bewegung zweier Stimmen ein:

Gegenbewegung Die beiden Stimmen bewegen sich in entgegengesetzte Richtungen.

Seitenbewegung Eine Stimme bleibt liegen bzw. repetiert. Die andere bewegt sich.

Gerade Bewegung Die beiden Stimmen gehen in die gleiche Richtung, jedoch mit unterschiedlichem Intervall, so daß sich das zusammenklingende Intervall ändert.

Parallelbewegung Die beiden Stimmen haben das gleiche melodische Intervall in die gleiche Richtung, so daß das zusammenklingende (tonale) Intervall gleich bleibt.

Für die Stimmführungsquadrupel ist eine Datenstruktur definiert, die den Zugriff auf die Komponenten ermöglicht. Außerdem sind Funktionen definiert, die das Stimmführungsquadrupel auf die verschiedenen Zusammenklänge und Bewegungen untersuchen.

3.7.2 Definitionssprache für Stimmführungsregeln

Die Stimmführungsregeln beschreiben eine Situation, und geben an, ob diese Situation verboten oder erlaubt ist. Zur Beschreibung der Situation werden Intervalle und Bewegungen verwendet. Die Voraussetzung der Regel ist also die Situationsbeschreibung, die Folgerung ist nur ein boolescher Wert.

Stimmführungsregeln tragen den Regeltypbezeichner `:v1`. Die Syntax zur Definition von Stimmführungsregeln sieht folgendermaßen aus:

```
(def-rule <regelname> :v1 [ :verboten | :erlaubt ]
  [ :reach <reachspec> ]
  [ :casedescr <casespec> * ])
```

Die Elemente der Regelbeschreibung haben folgende Bedeutung:

:verboten — **:erlaubt** Dies ist die Folgerung der Regeln. Stimmführungsregeln beschreiben Kombinationen von Intervallen und Bewegungen, die in bestimmten Situationen erlaubt oder verboten sind.

<reachspec> Die Stimmführungsregeln werden von Koch so formuliert, als handle es sich um unmittelbar aneinanderstoßende Töne. Manche Regeln gelten aber auch, wenn eine unbedeutende Nebennote oder eine kurze Pause dazwischen ist. Hier kann die Reichweite der Stimmführungsregel angegeben werden (vgl. Kap. 3.7.3).

<casespec> Jede Zeile der Fallbeschreibung muß erfüllt sein, damit die Regel anwendbar ist, d.h. die Zeilen werden mit *und* verbunden.

Folgende Zeilentypen sind definiert:

:intseq Folge von zusammenklingenden Intervallen. Ein Intervall kann man exakt, als Klasse (unabhängig von der Oktave) oder durch seine Konsonanzklasse angeben.

:bew Art der Bewegung wie oben beschrieben.

:expr Beliebiger Lisp-Ausdruck, der einen booleschen Wert zurückgibt.

Auch die Stimmführungsregeln werden zu einer Regelfolge zusammengefaßt. Es wird die erste Regel, deren Voraussetzung erfüllt ist, angewandt. So kann man eine Ausnahme zu einer Regel definieren, indem man die Ausnahme in der Regelfolge vor die Hauptregel stellt. Ist keine Regel anwendbar, ist die Stimmführung zulässig.

3.7.3 Reichweite von Stimmführungsregeln

Die Stimmführungsregeln betreffen nicht nur direkt aufeinander folgende Noten, sondern auch Noten, die durch Nebennoten oder Pausen getrennt sind. Dies wird im System durch die *Reichweite* einer Stimmführungsregel modelliert. Das entscheidende Kriterium ist die Hörbarkeit eines Satzfehlers (Verstosses gegen eine Stimmführungsregel). Für die wichtigsten Regeln erläutert Koch in [Koc82] die Reichweite durch positive und negative Notenbeispiele. Abbildung 3.9 erläutert Kochs Beispiele für die Oktavparallele.

Ansonsten fordert Koch den Leser in der Anmerkung zu §93 auf, durch Analyse von guten Kompositionen die nötige Kenntnis zu erwerben. Koch nennt auch Faktoren, die Einfluß auf die Hörbarkeit eines Satzfehlers haben. Zu diesen Faktoren gehört u.a.:

- Die metrische Betonung der beiden Zusammenklänge
- Die Dauern der Zusammenklänge
- Ob beide Töne der Zusammenklänge neu angeschlagen werden
- Ob die Harmonie zwischen den Zusammenklängen wechselt
- Die Bewegungsrichtungen (auf- oder absteigend) der Stimmen
- Ob die untersuchten Stimmen Außen- oder Mittelstimmen sind

Betonte und unbetonte Oktaven mit direkter Reichweite (§92 Fig. 1,2)

Besonderheiten			Mittelst.	unbetont	unbetont
		direkt	direkt	direkt	direkt
	Reichweite	verboten	verboten	verboten	verboten
Einstufung					

Oktaven mit größerer Reichweite (§93 Fig. 1-4)

Besonderheiten		eine Note		Harmoniewechsel
		unbetont		langames Tempo
	Reichweite	Schlagakzent	S.-Akzent	Taktakzent
Einstufung	verboten	verboten	verboten	erlaubt

Unbetonte, nicht direkte Oktaven (§93 Fig. 10,11)

Besonderheiten		unbetont	unbetont
		Nicht direkt	Nicht direkt
	Reichweite	erlaubt	erlaubt
Einstufung			

Notensatz Original [Koc82]

Abbildung 3.9: Beispiele für die Reichweite der Oktavparallele aus [Koc82]

Der Hintergrund dieser Festlegungen in [Koc82] ist, daß die traditionellen Satzregeln auf klassische Kompositionen mit Begleittexturen und durch Nebennoten verzierten Linien angewandt werden sollen. Bei den Menuetten, die das hier besprochene System generiert, kommen viele dieser Elemente nicht vor. Deswegen genügt für die Menuettkomposition ein einfacheres Modell der Reichweite, das die Noten auf benachbarten betonten Taktteilen berücksichtigt. Das Modell vernachlässigt beispielsweise den Fall von § 93 Fig. 2 (Abb. 3.9), da das Kompositionsverfahren keine kurzen harmoniefremden Nebennoten auf betonter Zeit erzeugt; es behandelt aber § 93 Fig. 10,11 korrekt.

In diesem Modell sind folgende Reichweitestufen definiert.

- :**direct** Die Regel gilt nur für direkt benachbarte Zusammenklänge. Dafür genügt, daß beim zweiten Zusammenklang eine Note neu anschlägt.
- :**acc-semibeat** Es werden die vertikalen Intervalle zweier benachbarter Halbschläge berücksichtigt, wenn beim ersten Zusammenklang beide Noten neu anschlagen.
- :**acc-beat** Es werden die vertikalen Intervalle zweier benachbarter Schläge berücksichtigt, wenn beim ersten Zusammenklang beide Noten neu anschlagen.
- :**acc-bar** Es werden die vertikalen Intervalle zweier benachbarter Taktanfänge berücksichtigt, wenn beim ersten Zusammenklang beide Noten neu anschlagen.

Jede Reichweitestufe schließt alle in der Beschreibungen vorangehenden Stufen mit ein, so daß eine Totalordnung¹⁶ entsteht. Die Totalordnung erleichtert es, bei der Prüfung der Stimmführungsregeln die Reichweite zu berücksichtigen. Ein Reichweitenmodell, das weitere der Kochschen Kriterien für die Hörbarkeit berücksichtigen würde, ist schwieriger zu handhaben, insbesondere wenn die Reichweitenstufen nur noch eine Halbordnung bilden, weil Paare von Reichweitestufen existieren, bei denen keine die andere einschließt.

Die Untersuchung der Stimmführung dient dazu, die Korrektheit einer neuen Note der gerade generierten Stimme festzustellen. Bei der Erstellung der Stimmführungsquadrupel wird vom zweiten Zweiklang ausgehend für jede Reichweitekategorie untersucht, welche Töne für den ersten Zusammenklang des Stimmführungsquadrupels in Frage kommen. Die bei der Erzeu-

¹⁶Bei einer Totalordnung ist von zwei Elementen a und b entweder a b untergeordnet, oder umgekehrt.

gung des Quadrupels verwendete Reichweite wird in der Datenstruktur des Stimmführungsquadrupels abgelegt.

Bei der Stimmführungsprüfung berücksichtigt die Stimmführungsregel alle die Stimmführungsquadrupel, bei denen die Reichweite des Stimmführungsquadrupels in der Reichweite der Regel eingeschlossen ist. Ist die Reichweite der Regel beispielsweise `:acc-beat` und die des Stimmführungsquadrupels `:direct`, dann berücksichtigt die Regel das Quadrupel, da die Reichweitenstufe `:acc-beat` die Reichweitenstufe `:direct` einschließt.

3.7.4 Stimmführungsregeln

Es werden hier die Stimmführungsregeln aus [Koc82] aufgeführt, die im Kompositionssystem eingesetzt wurden. Die Regeln werden in verbaler Form unter Verwendung der Begriffe der Regelsprache aus Kapitel 3.7.2 beschrieben. Die Codierung der Regel wird an einem Beispiel erläutert.

Regel 47 Direkte Repetition

Erlaubt ist:

- Bewegung parallel
- Beide Stimmen wiederholen ihren Ton, d.h. beide linearen Intervalle sind eine Prime.
- Reichweite: `:direct`

Diese Regel ist unter Experten bekannt, wird in der Literatur aber selten explizit genannt (WQ3). Ohne diese Regel würden Stücke aber sehr seltsam aussehen, da der Großteil der üblichen Akkorde vollkommene Konsonanzen (Oktaven und Quinten) enthält und damit wegen des Verbots von Oktav- und Quintparallelen nicht wiederholt werden dürfte.

Regel 48 Oktavparallelen

Verboten ist:

- Bewegung parallel
- Beide Zusammenklänge sind oktavbereinigt Oktaven
- Reichweite: `:acc-beat`

Regel 49 Quintparallelen

Verboten ist:

- Bewegung parallel
- Beide Zusammenklänge sind oktavbereinigt Quinten
- Reichweite: :acc-beat

Regel 50 Oktavgegenparallelen

Verboten ist:

- Bewegung gegen
- Beide Zusammenklänge sind oktavbereinigt Quinten
- Reichweite: :acc-beat

Die Gegenparallelen¹⁷ heißen so, weil sie trotz Gegenbewegung vom Höreindruck her wie Parallelen empfunden werden. Die beiden Regeln sind Vergrößerungen von §95 in [Koc82]. Dort werden Situationen beschrieben, bei denen Gegenparallelen im Gegensatz zu gewöhnlichen Parallelen erlaubt sind. Beispiele dafür sind der Satz mit mehr als vier Stimmen und kurze Noten in Begleitfiguren, die nicht zusammen anschlagen. Für die Komposition von zweistimmigen Menuetten sind die vergrößerten Regeln aber ausreichend.

Regel 51 Quintengegenparallelen

Verboten ist:

- Bewegung gegen
- Beide Zusammenklänge sind oktavbereinigt Quinten
- Reichweite: :acc-beat

In der Seitenbewegung ist eine Folge zweier gleicher vollkommenen Konsonanzen unproblematisch. Diese Regel kommt dem Musiker selbstverständlich vor und wird deswegen selten genannt. Interessanterweise ist sie bei Koch aufgeführt.

¹⁷Das Wort ist eigentlich ein Oxymoron.

Regel 52 Verdeckte Oktavparallelen

Verboten ist:

- Außenstimmen
- Obere Stimme springt um eine Terz oder mehr nach oben.
- Bewegung gerade
- Zweites Intervall ist oktavbereinigt eine Oktave
- Reichweite: `:direct`

Die Regel ist unter Berücksichtigung der Beispiele von Mattheson [Mat39] und der Regeln Scheibes¹⁸ etwas weiter gefaßt als [Koc82]. Bei Koch fehlt die erste Voraussetzung mit den Außenstimmen und er faßt die zweite Voraussetzung enger, so daß Kochs Variante der Regel strenger ist.

Die Regeldefinition für die verdeckten Oktaven wird folgendermaßen kodiert:

```
(def-rule verd-oktaven :vl :verboten
  :casedescr
  (:reach :direct)
  (:bew :gerade)
  (:expr (and (eql (vname1 vl) :melody) ;; Aussenstimmen
              (eql (vname2 vl) :bass)))
  (:expr (>= (szahl (hi-int vl)) 2)      ;; Oberstimme Terz aufwaerts
  (:intseq (:kon :unvollkom)           ;; Zweiklangfolge
           (:cls $(cintv :rn oktave :ab))))
```

Regel 53 Verdeckte Quintparallelen

Verboten ist:

- Außenstimmen
- Obere Stimme springt um Terz oder mehr nach oben.
- Bewegung gerade
- Zweites Intervall ist oktavbereinigt eine Quinte.
- Reichweite: `:direct`

¹⁸Das *Compendium Musices* von Johann Adolph Scheibe (entstanden zwischen 1728 und 1736) ist in [Ben60] abgedruckt

3.8 Muster und Patternmatching

3.8.1 Grundlagen von Patternmatcher und Patternsprache

Die Grundidee bei der Verwendung von Mustern ist es, eine Menge zulässiger (passender) musikalischer Elemente durch Ähnlichkeit mit einem Muster zu beschreiben. Das Muster ist eine abstrakte intensionale Beschreibung der Menge. Der Vorteil von Mustern gegenüber anderen Formen der abstrakten Beschreibung ist, daß sie von Menschen, die gewohnt sind, die Daten direkt zu betrachten, gut verstanden werden, da das Muster in Aufbau und Aussehen den Daten ähnlich ist.

Die Grundbestandteile eines Musters sind musikalische Musterelemente, die analog zu den musikalischen Elementen aus COMF aufgebaut sind, im Gegensatz zu letzteren aber Wildcards oder Variablen¹⁹ als Slotwert enthalten dürfen.

Der Test, ob ein Datenobjekt zu der vom Muster beschriebenen Menge gehört, heißt *Matching*. Die Struktur von Datenobjekt und Muster wird von einem *Patternmatcher* parallel durchwandert und dabei daraufhin untersucht, ob ein Element vom anderen abweicht. Der in dieser Arbeit verwendete Patternmatcher ist eine leicht angepaßte Variante des Patternmatchers für Musikdaten aus [Bec00]. Die Grundelemente des Musters sind entweder COMF-Objekte oder analog zu COMF-Klassen²⁰ definiert. Im Muster kann ein ganzes Objekt oder ein einzelner Slot in einem Objekt durch eine Variable oder eine Wildcard ersetzt werden. Zu den Grundelementen treten weitere Elemente zur abstrakten Beschreibung musikalischer Sachverhalte, z.B. das Anschlagselement (*touchel*), das die erste Note und die Gesamtdauer vorgibt, die restlichen Noten aber offenläßt oder das Betonungselement, mit dem man die Betonungsstärke des aktuellen metrischen Zeitpunkts prüfen lassen kann. Weitere Elemente, z.B. logische Verknüpfungen, dienen zum Aufbau des Musters.

3.8.2 Muster für Endformeln

Wenn bei der Stimmengenerierung ein Satz die geforderte Länge erhalten hat, dann überprüft der Fail-Test, ob das Endstück der generierten Stimme

¹⁹Sowohl Wildcards als auch Variablen stehen für ein beliebiges Objekt. Bei Variablen wird zusätzlich noch der Variablenname an den Wert gebunden. Bei erneutem Auftreten der Variable im Muster muß die alte Bindung mit dem neuen Wert matchen.

²⁰Die Definition solcher Analogklassen ist nötig, da die Konsistenzprüfungen von COMF Wildcards und Variablen nicht zulassen

mit dem entsprechenden Stimmuster aus einer der zulässigen Endformeln matcht. Das Muster kann Patternelemente für Noten, Stufen und Intervalle enthalten. Für das Matching von Stufen ist für das gesamte Endstück der generierten Stimme die Zieltonart der Endformel zu verwenden.

Die Schlußwirkung der Endformeln entsteht durch das Zusammenwirken melodischer, harmonischer und metrischer Aspekte. Eine Endformel wird durch ein Muster für die Melodielinie, ein Muster für die Baßlinie und die harmonische Basis beschrieben. Ein Beispiel für eine Endformel und die Repräsentation in COMF ist in Abb. 3.10 abgedruckt.

Bei der Endformel wird der Typ, nämlich Kadenz, Quintabsatz oder Grundabsatz mit abgespeichert. Grundkadenz und Quintkadenz verwenden die gleichen Formeln (relativ zur aktuellen Tonart des Schlusses). Außerdem können zur einer Endformel weitere Eigenschaften (z.B. daß die Endformel für akkordbrechungsbasierte Melodien geeignet ist) angegeben werden.

In der Makrostruktur ist durch die Tonordnung für jeden Satz der Typ der Endformel und deren Tonart festgelegt. So kann man aus der Menge aller überhaupt definierten Endformeln die Menge der für den Satz zulässigen Endformeln auswählen.

Für die erste Stimme sind alle Endformeln zulässig, die von Typ und Eigenschaften her passen. Die bei der ersten Stimme verwendete Endformel wird dann in der Chart abgelegt, und für die nächste Stimme ist nur noch das Muster aus dieser Endformel zulässig.

Eine Sammlung von Endformeln findet sich bei [Bud83]. Für diese Arbeit wurden davon Endformeln, die für die metrischen Verhältnisse des Menuetts geeignet sind, ausgewählt und durch solche, die bei der Analyse von Menuetten gefunden wurden, ergänzt.

Von den Möglichkeiten des Patternmatchers aus [Bec00] wird nur ein Teil ausgenutzt. Da bei der Stimmengenerierung immer eine Stimme neu generiert wird, wird immer nur eine Stimme mit einer Stimme aus dem Muster verglichen. Die in [Bec00] ebenfalls definierten Verbindungselemente für mehrstimmige Muster werden für die Stimmengenerierung nicht benötigt. Ebenso ist es beim Endmustervergleich nicht nötig, in den Daten nach einem Musteranfang zu suchen, da die metrische Position der Endformel a priori bekannt ist. Es genügt die Verwendung der `match`-Funktion, die feststellt, ob ein Muster auf gegebene Daten paßt.



```
(def-endformula :f4 (:kadenz)
  (:harm ($ (harmel 1/4 4 6) $(inversel 5 1/4 1) $(harmel 1/4 5)
    $(harmel 3/4 1)))
  (:melody ($ (degnote vi 4) $(degnote vi 3) $(degnote vi 2)
    $(degnote pha 1)))
  (:bass (:overlapping
    $(degnote vi 4) $(betwel-t)
    $(degnote vi 5)
    $(betweenel '(:or $(cintv :RN prime :auf)
      $(cintv :RN oktave :ab)))
    $(degnote vi 5) $(betwel-t)
    $(degnote 1/4 1 0)
    $(betweenel $(cintv :RN QUARTE :AB))
    $(degnote 1/4 5 0)
    $(betweenel $(cintv :RN QUINTE :AB))
    $(degnote 1/4 1 0))))
```

- Die harmonische Basis ist im Notenbeispiel durch die Bezifferung des Basses angegeben.
- Die Between-Elemente in der Baßstimme legen Intervalle und damit die Bewegungsrichtung fest. Die Stufen geben keine Oktavlage an, so daß die Baßstimme ohne die Intervallangaben ziellos springen würde.
- In der Melodie ist dies nicht nötig, da die Intervalle durch die Stimmengenerierungsregeln ohnehin beschränkt sind.

Abbildung 3.10: Ein Endformelmuster für eine Kadenz

Kapitel 4

Implementierung

Die im vorangegangenen Kapitel beschriebenen Verfahren und Vorgehensweisen werden nun in ein Computerprogramm umgesetzt.

Dieses Kapitel beschreibt die allgemeinen Prinzipien der Implementierung und gibt einen Überblick über den Aufbau des Systems auf programmtechnischer Betrachtungsebene. Für die einzelnen Komponenten werden jeweils die grundlegende Arbeitsweise, die Klassenhierarchie und die wichtigsten Schnittstellen beschrieben.

4.1 Prinzipien der Implementierung

Bei der Programmierung ist den in Kap 3.1 gestellten Forderungen an das Verfahren Rechnung zu tragen.

- Um die Wirkung des eingesetzten Wissen überprüfen zu können, ist es nötig, daß es explizit repräsentiert wird.
- Der Programmaufbau muß flexibel genug sein, um explorative Programmierung, zuzulassen.
- Die Datenstrukturen und Wissensdarstellungen müssen gut untersuchbar und lesbar sein, damit Fehler leicht gefunden werden können.

Die eingesetzten Techniken und Programmierwerkzeuge wurden so ausgewählt, daß sie diesen Anforderungen genügen.

4.1.1 Wahl der Programmiersprache und der Sprachmittel

Bei dieser Aufgabe ist es im Gegensatz zu manchen anderen Aufgabenstellungen der Informatik nicht möglich, das Verfahren vor Beginn der Imple-

mentierung im Detail zu planen. Vielmehr ist die konzeptuelle Arbeit und die technische Realisierung eng miteinander verbunden. Die Voraussetzungen für diese Vorgehensweise werden mit Hilfe von Programmiermethoden der symbolischen künstlichen Intelligenz, wie sie in [Nor92] beschrieben werden, geschaffen.

Die Programmiersprache Common Lisp hat sich für verschiedenste Aufgaben der künstlichen Intelligenz bewährt, da sie das Denken und Arbeiten in unterschiedlichen Programmierparadigmen (*paradigmaticity*) zuläßt und damit erlaubt, die Vorgehensweise in der Programmiertechnik dem Problem anzupassen. Neben dem zentralen Paradigma der funktionalen Programmierung kann man auch mit Techniken der prozeduralen und der objektorientierten Programmierung arbeiten. Für letzteres wird in Common Lisp die objektorientierte Erweiterung CLOS benutzt.

Bei dieser Aufgabe wird an mehreren Stellen das Modell für ein Problem deklarativ, in einer speziellen Repräsentationssprache notiert. Dabei wurde die Problemsprache und das in dieser Sprache erstellte Modell des Problems gemeinsam, beinahe parallel, entwickelt. Die Sprache wird erweitert oder abgeändert, wenn das Modell es verlangt, weil sich während der Arbeit am Modell zusätzliche Anforderungen herausstellen. Diese Vorgehensweise mit enger Verbindung zwischen Entwurf, Implementierung und experimentellem Arbeiten nennt man *exploratives Programmieren*.

So wichtig wie die Programmierparadigmen sind dabei einige nützliche Spracheigenschaften, mit denen Common Lisp die Definition problemspezifischer Modellierungssprachen erleichtert. Makros erlauben die Definition einer Eingabesyntax. Die Read-Write-Consistency schafft automatisch Möglichkeiten zur Ein- und Ausgabe verschiedenster Daten auf Bildschirm und Datei. Lisp-Programmcode wird durch Listen - also innerhalb des Typsystems von Lisp - dargestellt, so daß compilierende Techniken, die Definitionen aus der Problemsprache in Lisp-Code umsetzen, einfach zu realisieren sind. Das Laufzeitprogrammiersystem von Common Lisp ermöglicht es dann, die so entstandenen neuen Lisp-Funktionen zur Laufzeit bis hinunter zum Maschinencode zu compilieren.

Für das Menuettkompositionssystem werden Modelle von musikalischen Elementen und Vorgängen erstellt. Objektorientierte Systeme sind für Aufgaben dieser Art besonders geeignet - schließlich waren Simulationsaufgaben der Anstoß für die Entwicklung der objektorientierten Programmierung. Ein weiterer Vorteil des objektorientierten Paradigmas ist, daß es sich sowohl in die Grundstrukturen der prozeduralen Programmierung, nämlich dem Ändern von Variablen und Datenstrukturen, als auch in die funktionale Programmierung einfach integrieren läßt.

Das objektorientierte Programmierparadigma bietet verschiedene Vortei-

le bei der Strukturierung des Systems. Die Verwendung von Objekten sorgt dafür, daß Änderungen an Datenstrukturen sich nur lokal auswirken. Verhalten kann automatisiert werden, z.B. kann der Initialisierungsmechanismus für ein Objekt mit der Klassendefinition angegeben werden.

Bei der objektorientierten Programmierung mit CLOS hat man die Wahl zwischen verschiedenen Arten, das Programm in einzelne Komponenten aufzuteilen. Man kann es nach der Aufrufhierarchie der Funktionen oder nach der Klassenhierarchie strukturieren. Man kann sowohl die Klassen (durch Subklassenbildung) als auch die generischen Funktionen (durch Hinzufügen weiterer Methoden) in Einzelschritten aufbauen. Dabei erlaubt CLOS Mehrfachvererbung von Klassen und Multimethoden, d.h. generische Funktionen, die nach mehreren Argumenten spezialisiert sind.

Das Ziel all dieser Programmier Techniken und Methoden ist es, das Programm in inhaltlich zusammengehörige Einheiten zu gliedern. Solch eine inhaltliche Trennung, die sogenannten *separation of concerns* ist nicht nur aus akademischer Sicht wegen ihrer "stilistischen Eleganz" erstrebenswert. Vielmehr verbessert diese Trennung nach inhaltlichen Gesichtspunkten die Verständlichkeit des Codes und die Übersicht über das System und hilft dabei, daß inhaltlich motivierte Änderungen - wie sie durch die o.g. Vorgehensweise der explorativen Programmierung immer wieder nötig werden - lokal bleiben, d.h. möglichst wenige Programmteile betreffen.

Ein Beispiel ist die Suche im Zustandsraum, die Problembeschreibung und problemunabhängige Suchstrategie trennt. Zum problemunabhängigen Teil gehört die Basisklasse für den Zustand bei der Suche (vgl. Kapitel 3.4), die generischen Funktionen (ohne Methoden) für die Operatoren und die Suchalgorithmen, die die generischen Funktionen rufen. Um die Beschreibung eines konkreten Suchproblems zu erstellen, wird eine Subklasse für den Zustand definiert, die Slots für die Daten des Zustands enthält. Für diese Klasse werden zu den im problemunabhängigen Teil definierten generischen Funktionen dann Methoden implementiert.

4.1.2 Programmier technische Festlegungen

Damit das Programm einheitlich und überschaubar bleibt, ist es nötig, zentrale Entscheidungen bewußt zu treffen und zu dokumentieren. Die wichtigsten Entscheidungen sind hier, von der generellsten zur technischsten geordnet, aufgeführt:

1. Es wird Common Lisp nach dem Standard von *Common-Lisp, the Language*, 2. Auflage [Ste90]¹ mit seiner objektorientierten Erweiterung CLOS verwendet.

¹Es wurde unter Franz Allegro 5.0 auf Sun/Solaris entwickelt, das bereits den ANSI-

2. CLOS bietet verschiedene Möglichkeiten an, sein Klassensystem aufzubauen.

Das Metaobjektprotokoll erlaubt es, auf die Klassen- und Methodendefinitionen zuzugreifen und die Funktionalität von Klassen und Methodendefinition in CLOS zu erweitern. Beispielsweise gibt es eine Klasse `standard-class`, in der festgelegt wird, wie eine Klassendefinition bearbeitet wird. Diese Klasse `standard-class` bezeichnet man als *Metaklasse* der definierten Klasse.

Auf die Verwendung des Metaobjektprotokolls² (MOP) [KdRB91] wurde mit Ausnahme des Makros `cp-make` verzichtet. Da die Arbeitersparnis eher gering ist und das MOP noch nicht völlig standardisiert ist, wurde der Kompatibilität und dem einfacheren Programmaufbau der Vorzug gegeben.

Einige der im folgenden beschriebenen Festlegungen hätten automatisiert werden können, indem man als Metaklasse eine eigene Subklasse von `standard-class` verwendet, die eine Kopiermethode und die Namensvergabe für die Slotzugriffe automatisiert. Andere immer wiederkehrende Aufgaben bei der Klassendefinition, wie z.B. die Definition der externen Darstellungen können allerdings nicht mehr auf einfache Weise automatisiert werden.

3. Das System arbeitet sequentiell. Nebenläufige Prozesse werden nicht benötigt. Das System muß nicht in Echtzeit arbeiten.
4. Auf eine grafische Benutzungsschnittstelle wird verzichtet, da sie in diesem Fall keinen Vorteil an Bedienkomfort oder Übersichtlichkeit bietet. In der Lisp-Interaktionsshell (Fachbegriff *read-eval-print-Loop*) kann man Funktionen aufrufen. Wenn man deren Aufrufe einfach gestaltet, ist das System bequem zu bedienen.

Durch die Verwendung der Lisp-Programmierungsumgebung als Bedienoberfläche wird Entwicklungsaufwand eingespart und die Kompatibilität mit allen gängigen Plattformen bleibt erhalten. Die Programmierungsumgebung von Allegro Common Lisp bietet über den Debugger verschiedene Mittel zur Untersuchung des Ablaufes und der Datenstrukturen von Lisp an.

Standard (Nachfolger des Standards von *Common-Lisp, the Language 2. Auflage* und abwärtskompatibel dazu.) realisiert. Eine Portierung auf andere Lispsysteme sollte unproblematisch sein. Sie erfordert höchstens die Reimplementierung des Makros `cp-make`, falls das MOP des Zielsystems abweicht.

²Das Metaobjektprotokoll erlaubt Zugriff auf den Klassendefinitionsmechanismus von CLOS und Erweiterungen desselben.

5. Zum Betrachten der Ergebnisse als Notenblatt wird das System Lilypond benutzt.
6. Die Datenstrukturen werden nichtdestruktiv bearbeitet, d.h. ein Objekt wird für das Ergebnis bei Bedarf kopiert, damit man das Originalobjekt nicht ändern muß. Hier werden die Konventionen, die bei den Listenfunktionen von Common Lisp üblich sind, auch auf verzeigerte Objekte übertragen.

Diese Vorgehensweise erlaubt es, komplizierte Datenstrukturen nach dem funktionalen Programmierparadigma zu behandeln, d.h. sie als Wert an Funktionen zu übergeben und als Ergebnis zurückzuerhalten, ohne daß Nebenwirkungen berücksichtigt werden müssen. Außerdem erleichtert es die Verwendung von Backtracking-Algorithmen, da das Zurücknehmen von Entscheidungen einfach wird.

7. Um dieses Vorgehen zu erleichtern, gibt es ein Makro `cp-make`, das Objekte kopiert. Dieses benötigt bei seiner Expansion die Liste der Slotnamen einer Klasse und erhält diese über das Metaobjektprotokoll. Beim Portieren des Systems ist deswegen möglicherweise `cp-make` zu reimplementieren.
8. Für einen Teil der Klassen ist die Ausgabe einer Instanz im externen Format über die `print-object`-Methode der Klasse definiert.
9. Um Wiedereinlesbarkeit (*read-write-consistency*) zu erhalten, wird für solche Klassen ein Konstruktor (nach Bedarf Funktion oder Makro) definiert, so daß die Evaluation der externen Darstellung (Liste) das Objekt ergibt.

Als Name des Konstruktors und damit als Klassenbezeichnung in der externen Darstellung wird der Klassenname verwendet. Der Konstruktor selbst sollte möglichst wenig Verarbeitung und Initialisierung übernehmen, sondern dies den Initialisierungsmethoden der Klasse überlassen (vgl. Bemerkung 12), damit die Initialisierungsmechanismen auch vollständig gerufen werden, wenn die Instanz von einer Funktion des Programms mit `make-instance` erzeugt wird.

10. Es ist ein Readermacro-Zeichen `$` definiert, das den darauffolgenden Ausdruck evaluiert. Readermacros verbinden ein Zeichen mit einer Funktion. Findet der Lisp-Reader beim Einlesen eines Ausdrucks das Zeichen, so ruft er die damit verbundene Funktion.

Beispiel einer externen Darstellung:

```
$(track () $(note vi g 4)$(note vi h 4))
```

Ohne das innere \$ würde die Liste (note vi g 4) in das track-Objekt eingebaut. Mit dem \$ wird während des Einlesens der Konstruktor aufgerufen und somit das durch die Listennotation beschriebene CLOS-Objekt der Klasse note erzeugt, korrekt initialisiert und in das track-Objekt eingebaut.

Das Readermacro \$ ermöglicht es also, eine Hierarchie von Objekten auszugeben und wieder einzulesen. Damit kann man die Objekte mit externer Darstellung genauso freizügig verwenden, wie die Lisp-Datentypen, die eine vordefinierte externe Darstellung besitzen.

11. Slotnamen, Initargs und Accessor-Funktionen eines Slots haben den gleichen Namen.
12. Die Normalisierung und Konsistenzprüfung neuer Instanzen übernimmt eine :after-Methode zu initialize-instance. Es werden alle anwendbaren :after-Methoden einer generischen Funktion gerufen^{3 4}.

Damit werden nacheinander die Initialisierungsmechanismen der Superklassen von der generellsten bis zur speziellsten (der Klasse selbst) ausgeführt. So kann man die Normalisierungsmechanismen slotweise planen.

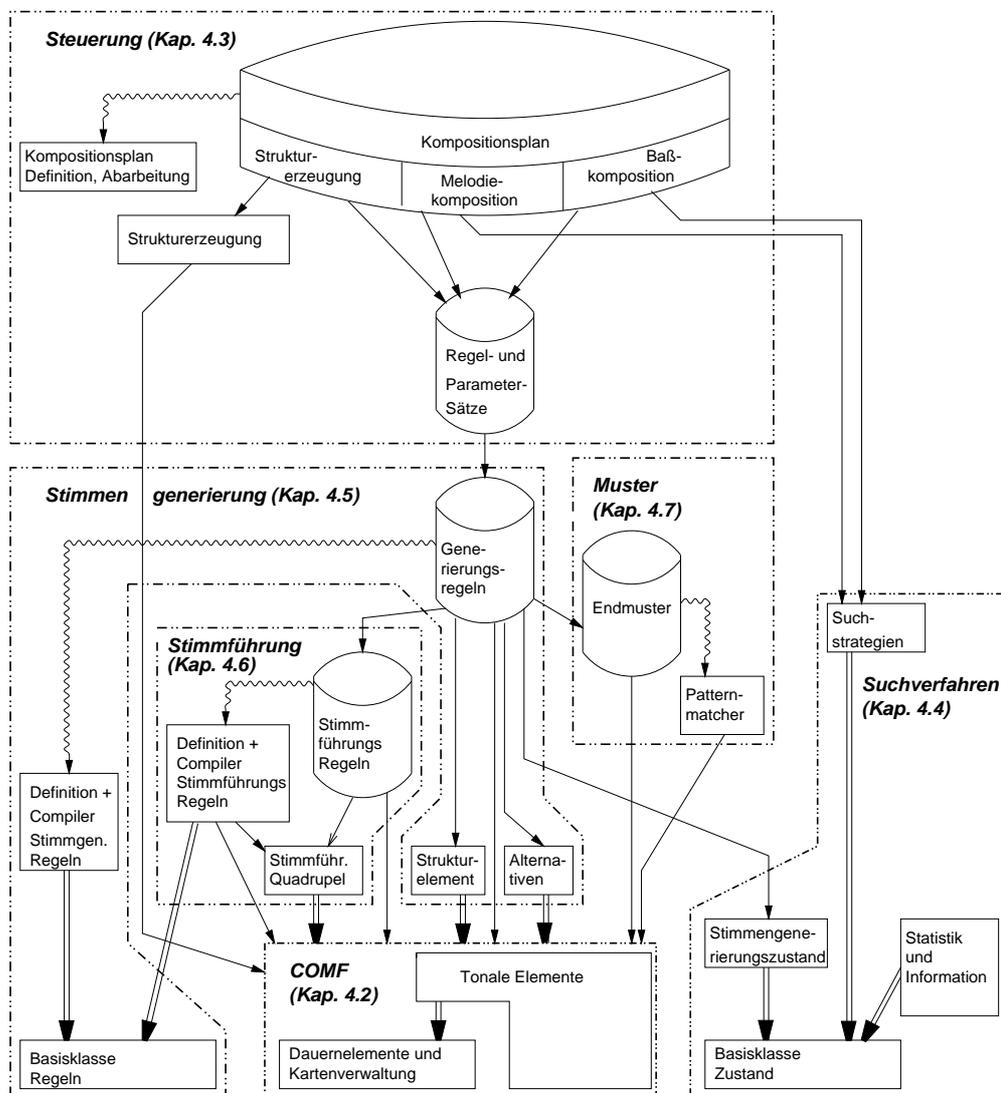
4.1.3 Überblick über den Systemaufbau auf technischer Ebene

Ein wichtiges Mittel zur Beherrschung komplexer Systeme ist es, das System in Module mit wenigen, klar definierten Außenbeziehungen zu gliedern. Nachdem die Beschreibung in Kapitel 3.1 das Verfahren nach inhaltlich-logischen Gesichtspunkten gegliedert hat, werden hier nun die Komponenten aus programmtechnischer Sicht beschrieben.

Durch die Verwendung der im vorangegangenen Abschnitt genannten Programmier Techniken gibt es verschiedene Arten, wie eine Komponente A

³ :before und :after-Methoden werden in der Standardmethodenkombination von CLOS [Kee89] vor bzw. nach der Hauptmethode gerufen. Im Gegensatz zu den Hauptmethoden ersetzt bei :before und :after-Methoden eine speziellere die allgemeinere nicht, sondern wird zusätzlich gerufen.

⁴In dieser Arbeit wird nur die *Standardmethodenkombination* von CLOS verwendet.



Legende:

- A → B Funktionaler Aufruf
Wissensverwendung
- A ==> B Erbt von
- A ~~~~~> B Hat Inferenzmaschine
interpretiert durch

- <Name> Funktionale Komponente
Funktionen/Klassen
- <Name> Wissenskomponente
Daten
- <Name>
<Abschnitt> Komponentengruppe

Abbildung 4.1: Systemkomponenten auf physischer Ebene

eine andere Komponente B benutzen kann. In der Beschreibung des physischen Aufbaus werden die folgenden Beziehungen unterschieden:

Funktionaler Aufruf Dies ist die übliche Art. Die Komponente A ruft Funktionen der Komponente B auf oder verwendet globale Variablen, die in der Komponente B definiert sind.

Erbt von Die Komponente A bildet Subklassen von Klassen, die in der Komponente B definiert sind und/oder A erweitert generische Funktionen der Komponente B um weitere Methoden.

Hat Inferenzmaschine Die Komponente A definiert Wissen in einer Definitionssprache. Die Komponente B implementiert die Definitionssprache und stellt einen Interpreter bzw. Compiler zur Verfügung.

Verwendung von Wissen Die Komponente A verwendet das in der Komponente B definierte Wissen. Es handelt sich um eine Art Aufruf - mit dem Unterschied, daß anstelle des Aufrufs einer Funktion Wissen von einer Inferenzmaschine abgearbeitet wird. Daher wird in Diagramm 4.1 das gleiche Symbol wie für den funktionalen Aufruf benutzt.

Abb. 4.1 gibt einen Überblick über die Komponenten des Systems. Die Komponenten werden dort zu inhaltlich zusammengehörigen Komponentengruppen zusammengefaßt. Die folgenden Abschnitte dieses Kapitels beschreiben jeweils eine dieser Komponentengruppen (die Abschnittsnummer ist in Abb. 4.1 angegeben).

4.1.4 Bedeutung der Effizienz

Für diese Aufgabe ist Berechnungseffizienz gegenüber den Zielen der Flexibilität und Änderbarkeit zweitrangig, da es keine strikten Vorgaben für die Laufzeit gibt. An besonders häufig ausgeführten Programmteilen wird dagegen gezielt auf gute Effizienz geachtet.

Die Effizienz von Lisp-Programmen betrachtet man üblicherweise auf drei Ebenen.

- Bei großen Datenmengen hat der asymptotische Aufwand den größten Einfluß auf die Laufzeit eines Verfahrens. Verbesserungen erhält man üblicherweise durch ein komplizierteres (intelligenteres) Verfahren oder durch Erweiterungen an der Datenstruktur, z.B. indem man kürzere Zugriffswege definiert oder Zwischenergebnisse zur Wiederverwendung

aufbewahrt (Redundanz). Bei der Menüettkomposition sind die Datenstrukturen allerdings nicht besonders groß. Mit Rücksicht auf die Flexibilität wurde zumeist auf Redundanz in den Datenstrukturen zugunsten eines einfacheren Aufbaus verzichtet. Bei Zugriffen auf Elemente wird die Datenstruktur linear durchwandert.

Eine Ausnahme ist der Zustand für die Stimmengenerierung. Hier werden häufig benutzte Angaben vorberechnet und in Slots des Suchzustands abgelegt. Da nichtdestruktiv vorgegangen wird, d.h. der Zustand wird nur zu Beginn erzeugt, dann aber nicht mehr geändert, ist es hier kein Problem, die Datenstruktur konsistent zu halten.

Bei Lisp-Funktionen sieht man den asymptotischen Aufwand nicht auf den ersten Blick, so daß zwei ähnlich kompliziert aussehende Lösungen einen sehr unterschiedlichen Aufwand haben können. Sofern dies keine Nachteile hinsichtlich der Flexibilität hat, wird natürlich auf die Wahl der effizientesten Funktion geachtet.

- Lisp definiert abstrakte Grundoperationen, wie das Erzeugen einer Listenzelle oder eines Objekts. Der abstrakte Aufwand befaßt sich mit der Zahl dieser Grundoperationen. Eine Verbesserung auf dieser Ebene ist beispielsweise das Ersetzen einer kopierenden Funktion durch eine destruktive Funktion, so daß das Erzeugen von Listenzellen gespart wird.

An häufig ausgeführten Stellen wird auf einen geringen Grundoperationsaufwand Rücksicht genommen. So ist in den Zugriffsfunktionen von COMF Code verdoppelt, anstatt andere COMF-Funktionen zu rufen. Dies spart das Anlegen von Objekten für Zwischenergebnisse ein. Ebenso werden in den COMF-Objekten die Werte intern durch Integerzahlen repräsentiert, um möglichst effiziente Grundoperationen der musikalischen Repräsentation zu erhalten. Destruktive Funktionen für die Listenverarbeitung werden an einigen Stellen verwendet, wenn die Listenzellen frisch erzeugt sind und daher garantiert an keiner anderen Stelle referenziert werden.

- In den *physischen Aufwand* gehen auch noch die Kosten der einzelnen Grundoperationen ein. In diesem Bereich wurde nur darauf geachtet, daß häufig ausgeführte und sorgfältig getestete Teile des Programms - insbesondere der von compilierenden Programmteilen erzeugte Code - mit maximaler Geschwindigkeit und niedrigster Sicherheit (Verzicht auf Typprüfungen) compiliert werden. Falls dabei Probleme entstehen, können diese Prüfungen ohne Änderungen am Quellcode wieder ein-

geschaltet werden. Auf weitere Verbesserungen, z.B. durch Deklarationen⁵ wurde verzichtet.

Einen starken Einfluß auf den Aufwand von Suchvorgängen hat die Struktur des Suchraums, die durch die Übergangsfunktion bestimmt wird. Es ist daher nötig, die Suchstrategie an den Suchraum anzupassen. Auch kann man bei der Erstellung der Zustandsübergangsfunktion auf die Effizienz Rücksicht nehmen, beispielsweise indem man Regeln einbaut, die versuchen, Sackgassen möglichst früh zu erkennen.

An einigen entscheidenden Stellen wird mit Makros und compilierenden Techniken die Effizienz verbessert, ohne die geforderte Flexibilität zu beeinträchtigen. Wenn man das Wissen von Hand ausprogrammiert, „hart verdrahtet“, dann kann man ein sehr effizientes Programm erhalten. Man verliert aber an Flexibilität, da man bei jeder Änderung am Wissen größere Änderungen am Programm durchführen muß.

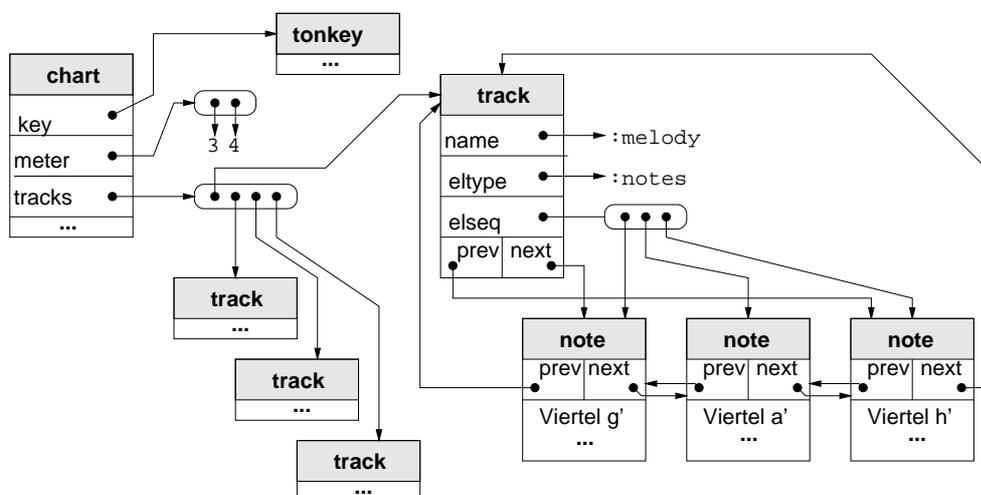
Die Flexibilität gewinnt man durch explizite Wissensdarstellungen. Eine Abarbeitung der Wissensdarstellung in einem Interpreter kostet aber Rechenzeit, da die Datenstrukturen des Wissens bei jedem Zugriff durchwandert werden. Eine compilierende Lösung erzeugt aus der Wissensdarstellung Lisp-Quellcode, der vom Lisp-Compiler übersetzt werden kann und kombiniert so Flexibilität und Effizienz. Der Vorteil im Vergleich zu einer Interpreterlösung liegt darin, daß die Durchwanderung der Datenstruktur der Darstellung nur einmal, bei der Definition des Wissens geschieht, und nicht jedesmal bei dessen Verwendung erfolgen muß. Der Nachteil der compilierenden Lösung ist, daß sie aufwendiger zu erstellen ist.

4.2 COMF - ein Datenformat zur Repräsentation tonaler Musik

Das Datenformat COMF definiert Repräsentationen für die Grundelemente tonaler, notierter Musik und eine Partiturkarte, die solche Elemente in Stimmen und diese Stimmen in einer Karte anordnet. Die Klassen für die Elemente werden durch Mehrfachvererbung aus Kartenelementen, Dauerelementen und tonaler Information, z.B. Tonhöhen, Intervallen und Harmonien zusammengefügt. Abbildung 4.2 gibt einen Überblick über die Klassenhierarchie von COMF. Ebenso werden Elemente für strukturelle Information definiert.

Die Slotwerte der musikalischen Elemente werden aus Effizienzgründen intern soweit als möglich durch Zahlen dargestellt. Für die externe Darstellung wird, soweit gebräuchlich, die deutsche Terminologie verwendet. Davon

⁵Typdeklaration sind in Lisp optional.



Das Box-Pointer-Diagramm ist aus Übersichtlichkeitsgründen vereinfacht. Bei Objekten werden Attribute weggelassen und die Zellen der Listen sind zu einem Oval zusammengezogen.

Abbildung 4.3: Datenstruktur der Partiturkarte im Box-Pointer-Diagramm

angegeben werden. Das für die ganze Karte gültige Metrum wird als Liste aus Zähler und Nenner angegeben.

Über die Karte hat man Zugriff auf alle Elemente, die zu einem Zeitpunkt laufen und alle Elemente, die zu einem Zeitpunkt gerade beginnen. Umgekehrt kann man zu einem gegebenen Element die Spur und den Zeitpunkt eines Elements im Kartenzusammenhang bestimmen.

4.2.2 Tonale Elemente

Die tonalen Elemente repräsentieren Töne, Tonarten, Akkorde und Intervalle. Hierbei gibt es vielfältige Beziehungen und Gemeinsamkeiten. Eine ausführliche Studie über die objektorientierte Modellierung der tonalen Begriffe und die inhaltlichen Zusammenhänge findet sich bei [Mah87]. Das Klassensystem von COMF ist demgegenüber deutlich vereinfacht.

Tonhöhen (Klasse `pitch`) werden durch den Tonnamen, Oktave und Vorzeichen dargestellt. Durch Verbindung einer Tonhöhe mit einem Dauernelement erhält man die Note (Klasse `note`), die in einer Stimme verwendet werden kann.

Tonarten (Klasse `tonkey`) sind eine Subklasse der Tonhöhe, bei der das Tongeschlecht (Dur oder Moll) hinzukommt. Töne können durch Stufen (Klasse `degr`) relativ zu einem Bezugspunkt dargestellt werden. Die Stufennummer

gibt die Zahl der diatonischen Schritte vom Grundton aus an, dazu kommt noch die Alteration gegen den durch die Tonart bestimmten Ton, die durch die Zahl der Halbtöne angegeben wird.

Intervalle (Klasse `interv`) werden durch ihre Schrittzahl charakterisiert. Die Schrittzahl ist so, daß die Prime den Wert 0 hat, so daß man bei der Bestimmung des Zieltons einfach rechnen kann. Intervalle können entweder absolut im Tonraum angegeben werden (Klasse `cintv`), dann kommt ein Modifikator (groß, klein, vermindert, ...) dazu oder relativ zur Tonart (Klasse `dintv`), dann kommt die Alteration gegen die Tonart und ein Verweis auf die Tonart dazu.

Harmonien (Klasse `harm`) werden durch den Akkordgrundton und die Liste der restlichen Akkordtöne repräsentiert. Der Akkordgrundton wird durch die Stufe bezüglich der aktuellen Tonart dargestellt, während die Akkordtöne durch Stufen bezüglich des Akkordgrundtons angegeben werden. Eine Erweiterung ist die Umkehrung (Klasse `invers`). Hier wird noch die Stufe des Baßtons relativ zum Akkordgrundton angegeben.

Ebenso wie Töne können Stufen, Intervalle und Harmonien mit einer Dauer verbunden werden und so als Träger von Dauerninformation und Elemente von Stimmen einer Partiturbkarte auftreten. Für die verschiedenen Klassen von COMF, die eine Note beschreiben, sind Konversionsfunktionen definiert, z.B. die Bestimmung der Stufe eines Tons in der Tonart, der Zielton eines Intervalls von einem Ton ausgehend, usw. .

4.2.3 Verwendung der definierten Klassen

Durch Mehrfachvererbung werden tonale Elemente mit einem Dauerelement kombiniert. Da ein Dauerelement gleichzeitig Kartenelement ist, können Instanzen der Subklassen von Dauerelement auch in die Karte eingetragen werden. Die tonalen Elemente ohne das Dauerelement kommen im Programm gelegentlich als Zwischenergebnisse vor.

Die Chart, die Tracks und die Elemente mit Dauern, d.h. Noten, Intervalltöne, Stufentöne und Harmonieelemente haben alle eine externe Darstellung, die vom Rechner wieder eingelesen werden kann. Die Verzeigerung der Chart und der Zusammenhang von Harmonieelementen mit der Grundtonart wird dabei nicht mit auf die Datei geschrieben, da zyklische Strukturen entstehen würden, die man so nicht auf Datei schreiben kann. Zudem ist es nicht nötig, diese Verbindungen auf Datei zu speichern, da sie rekonstruiert werden können. Die externe Darstellung der Chart enthält einfach eine Liste von Spuren, die wiederum die Elemente enthalten.

Bei der externen Darstellung wurde darauf geachtet, daß die Elemente für den Menschen gut lesbar sind. Es wird eine Tabelle zur Umsetzung der

äußeren Darstellung in die interne Darstellung durch Zahlen verwendet. Die äußere Darstellung verwendet die deutschen Bezeichnungen bzw. Abkürzungen davon. Durch Verwendung einer anderen Tabelle kann die Darstellung auf die Bezeichnungen einer anderen Sprache umgestellt werden.

4.2.4 Ausgabe der Noten mit Lilypond

Das Notensatzprogramm Lilypond [NN99] ist Teil des GNU-Projekts und steht in der Tradition von $\text{T}_{\text{E}}\text{X}$ und $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. Lilypond erzeugt aus einer textuellen Darstellung der Musik auf logischer Ebene ein hochwertiges Druckbild. Aus der Quelldatei im Lilypond-Format erstellt Lilypond $\text{T}_{\text{E}}\text{X}$ -Dateien, die dann unter Verwendung von speziellen, mit Lilypond mitgelieferten Fonts wie jede $\text{T}_{\text{E}}\text{X}$ -Datei zu dvi-Dateien und anschließend zu PostScript-Dateien weiterverarbeitet werden.

Das Schreiben von Lilypond-Dateien ist verhältnismäßig einfach, da das Datenformat von Lilypond ebenso wie die COMF-Partiturskarte in Stimmen strukturiert ist. Der Kopf einer Lilypond-Datei legt die Parameter des Papiers und des Seitenaufbaus fest. Die Partitur besteht aus Noten- und Textzeilen. Bei den Notenzeilen wird jeweils Tonart und Metrum angegeben, damit Lilypond die Vorzeichen und die Taktart richtig setzen kann. Notenzeilen enthalten dann Noten und Pausen. Die Ausgabefunktion von COMF kann Textzeilen zur Ausgabe von Strukturelementen nutzen. Sie gibt Tracks vom Elementtyp `:annotation` als Textzeile aus. Die Elemente eines solchen Tracks müssen Objekte der Klasse `tagel` sein. Der Wert des Slots `tag` wird als $\text{T}_{\text{E}}\text{X}$ -Code interpretiert. Damit kann man in der Ausgabe Annotationen an der metrisch korrekten Position anbringen.

Ebenso wie COMF kennt Lilypond die Bedeutung von Metrum und Tonart. Lilypond setzt Vorzeichen, Taktstriche, Notenhäse und -balken, horizontale Abstände, Umbruch und Akkoladenklammern von selbst korrekt, so daß ein Großteil der Noten direkt nach Lilypond umgesetzt werden kann. Eine Ausnahme sind Dauern, die im Notenbild durch eine Überbindung dargestellt werden. Für Lilypond müssen die Überbindungen zuerst drucktechnisch korrekt in Teildauern aufgelöst werden. Außerdem werden Triolen in Lilypond durch ein Triolenelement, das die notierten Dauern enthält, dargestellt, während COMF Einzelnoten mit Primfaktor 3 im Nenner hat.

4.3 Kompositionssteuerung

Die Kompositionssteuerung arbeitet die Schritte des Kompositionsplans der Reihe nach ab, verwaltet die anfallenden Lösungen und versorgt die Funktio-

nen der Stimmengenerierung über den Regelsatz und die Strukturbeschreibung mit den korrekten Parametern. Sämtliche Operationen sind nichtdestruktiv realisiert, damit bei Bedarf eine andere Abarbeitungsstrategie, z.B. mit Backtracking, eingesetzt werden kann.

4.3.1 Aufbau eines Regelsatzes

Der Regelsatz ist ein Objekt der Klasse `ruleset`. Es wird mit dem Definitionsmakro `def-ruleset` definiert. Dabei wird das Objekt erzeugt und in die globale Liste der Regelsätze eingetragen, wobei ein Objekt mit gleichem Namen ein bisher eingetragenes überschreibt.

Format des Regelsatzes

```
(def-ruleset <name>
  (:score ((<attr> <wert>)...))
  (:melody ((<attr> <wert>) ...))
  (:bass ((<attr> <wert>) ...)))
```

Es gibt also unter `:score` eine Assoziativliste⁶ mit Angaben, die für das ganze Stück gelten und für jede Stimme eine Assoziativliste mit weiteren Angaben. Die Attributnamen sind Keywords.

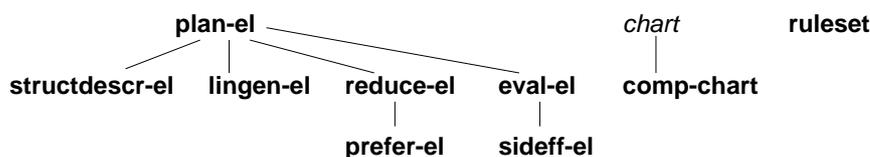
Die Regelfolgen für die Stimmengenerierung sind bei der entsprechenden Stimme mit abgelegt; die Folge für die Startnote unter dem Attributnamen `:startrules` und die Regelfolge für die Folgeintervalle unter dem Attributnamen `:intrules`. Für den Baß sind die Stimmführungsregeln unter `:vlseq` abgelegt.

4.3.2 Aufbau der Strukturbeschreibung

Die Strukturbeschreibung ist syntaktisch wie der Regelsatz aufgebaut: Sie enthält Angaben für die ganze Partitur und für die einzelnen Stimmen. Die Vorgehensweise bei der Erzeugung der Strukturbeschreibung ist in den Kapiteln 3.3.3 und 3.3.4 zu finden.

Die Makrostruktur ist ein Teil der Strukturbeschreibung und wird in der Strukturbeschreibung in den Angaben für die Partitur (`:score`) unter dem Parameternamen `:secpars` abgelegt. Die Makrostrukturbeschreibung besteht aus einer Abfolge von Beschreibungen für jeden Satz. Solch eine Satzbeschreibung besteht aus dem Satznamen und einer Assoziativliste mit Angaben für diesen Satz (Beispiel siehe Kapitel 3.3.3).

⁶Eine Assoziativliste (assoc list) enthält Sublisten mit Attribut und Wert.



Klassennamen in Kursivdruck sind in einem anderen Diagramm bereits eingeführt.

Abbildung 4.4: Klassenhierarchie der Elemente des Kompositionsplans

Bei der Erzeugung der Angabensequenzen für die Mikrostruktur hängen viele Entscheidungen oftmals von der Position in der Makrostruktur ab. Deswegen wurde die Funktion `iter-sections` als allgemeines Werkzeug zur Durchwanderung der Makrostruktur definiert. Sie geht die Makrostruktur durch und ruft für jeden Takt eine als Parameter übergebene Angabefunktion auf, die den Tag-Wert für die Angabensequenz zurückgibt. `iter-sections` erzeugt damit das `tagel` und sammelt alle Elemente in einer Liste.

Die Angabefunktion erhält als Parameter die Satzbeschreibung des aktuellen Satzes, die Zahl der Takte seit Beginn des aktuellen Satzes und das letzte erzeugte mikrostrukturelle Element. Auf diese Weise kann durch Definition einer neuen Angabefunktion eine neue Sequenz erzeugt werden.

4.3.3 Abarbeitung von Kompositionsplänen

Ein Kompositionsplan ist eine Abfolge von Planelementen, von denen jedes einen Schritt beschreibt. Die Funktion `cascad-linplan` arbeitet diesen Plan ab und aktualisiert nach jedem Schritt die Lösungsmenge.

Alle Planelemente sind Subklassen der Klasse `plane1`. Einen Überblick über die für die Kompositionssteuerung wichtigen Klassen findet man in Abb. 4.4. Die Bearbeitung eines Planschritts übernimmt die generische Funktion `exec-plane1`. Diese erhält die bisherige Lösungsmenge und das Planelement als Parameter und gibt die aktualisierte Lösungsmenge zurück. Für jede Planelementklasse gibt es eine Methode von `exec-plane1`, die die entsprechende Aktion ausführt.

Die Lösungen sind Objekte der Klasse `comp-chart`, einer Subklasse von `Chart`, die zusätzliche Slots für den Regelsatz, die Strukturbeschreibung und den Suchzustand, mit dem diese Chart erreicht worden ist, enthält. Über den letzten Suchzustand ist der gesamte Kompositionsweg zugänglich. Der Startwert für die Lösungsmenge ist eine Liste mit einer leeren Chart, die Regelsatz enthält. Der erste Kompositionsschritt fügt die Strukturbeschreibung hinzu.

Folgende Klassen von Planelementen sind definiert:

structdescr-el Der Konstruktor für das Strukturbeschreibungselement benötigt keine Argumente. Die `exec-planet`-Methode legt die erstellte Strukturbeschreibung in den entsprechenden Slot der Chart.

lingen-el Diese Klasse steht für einen Stimmengenerierungsschritt. Es muß der Name der Stimme, der Name des Satzes und die gewünschte Lösungszahl angegeben werden.

Die `exec-planet`-Methode dieser Klasse ruft für jede Chart in der Lösungsmenge die Suchstrategie „Zufallsschuß“ (vgl. Kap 4.4), die versucht, die im Planelement angegebene Zahl von Lösungen zu erzeugen.

Gibt es beispielweise in der Lösungsmenge 5 Charts und das Element fordert eine Lösungszahl von 3, dann hat man nach der Abarbeitung des Planelements im Falle eines vollen Erfolgs aller 5 Suchvorgänge 15 Lösungen. Es kann jedoch vorkommen, daß für eine Chart weniger als die angegebene Zahl von Lösungen gefunden wird und das Zufallsschußverfahren abbricht, weil das Limit für die Zahl der Zufallschüsse erreicht wurde. Dies kommt dann vor, wenn die bisherige Lösung mit ihrer Strukturbeschreibung so nicht fortgesetzt werden kann.

reduce-el Das Reduktionselement enthält eine Lösungszahl. Die Methode `exec-planet` reduziert die Lösungsmenge auf diese Anzahl per Zufall.

prefer-el Dieses Planelement verkürzt ebenfalls die Lösungsmenge auf die angegebene Zahl. Statt zufällig auszuwählen, wird jedoch das Ergebnis der Bewertungsfunktion berücksichtigt.

eval-el Bei der Definition des Planelements wird ein Codestück angegeben, dem die aktuelle Lösungsmenge in der Variable `charts` zur Verfügung steht.

Die `exec-planet` für diese Klasse stellt dem Codestück die Variable `charts` zur Verfügung, indem sie es in `(let ((charts . .)) <code>)` einpackt und führt den gesamten Ausdruck dann mittels `eval` aus. Das Ergebnis wird als neue Lösungsmenge zurückgegeben.

Dieses Element ist für Fehlersuche und Experiment mit dem Programm nützlich, da man an Zwischenergebnissen Änderungen vornehmen kann, und so mit dem folgenden Kompositionsschritt beliebig experimentieren kann.

sideeff-el Dieses Element verhält sich wie ein `eval-el`, läßt aber die Lösungsmenge unverändert.

Dieses Verhalten wird durch eine `:around`-Methode auf `exec-planet` realisiert, die mittels `call-next-method` die nächste Methode, nämlich die der Superklasse `eval-el`, ruft und dann die unveränderte Lösungsmenge zurückgibt.

Dieses Element ist für Debug-Zwecke nützlich, beispielsweise kann man für einen Kompositionsschritt mittels eines `sideeff-el` die Ausgabe von Debuginformation einschalten und anschließend wieder abschalten.

4.3.4 Initialisierung der Stimmengenerierung

Um für ein `lingen-el`-Element die Stimmengenerierung aufzurufen, wird zuerst der Startzustand - ein Objekt der Klasse `lingen-st` erzeugt, den dann die Suchstrategie als Startzustand erhält. Der zu komponierende Satz in einer Stimme ist zu Beginn leer.

Aus den im Regelsatz angegebenen Regelfolgen werden nun die Alternativenfunktionen für die Startnoten und die Folgeintervalle erzeugt. Außerdem werden die für diesen Satz gültigen Parameter aus den globalen und stimmbezogenen Teilen von Strukturbeschreibung und Regelsatz zusammengestellt. Während der Stimmengenerierung wird häufig auf die Parameter zugegriffen. Damit der Zugriff effizient ist, werden die Parameter in einer Hashtabelle abgelegt.

Reihenfolge, in der die Parameter eingetragen werden

Nr.	Quelle	einzutragende Werte
1	bisherige Chart	alle bereits komponierten Stimmen werden unter ihrem Namen eingetragen
2	Strukturbeschreibung	satzspezifische Parameter
3	Strukturbeschreibung	stimmspezifische Parameter
4	Strukturbeschreibung	globale Parameter
5	Regelsatz	stimmspezifische Parameter
6	Regelsatz	globale Parameter
7	Makrostruktur	wird unter <code>:secpars</code> eingetragen
8	aktueller Satzname	wird unter <code>:phasename</code> eingetragen
9	Endformel	Muster für aktuelle Stimme und Satz wird als Parameter <code>:endform-lin</code> eingetragen

Beim Aufbau der Hashtabelle wird nach der Rangfolge vorgegangen. Bei jedem Schritt werden die Parameterwerte in die Hashtabelle übernommen, falls nicht schon ein gleichnamiger Wert in der Tabelle eingetragen ist.

Um den Zugriff auf Parameter zu vereinheitlichen, ist eine generische Funktion `get-par` definiert, die im Stimmengenerierungszustand den Parameter in der Hashtabelle sucht und bei Charts und Regelsätzen in den entsprechenden Assoziativlisten nachschaut.



Abbildung 4.5: Klassenhierarchie der Zustände

4.4 Suchstrategien und Zustandsverwaltung

Der Zustand für Suchvorgänge enthält sowohl Information, die für die aufgabenunabhängige Suchstrategie benötigt wird, als auch solche, die von der strategieunabhängigen Problembeschreibung stammt. Die Basisklasse definiert den aufgabenunabhängigen Anteil und erlaubt somit die Definition der Suchverfahren. Einen Überblick über die Klassen gibt Abbildung 4.5.

4.4.1 Basisdefinitionen zur Suche

Die Basisklasse `searchstate` legt die Schnittstelle zwischen Problem und Strategie fest. Sie ist eine abstrakte Klasse⁷, d.h. es ist nicht vorgesehen, sie zu instanziiieren. Die Klasse `searchstate` enthält die Slots für die Information, die bei jeder Problembeschreibung anfällt. Dies sind der Verweis auf den Vorgängerzustand und der Übergangoperator.⁸

Außerdem werden die generischen Funktionen definiert, die von den Suchstrategien an geeigneter Stelle gerufen werden. Das sind die Zustandsübergangsfunktion (`successors`), der Erfolgstest (`goal-p`) und der Fehlschlagtest (`fail-p`). Die Definitionen der generischen Funktionen legen nur den Parametersatz fest. Für die Basisklasse des Zustands sind keine Methoden implementiert. Die Methoden, die das konkrete Verhalten realisieren, werden erst in der Problembeschreibung, d.h. für die spezielle Zustandsklasse des Problems, implementiert.

Für Zustände, die einen Gewichtungsfaktor tragen, ist die Klasse `weight-mixin` definiert. Sie ist eine Mix-in-Klasse, d.h. sie ist nicht dafür vorgesehen, daß sie oder eine ihrer durch Einfachvererbung entstandenen Subklassen instanziiert wird, sondern dient lediglich dem Aufbau neuer Klassen durch Mehrfachvererbung. Für Zustandsklassen, die kein `weight-mixin` enthalten, hat die Basisklasse `search-state` eine Defaultmethode, die als Gewichtungsfaktor den Wert 1 zurückgibt.

⁷Von der Definition der Programmiersprache her kann man jede Klasse von CLOS instanziiieren. Die Begriffe *abstrakte Klasse* und *Mix-in-Klasse* geben hier an, ob dies von der Anwendung her sinnvoll ist.

⁸Fußnote 7 aus Kapitel 3 auf Seite 58 gilt sinngemäß.

4.4.2 Suchstrategien

Die Suchstrategien sind Funktionen. Sie erhalten einen Startzustand als Argument und durchwandern den Suchraum. Bei den einzelnen Bearbeitungsschritten ruft die Funktion der Suchstrategie die drei oben genannten generischen Funktionen auf.

Von den Suchverfahren wird eine Variante verwendet, die eine gegebene Zahl von Lösungen erzeugt. Bei der Tiefensuche und der zufallsgesteuerten Tiefensuche sucht diese Variante einfach weiter, bis die gewünschte Zahl von Lösungen erreicht ist oder der Suchbaum fertig durchsucht ist.

Beim Zufallsschußverfahren werden so lange Zufallsschüsse durchgeführt, bis die geforderte Zahl erreicht ist. Wie in Kapitel 3.4.4 beschrieben, kann die Zufallsschußstrategie nicht erkennen, wenn es keine Lösung gibt. In diesem Fall oder im Fall, daß die Lösungen im Suchraum so selten sind, daß das Verfahren keine findet, würde das Verfahren endlos laufen. Um dies zu verhindern, wird durch die Variable **shots-per-sol** eine Obergrenze für die Zahl der Versuche pro geforderter Lösung angegeben. Überschreitet die Zahl der durchgeführten Versuche die Obergrenze (Produkt aus der Zahl der geforderten Lösungen und **shots-per-sol**), wird der Suchvorgang abgebrochen. Dies ist wichtig, damit der Suchvorgang auf jeden Fall terminiert. Die Termination ist nötig, da das Suchverfahren zur Stimmengenerierung eingesetzt wird und nur ein Teilschritt bei der Abarbeitung des Kompositionsplans ist.

4.4.3 Problembeschreibung für die Stimmengenerierung

Die Stimmengenerierung ist als Suchproblem formuliert. Die Problembeschreibung definiert eine Zustandsklasse, die den Stimmengenerierungsregeln die benötigte Information über den Stand der Suche zur Verfügung stellt. Außerdem definiert sie Methoden für die drei generischen Funktionen der Suche im Zustandsraum.

Für die Verwendung in der Zustandsübergangsfunktion werden aus den Regeln zwei Funktionen erzeugt, eine für die erste Note zu Beginn des Abschnitts, die andere für die folgenden Noten. Die beiden Funktionen sind im Zustand abgespeichert, so daß die *successors*-Methode die richtige Funktion rufen kann. Auch die von der Basisklasse geerbten Slots werden passend belegt: Die Zustandsübergangsfunktion speichert das Fortsetzungselement, mit dem der Zustand erreicht wurde, als Operator (Slot *oper*) ab und belegt die Referenz auf den Vorgängerzustand. Damit ist der Suchvorgang explizit repräsentiert und rekonstruierbar.

Die generierte Stimme ist im Slot *notelist* abgelegt. Sie ist anfangs leer und wird bei jedem Schritt verlängert. Außerdem werden die metrischen

Längen der Komposition insgesamt (seit Beginn des Stücks) und ab Beginn des laufenden Suchvorgangs mitgeführt. Dies erspart die Neuberechnung des metrischen Gewichts der zu komponierenden Note, was häufig benötigt wird.

Für einige Zugriffsfunktionen von COMF wurden Methoden für den Zustand definiert, z.B. für die aktuelle Tonart, deren Aufrufe weiterdelegiert werden. Häufig wird auch auf die Strukturbeschreibung zugegriffen. Für den Zugriff auf das aktuelle Element einer Elementfolge aus der Strukturbeschreibung gibt es eine spezielle Funktion. Diese Definitionen vereinfachen die Implementierung der Regelvoraussetzungen und -folgerungen und sorgen für eine gut lesbare Regeldefinition.

Um heuristische Suchstrategien zuzulassen, ist der Stimmengenerierungszustand eine Subklasse von `weight-mixin`. Die Alternativen sind ebenfalls eine Subklasse von `weight-mixin`, so daß aus ihnen die benötigten Gewichtswerte entnommen werden können.

4.4.4 Benutzerinformation während der Suchvorgänge

Während die Suchstrategie den Ablauf des Suchvorgangs bestimmt, ist das Wissen in den Regeln repräsentiert. Die Auswirkungen von Änderungen an den Regeln kann man nur entweder indirekt über die Auswirkungen auf das Ergebnis oder durch den Ablauf des Suchvorgangs beobachten. Daher sind in den Suchstrategien Möglichkeiten zur Beobachtung des Suchvorgangs eingebaut.

Für alle Suchstrategien kann man über globale Variablen angeben, welche Information man bei der Suche erhalten möchte. Folgende Möglichkeiten sind definiert:

- Bei jeder gefundenen Lösung wird eine Meldung ausgegeben.
- Bei der Zufallsschußstrategie kann man sich regelmäßig (z.B. bei jedem 100. Versuch) eine Meldung ausgegeben lassen. Die Häufigkeit steuert man über die Variable `*shot-verbose-frequency*`. So kann man erkennen, daß die Suche noch läuft und das Programm nicht abgestürzt ist.
- Eine Beobachtung der Umstände unter denen eine Versuch nicht zum Erfolg führt, ist oft lehrreich. Setzt man die Variable `*death-report-action*` auf `t`, so wird eine Meldung über jeden Mißerfolg angezeigt.

Die Ausgabe gibt den Grund des Scheiterns an, nämlich entweder, daß es keine Alternativen (Folgezustände) gab oder daß der Fehlschlagtest

`fail-p` wahr war. Außerdem wird der metrische Zeitpunkt, zu dem die Lösung gescheitert ist, angegeben.

Für eine gezieltere Untersuchung des Suchvorgangs gibt man in `*death-report-action*` noch eine zusätzliche Bedingung an, die gelten muß damit die Meldung angezeigt wird. Außerdem kann man eine Funktion angeben, mit der eine selbstdefinierte, detailliertere Ausgabe erzeugt wird.

Alle Ausgaben werden auf einen Stream ausgegeben, der per Default auf die Lisp-Interaktion geht, aber auch in eine Datei umgelenkt werden kann.

4.4.5 Statistikkomponente

Die Statistikkomponente protokolliert für einen Suchlauf Daten über den Suchvorgang mit.

Dies geschieht mittels `:around`-Methoden für die drei generischen Funktionen der Suchstrategie, die auf die Klasse `search-state` spezialisiert sind. Diese Methoden rufen mittels `call-next-method` die Hauptmethode. Diese Technik erlaubt es, die Statistik zum bestehenden System dazuzuladen, ohne etwas an den bestehenden Funktionen ersetzen zu müssen.

Es werden folgende Daten über den Suchvorgang erfaßt:

- Zustandsübergänge werden gezählt.
- Es wird eine Liste der Längen (Notenanzahlen) von erfolgreichen Lösungen geführt.
- Es wird eine Liste der Hypothesenlängen bei Fehlschlägen durch den Fail-Test und bei Aussterben der Zustände (Nachfolgerfunktion liefert keine Nachfolgezustände) geführt.
- Die Verzweigungsgrade (Zahl der Folgezustände) bei jedem Aufruf von `successors` werden in einer Liste abgelegt.
- Die CPU-Zeit für den Versuch wird gestoppt.

Aus diesen Absolutzahlen werden dann Auswertungen vorgenommen, die von der absoluten Größe des Suchvorgangs (Anzahl der Zustandsübergänge) abstrahieren sollen. Damit kann man Aussagen über das bei diesem Suchvorgang eingesetzte Wissen treffen. Einzelheiten über Auswertungen und Beispiele findet man in Kapitel 5.5

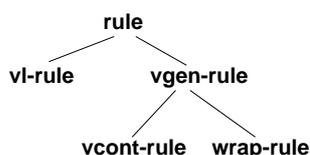


Abbildung 4.6: Klassen für die verschiedenen Regeln

4.5 Regeln und Stimmengenerierung

An verschiedenen Stellen im System wird Wissen durch Regeln dargestellt.

4.5.1 Gemeinsame Elemente aller Regeln

Die Klassen für die verschiedenen Arten von Regeln sind alle Subklassen der Klasse `rule` (siehe Abbildung 4.6), die gemeinsame Eigenschaften aller Regeln implementiert. Regeln aller Regeltypen werden mit dem Definitionsmakro `def-rule` definiert. Die Definitionssyntax aller Regeln beginnt mit dem Regelnamen gefolgt vom Regeltyp, einem Symbol. Die danach folgenden Elemente der Regeldefinition werden einer regeltypspezifischen Bearbeitung überlassen. Für verschiedene Anwendungsbereiche kann man so jeweils eigene Regeltypen mit eigener Syntax definieren.

Man kann weitere Regeltypen hinzufügen, ohne dafür das Definitionsmakro `def-rule` ändern zu müssen, da die dem Regeltyp zugeordnete Regelklasse über eine globale Tabelle bestimmt wird. Die Funktion `add-ruletypes-for-class` ordnet einem Regeltypsymbol die zu erzeugende Klasse zu. Dabei können verschiedene Regeltypen die gleiche Regelklasse erzeugen. Den Rest der Regeldefinition verarbeitet die Initialisierung der Regelklasse.

Die Basisklasse implementiert die Verwaltung der Regeln. Die Regeln können mittels `get-rule` über ihren Namen referenziert werden. Dafür führt die `:after`-Methode von `initialize-instance` für die Klasse `rule` eine Hashtabelle aller definierten Regeln. Anschließend ruft diese Methode die generische Funktion `parse-rule`, deren `:before`-Methode den Regelnamen prüft und übernimmt.

Um die speziellen Syntaxen der verschiedenen Regelklassen zu implementieren, sind für diese Klassen Hauptmethoden zu `parse-rule` und Slots für die beim Parsen der Syntax gefundenen Daten definiert. Die Originaldefinition wird im Slot `orig-def` abgelegt, so daß die Ausgangsgestalt der Regeldefinition wiederhergestellt werden kann. Damit wird für die Regelklasse immer die `:before`-Methode von `parse-rule` auf `rule` und die Hauptmethode der speziellsten Regelklasse gerufen.

4.5.2 Regeln für die Stimmengenerierung

Die Regelklasse für Stimmengenerierungsregeln wird in zwei Schritten definiert. Der erste Zwischenschritt, die abstrakte Klasse `vgen-rule`, definiert einen Slot für den der Regel zugeordneten Code. Von dieser werden dann die Klassen für die eigentliche Stimmengenerierungsregel `vcont-rule` und die Klasse für die Hüllregel `wrap-rule` abgeleitet.

Die Stimmengenerierungsregeln bestimmen die Auswahl der Alternativen für die Fortsetzung einer Stimme um eine Note. Diese Klasse definiert die Slots für die in Kapitel 3.5.5 beschriebenen Regelbestandteile Bedingung, Folgerungstyp und Folgerungsausdruck.

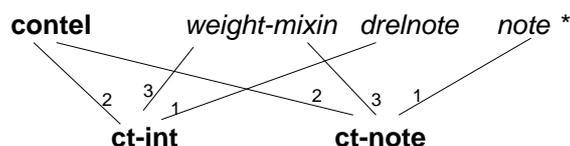
Aus der Folge der Stimmengenerierungsregeln wird die Funktion gebildet, die die Menge der zulässigen Alternativen bestimmt. Dafür wird für jede Regel Code erzeugt, der die Menge der möglichen Fortsetzungen bearbeitet und das Ergebnis der Variable `cs` zuweist. Für die verschiedenen Fortsetzungstypen ist jeweils eine Codeschablone definiert, in die der gegebene Folgerungsausdruck mittels Backquote und Komma eingesetzt wird.

Die Schablone für den Folgerungstyp `:filter` lautet beispielsweise:
`'(remove-if-not #'(lambda (ctel),(conseq-expr rule)) cs)`
 Der Ausdruck mit dem Komma `,(conseq-expr rule)` setzt den bei der Regeldefinition angegebenen Folgerungsausdruck in die Codeschablone ein. Auf diese Art ist in den elementweise gerufenen Folgerungsausdrücken das Element über die Laufvariable `ctel` zugänglich.

Aus dem Code der Regeln wird dann ein Lambdaausdruck - eine namenlose Funktion - zusammengebaut. Die Lambdaliste definiert die Parameter `cs` und `st`, die dann im Regelcode zugänglich sind. Als Funktionsrumpf folgt der Code der Regeln in der durch den Regelsatz vorgegebenen Reihenfolge. Rückgabewert ist der Wert von `cs` nach der Ausführung aller Regelcodes.

Da die Zugriffe auf die Alternativenobjekte in Bedingung und Folgerungsausdruck durch Codestücke angegeben werden, ist es für die Arbeit des Regelcompilers unerheblich, ob die Alternativenmenge aus Fortsetzungenoten (die erste Note) oder aus Fortsetzungsintervallen (die folgende Note) besteht. Die Codestücke in der Regeldefinition funktionieren allerdings möglicherweise nicht für Alternativen beider Klassen. Das Gültigkeitsflag ist dafür da, daß der Regelcompiler den korrekten Einsatzbereich der Regel überprüfen kann und so Laufzeitfehler vermeidet.

Die Hüllregeln definieren zu einer Regel eine zusätzliche Voraussetzung. Die Code-Methode für Hüllregeln ruft die Code-Methode der eingehüllten Regel und fügt diesem Code die zusätzliche Voraussetzung hinzu.



*Klassennamen in Kursivdruck sind in einem anderen Diagramm bereits eingeführt.

Abbildung 4.7: Klassen für Fortsetzungselemente

4.5.3 Fortsetzungselemente

Die Elemente der Alternativenmenge bei der Stimmengenerierung sind die Fortsetzungselemente. Die Klasse `contel` ist eine Mix-in-Klasse, die Träger der Information ist, die für die Rolle als Alternativenelement wichtig ist. Diese besteht aus einem Verweis auf den Suchzustand, von dem die Fortsetzung ausgeht und eine Liste der Bottom-up-Strukturelemente, die durch diese Fortsetzung begonnen werden (vgl. Kapitel 3.5.2).

Wie in Kapitel 3.5.5 beschrieben, sind die Fortsetzungselemente beim ersten Zustandsübergang Noten, bei den folgenden dann Relativtöne (Intervalle mit Dauer). Dementsprechend erbt die Klasse `ct-note` für die Fortsetzungsnote von `note`, vom Fortsetzungselement `ct-el` und von der Klasse `weight-mixin`. Die Klasse `ct-int` für das Fortsetzungsintervall kombiniert Fortsetzungselement `ct-el` und `weight-mixin` mit dem Relativton `drelnote`.

Damit unterschiedliche Suchstrategien möglich sind (z.B. Zustände verworfen werden können), darf die Suche keinerlei Änderungen am statischen Teil des Wissens vornehmen. Für die Fortsetzungselemente ist deswegen der begriffliche Unterschied zwischen folgenden zwei Verwendungsarten zu beachten:

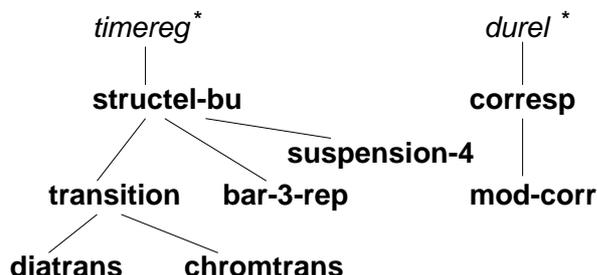
- Ein Fortsetzungselement kann im Folgerungsausdruck einer Regel stehen, z.B. als Teil der Startmenge oder in einem Folgerungsausdruck für den Folgerungstyp `:add`. Das Element wird als Beschreibung eines Fortsetzungselements in der Regel eingesetzt.

Hier gehört das Fortsetzungselement zum Regelcode und damit zum festen Teil des Wissens.

- Während der Suche werden Fortsetzungselemente als Element der Alternativenmenge benutzt.

Hier gehört es zum variablen Teil des Suchzustands.

Um den Aufbau des Klassensystems nicht zu sehr zu verkomplizieren, werden die beiden Verwendungsweisen durch die gleiche Klasse dargestellt. Für



**Klassennamen in Kursivdruck* sind in einem anderen Diagramm bereits eingeführt.

Abbildung 4.8: Klassen für Strukturelemente

die Verwendung als Beschreibung ist kein Suchzustand angegeben. Für die Verwendung in der Alternativenmenge wird das beschreibende Fortsetzungselement aus der Regeldefinition unter Hinzufügen des aktuellen Suchzustands kopiert, damit mutierende Änderungen des Fortsetzungselements bei der weiteren Bearbeitung nicht zu Änderungen des beschreibenden Elements führen. Im Extremfall würde dies nämlich zu einer Änderung des Regelquelltexts führen, dessen Teil das beschreibende Fortsetzungselement ja ist. Die externen Darstellungen der Fortsetzungselementklassen sind für die beschreibende Verwendung ausgelegt, da Elemente für den Einsatz in der Alternativenmenge nicht ausgegeben oder auf Datei gespeichert werden müssen.

Die Mehrzahl der Stimmengenerierungsregeln befaßt sich nur mit der Tonhöhe, einige nur mit der Dauer und einige kombinieren Tonhöhen und Dauern. Einige der Regeln befassen sich mit der Tonhöhe und arbeiten unabhängig von der Dauer. Damit man nicht die Fortsetzungselemente für eine Tonhöhe mit sämtlichen denkbaren Dauern erzeugen muß, kann die Dauer verkürzt durch eine Beschränkungsangabe dargestellt werden. Eine der Regeln zur Stimmengenerierung ist die Dauernvergaberegeln. Diese generiert dann unter Berücksichtigung der Beschränkungsangabe und der Parameter im Regelsatz die Menge der zulässigen Dauern.

4.5.4 Strukturelemente

Strukturelemente geben Metainformation über die Komposition an. In den meisten Fällen geht es nur darum, zu repräsentieren, daß für einen bestimmten Zeitbereich ein bestimmtes Attribut gilt; beispielsweise, daß dieser Zeitbereich ein vollkommener Takt (vgl. Kapitel 3.6) sein muß. Dafür genügt die in COMF definierte Klasse `tagel`, die eine beliebige Angabe (`tag`) mit einer Dauer verknüpft.

Eigene Klassen werden nur für solche Strukturelemente benötigt, bei de-

nen entweder noch spezifisches Verhalten definiert werden muß, oder deren Dauer bzw. Lage in der Komposition nicht vorher bekannt ist.

Korrespondenzen

Die Korrespondenzen beschreiben Ähnlichkeiten zwischen einem Abschnitt und einem Vorbild, dem *Modell*. Die Korrespondenz ist ein Dauerelement, für das in der Korrespondenzspezifikation, einer Assoziativliste, zusätzliche Angaben gemacht werden. Bei der Generierung der Stimme kann durch die Strukturbeschreibung eine Korrespondenz gefordert werden. Im Normalfall ist das Modell ein bereits erstellter Teil der Komposition und wird entweder absolut, durch den Anfangszeitpunkt, gegeben oder relativ durch den zeitlichen Abstand der Anfangszeitpunkte von Modell und Korrespondenz.

Ein Spezialfall ist die Korrespondenz mit externem Modell. Sie wird durch die Klasse `mod-corr` repräsentiert. Hier ist das Modell nicht ein Teil der Komposition, sondern eine Notenfolge, die in einem Slot des Korrespondenzelements angegeben wird.

Die Korrespondenz wird durch eine spezielle Regel geprüft, die die Einhaltung aller für den aktuellen Zeitpunkt geltenden Korrespondenzen überwacht. Die Regel ruft die generische Funktion `corresponds-p`, die prüft, ob ein Fortsetzungselement in einen Zustand und für eine gegebene Korrespondenz zulässig ist. In der Korrespondenzspezifikation wird mit dem Schlüsselwort `:startfilter` das Vergleichsprädikat für die erste Note der Korrespondenz und mit dem Schlüsselwort `:intfilter` das Vergleichsprädikat für die folgenden Intervalle der Korrespondenz angegeben. Das Vergleichsprädikat wird dann gerufen, um zu prüfen, ob die aktuelle Fortsetzung mit der entsprechenden Stelle im Modell korrespondiert. Da die Korrespondenz sowohl auf den ersten Ton eines Satzes als auch auf folgende Intervalle zugreifen kann, muß das Vergleichsprädikat für die erste Note sowohl für Fortsetzungsnoten als auch für Fortsetzungsintervalle geeignet sein.

Bottom-Up-Strukturelemente

Die Bottom-up-Strukturelemente werden verwendet, um kleine (aus einigen Noten bestehende) Figuren (z.B. Durchgänge) bei der Stimmengenerierung zu beschreiben, für die Ausnahmen von einem Teil der üblicherweise geltenden Regeln gemacht werden. Da die Dauern der Noten in dieser Figur vorher nicht bekannt sind, ist die Gesamtdauer des Elements vor seiner Vollendung nicht bekannt. Bottom-up-Strukturelemente werden von einem Zustandsnetzwerk - vergleichbar mit einem endlichen Automaten - erkannt.

Ein Bottom-Up-Strukturelement wird durch eine Stimmengenerierungsregel eingeführt. Das Fortsetzungselement hat einen Slot, in dem neu hinzukommende Strukturelemente eingetragen werden können. Auch für die korrekte Fortführung der Figuren sind Stimmengenerierungsregeln zuständig, die die Bottom-Up-Strukturelemente untersuchen.

Basisklasse für Bottom-Up-Strukturelemente ist `structel-bu`, eine Subklasse von `timereg`. Sie enthält einen internen Abarbeitungszustand⁹, der durch ein Symbol bezeichnet wird, und eine Liste der zur Figur gehörenden Suchzustände, damit man nach Vervollständigung der Figur den Zeitraum berechnen kann, den das Bottom-up-Strukturelement überspannt.

Auch bei Bottom-Up-Strukturelementen ist im Prinzip zwischen Beschreibung im Regelrumpf und Einsatz im Suchvorgang zu unterscheiden. Deswegen erzeugen die Regelrumpfe Bottom-Up-Strukturelemente bei jeder Regelanwendung durch einen Aufruf des Konstruktors. Der Konstruktor ist so gestaltet, daß der Klassenname angegeben werden kann und über zusätzliche Angaben weitere Slots belegt werden können, so daß dieser Konstruktor für beliebige Subklassen von `structel-bu` einsetzbar ist und trotzdem alle nötigen Initialisierungen durchgeführt werden. Dies soll das Erstellen weiterer Klassen von Bottom-Up-Strukturelementen und damit die Realisierung weiterer Figuren in der Stimmengenerierung erleichtern.

Wenn aus dem Fortsetzungselement der Folge(such)zustand bestimmt wird, werden alle Bottom-up-Strukturelemente im Suchzustand verlängert, indem die `successor`-Methode des Suchzustands für jedes Strukturelement die generische Funktion `sr-cont` ruft. Die `sr-cont`-Methoden der verschiedenen Klassen von Bottom-up-Strukturelementen definieren den Übergang von Elementzuständen. Die Stimmengenerierungsregeln haben Zugriff auf Elementzustand und können so berücksichtigen, an welcher Stelle man in der zu generierenden Figur ist.

Für die verschiedenen Arten von Bottom-Up-Strukturelementen definiert man nun Subklassen von `structel-bu` (vgl. Abb. 4.8) und Methoden für `sr-cont`. Wenn das Strukturelement noch nicht vollständig ist, gibt `sr-cont` das verlängerte Strukturelement zurück, wenn das Element vollständig ist, NIL. Eine `:after`-Methode von `sr-cont` zur Basisklasse `structel-bu` aktualisiert automatisch die Liste der vom Strukturelement überspannten Suchzustände (Gegenverzeigerung), damit man nach Vervollständigung der Figur über die Suchzustände Beginn und Dauer des Strukturelements bestimmen kann.

⁹Zur Unterscheidung wird in diesem Abschnitt der interne Zustand des Bottom-Up-Strukturelements *Elementzustand* und der Zustand der Stimmengenerierung *Suchzustand* genannt.

Strukturelemente für Durchgänge (Klasse `transition`) haben drei Elementzustände. Der leere Durchgang (vor der Durchgangsnote) hat den Elementzustand `:start`, nach Hinzunahme der Durchgangsnote hat sie den Zustand `:trans` und nach Hinzunahme der Fortführungsnote den Zustand `:completed`. In der Durchgangsklasse wird die Richtung abgespeichert, da der Durchgang in der gleichen Richtung fortgesetzt werden muß, wie er betreten wird.

Vom Durchgang ist wiederum eine Subklasse für den diatonischen Durchgang (Klasse `dia-trans`) implementiert. Es wurden auch Versuche mit einem chromatischen Durchgang (Klasse `chrom-trans`, ebenfalls Subklasse von `transition`) durchgeführt, dieser hat sich jedoch musikalisch nicht bewährt. Die Übergangsmethode für den Elementzustand ist bei der Klasse `transition` angesiedelt, da sie für alle Durchgänge gleich arbeitet.

In Menuettmelodien findet man häufig einen Takt, in dem drei Viertelnoten gleicher Tonhöhe stehen. Die Verwendung von Tonrepetitionen in der Melodie ist im Normalfall durch die Stimmengenerierungsregeln eingeschränkt, aber diese Figur ist eine Ausnahme von der Beschränkung der Repetitionen. Diese Figur wird durch Strukturelemente der Klasse `bar-3-rep` dargestellt. Die Klasse hat die Elementzustände `:start`, `:nr2` und `:nr3`.

Der Quartvorhalt wird durch die Klasse `suspension-4` dargestellt. Der Vorhalt durchläuft nach der zeitgenössischen Theorie [Koc82] die Schritte Vorbereitung, Vorhaltsnote (Dissonanz) und Auflösung. Dies entspricht den Elementzuständen `:start`, `:susp-note` und `:complete`.

4.6 Stimmführung

Die Stimmführungsregeln geben an, welche Fortführungen zweier Stimmen zulässig sind. Sie betrachten also je zwei Töne in zwei Stimmen. Bei der Generierung einer Begleitstimme wird untersucht, ob eine Alternative die Stimmführungsregeln erfüllt. Dabei stößt eine Stimmgenerierungsregel die Überprüfung der Stimmführungsregeln an.

Die Stimmführungsprüfung geschieht in zwei Schritten:

- Die aufeinanderstoßenden Töne der beiden Stimmen und die Töne, die innerhalb der größten Reichweite einer Stimmführungsregel sind, werden in Stimmführungsquadrupeln abgelegt
- Für jedes Stimmführungsquadrupel wird die Stimmführungsfunktion gerufen. Sie gibt einen booleschen Wert zurück. Die Stimmführungsfunktion wird aus der Folge der Stimmführungsregeln erzeugt.

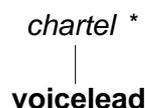


Abbildung 4.9: Klasse für Stimmführungsquadrupel

Die Stimmführung ist insgesamt zulässig, wenn die Stimmführungsfunktion für alle Stimmführungsquadrupel „wahr“ ergibt.

4.6.1 Stimmführungsquadrupel und Reichweiten

Im ersten Schritt wird das Stimmführungsquadrupel - die Kombination von 4 aufeinandertreffenden Noten in zwei Stimmen - als explizite Datenstruktur angelegt. Dies vereinheitlicht die Zugriffswege in den Stimmführungsregeln und erleichtert die Erweiterung des Kompositionsverfahrens um zusätzliche Stimmen.

Die Klasse für das Stimmführungsquadrupel `voicelead` erbt von `chartel` und hat Slots für die 4 beteiligten Noten, den Zeitpunkt, an dem der zweite Zweiklang des Stimmführungsquadrupels beginnt, die Namen der beteiligten Stimmen, den Suchzustand, und die Reichweite des Stimmführungsquadrupels.

Das Stimmführungsquadrupel stellt Funktionen für die Formulierung der Voraussetzungen der Stimmführungsregeln zur Verfügung:

- Zugriff auf die linearen Intervalle der beiden Stimmen erhält man mit `hi-int` und `lo-int`.
- Die beiden Zweiklänge (zusammenklingenden Intervalle) erhält man mit `ha-int1` und `ha-int2`.
- Die Funktion `codis-int` stuft das Intervall nach Konsonanz und Dissonanz (Definitionen nach [Koc82]) ein. Es wird eine Liste von Schlüsselwörtern zurückgegeben.
- Die Funktion `bewtype` klassifiziert die Bewegung des Stimmführungsquadrupels nach den Bezeichnungen aus [Koc82].

Das Stimmführungsquadrupel als Zwischenschritt der Stimmführungsprüfung vereinheitlicht die Zugriffswege für unterschiedliche Reichweiten. Wie in Kapitel 3.7.3 beschrieben gelten die Stimmführungsregeln nämlich nicht nur für direkt aufeinander folgende Töne, sondern auch für Töne, die durch Nebennoten oder kurze Pausen getrennt werden. Die Reichweitenstufen sind

durch Keywords repräsentiert. Die Reichweiten bilden eine Totalordnung, für die das Ordnungsprädikat `reach-includes-p` definiert ist. Eine Stimmführungsregel berücksichtigt ein Quadrupel dann, wenn die Reichweitenstufe des Quadrupels in der Reichweitenstufe der Regel eingeschlossen ist.

Wenn die Stimmengenerierungsregel eine Alternative auf korrekte Stimmführung untersucht, läuft die Erzeugung der Stimmführungsquadrupel nach folgendem Verfahren ab:

- Es werden alle Stimmenpaare aus der neu hinzutretenden Stimme und jeder bestehenden Stimme betrachtet.
- Für jedes Stimmenpaar werden alle Reichweiten untersucht, die in den Stimmführungsregeln vorkommen. Die Menge der zu berücksichtigenden Reichweiten wird im Regelsatz angegeben.
- Für jede solche Konstellation wird das Quadrupel mit den Stimmennamen, dem Suchzustand, der Reichweite und dem Zielton der Alternative als zweitem Ton der zweiten Stimme instantiiert.
- Für jedes dieser Rohquadrupel wird die generische Funktion `v1-contra` gerufen. Für die Reichweitestufen sind jeweils Methoden für `v1-contra` definiert. Die spezifische Methode der Reichweite trägt die fehlenden drei Töne in das Quadrupel ein.
- Alle Stimmführungsquadrupel, bei denen Töne unbelegt sind, werden entfernt. Dies geschieht beispielsweise, wenn bei Reichweite `:acc-beat` zum entsprechenden Zeitpunkt eine der Noten nicht neu anschlägt.
- Die Stimmführungsquadrupel werden normalisiert. Wenn der erste Ton der oberen Stimme tiefer als der erste der unteren ist, werden die Stimmen getauscht.
- Aus der Menge der Stimmführungsquadrupel werden die Doubletten eliminiert. Sind bei zwei Quadrupeln alle vier Töne gleich, dann wird das Quadrupel mit der *größeren* Reichweite, d.h. das, dessen Reichweite die des anderen einschließt, entfernt.

Wenn beispielsweise Melodie und Baß in Vierteln fortschreiten, werden bei Reichweite `:direct` und bei Reichweite `:acc-beat` die gleichen Töne erreicht. In diesem Fall wird das Quadrupel mit der Reichweite `:direct` behalten.

Die Implementierung ist auf mögliche Erweiterung des Reichweitenmodells ausgelegt. Um dem System eine weitere Reichweitenstufe hinzuzufügen, muß

das Ordnungsprädikat `reach-includes-p` erweitert und eine neue Methode für `v1-contra` erstellt werden.

4.6.2 Definition von Stimmführungsregeln

Die Klasse `v1-rule` für die Stimmführungsregeln ist von der Regelklasse `rule` abgeleitet (vgl. Abb. 4.6). Außerdem wird ein Regeltyp `:v1` für das Definitionsmakro `def-rule` eingeführt. Die `parse-rule`-Methode der Klasse `v1-rule` belegt die Slots des Regelobjekts (Syntax siehe Kapitel 3.7.2):

casedescr Die Fallbeschreibung besteht aus Zeilen. Die zulässigen Optionen werden im folgenden genauer beschrieben.

conseq Die Folgerung ist ein konstanter boolescher Wert, denn der in der Regel beschriebene Fall ist entweder erlaubt oder verboten.

reach Die Reichweite der Regel gibt an, welche Stimmführungsquadrupel der Untersuchung durch die Regel unterzogen werden.

Im Regelsatz ist für die Baßstimme eine Folge von Stimmführungsregeln angegeben. Die Stimmführungsprüfung wird häufig ausgeführt, so daß auf Effizienz geachtet wird. Der Code für die Regeln wird zu einer Funktion zusammengesetzt, die dann von Lisp kompiliert wird. Der Code für den Rumpf der Stimmführungsfunktion wird folgendermaßen aus der Folge der Stimmführungsregeln erzeugt:

- Im Code ist das Stimmführungsquadrupel über die Variable `v1` zugänglich.
- Aus jeder Stimmführungsregel wird eine Zeile des Verzweigungsstrukts `cond`, wobei die Reihenfolge der Regeln beibehalten wird.
- Aus Reichweite und Fallbeschreibung wird der Code für die Bedingung des `cond`-Zweigs erstellt.
- Der Code für die Reichweite ist ein Vergleich der Reichweiten von Stimmführungsquadrupel und Regel mittels `reach-includes-p`.
- Als Rückgabewert des Zweigs wird der Folgerungswert der Regel eingesetzt.
- Falls keine Bedingung einer Regel zutrifft, wird der Wert `:nicht-verboten` zurückgegeben. Dieser wird von Lisp als wahr interpretiert. Die Unterscheidung dient nur zur Benutzerinformation und zur Erleichterung der Fehlersuche.

Es wird also die Konsequenz der ersten zutreffenden Voraussetzung zurückgegeben.¹⁰ Dies erlaubt es, in einer Regelfolge Ausnahmen von Standardregeln zu definieren. Die Regel für die Ausnahme muß dabei vor der Regel für den Standardfall stehen.

Die Fallbeschreibung besteht aus mehreren Zeilen der drei Zeilentypen `:intseq`, `:bew` und `:expr`. Für jede Zeile wird der entsprechende Code erzeugt und mit *und* verbunden.

Die drei Zeilentypen werden folgendermaßen in Code umgesetzt:

`:expr <ausdr>` Der Ausdruck aus der Regel wird als Zeilencode zurückgegeben.

`:bew <wert>` Zeilencode ist ein Vergleich des Bewegungstyps des Stimmführungsquadrupels mit dem angegebenen Wert.

`:intseq <intspec> <intspec>` Die zwei Intervallspezifikationen müssen zu den beiden zusammenklingenden Intervallen des Stimmführungsquadrupels passen. Der Vergleichscode wird mit *und* verbunden.

Der Vergleich zwischen einer Intervallspezifikation aus der Regeldefinition und dem betreffenden Intervall des Stimmführungsquadrupels ist folgendermaßen realisiert:

- Ist die Intervallspezifikation ein Intervallobjekt, muß es gleich dem aktuellen Intervall sein.
- Ist die Intervallspezifikation eine Liste aus `:cls` und einem Intervallobjekt, müssen die beiden oktavbereinigt gleich sein, beispielsweise ist eine Dezime und eine Terz oktavbereinigt gleich.
- Ist die Intervallspezifikation ein Schlüsselwort, so muß sie in der Ausgabe von `codis-int` des Intervalls enthalten sein.
- Ist das Intervall eine Liste von Schlüsselwörtern, so muß sie Teilmenge der Ausgabe von `codis-int` des Intervalls sein.

¹⁰Die übliche Semantik des Verzweigungskonstrukts `cond`

Beispiel: Umsetzung einer Regeldefinition in Code

Die Regel für die verdeckte Oktavparallele sieht in der Regelsprache folgendermaßen aus:

```
(def-rule verd-oktaven :vl verboten
  :reach :direct                ;; Reichweite
  :casedescr
  (:bew :gerade)                ;; Bewegung
  (:expr (and (eql (vname1 vl) :melody) ;; Aussenstimmen
              (eql (vname2 vl) :bass)))
  (:expr (>= (szahl (hi-int vl)) 2)) ;; Oberstimme Terz aufwaerts
  (:intseq (:kon :unvollkom)      ;; Zweiklangfolge
           (:cls $(cintv :rn oktave :ab))))
```

Für eine Regelfolge, die nur diese Regeldefinition enthält, wird folgender Code erzeugt:

```
(lambda (vl)
  (cond ((and (reach-include-p :direct (reach vl)) ;; Reichweite
             (eql (bewtype vl) :gerade)           ;; Bewegung
             (and (eql (vname1 vl) :melody)      ;; Aussenstimmen
                  (eql (vname2 vl) :bass)))
             (>= (szahl (hi-int vl)) 2)           ;; Oberst. aufw.
             (and (subsetp '(:kon :unvollkom)     ;; unvollk. Konson.
                      (codis-int (ha-int1 vl)))
                  (int= $(cintv :rn oktave :ab)   ;; zur Oktave
                        (oktber (ha-int2 vl) nil))))
        nil)
    (t :not-forbidden)))
```

Die Voraussetzung besteht aus der Überprüfung der Bewegung, einer Untersuchung der Konsonanzeinstufung des ersten Intervalls und einem oktavbereinigten Vergleich des zweiten Intervalls.

4.7 Endmuster und Pattern matching

Bei der Stimmengenerierung wird nach jedem Zustandsübergang das Zielprädikat `goal-p` gerufen. Stellt dieses fest, daß der aktuelle Satz die geforderte Länge erreicht hat, dann ruft es die Endformelprüfung `end-ok-p`. Das Ergebnis wird im Zustand gecacht, weil es mehrfach benötigt wird. Die Endformelprüfung `matcht` das Ende des aktuellen Satzes mit der entsprechenden Stimme der Endformel. Bei der Komposition der Melodie stehen verschiedene Endformeln zur Verfügung, die den richtigen Typ (siehe Makrostruktur)

haben und zur Harmonie passen. Der Baß muß dann die von der Melodie festgelegte Endformel benutzen.

4.7.1 Definition von Endformeln

Eine Endformel ist ein Objekt der Klasse `endformula` und wird mit dem Definitionsmakro `def-endformula` definiert. Aufgabe dieser Klasse ist es, die Endformeln zu speichern, zu verwalten und der Stimmengenerierung die Endmuster für die aktuelle Stimme zur Verfügung zu stellen.

Syntax von `def-endformula`.

```
(def-endformula <name> (<eftype> <spec> ...)
  <harm>
  (<voicename> <voicepat>)
  (<voicename> <voicepat>)
  ...)
```

Das Endmuster wird unter dem eindeutigen Namen `<name>` in einer globalen Liste aller Endmuster abgelegt. Es wird über den Typ `<eftype>` und die folgenden weiteren Angaben `<spec>` gesucht. Der Typ ist entweder `:grundabsatz`, `:quintabsatz` oder `:kadenz`. Die Angaben in `<spec>` sind Schlüsselwörter, die z.B. angeben, daß die Endformel für Dreiklangsmelodien zu verwenden ist.

4.7.2 Matching-Verfahren

Für die Menuettkomposition wird ein Patternmatcher für Musikdaten verwendet, der im Rahmen der Diplomarbeit von Ina Becker [Bec00] entstand. Er ist Teil eines Systems zur Suche nach Mustern in tonaler Musik. Das System besteht aus einer Komponente, die MIDI-Dateien in COMF umwandelt, dem Suchalgorithmus, der das Muster an verschiedenen Stellen der Komposition anlegt und dem Patternmatcher, der entscheidet, ob das Muster an der aktuellen Position auf die Komposition paßt oder nicht.

Der Patternmatcher geht nach dem Grundalgorithmus des Patternmatching auf geschachtelten Ausdrücken (z.B. Listen) mit Variablenbindung nach [Nor92] vor. Beim Patternmatcher für Musikdaten sind sowohl Daten als auch Muster CLOS-Objekte.

Das Matching wird von der generischen Funktion `match` vorgenommen. Sie erhält als verpflichtende Parameter das Datenobjekt `obj` und das Muster `pat` und als optionalen Parameter die Bindungsliste `bind`. Die generische Funktion besteht aus Multimethoden, die auf die beiden verpflichtenden Parameter spezialisiert sind. Die verschiedenen Fälle beim Patternmatching -

mögliche passende Kombinationen von Daten und Muster - werden jeweils von einer Methode für eine Kombination von Klassen behandelt. Für jedes Paar von Objekt- und Musterklasse, bei dem ein erfolgreicher Match möglich ist, wird eine entsprechende Match-Methode definiert. Ergebnis des Aufrufs der generischen Funktion `match` ist eine Bindungsliste. Im Erfolgsfall enthält sie Paare von Variablenbindungen oder $(\$(\text{vare1 NIL}))$ ¹¹, wenn keine Variablenbindungen nötig waren. Bei einem Mißerfolg wird als Bindungsliste `NIL` zurückgegeben.

Jeder Methodenaufruf von `match` erhält die augenblickliche Bindungsliste als Parameter und gibt die aktualisierte Bindungsliste zurück, wobei die Aktualisierung nichtdestruktiv vorgenommen wird. Vorteil gegenüber einer globalen Bindungsliste ist, daß Kombinationen von Mustern mit *oder* leicht zu realisieren sind. Solche Muster sind speziell für Endmuster nützlich.

Die generische Funktion `match` hat eine Default-Methode, die `NIL` zurückgibt. Ein Datenobjekt und ein Muster können nur zusammenpassen, wenn für diese Kombination von Klassen auch eine `match`-Methode definiert ist. Eine Kombination unbekannter Klassen `matcht` nicht.

4.7.3 Einsatz des Patternmatchers für das Matching von Endformeln

Beim Matching von Endformeln kann im Gegensatz zu der in [Bec00] beschriebenen Suche nach Mustern in Musikdaten auf die Durchwanderung der Komposition verzichtet werden, da die Stelle, an der das Muster an die Komposition angelegt wird, bekannt ist - nämlich das Ende des aktuellen Satzes in der aktuellen Stimme.

Von den Musterklassen des Patternmatchers wird nur ein Teil benötigt. Beispielsweise werden die Klassen für mehrstimmige Muster nicht benötigt, da die Menuettkomposition stimmenweise vorgeht.

Andererseits werden dem Patternmatcher aus [Bec00] für den Einsatz in der Menuettkomposition ein paar Klassen mit entsprechenden Matchmethoden hinzugefügt, die notwendige, kleine Erweiterungen realisieren. Das Muster für die Endformel besteht aus Stufen. Zusätzlich sind stellenweise durch Verbindungselemente (`:overlapping-Elemente`) für die Intervalle zwischen den Stufentönen noch Beschränkungen angegeben.

¹¹Ein Wert, der angibt, daß keine Variablenbindung stattfand, der aber als wahr interpretiert wird, da $\neq \text{NIL}$.

Um das Matching mit solchen Endformeln zuzulassen, wurde der Pattermatcher aus [Bec00] folgendermaßen erweitert:

- Die Eingabesyntax für das Verbindungselement **betweene1** wurde erweitert, so daß im Verbindungselement statt einzelnen Intervallen auch Ausdrücke mit **:or** stehen dürfen.
- Die Klasse **betwel-t** ist ein Verbindungselement, das mit jedem Datenelement **matcht**.
- Wenn untersucht wird, ob das Ende der generierten Stimme der Endformel genügt, werden für das direkte Matching die Stufen und für das Bestimmen der Intervalle die absoluten Noten benötigt. Die Stufen müssen für die gesamte Endformel relativ zur Zieltonart der Endformel bestimmt werden, auch wenn diese von der in der Harmoniebasis angegebenen augenblicklichen Tonart abweicht. Dieser Fall kann beispielsweise eintreten, wenn die Ausweichung, die zur Zieltonart der Endformel hinführt, erst nach Beginn der Endformel vollzogen wird. Man kann also nicht einfach bei Bedarf aus der Note die Stufe bestimmen, da dann über die augenblickliche Tonart in der Chart zugegriffen wird.

Deswegen wird das Melodieende vor dem Matching in eine Folge von **keyed-dnote** relativ zur Zieltonart der Endformel überführt. Die Klasse **keyed-dnote** verbindet Ton und Stufe. Sie erbt von der Stufe **degr** und von der Note **note**. Die Matchmethoden sorgen dafür, daß ein Objekt dieser Klasse sich beim Matching gegen Stufenmuster wie eine Stufe verhält. Ansonsten wird es automatisch wie eine Note behandelt.

Kapitel 5

Ergebnisse

Dieses Kapitel stellt verschiedene Menuette vor, die das System komponiert hat. Die Beispiele im ersten Abschnitt sind Kompositionen mit unterschiedlichen strukturellen Beziehungen in der Melodie. In Kapitel 5.2 werden die Lösungen in der Lösungsmenge eines Kompositionslaufs untersucht und es wird betrachtet, wie stark ihre Qualität schwankt.

Interessant ist es, zu beobachten, welche Auswirkungen es hat, wenn man einzelne Wissens Elemente (z.B. Regeln oder strukturelle Beziehungen) hinzunimmt oder wegläßt. Beispiele dafür findet man in Kapitel 5.3. Der modulare Aufbau des Systems wird ausgenutzt, um zu einer Originalmelodie aus der Frühklassik einen Baß zu komponieren (Kapitel 5.4).

Abschließend zeigt Kapitel 5.5 mittels Statistiken, wie für unterschiedliche Regelsätze der Suchraum bei der Komposition eines viertaktigen Satzes aussieht.

5.1 Menuette mit strukturellen Beziehungen in der Melodie

Die Musikbeispiele in diesem Abschnitt sind typisch für die Ergebnisse des Menuettkompositionssystems. Als Makrostruktur wird die in Kap. 2.2 beschriebene prototypische Form verwendet. Das Menuett besteht aus vier viertaktigen Sätzen mit der Tonordnung (Folge der Endformeln) Grundabsatz - Quintkadenz - Grundabsatz oder Quintabsatz - Grundkadenz. Vom Melodietypus her werden dreiklangs- und skalenbasierte Abschnitte gemischt. Sofern nichts anderes angegeben ist, wird im dritten Vierer eine Fonte-Sequenz gesetzt.

Die Einhaltung der in Kap. 3.6 beschriebenen harmonischen und melodischen Regeln, die die Größe der zulässigen Intervalle beschränken, sind für

die stilistische Korrektheit wichtig. Verstöße gegen diese Regeln werden als deutlich lokalisierbare Fehler empfunden.

Strukturelle Beziehungen (Korrespondenzen, vgl. Kap. 3.5.4) im Stück sorgen für eine Verbesserung der stilistischen Qualität. Fehlen diese Verwandtschaften völlig, so wird das Stück als ziellos oder unsystematisch empfunden, ohne daß ein einzelner Kompositionsfehler lokalisiert werden kann. Solche Korrespondenzen sind durch spezielle Strukturelemente repräsentiert (s. Kap. 3.5.4). Das Strukturelement fordert, daß der von ihm regierte Teil des Stücks zu einem Modell ähnlich ist. Das Modell ist entweder ein vorangegangener Teil des Stücks (internes Modell) oder vorgegebenes Notenmaterial (externes Modell).

Wie stark eine Komposition strukturell bestimmt hat, hängt von zwei Faktoren ab:

- Die von Korrespondenzen regierten Bereiche der Komposition.

Damit die Korrespondenz zur Makrostruktur paßt, korrespondieren bei den folgenden Beispielen entweder Takte innerhalb eines Vierers oder Takte in verschiedenen Vierern, die an gleicher Position innerhalb des Vierers stehen. Beispielsweise sind Takt 2 und Takt 6 in ihrem jeweiligen Vierer der zweite Takt.

Die Endformeln sind ohnehin stark festgelegt. Daher werden die Bereiche der Endformeln bei den meisten der folgenden Beispiele nicht von Korrespondenzen regiert (Ausnahme: Fonte-Sequenz).

- Das Vergleichsprädikat für die Ähnlichkeit, das von den Korrespondenzen verwendet wird.

Es sind folgende Vergleichsprädikate definiert (vgl. Kap. 3.5.4):

Dauerngleichheit ist die schwächste Form der Korrespondenz. Die anderen Korrespondenzen schließen die Dauerngleichheit ein, da über Metrum und Dauer die korrespondierenden Noten gesucht werden.

Intervallrichtungsgleichheit Intervalle¹ gleicher Richtung korrespondieren.

Intervallrichtungsumkehrung Intervalle mit verschiedener Richtung korrespondieren. Intervalle mit gleicher Richtung korrespondieren nicht.

Intervallähnlichkeit Intervalle, deren Größe maximal um 1 abweicht, korrespondieren.

¹Hier werden immer tonale Intervalle (s. Kap.3.2.3) betrachtet.

Diese Korrespondenz erzeugt dann die gleiche Grundbewegung der Melodie wie im Modell. Bei dreiklangs-basierten Melodieabschnitten entsteht die gleiche Bewegung im gebrochenen Akkord (Dreiklangs-analogon). Auch wenn die Intervalle leicht voneinander abweichen, bleibt die Schrittzahl innerhalb der Akkordtöne gleich. Diese Verwandtschaft ist in der Klassik sehr verbreitet.

Ähnlichkeitsumkehrung Es wird ähnlicher Betrag (Abweichung maximal 1) des Intervalls, aber eine umgekehrte Richtung gefordert.

Intervallgleichheit Gleiche Intervalle korrespondieren.

Umkehrung Intervalle mit gleichem Betrag und entgegengesetzter Richtung korrespondieren.

Gelegentlich entstehen bei der Komposition durch den Computer auch zufällig solche Verwandtschaften, ohne daß Korrespondenzelemente dafür verantwortlich sind. Bei den Notenbeispielen wird die Korrespondenzstruktur angegeben, die dem Kompositionslauf vorgegeben ist und nicht die, die zufällig entstanden ist.

Es werden nun vom System komponierte Menuette mit unterschiedlichen strukturellen Vorgaben vorgestellt:

Notenbeispiel 1 Dauerngleichheit

Korrespondenzstruktur:

- Der erste Takt dient als (internes) rhythmisches Modell für das sämtliche Takte des Menuetts außerhalb der Endformeln. Die Tonhöhen werden nicht durch Korrespondenzen bestimmt.
- Im dritten Vierer ist eine Fontesequenz

In den Takten 3 und 14 treten von Schlag 2 zum Schlag 1 der Folgetakts Akzentparallelen auf, die vom Reichweitenmodell nicht erfaßt werden. Sie sind in dieser speziellen Situation leicht hörbar, obwohl ein ganzer Schlag dazwischen ist, da der Baßton den ganzen Takt lang der selbe ist.

Notenbeispiel 2 Dauerngleichheit mit Monte

Korrespondenzstruktur:

- Der erste Takt dient als rhythmisches Modell für alle Takte außerhalb der Endformeln.
- Der dritte Vierer enthält im Gegensatz zum vorangegangenen Beispiel eine Montesequenz. Diese führt zu einem Quintabsatz.

Notenbeispiel 3 Intervallähnlichkeiten

Korrespondenzstruktur:

- Die Takte 5 und 13 sind zu Takt 1 intervallähnlich.
- Die Takte 6 und 14 sind zu Takt 2 dauerngleich.
- Takt 14 ist zu Takt 6 intervallrichtungsgleich.

Außer den rhythmischen Korrespondenzen werden hier noch intervallische Verwandtschaften gefordert. Zwischen den Takten 6 und 14 wird also eine Verwandtschaft zwischen einem skalenbasierten und einem akkordbasierten Melodieteil aufgebaut.

Notenbeispiel 4 Menuett mit Umkehrung

Korrespondenzstruktur:

- Takt 3 und 5 haben eine Korrespondenz mit Intervallähnlichkeit zu Takt 1.
- Takt 6 und 14 haben eine Korrespondenz mit Dauerngleichheit zu Takt 2.
- Takt 13 hat eine Korrespondenz mit umgekehrter Intervallähnlichkeit zu Takt 1.
- Im dritten Vierer ist eine Fontesequenz.

Da die Intervallähnlichkeiten auf einen dreiklangs-basierten Bereich fallen, ist die Verwandtschaft zwischen den Teilen das Dreiklangs-analogon (gleiche Zahl von Schritten im Akkord). Ebenso betrifft die Umkehrung den dreiklangs-basierten Bereich. Damit sind die ersten beiden Takte ein Modell für das gesamte Stück außerhalb des dritten Vierers und der Endformeln. Der dritte Vierer enthält die übliche Fontesequenz.

In Takt 7 auf Schlag 1 findet man einen Quartvorhalt, der auf Schlag 2 aufgelöst wird.

Notenbeispiel 5 Menuett mit Viertelrepetitionstakt

Korrespondenzstruktur:

- Die ersten Takte des zweiten und vierten Vierers korrespondieren durch Intervallähnlichkeit mit Takt 1
- Die zweiten Takte des zweiten² und vierten Vierers korrespondieren rhythmisch mit Takt 2.
- Takt 3 korrespondiert rhythmisch mit Takt 1.
- Im dritten Vierer ist eine Fontesequenz.

Normalerweise ist bei der Melodiebildung die Verwendung von Repetitionen beschränkt (Regeln RN8, RN20 und RN21). Als Ausnahme von diesen Beschränkungen wird hier eine bei Menuetten gelegentlich anzutreffende Figur, der Viertelrepetitionstakt, zugelassen. Dieser ist ein Takt mit drei Viertelnoten gleicher Tonhöhe (RN38 und RN39). Er wird durch ein bottom-up-Strukturelement realisiert, d.h. die Entscheidung, ob ein Viertelrepetitionstakt gesetzt wird, fällt erst bei der Komposition der Melodie. Dieses Menuett hat in Takt 6 einen Viertelrepetitionstakt. Der Viertelrepetitionstakt überdeckt auch die eigentlich vorgesehene rhythmische Korrespondenz.

In Takt 6 überlagern sich zwei bottom-up-Strukturelemente auf ungünstige Art. Auf den Schlägen 1 und 2 ist die Harmonie A-Dur (5. Stufe zur Ausweichung³ nach D-Dur) gefordert, daher auch das a im Baß. Das d'' in der Melodie stammt aus dem Versuch, einen Quartvorhalt zu setzen. Die korrekte Auflösung zu cis'' wird durch den Viertelrepetitionstakt unterdrückt.

Bei den strukturarmen Menuetten in Kap. 5.3.1 macht sich der Viertelrepetitionstakt positiv bemerkbar. Hier gibt es hingegen Unverträglichkeiten zu anderen Strukturelementen. Solche Probleme sind typisch für die Integration unterschiedlicher Wissensselemente.

²Wird vom Repetitionstakt überdeckt, s.u.

³Ausweichung: zeitgenössischer Begriff [Koc82]; modern: Modulation, vgl. Kap. 2.2

Notenbeispiel 6 Menuett mit externem Rhythmusmodell



Korrespondenzstruktur:

- Alle Takte außerhalb der Endformeln korrespondieren mit einem externen Rhythmusmodell: zwei Achtel gefolgt von zwei Vierteln⁴.
- Ansonsten gibt es keine Korrespondenzen - auch keine Fonte- oder Montesequenz im dritten Vierer.

Dieses Beispiel verwendet für den dritten Vierer ein anderes Harmoniemodell als für die anderen Beispiele und läßt auch die Fontesequenz weg.

5.2 Untersuchung einer Lösungsmenge

Bei einem wissensbasierten System darf ein korrektes Ergebnis kein Zufallsprodukt sein. Vielmehr muß es zuverlässig als Folge von korrekten Inferenzen auf dem angemessen modellierten Wissen entstehen. Deswegen ist zu erwarten, daß bei einem Menuettkompositionslauf alle verschiedenen Lösungen in der Ergebnismenge stilistisch korrekt sind und in der Qualität nicht zu stark voneinander abweichen.

Dieses System steht damit im Kontrast zu Systemen mit stärkerer künstlerischer Ausrichtung, wie z.B. das System SARA von David Cope [Cop95] (vgl. Kap 2.3.5). Bei SARA ist ein starker Qualitätsunterschied zwischen den verschiedenen Lösungen eines Laufs mit einem Parametersatz zu verzeichnen. David Cope, der Autor des Systems SARA, sucht unter einer großen Zahl von Ergebnissen die geeigneten Lösungen von Hand aus.

Im folgenden wird die komplette Lösungsmenge eines Programmlaufs des Menuettkompositionssystems abgedruckt.

⁴Aus Mozart KV2, Abb. 2.2

Die Lösungsmenge wurde mit einem Kompositionsplan erzeugt, der für Komposition mit strengen strukturellen Vorgaben günstig ist. Interne strukturelle Beziehungen zum ersten Satz schränken den Lösungsraum für den zweiten und die folgenden Sätze ein. Damit besteht auch die Gefahr, daß eine Lösung für den ersten Satz überhaupt nicht mit einem zweiten fortgesetzt werden kann. Es ist daher günstig, für den ersten Satz viele verschiedene Alternativen zu erzeugen, jedoch für die folgenden Sätze nur wenige (vgl. Kap 3.3).

Notenbeispiel 7 Ergebnismenge mit Strukturvorgabe

Korrespondenzstruktur:

- Takt 3, 5 und 13 sind intervallrichtungsgleich zu Takt 1
- Takt 6 und 14 sind dauerngleich zu Takt 2
- Im dritten Vierer ist eine Fonte-Sequenz

Beim Baß des vierten Satzes gab es für ein Menuett keine Lösung, so daß die Ergebnismenge insgesamt nur 5 Lösungen enthält.

7/1

The musical score consists of two systems, each with a treble and bass staff. The key signature is one sharp (F#) and the time signature is 3/4. The first system contains measures 1 through 8, and the second system contains measures 9 through 16. The notation includes various rhythmic values such as quarter notes, eighth notes, and sixteenth notes, along with accidentals like sharps and naturals.

Die für Menuett 3 angegebene Verbesserung der Baßlinie in Takt 6 durch ein H ist hier auch denkbar. Das fis ist bei diesem Menuett aber besser motiviert als in Menuett 3, weil der Oktavsprung vorausgeht. Das fis in der Baßlinie ist ein Gegensprung (RN32) zum Oktavsprung und gleichzeitig in Gegenbewegung zur Melodie(RN33).

7/2

The musical score for 7/2 consists of two systems. The first system contains measures 1 through 8, and the second system contains measures 9 through 16. The music is written in treble and bass clefs with a key signature of one sharp (F#) and a 7/2 time signature. The melody in the treble clef is primarily eighth-note based, while the bass clef provides a steady accompaniment of quarter notes.

In Takt 6 Schlag 1 steht im Baß der Leitton cis', der die Ausweichung markiert. Dieser Leitton muß, da die Harmonie noch länger steht, nicht sofort aufgelöst werden (akkordische, abgeschwächte Leittonregel RN13). Eine Auflösung auf Distanz durch ein d' statt des tiefen d auf Schlag 3 würde die Baßlinie in Takt 6 verbessern. Das System hat das tiefe d bevorzugt, weil es in Gegenbewegung zur Melodie steht.

7/3

The musical score for 7/3 consists of two systems. The first system contains measures 1 through 8, and the second system contains measures 9 through 16. The music is written in treble and bass clefs with a key signature of one sharp (F#) and a 7/3 time signature. The melody in the treble clef is primarily eighth-note based, while the bass clef provides a steady accompaniment of quarter notes.

In Takt 6 Schlag 3 wäre es besser und harmonisch interessanter, im Baß H statt fis zu setzen. Der Sextsprung zum fis nach oben wirkt nach dem vorangehenden Schritt nach oben unmotiviert (aber nicht falsch). Diese Verbesserung greift aber in die Harmonik ein und ist mit den harmonischen Vorgaben des Kompositionslaufs nicht möglich. Die Parallelstelle in Ergebnis 1 ist durch die (lineare) Gegenbewegung besser.

An dieser Stelle würde ein erweitertes harmonisches Modell helfen, bei dem die Entscheidung zwischen der 1. Stufe D-Dur⁵ und seiner Mollparallele, der 6. Stufe h-moll, erst bei der Stimmengenerierung endgültig getroffen wird.

⁵Augenblickliche Tonart ist D-Dur, die Ausweichung ist bereits vollzogen.

7/4

Hier, wie auch in Beispiel 2 und 3 endet im dritten Vierer der Takt 10 im Baß nicht auf den Akkordgrundton a, sondern auf die Terz c. Außerhalb von Endformeln meidet das System vollkommene Konsonanzen zwischen Melodie und Baß da sie im zweistimmigen Satz leer klingen (RN34). Dies führt dazu, daß in Takt 10 der Baß den Harmoniegrundton a vermeidet, weil er schon in der Melodie vorkommt. In dieser Situation wäre es aber schön, die Fonte-Sequenz folgerichtig durchzuziehen, d.h. auch den Baß zu sequenzieren. Die Verwandtschaft der Baßline in Takt 10 mit der Endformel in Takt 12 hört sich dabei gut an, da die Dominante-Tonika Beziehung von Takt 9 auf Takt 10 auch einen gewissen Endcharakter hat. Die vollkommenen Konsonanzen sind hier das kleinere Übel.

Zur Verbesserung könnte man bei den Fontesequenzen an der Parallelstelle zur Endformel den Regelsatz für den Baß durch Hüllregeln auf die in den Endformeln geltende Regeln umschalten.

7/5

Wie beim Menuett 2 wäre hier der Fernleitton im Takt 6 des Basses wünschenswert. Gut ist, daß in Takt 10, an der Stelle, die mit der Endformel korrespondieren könnte (s. o.), der Baß immerhin den Grundton A erreicht. Dies ist auf Schlag 3 erlaubt, weil die schwache

Zusammenklangsregel die unvollkommene Konsonanz nur auf Schlag 1 fordert.

Insgesamt hat bei diesem Menuett im dominantische geprägten Takt 9 (E^7 vor a-moll) die Terz ein starkes Gewicht. Der Baß könnte in dieser Situation die akkordische Spannung des Dominantseptakkords stärker betonen, wenn er in Schlag 3 von Takt 9 im Baß die Septime d gegen die Terz gis' setzt.⁶ Dies würde auf eine für die Klassik besonders typische Weise die Dominantseptakkordspannung betonen. Dafür müssen aber zuerst Regeln zur korrekten Behandlung des Sekundakkords implementiert werden.

Vergleicht man die Ergebnisse, dann sind sie verhältnismäßig homogen. Bei den 5 Menuetten sind einzelne Stellen besser oder schlechter gelungen. Unterschiede im Gesamteindruck entstehen nur durch die Häufung etwas besserer oder schlechterer Partien, nicht durch grundsätzliche Abweichungen.

Am besten ist Nr. 1. Der Baß ist im dritten Vierer interessant belebt, bleibt aber in den anderen Vierern unauffällig und enthält sich aller Seltsamkeiten. Damit streicht er die Andersartigkeit des dritten Vierers heraus. Dafür ist die melodische Grundfigur in Takt 1 und 2, die ja durch die Korrespondenzen im ganzen Stück fortgesetzt wird, bei Nr. 5 etwas interessanter als bei Nr. 1. Bei Nr. 1 springt der erste Takt der Melodie nur zwischen zwei Noten hin- und her. Gut ist bei Nr. 5 auch der Takt mit Viertelnoten an zweiter Position in den Vierern (Takt 2,6,14) der dem Stück eine menuetttypische Bewegung gibt. Nr. 5 fehlt dagegen die (Fern-)Auflösung des Leittons cis' im Takt 6 des Basses.

5.3 Variation des Musikwissens

Die Ergebnisse in diesem Abschnitt zeigen, welchen Einfluß die Verwendung bzw. das Weglassen einzelner Wissens Elemente und die Variation von Parametern auf die Kompositionsergebnisse haben. Die Variation von Parametern führt teilweise zu schlechten oder sogar unstilistischen Ergebnissen. Unstilistische (eindeutig fehlerhafte) Ergebnisse sind durch * gekennzeichnet.

5.3.1 Menuettmelodien ohne Korrespondenzen

Die inneren Korrespondenzen (Verwandtschaften) sorgen für Ähnlichkeiten und Zusammenhalt des Stücks. Bei den folgenden Experimenten werden die-

⁶An der Parallelstelle in Takt 11 wäre ein analoges Vorgehen stilistisch denkbar, die geforderte Fortführung steht aber im Widerspruch zu den im System definierten Endformeln.

se Korrespondenzen abgeschaltet. So kann man die Auswirkung der Struktur auf den Gesamteindruck betrachten. Außerdem sieht man hier die Grundtypen der Melodie, die skalen- und die dreiklangsbasierte Melodie ohne den Einfluß von Strukturelementen. Zudem wurde hier die punktierte Viertel als Dauer freigegeben.

Bemerkenswert ist, daß auch bei diesen Beispielen an mehreren Stellen strukturelle Beziehungen und Urlinien hörbar sind. Sie sind durch Zufall, nicht durch Regeln, entstanden. Wiederholt man das Experiment mit den gleichen Einstellungen, dann kann man mit diesen Strukturen nicht mehr rechnen.

Notenbeispiel 8 Dreiklangsmelodie

The musical notation is presented in two systems. Each system contains a treble and a bass staff. The first system covers measures 1 through 8, and the second system covers measures 9 through 16. The key signature is one sharp (F#), and the time signature is 3/4. The melody in the treble staff is primarily based on triads, while the bass staff provides harmonic support with various chords and bass lines.

Dieses Beispiel ist außerhalb der Endformeln dreiklangsbasiert. Die Rhythmusverwandtschaft zwischen Takt 2 und Takt 3 ist Zufall. Insgesamt wirkt der erste Vierer wie erwartet etwas ziellos. Dazu trägt auch der Baß bei, der zu oft den Sextakkord wählt und in Takt 2 auf den Nachschlag von Schlag 1 und 2 trotzdem die leere Oktave hat.

Der zweite Vierer wirkt recht gut, da die Einsen der Takte in Gegenbewegung gesetzt sind, in der Melodielinie das die Modulation einleitende *cis* vorkommt und zudem noch diese Linie zum melodischen Höhepunkt am Beginn der Endformel in Takt 10 hinführt. Dies gleicht die fehlenden Beziehungen völlig aus.

Diese Linienführung ist nicht durch Regeln gefordert und ist Zufall.

Auch der dritte Vierer wirkt erstaunlich gut, da die ohnehin deutliche harmonische Spannung der *Fonte*-Sequenz durch die Linie der Einsen von Takt 9 und Takt 11 unterstützt wird.

Der vierte Vierer wirkt dagegen chaotisch, wie es bei diesem Experiment zu erwarten ist.

Notenbeispiel 9 Skalenbasierte Melodie

The musical score consists of two systems of two staves each. The first system covers measures 1 through 8, and the second system covers measures 9 through 16. The key signature is one sharp (F#) and the time signature is 3/4. The melody is primarily scalar, moving mostly in ascending and descending steps. The bass line provides a steady accompaniment with a mix of eighth and quarter notes.

Beim ersten Vierer fällt der etwas große Ambitus auf. Ursache ist Regel RN31, die Sekunden in gleicher Richtung bevorzugt. Im Rest des Stücks macht sich diese Regel aber positiv bemerkbar. Sie sorgt dafür, daß der dritte und vierte Vierer jeweils einen höchsten Punkt in der Mitte der Melodie haben.

Der Rhythmus ist gleichförmiger als der des letzten Musikbeispiels. Als Ursache kommt neben zufälligen Effekten die Abhängigkeit der Dauer der Durchgangsnote von der der Nachbarnoten (vgl. RN19) im Zusammenspiel mit dem Synkopenverbot in Frage. (Durchgänge kommen im Regelsatz für dreiklangs-basierte Melodien nicht vor.)

Zwischen Takt 11 Schlag 2 und Takt 12 Schlag 1 findet man eine leicht unschöne Akzentparallele, auf die keine der im System verwendeten Reichweitenstufen (vgl. Kap. 3.7.3) passt.

Notenbeispiel 10 Mischmelodie

The musical score consists of two systems of two staves each. The first system covers measures 1 through 8, and the second system covers measures 9 through 16. The key signature is one sharp (F#) and the time signature is 3/4. The melody is more varied than in the previous example, featuring some leaps and non-scalar passages. The bass line is more active, with many eighth notes and some triplets.

Diese Mischmelodie wirkt recht zusammenhanglos, insbesondere im dritten Vierer, wo auf dem üblichen Harmonischemata der Fonte-Sequenz keine der erwarteten Beziehungen zu hören sind.

Eigentlich sollte man bei der strukturlosen Mischmelodie erwarten, daß die Schwächen der Reinformen gemildert auftreten. Dies ist hier aber nicht der Fall. Bei den beiden Reinformen wirkten nämlich die klar skizzierte Harmonik der Dreiklangsmelodie einerseits und die langen Sekundenzüge der Skalenmelodie andererseits strukturierend. Bei der Mischform wird dagegen die Inhomogenität der Rhythmen und die Zusammenhanglosigkeit der Bewegungen hörbar.

5.3.2 Baßstimme

Nach Riepel [Rie52] ist der Ausgangspunkt für den Baß die Folge der Harmoniegrundtöne. Die anderen Harmonietöne können verwendet werden, um den Zusammenklang und die Linie zu verbessern.

Speziell für den zweistimmigen Satz fordert Riepel, daß der Baß - er bezeichnet dies als *jung* - durch Verwendung der anderen Harmonietöne eine gute Baßlinie und möglichst viele unvollkommene Konsonanzen zur Melodie erhält. Riepel warnt andererseits davor, dies zu weit zu treiben, den dann *verliere der Baß das Mannbare*. Im Bereich der Endformeln werden deswegen (von den Quartsextvorhalten abgesehen) Grundtöne verwendet.

Die folgenden Kompositionsbeispiele sollen illustrieren, wie sich der Baß verändert, wenn man aus der Regelfolge sukzessive die o.g. Regeln für die Verbesserung des Basses abschaltet.

Als erster Schritt wird der Baß ohne die Regeln zur Verbesserung der Baßlinie erstellt. Folgende Regeln wurden dafür deaktiviert: die Bevorzugung der Gegensprünge (in der Linie) RN32, die Bevorzugung der Gegenbewegung RN33, das Intervallsummenlimit RN9, die Bevorzugung kleiner Intervalle RN30 und das Verbot gleichzeitiger langer Noten RN26.

Notenbeispiel 11 Baß ohne Linienverbesserung

The image displays two systems of musical notation for a two-staff piece in 3/4 time with a key signature of one sharp (F#). The first system consists of two staves: the upper staff is in treble clef and the lower staff is in bass clef. The melody in the treble staff is composed of eighth and quarter notes, while the bass line in the bass staff consists of quarter notes. The second system, starting at measure 9, continues the piece with similar rhythmic patterns and note values in both staves.

Die befürchteten unmotivierten großen Sprünge treten tatsächlich auf. Besonders ungünstig ist die Folge zweier Sexten in Takt 9/10, insbesondere, weil sie zusammen mit einer großen harmonischen Spannung auftritt.

Notenbeispiel 12 Baß ohne Zusammenklangsregeln

Bei diesem nächsten Beispiel wird zudem noch auf die Regeln zur Verbesserung des Zusammenklangs verzichtet

Das Ergebnis enthält mehrfach Oktaven zwischen Melodie und Baß. Im zweistimmigen Satz klingen diese leer und unvollständig und entsprechen damit nicht dem geforderten Ideal eines „jungen“ Basses. Das Beispiel hinterläßt aber keinen direkt falschen Eindruck, schließlich wäre dieser Baß für einen vierstimmigen Satz, bei dem der Akkord durch die Mittelstimmen vervollständigt wird, durchaus geeignet.

Im zweiten Vierer findet man zwei unschöne Akzentparallelen, die der Stimmführungsprüfung durchgeschlüpft sind. Die erste von Takt 5 Schlag 3 zum Takt 6 Schlag 1 wird nicht erkannt, weil nicht direkt ist und der erste Zusammenklang nicht zusammen anschlägt. Die zweite (Takt 6 Schlag 1 zu Takt 6 Schlag 3) ist trotz des Abstands von 2 Schlägen hörbar, weil die trennende Note nur eine Achtelnote ist. Um diese beiden Fälle abzustellen, wäre ein verfeinertes Reichweitenmodell notwendig (vgl. Kap. 3.7.3).

Notenbeispiel 13 Baß aus Akkordgrundtönen

Dieses Beispiel geht einen Schritt weiter und simuliert Riepels Ausgangspunkt für die Baßkomposition, nämlich den Baß, der aus der Folge der Akkordgrundtöne besteht. Dadurch entstehen noch mehr vollkommene Konsonanzen zur Melodie, so daß der Satz unvollständig klingt.

Notenbeispiel 14 * Baß ohne Stimmführung

Bei diesem Beispiel wird die Stimmführungsprüfung abgeschaltet.

Hier finden sich einigen Stellen verbotene Parallelen, beispielweise direkte Parallelen an den Taktstrichen von Takt 3 auf Takt 4, von Takt 11 auf 12 und von Takt 13 auf 14. In Takt 14 findet man zwischen Schlag 2 und 3 Akzentparallelen. Man erkennt also, daß der Aufwand, den das Verfahren mit der Stimmführungsprüfung betreibt, wirklich notwendig ist.

5.3.3 Spezielle Melodien

Nach Riepel ist der Ausgangspunkt für den Menuettrhythmus ein Takt mit drei Viertelnoten. Es ist daher naheliegend, die Komposition eines Menuetts aus lauter Viertelnoten zu versuchen.

Notenbeispiel 15 Menuett aus Viertelnoten

Das Ergebnis ist ziemlich gut, ohne daß das Wissen speziell an die Einschränkungen angepaßt wurde. Das Experiment stützt damit die Auffassung, der Menuettrhythmus sei durch Verfeinerung und Variation aus einer Folge von Viertelnoten hergeleitet.

Im dritten Vierer wird die Melodie sequenziert. Bei diesem besonders einfachen Menuett wäre es besser, diese Beziehung auch im Baß zu haben (siehe Kap. 5.2). Nimmt man aber die entsprechenden Strukturelemente zu den Regeln für den Baß hinzu, dann gibt es keine Lösungen. Es ist nämlich eine Korrespondenz zwischen einem Endformelbereich und einem Nichtendformelbereich gefordert und die Regeln der beiden Bereiche widersprechen sich in manchem.

Notenbeispiel 16 Menuett mit Vierteln und Halben

Hier werden zusätzlich noch halbe Noten zugelassen. Auch dies ergibt gute Ergebnisse.

Etwas unschön ist der Baß in Takt 6 Schlag 3. Schön wäre hier das obere d', da dies eine Fernauflösung des Leittons cis' auf Schlag 1 wäre. Stattdessen hat sich wegen des Vorrangs der Gegenbewegung zwischen Stimmen das untere d durchgesetzt.

Die Endformeln schränken die Zahl der Alternativen für die von ihnen regierten Bereiche stark ein. Es ist daher interessant, zu untersuchen, ob die Prüfung der Endformeln wirklich nötig ist, oder ob die wesentlichen Aufgaben der Endformeln nicht von anderen Gesetzmäßigkeiten, beispielsweise von der Harmonie und von den melodischen Regeln, ohnehin erledigt werden.

Notenbeispiel 17 * Verzicht auf Endformeln

Die fehlende Schlußwirkung der Endformeln stört den Gesamteindruck de Stücks erheblich, so daß das Ergebnis unstilistisch wird.

Es ist also notwendig, die Endformeln gesondert zu modellieren und prüfen. Die anderen harmonischen Vorgaben und die Melodieregeln beschreiben die linearen Aspekte der Schlußwirkung eines Satzes nicht mit.

5.4 Baß zu einer zeitgenössischen Melodie

Beim Design des Systems wurde Wert darauf gelegt, daß die einzelnen Schritte des Kompositionsprozeß eigenständig ausführbar sind. Jeder Kompositionsschritt erstellt eine Stimme eines Satzes. Diese Möglichkeit wird nun ausgenutzt, um zu einer gegebenen Melodie einen Baß setzen zu lassen. Es wurde eine zeitgenössische Menuettmelodie, ein (positives) Lehrbeispiel aus Riepel [Rie52] ausgewählt.

Durch die Melodie ist implizit die Makrostruktur des Stücks und auch grob (nicht ganz eindeutig) die Harmoniefolge vorgegeben. Die Makrostruktur unserer Beispielmelodie entspricht der prototypischen Menuettstruktur (siehe Kap. 2.2). Die Grundtonart C-Dur der Originalmelodie wird für dieses Kompositionsbeispiel beibehalten, obwohl die anderen Beispiele in diesem Kapitel der Übersichtlichkeit halber einheitlich in G-Dur stehen.

Das System erhält als Eingabe einen Kompositionsplan für die vier Sätze des Basses, einen Regelsatz und eine Partiturokarte (Klasse *comp-chart*), die bereits die Melodie und eine durch Analyse der Melodie von Hand erstellte Strukturbeschreibung mit Makrostruktur und Harmoniefolge enthält (vgl. Kap. 3.3.1). Bei der Erstellung der Strukturbeschreibung kann man hier auf die Modellierung der strukturellen Eigenschaften, die nur von der Melodie verwendet werden, verzichten. Die meisten Teile der Mikrostruktur betreffen nur die Melodie, so daß die Strukturbeschreibung im Wesentlichen aus der Makrostruktur mit der Takt- und Tonordnung besteht.

Notenbeispiel 18 Baß zu gegebener Melodie

The musical score for Example 18 is written in 3/4 time. It consists of two systems of two staves each. The top staff of each system is in the treble clef and contains the melody. The bottom staff is in the bass clef and contains the bass line. The melody starts with a quarter note G4, followed by quarter notes A4, B4, and C5. The bass line starts with a dotted half note G2. The score ends with a final cadence in the melody and a whole note G2 in the bass line.

Das erste Beispiel fordert nach der strengeren Fassung der Zusammenklangsregeln unvollkommene Konsonanzen auf jeden Anschlag des Basses (außerhalb von Endformeln und Satzanfängen).

Notenbeispiel 19 Baß zu gegeb. Melodie, unvollk. Konson. auf 1

The musical score for Example 19 is identical to Example 18, showing a melody and a bass line in 3/4 time. It consists of two systems of two staves each. The top staff of each system is in the treble clef and contains the melody. The bottom staff is in the bass clef and contains the bass line. The melody starts with a quarter note G4, followed by quarter notes A4, B4, and C5. The bass line starts with a dotted half note G2. The score ends with a final cadence in the melody and a whole note G2 in the bass line.

Bei diesem Beispiel werden dem Baß mehr Freiheiten gegeben, indem die unvollkommenen Konsonanzen nur auf der Eins der Takte gefordert werden. Der Baß wird dadurch grundtöniger und so vom Zusammenklang im zweistimmigen Satz etwas schlechter. Da so aber mehr Alternativen entstehen, können die Regeln zur Verbesserung der Baßlinie besser greifen und die Baßlinie wird besser.

Das Kompositionssystem prüft nicht auf Parallelen zwischen zwei Sätzen, da diese wegen der Schlußwirkung und dem Neuanfang kaum wahrgenommen wird. Der dritte Vierer von Riepels Melodie hat eine melodisch schwache Endformel. Deswegen wird die Parallele hier als leicht störend wahrgenommen.

Im Gesamteindruck sind beide Beispiele - wie erwartet - sehr gut, da für die Melodie Originalmaterial verwendet wird, das zudem noch von Riepel als menuetttypisch deklariert wurde. Nach Riepel ist gegenüber der Melodie die Linienführung der anderen Stimmen untergeordnet. Damit ist für den Gesamteindruck, den das Menuett hinterläßt, die Qualität der Melodie maßgeblich, sofern der Baß einigermaßen korrekt ist. Nur grobe Satzfehler oder gravierende Probleme im Baß stören den Gesamteindruck.

5.5 Statistiken zum Suchvorgang

Aus der Sicht des Informatikers sind besonders die Eigenschaften des Suchraums von Interesse. Die Struktur des Suchraums bei der Stimmengenerierung hängt von der verwendeten Regelfolge, den Parametern und strukturellen Vorgaben ab. Da der Suchvorgang wegen seiner Größe nicht zu überblicken ist, wird nun versucht, ihn mittels Kenngrößen zu beschreiben.

Diese Kenngrößen sind teils absolute Werte, die als Maß für die Größe des Suchvorgangs dienen und teils Durchschnitte und Quotienten, die von der absoluten Größe des Suchvorgangs abstrahieren. Alle Werte der in diesem Abschnitt beschriebenen Statistiken entstanden bei der Komposition eines 4-taktigen Satzes einer Stimme mit dem Zufallsschußverfahren.

Die folgenden Werte werden angegeben:

- (A) Lösungszahl** *Absolutwert* Das Element des Kompositionsplans gibt eine gewünschte Gesamtzahl von Lösungen vor. Hier wird die wirklich erzeugte Lösungszahl angegeben. Sie ist niedriger als die geforderte Lösungszahl, wenn innerhalb der vorgegebenen Maximalzahl von Versuchen weniger Lösungen erzielt werden.

- (B) **Schußzahl** *Absolutwert* Die Schußzahl ist die Gesamtzahl der Versuche, die nötig waren, um die gewünschte Zahl von Lösungen zu erhalten. Sie enthält erfolgreiche Versuche genauso wie erfolglose.
- (C) **Schüsse pro Lösung** *Quotient* Die Anzahl der Schüsse pro Lösung beschreibt die Dichte der Lösungen im Suchraum. Je mehr Schüsse pro Lösung benötigt werden, desto dünner sind die Lösungen gesät.
- (D) **Lösungslänge** *Durchschnittswert* Die durchschnittliche Länge der Lösungen gibt die Anzahl der Zustände und damit die Zahl der Noten in einer Lösung an. Vergleicht man die Komposition mit verschiedenen Regelsätzen bei gleicher metrischer Länge, dann kann man an der Notenzahl ablesen, ob der Regelsatz eher zu kleinen oder zu großen Notenwerten hin tendiert.
- (E) **Schußlänge** *Durchschnittswert* Die Schußlänge gibt die durchschnittliche Zahl der Zustandsübergänge in einem Zufallsschuß (sei er nun erfolgreich oder erfolglos) an. An der Schußlänge kann man erkennen, ob bei dem betreffenden Regelsatz Fehlschläge eher frühzeitig oder eher spät erkannt werden.
- (F) **Zustandsübergangszahl** *Absolutwert* Die Zustandsübergangszahl ist die Zahl der erzeugten Zustände. Ein Zustandsübergang kann als Grundoperation der Suche mit konstantem Aufwand gesehen werden. Die Anzahl der erzeugten Zustände ist daher ein Maß für die absolute Größe des Suchvorgangs und damit für den Aufwand der Suche. Eine hohe Zustandszahl kommt in erster Linie durch eine hohe Zahl an Versuchen, in zweiter aber durch eine große Schußlänge - oft durch ein spätes Entdecken von Fehlschlägen verursacht - zustande.
- (G) **Verzweigungsgrad** *Durchschnittswert* Der Verzweigungsgrad gibt die durchschnittliche Zahl an Alternativen (Folgezustände eines Zustands) an. Am Verzweigungsgrad kann man ablesen, wie restriktiv ein Regelsatz ist.
- (H) **Zustandsausnutzung** *Quotient* Die Zustandsausnutzung (in Promille) ist das Verhältnis der Anzahl an Zuständen, die im Weg zu einer Lösung enthalten sind, zur Gesamtzahl der Zustandsübergänge (F). Je größer dieser Quotient ist, desto zielstrebig ist die Suche, da weniger vergebliche Berechnungen vorgenommen werden. Ein deterministisches Verfahren hat eine Zustandsausnutzung von 1 (1000 Promille).

(I) Laufzeit *Absolutwert* Die Laufzeit wird in CPU-Sekunden angegeben.

Die bisherigen Kenngrößen betrachten nur den Suchvorgang, abstrahieren jedoch vom Aufwand bei der Bestimmung der Alternativenmenge für einen einzelnen Zustandsübergang. Ziel der Laufzeitmessung ist es, den spezifischen Aufwand eines Zustandsübergangs mit zu erfassen.

Damit die Laufzeiten der verschiedenen Tests vergleichbar sind, wird versucht, Störeinflüsse so weit wie mit vertretbarem Aufwand möglich zu vermeiden. So werden für die Messungen folgende Ausgangsbedingungen gewählt:

- Für alle Messungen wird die gleiche Testplattform verwendet. Das Lispsystem ist Allegro Common Lisp 5.0.1 auf einer Sun SPARC Ultra 80 unter Solaris 5.7. Der Lispcompiler ist bei allen Versuchen gleich eingestellt (Sicherheit, Geschwindigkeit, Debugbarkeit, Speicherplatz).
- Es wird mittels der eingebauten Funktionen von Allegro CL die für den Prozeß aufgewendete CPU-Zeit gemessen (nicht die Realzeit), so daß das Vorhandensein anderer, konkurrierender Prozesse auf dem Rechner das Ergebnis nicht (bzw. nur geringfügig) beeinflussen sollte. Natürlich werden im Allegro-Prozeß keine weiteren Threads gestartet.
- Vor Beginn der Messung wird mit `(excl::gc t)` eine globale Garbage-Collection ausgelöst, so daß der Speicher zu Beginn jedes Testlaufs bereinigt ist und vorangegangene Berechnungen das Ergebnis nicht beeinflussen. Die Garbage-Collections, die dann während des Testlaufs nötig werden, gehören dann zu Recht zum Zeitaufwand des Verfahrens.

(J) Zeit pro Zustandsübergang *Quotient* Der Zeitbedarf pro Zustandsübergang ist der Quotient aus dem absoluten Zeitbedarf (I) und der Zustandsübergangszahl (F). Dieser Quotient ist ein Maß dafür, wie aufwendig die Berechnung der Alternativenmenge für eine bestimmte Regelfolge ist.

Abbildung 5.1 zeigt Werte für unterschiedliche Regelsätze und verschiedene Strukturvorgaben. Vergleicht man nun die von der absoluten Umfang des Suchvorgangs abstrahierenden Kenngrößen, läßt sich folgendes feststellen:

- Die Zahl der Schüsse pro Lösung ist bei den verschiedenen Läufen stark unterschiedlich. Strenge Strukturvorgaben können entweder (wie beim 2. Satz in Messung 2) die Zahl der Fehlschüsse erhöhen oder sie wie bei

1. Melodie, Mischtyp mit Dreiklangs- und Skalenbereich, 1. Satz
2. Melodie, Mischtyp, 2 Satz mit Quintkadenz.
3. Melodie, 3.Satz
4. Baß, 1.Satz
5. Mischmelodie, keine Strukturvorgaben, 1.Satz
6. rein skalenbasierte Melodie, keine Strukturvorgaben, 1.Satz
7. rein dreiklangsbasierte Melodie, 1.Satz
8. Mischmelodie, 1.Satz, keine Strukturvorgaben und keine Verwendung der Endformeln

	A	B	C	D	E	F	G	H	I	J
1.	5	583	116.60	13.60	6.52	4377	3.82	20.00	18.59	4.25
2.	12	9959	829.92	11.75	2.48	34647	1.52	4.06	165.06	4.76
3.	8	407	50.87	12.87	6.45	3020	3.98	34.10	20.67	6.84
4.	15	232	15.47	10.13	8.29	2016	2.67	75.39	33.60	16.67
5.	8	996	124.50	13.50	6.07	7030	4.48	15.36	29.47	4.19
6.	8	4863	607.87	13.87	4.27	25581	5.62	4.33	122.11	4.77
7.	8	97	12.12	14.25	9.92	1051	3.83	108.46	4.48	4.26
8.	8	522	65.25	14.25	6.21	3754	4.43	30.36	16.00	4.26

Abbildung 5.1: Statistiken zur Suche bei der Komposition

Messung 3 (Mischmelodie, 3. Satz) reduzieren. Die Fehlschußrate für den 1. Satz ist bei der strukturbehafteten Mischmelodie (Messung 1) und der strukturlosen Variante der Mischmelodie (Messung 5) recht ähnlich.

Vergleicht man die Melodtypen ohne den Einfluß von Strukturforderungen, dann gibt es bei dreiklangsbasierten Melodien (Messung 7) weniger Fehlschüsse als bei skalenbasierten, Mischformen (Messung 5) liegen in der Mitte.

Zusammenfassend kann man sagen, daß Baß und Dreiklangsmelodien einfach zu komponieren sind, skalenbasierte Melodien schwieriger. Strukturelemente können den Suchvorgang zielstrebig machen und somit Berechnungsaufwand sparen, sie können ihn aber genausogut drastisch erschweren. Die Auswirkungen von Strukturforderungen des Regelsatzes auf den Suchvorgang sind also sehr schwer vorherzusagen. Da ein höherer Aufwand beim Suchvorgang bis hin zur praktischen Unlösbarkeit gehen kann, erschwert dieser Effekt die experimentelle Arbeit mit strukturellen Beziehungen sehr.

Das Weglassen der Endformelprüfung in Messung 8 sorgt natürlich für eine geringere Fehlschußrate im Vergleich zur Referenzmessung Nr. 5.

- Wenn man die Lösungslänge (D) betrachtet, stellt man für die Baßstimme und für stark strukturbehaftete Melodieteile eine leichte Tendenz zu längeren Notenwerten gegenüber der strukturlosen Komposition (Melodie 5-8.) fest.
- Die Schußlänge E ist bei den Messungen 4 und 7 am größten, da der verhältnismäßig hohe Anteil an Erfolgen hier die durchschnittliche Schußlänge beeinflusst. Interessant ist auch, daß bei den Messung 2, bei der ja durch konfliktierende Strukturforderungen die Fehlschußrate groß ist, die Schußlänge gering ist, d.h. die Fehlschüsse frühzeitig erkannt werden.
- Vergleicht man den Verzweigungsgrad (G) der Melodietypen in Reinkultur, d.h. ohne Struktur (Messungen 5-7), dann liegt Verzweigungsgrad bei der skalenbasierten Melodie etwas höher als der der dreiklangsbasierten, Mischmelodien liegen dazwischen. Aus der im Schnitt größeren Zahl von Alternativen kann man allerdings wie gesagt nicht folgern, daß die Komposition insgesamt weniger aufwendig wird. Der Suchraum ist einfach insgesamt größer, die Lösungsdichte hängt davon nicht ab.

Bei den strukturbehafteten Kompositionsläufen ist der Verzweigungsgrad geringer. Da die Endformeln erst nach der Suche geprüft werden, hat das Weglassen der Endformelprüfung (Messung 8) keinen Einfluß auf den Verzweigungsgrad (verglichen mit der üblichen Mischmelodie Nr. 5).

Beim Baß sorgen die Regeln für Zusammenklangs- und Stimmführungsprüfung, zur Verbesserung der Baßlinie und die Hilfen zur Einhaltung der Endformeln für einen geringeren Verzweigungsgrad als bei der Melodie. Dies ist besonders auffällig, weil der Baß die Überprüfung der Regelfolge mit einer viel größeren Startintervallmenge als die Melodie beginnt. Die Regeln lassen dann beim Baß aber eine kleinere Zahl von Alternativen übrig.

- Die Zustandsausnutzung (H) hängt im Wesentlichen von der Fehlschußrate (C) ab. Da Zustände von Fehlschüssen nicht in Lösungen verwendet werden, ist bei einer hohen Fehlschußrate die Zustandsausnutzung gering. Die Unterschiede in der Schußlänge (E) beeinflussen ebenfalls die Zustandsausnutzung, die Fehlschußrate hat aber den stärkeren Effekt.
- Der spezifische Aufwand für einen Zustandsübergang (J) ist für die Mehrzahl der Melodiekompositionen ähnlich, nur bei Messung 3 (3. Satz)

mit der Fonte-Sequenz etwas erhöht. Der Aufwand pro Zustandsübergang für die Baßkomposition ist dagegen deutlich höher. Dies läßt sich leicht durch die Prüfung des Zusammenklangs und der Stimmführung erklären. Beide erfordern den Zugriff auf die andere Stimme über die Chart (und damit aufwendige Durchwanderung von Datenstrukturen). Die Stimmführungsprüfung erzeugt darüberhinaus noch bei jedem Zustandsübergang mehrere Stimmführungsquadrupel als Zwischendatenstruktur.

Zusammenfassend ist auffällig, daß der Aufwand eines Zustandsübergangs über verschiedene Melodieregelsätze hinweg als recht konstant angesehen werden kann. Man kann also die Zustandszahl als Maß für den Aufwand der Melodiekomposition benutzen. Die Fehlschußraten und die Zustandsübergangszahlen schwanken dagegen stark, insbesondere dann, wenn strukturelle Beziehungen innerhalb der Melodie mit im Spiel sind. Im Extremfall kann das System für einen Satz dann keine Lösungen finden.

Kapitel 6

Ausblick

Im Rahmen dieser Arbeit wurde ein System erstellt, das musikalisches Wissen aus verschiedenen Bereichen der Komposition, wie Melodik, Harmonik, Form und Stimmführung einsetzt, um frühklassische Menuette zu komponieren. Im folgenden wird nun beschrieben, auf welche Art das System um weitere musikalische Möglichkeiten erweitert werden kann und welche Perspektiven diese Arbeit für weitere Forschungsarbeiten eröffnet.

6.1 Erweiterungen und Verbesserungen

Erweiterungen und Verbesserungen des Systems sind in verschiedenen Bereichen denkbar.

Für die unterschiedlichen Teilaufgaben der Menuettkomposition wurden Verfahren zur Wissensrepräsentation und Verarbeitung realisiert, die an die spezielle Problemstellung angepaßt sind. Das Darstellungsformat COMF hat sich bei der Menuettkomposition aber auch beim Patternmatching [Bec00] und bei der harmonischen Analyse [Hö00] bewährt. Eine hilfreiche Erweiterung wäre es, eine Spur (track) nicht nur als Folge von Dauerelementen darzustellen, sondern die metrischen Anfangszeitpunkte mit abzuspeichern, da häufig über den Anfangszeitpunkt auf ein Element zugegriffen wird.

6.1.1 Erweiterungen der musikalischen Mittel

Die interessanten Erweiterungsmöglichkeiten liegen weniger bei den Verfahren, sondern mehr beim repräsentierten musikalischen Wissen. Hier gibt es noch verschiedene Ausbaumöglichkeiten.

Das Verfahren erzeugt nur einen kleinen Teil der denkbaren stilistischen Menuette (vgl. Kap. 1, Abb. 1.2). Die im folgenden skizzierten Erweiterungen

zielen darauf hin, in das Verfahren weitere musikalische Mittel einzubauen. Dies macht Stücke interessanter und vergrößert die Abdeckung von Stil und Genre.

Die technische Seite ist dabei eher beherrschbar, da beim Entwurf die Notwendigkeit von Erweiterungen berücksichtigt wurde. Die Komponenten des Kompositionsverfahrens sind flexibel gehalten, so daß das Programm oft mit wenig Aufwand erweitert werden kann. Die Datenstrukturen sind ebenfalls „auf Zuwachs“ angelegt und sind dafür ausgelegt, weitere Elemente aufzunehmen und bei Bedarf auch (im Sinne der nichtdestruktiven Modifikation) mitzukopieren. Die Schwierigkeit bei Erweiterungen liegt dagegen oftmals darin, daß es an musikalischem Wissen fehlt. Durch Unverträglichkeiten mit dem bestehenden Wissen laufen Erweiterungen Gefahr, den stilistisch vorgesehenen Bereich zu überschreiten. Ein Beispiel dafür sieht man in Notenbeispiel 5, wo sich ein Viertelrepetitionstakt und ein Quartvorhalt ungeschickt überlagern. Um solche Konflikte zu verhindern, bedarf es vor Erweiterungen genauer musikalischer Untersuchungen, deren Ergebnisse wieder technisch umgesetzt werden können.

Im Folgenden werden einige denkbare Erweiterungen vorgestellt und soweit wie möglich die Probleme bei der programmtechnischen Realisierung diskutiert.

Erweiterungen der Melodieregeln

Für die Melodie sind einige zusätzliche musikalische Elemente interessant:

- Einführung eines Auftakts. Auftaktige Menuette kommen häufig vor. Diese Erweiterung ist verhältnismäßig aufwendig, da der auftaktige Beginn sich im Stück fortsetzt, d.h. der zweite Satz beginnt ebenfalls auftaktig. Dementsprechend müssen einige Regeln, in denen bisher die Position im Stück über die Angabe des metrischen Schwerpunkts angegeben wird, auf eine abstrakte Angabe der Position im Stück relativ zum Auftakt umgestellt werden. Andere Regeln beziehen sich dagegen wirklich auf den metrischen Schwerpunkt. Durch die Verschiebung zwischen metrischen Schwerpunkten und der Position im Stück (bzw. im Satz) sind auch Endformeln mit anderen Dauern nötig.
- Einführung von Pausen. Es ist verhältnismäßig einfach, ein spezielles Fortsetzungselement für die Pause zu definieren und Regeln zu erstellen, die Pausen einführen. Allerdings muß für alle Regeln, die die Vorgängernote berücksichtigen, der Fall, daß die Vorgängernote eine Pause ist, definiert werden, (z.B. dadurch, daß die Note vor der Pause verwendet wird). Im Endeffekt läuft dies auf eine Modellierung der

Reichweite von Stimmgenerierungsregeln analog zur Reichweite von Stimmführungsregeln (vgl. Kap. 3.7.3) hinaus.

- Einsatz von großen Intervallen (Quarten, Quinten, Oktaven) in skalenbasierten Melodien. Große Intervalle haben meist wichtige Stufen (1. oder 5.) als Anfangs- oder Zielpunkt und werden danach gerne ausgeglichen, d.h. man geht in Sekunden der Sprungrichtung entgegen. Auch das metrische Gewicht spielt eine Rolle dabei, ob das große Intervall als Fremdkörper in der Melodie empfunden wird.

Erweiterungen der Strukturerstellung

Ein größeres Repertoire an Strukturen sorgt dafür, daß die Kompositionen interessanter und unterschiedlicher werden. Die Stimmgenerierung kann problemlos mit solchen Strukturbeschreibungen arbeiten. Unter anderem sind folgende Erweiterungsmöglichkeiten interessant.

- Verwendung von Formen mit einer größeren Zahl von Sätzen. Für die Tonordnung müssen dann entweder Schablonen für die möglichen anderen Formen aufgestellt werden oder - was deutlich aufwendiger ist - die Prinzipien, die zu den in [Bud83] beschriebenen Takt- und Tonordnungen führen, im Programm realisiert werden.
- Realisierung von den in der zeitgenössischen Literatur beschriebenen Mitteln zur Satzerweiterung, wie Einschiebsel oder Sequenzierung von Takten. So entstehen längere Sätze als die prototypischen Vierer.
- Stärkere Automatisierung bei der Erstellung der Mikrostruktur (z.B. Harmoniebasis) statt Verwendung von Bausteinen.

Die Realisierung des letzten Punkts erleichtert die Erweiterungen der makrostrukturellen Möglichkeiten, da die bisherigen Schablonen und Muster implizite Annahmen über die Makrostruktur enthalten. Problem bei der Automatisierung ist es, die Mikrostruktur so zu definieren, daß sie zur gewünschten Makrostruktur passt und sie noch komponierbar bleibt, d.h. daß keine Widersprüche zu den Stimmgenerierungsregeln oder anderen Parametern entstehen und die Lösungsdichte nicht zu gering wird.

Definition zusätzlicher Bottom-up-Strukturelemente

In Menuettmelodien sind verschiedene nichtharmonische Töne zu finden, die die Melodie interessanter gestalten. Nichtharmonische Tönen müssen auf eine

vorgegebene Art behandelt werden. Bislang ist von diesen nichtharmonischen Tönen der diatonische Durchgang und der Quartvorhalt realisiert.

Diese Elemente sind in verschiedenen Quellen der zeitgenössischen Literatur beschrieben, wobei sie meist am strengen vierstimmigen Satz demonstriert werden. Die musikalisch sinnvolle Übertragung dieser Regeln auf den zweistimmigen freien Satz in einem Menuett erfordert vertiefte musikalische Kenntnisse und Stilgefühl. Die technischen Mittel zur Realisierung sind durch die Bottom-Up-Strukturelemente vorhanden.

Interessant ist die Realisierung von weiteren Arten von Vorhalten, Wechselnoten, harten Durchgängen und chromatischen Durchgängen. Zu beachten ist allerdings, daß einige Elemente wie z.B. Vorhalte mit großen Notenwerten Einfluß auf Akzent und Betonung im Stück haben. Damit sie sinnvoll verwendet werden, ist es nötig, daß deren Existenz (nicht unbedingt die genaue Länge) von der Strukturbeschreibung berücksichtigt wird.

Urlinien

Bei manchen Menuetten entdeckt man, daß einzelne Töne, z.B. die höchsten Töne jedes Takts oder die Einsen jedes Takts, wieder eine Melodie ergeben. Dieses Prinzip wird durch die Schenkersche Ursatztheorie auf die Spitze getrieben. Ebcioğlu beruft sich auf diese [Ebc86] (vgl. Kap. 2.1.2,2.3.2). Auch bei der Analyse von Mozarts Menuett KV2 [Moz91] in Kap. 2.2.4 treten solche Linien zu Tage.

Der Einbau von Urlinien in die Stimmengenerierung kann durch eine Regel geschehen, die bei der Komposition der entsprechenden Noten fordert, daß die Note aus der vorgegebenen Urlinie verwendet wird. Die anderen Stimmengenerierungsregeln prüfen, ob die geforderte Note überhaupt zulässig ist. In einem zweiten Schritt kann man die Urlinie durch Stimmengenerierung erzeugen lassen.

Hinzufügen von Mittelstimmen

Das Verfahren komponiert einzelne Stimmen. Manche der originalen Klaviermenuette sind durchgängig zweistimmig, andere sind dreistimmig oder die Stimmenzahl wechselt, so daß beispielsweise an den Endformeln der Satz durch eine zusätzliche Stimme verstärkt wird. Bei einigen späteren Menuetten findet man statt einem an Stimmen orientierten Satz eine klaviermäßige Begleittextur vor.

Für das Hinzufügen von Mittelstimmen sind am System folgende Erweiterungen vorzunehmen:

- Verlängern des Kompositionsplans um Planelemente für die Generierung der einzelnen Sätze der zusätzlichen Stimme.
- Erweitern des Regelsatzes um einen Eintrag für die zusätzliche Stimme mit deren speziellen Angaben. Die Datenstruktur und das Eingabeformat sind bereits dafür vorbereitet.
- Definition einer Stimmengenerierungsregel für die Füllstimme. Man kommt größtenteils mit den bestehenden Stimmengenerierungsregeln aus. Der Regelsatz wird verhältnismäßig einfach, er wird typischerweise kaum mehr tun, als die Größe der Intervalle begrenzen, Stimmkreuzungen mit Melodie oder Baß verhindern und die Einhaltung der Harmonie zu sichern. Regeln, die für eine gute Melodie sorgen, wie z.B. die Reputationsbeschränkungen werden für Mittelstimmen weggelassen.
Um ein musikalisch gutes Ergebnis zu erhalten, kann man typischen Verwendungsweisen einer Mittelstimme, wie z.B. dem Austerzen der Melodiestimme, ein verstärktes Gewicht geben.
- Definition einer Folge von Stimmführungsregeln. Hier gibt es leichte Abweichungen, beispielsweise ist die Quarte einer Stimme gegen den Baß dissonant, die Quarte zwischen zwei Oberstimmen konsonant.
- Soll die Füllstimme nur an strukturell wichtigen Teilen der Komposition hinzukommen, muß dies in der Strukturbeschreibung festgelegt werden.
- An den Endformeln sind keine Erweiterungen nötig, da diese durch die Harmonien und die Stimmführung der Außenstimmen definiert sind. Für Mittelstimmen gibt es keine Einschränkungen durch die Endformel.

Nachträgliche Bewertung der Komposition

Die Bewertung fertig komponierter Sätze ist - neben der bereits implementierten Verwendung von Gewichten während der Stimmengenerierung - eine Möglichkeit, weiche Regeln in das System einzubringen. Die Bewertung bestimmt zu einer Lösung (Chart) eine Bewertungszahl, die dann bei der Reduktion der Lösungsmenge berücksichtigt wird.

Die Bewertung ist nützlich, um Phänomene zu berücksichtigen, die über eine Note und ihre direkten Nachbarn hinausreichen.

Folgende Ideen zur Bewertung der Melodie sind denkbar:

- Einhalten eines Melodiebogens mit höchster Note in der Mitte.
- Positive Bewertung der Verwendung gattungs- und stiltypischer melodischer Wendungen.

- Positive Bewertung von Ähnlichkeiten, die bei der Komposition zufällig entstanden sind, ohne daß sie von einem Strukturelement gefordert werden.
- Betrachtung auffälliger Töne, z.B. lokaler Spitzentöne oder metrisch betonter Noten als melodische Linie.

Ist beispielsweise der höchste Ton eines Taktes die 7. Stufe, so ist es positiv zu bewerten, wenn im Verlauf des aktuellen Satzes der Melodie die 8. Stufe als Fortführung des Leittons nach oben folgt. Solche eine Fernauflösung des Leittons wäre im Baß bei Nr. 2 und Nr. 5 von Notenbeispiel 7 wünschenswert gewesen.

Für die Suche nach den Phänomenen kann der Patternmatcher aus [Bec00] verwendet werden, der bereits für den Endmustervergleich benutzt wird. Die Suchfunktion des Patternmatchers (die beim Endmustervergleich nicht eingesetzt wird) ist für die Bewertung dann hilfreich.

6.1.2 Verbesserung von Verfahren und Strategien

Im Verfahren kommen auf zwei Ebenen Strategien vor: Die Stimmengenerierung arbeitet notenweise und die Abarbeitung des Kompositionsplans satzweise.

Verfahren für die Suche im Zustandsraum

Das Verfahren ist durch die Auftrennung in Suchstrategie und Problembe-
schreibung (vgl. Kap 3.4) sehr flexibel und erlaubt es, die Strategien abhängig
von der Struktur des Lösungsraums zu wählen. Durch die explizite Repräsen-
tation der Bearbeitungsschritte und die geeignet angelegten Datenstruktu-
ren für Lösungen und Zwischenergebnisse ist der Wechsel der Strategie ohne
Programmieraufwand möglich. Es ist aber für den Menschen schwierig, die
Struktur des Lösungsraums vor dem Lauf des Programms abzuschätzen -
schließlich ist es gerade das Ziel solch einer Trennung zwischen Problembe-
schreibung und Strategie, daß man nicht mehr über die Abarbeitungsweise
nachdenkt.

Für die bisher verwendeten Regelsätze hat sich die Zufallsschußstrate-
gie bewährt (vgl. Kap. 3.4.4, Kap. 5.5). Bei Änderungen des Wissens und
Verstärkung der strukturellen Beziehungen kann es passieren, daß die Lösungs-
dichte für das Zufallsschußverfahren zu gering wird und eine andere Strategie
vorteilhafter ist.

Eine interessante Erweiterung, ist es, dieses Herumexperimentieren nicht von Hand zu machen, sondern die Entscheidung über die Strategie den Rechner treffen zu lassen. Wenn eine Strategie keine Ergebnisse liefert oder lange Zeit benötigt, wiederholt er der Rechner den Versuch mit einer anderen Strategie.

Abarbeitungsstrategie für den Kompositionsplan

Bisher ist der Kompositionsplan eine lineare Abfolge von Planschritten, die nacheinander abgearbeitet werden. Die Planelemente des Kompositionsplans können als Problembeschreibung im Sinne eines Suchproblems aufgefaßt werden, denn sie definieren eine Menge von möglichen Nachfolgern für eine gegebene Lösung.

Bei der Suche im Zustandsraum kann es vorkommen, daß die Abarbeitung eines Planschritts keine Lösung ergibt. Mögliche Gründe sind, daß es wegen eines Widerspruchs zwischen den verschiedenen Vorgaben, dem Regelsatz, der Strukturbeschreibung und der bisherigen Komposition keine Lösungen gibt, oder daß die Lösungsdichte im Suchraum so gering ist, daß keine gefunden wurde.

Eine sinnvolle Erweiterung ist es, die Abarbeitung des Kompositionsplans ebenfalls mit einer Suchstrategie vorzunehmen, die Entscheidungen zurücknehmen kann. Die Datenstrukturen und die Verwaltung der Lösungsmenge ist durch das nichtdestruktive Vorgehen dafür bereits geeignet.

Bisher wurde das musikalische Wissen so gestaltet, daß solche Widersprüche soweit wie möglich vermieden werden. Die erweiterte Strategie würde es erlauben, Wissen mit schärferen Restriktionen einzusetzen, da es nicht mehr nötig ist, Widersprüche im Vorfeld zu vermeiden. So wird das Strategie tragfähig für die Verwendung weiterer musikalischer Elemente.

Der Einsatz von Suchverfahren auf Satzebene ist auch Voraussetzung für die Beherrschung größerer Formen mit vielen Sätzen und dynamisch generierter (statt fest vorgegebener) Makrostruktur.

6.1.3 Verbesserung der Vorbereitung des Wissens

Eine sinnvolle Erweiterung ist die Suche nach Gesetzmäßigkeiten mittels Computerverfahren. Durch den Patternmatcher aus [Bec00] und die Reimplementierung von Maxwells System zur harmonischen Analyse in [Hö00] stehen nun Programme zur Verfügung, die nach Mustern unter Berücksichtigung der augenblicklichen Tonart suchen können.

6.2 Perspektiven

Diese Arbeit hat sich für Stil und Gattung enge Grenzen gesetzt, um klare Kriterien zur Bewertung der Ergebnisse zu erhalten. Trotzdem sind große Teile des Verfahrens und des im System repräsentierten Wissens im System nicht für die Aufgabe, frühklassische Klaviermenuette zu komponieren, spezifisch. Es ist deswegen denkbar, Teile des Systems für die Komposition in anderen Stilen, von anderen Gattungen und für andere Besetzungen wiederzuverwenden. In manchen Fällen ist es sogar möglich, Komponenten für andere Aufgaben (z.B. Analyse statt Komposition) einzusetzen. Dabei ist der flexible Aufbau des Systems und die explizite Repräsentation des Musikwissens nützlich.

Die einzelnen Teilaufgaben der Menuettkomposition sind bewußt getrennt definiert und implementiert. Dies ermöglicht die Verwendung einzelner Komponenten für andere Aufgaben:

- Das Verfahren kann - wie in Kap. 5.4 gezeigt - dazu verwendet werden, eine gegebene Melodie mit einer Begleitstimme zu versehen. Hierzu wird die gegebene Melodie mit der zu ihr passenden Strukturinformation versehen und beides in eine Chart eingegeben. Dann startet man das Verfahren mit einem Kompositionsplan, der nur die Planelemente für die Begleitkomposition enthält.
- Die Stimmführungsprüfung kann dazu benutzt werden, die Stimmführung eines gegebenen Stücks zu untersuchen. Man kann sie in einem System zur Korrektur von Tonsatzübungen ähnlich wie die Musik Theory Workbench nach [Tau99] einsetzen.
- Der Patternmatcher stammt aus einem System zur Analyse von Musikstücken [Bec00]. Diesem System kann man weitere Suchmöglichkeiten hinzufügen, indem man die Ergebnisse der harmonischen Analyse ([Hö00], [Max92]) und die im Rahmen dieser Arbeit implementierte Untersuchung der Stimmführung verwendet. So kann man die Stimmführung von Stücken mit dem Rechner analysieren.

Wenn man die Übertragbarkeit von Teilen des Systems auf andere Stile und Gattungen untersuchen will, ist es nötig, sich zu vergegenwärtigen, aus welchen Quellen das Wissen des Systems stammt, und welche Eigenschaften der frühklassischen Menuette für Entscheidungen beim Aufbau des Systems verantwortlich sind. Hier sind verschiedene Übertragungsmöglichkeiten vorstellbar:

- Die Partiturrepräsentation COMF und die Darstellung der Harmonien durch Stufen ist für metrisch und tonal gebundene Musik angemessen. In der hier vorgestellten Form ist das der Bereich vom Hochbarock (ab ca. 1700) bis in die Frühromantik (ca. 1820).

Für spätere Stile mit abendländisch tonaler Harmonik, wie z.B. die spätere Romantik oder die heutige U-Musik ist die Darstellung der erweiterten harmonischen Möglichkeiten in COMF zwar möglich, aber unangemessen.

- Die Sichtweise auf die Stimmführung durch Betrachtung von Stimmführungsquadrupeln ist vom Grundprinzip her für die vorangegangenen Epochen, z.B. die Komposition des Barock, angemessen.

Jedoch wurde für die Komposition von Menuetten nur ein kleiner Teil der in der Literatur beschriebenen Stimmführungsregeln realisiert, da bei Menuetten die harmonischen Verhältnisse und die verwendeten musikalischen Mittel einfach sind. Wenn das Verfahren auf Paradebeispiele der Satztechnik, wie z.B. Fugenkomposition oder Generalbaßsatz im Stil von Bach angewandt werden soll, muß das Stimmführungswissen deutlich ausgebaut werden. Eventuell wird es in diesen Fällen nötig, auch strukturelle Elemente bei der Stimmführungsprüfung mit einzubeziehen, damit großräumigere Gesetzmäßigkeiten berücksichtigt werden können.

- Das Verfahren zur Stimmengenerierung unter Berücksichtigung struktureller Randbedingungen kann zur Komposition von stark stimmenorientierter Musik, z.B. Chor- oder manche Instrumentalmusik eingesetzt werden. Die Stimmengenerierung ist für Musik mit konstanter Stimmenzahl sogar besser geeignet als für Klavierstücke.
- Die intervallweise Formulierung der Stimmengenerierungsregeln berücksichtigt nicht großräumige Phänomene wie melodische Spannungsverläufe. Sie ist dagegen gut geeignet für intervallweise Stimmengenerierung bei Kompositionsaufgaben mit starken gattungs- und instrumententypischen Einschränkungen des Intervallvorrats.
- Die eingesetzte Suchstrategie ist auf den Freiheitsgrad der Menuettkomposition abgestimmt. Bei stärker einschränkenden Kompositionsaufgaben kann dies dazu führen, daß das Zufallsschußverfahren keine Lösung findet. Bei starken Einschränkungen durch den Stil kann es sinnvoll sein, den Suchvorgang deterministisch zu einer Lösung hin zu steuern und verstärkt die Steuerung durch das Suchgewicht zu nutzen.

Soll das Kompositionsverfahren für die Komposition von Stücken mit größeren Formen verwendet werden, so wird auch ein reichhaltigerer Vorrat an Verwandtschaftsbeziehungen zwischen Sätzen oder Stimmen (Imitation) gewünscht sein. Ein vermehrter Einsatz von Korrespondenzen über die Satzgrenzen hinweg führt dazu, daß die Komposition eines Satzes wegen Entscheidungen, die während der Komposition vorangegangener Sätze bzw. Stimmen getroffen wurden, fehlschlägt. Damit dies nicht zum Scheitern des gesamten Kompositionsprozesses führt, ist für die Komposition in größeren Formen, wie in Kap. 6.1.2 skizziert, auch auf Makrostrukturebene eine Suchstrategie zu verwenden.

Eine Stärke des vorliegenden Systems sind die expliziten Darstellungen von Zwischenergebnissen, die die Beobachtung des Kompositionsprozesses und Eingriffe in denselben ermöglichen. Damit eröffnen sich interessante Möglichkeiten zur interaktiven Verwendung des Systems. Beispielsweise kann sich ein Benutzer die Menge der Zwischenergebnisse anzeigen lassen und selbst die Auswahl treffen, welche Komposition weiter verfolgt wird. Damit kann der Mensch ein z.B. künstlerisches Ziel mitbestimmen, während der Rechner die Einhaltung des handwerklichen Regelwerks übernimmt.

6.3 Fazit

Diese Arbeit hat gezeigt, daß die Komposition von frühklassischen Menuetten tatsächlich vom Rechner ausgeführt werden kann. Es ist nicht - wie manchmal behauptet - eine mystische „*künstlerische Inspiration*“ erforderlich, die die Komposition durch einen Rechner grundsätzlich unmöglich macht.

Die in Kap. 2.3 beschriebenen Vorarbeiten im Bereich der Stilkopie haben gezeigt, daß einzelne Kompositionsaufgaben vom Rechner ausgeführt werden können. Diese Arbeit zeigt, daß auch die Integration der verschiedenen Kompositionsaufgaben durch den Rechner möglich ist. Das reiche Repertoire an Modellierungs- und Inferenztechniken, das die symbolische künstliche Intelligenz in den letzten Jahrzehnten entwickelt hat, ist dafür solche Kompositionsaufgaben vollkommen ausreichend.

Der limitierende Faktor bei dieser Arbeit sind nicht die Programmier-techniken, sondern die Beschaffung modellierbaren und korrekt modellierten Musikwissens. Das verwendete Wissen muß beschafft und bei Änderungen anderer Wissensselemente gewartet werden. Die Wirkung eines Wissensselements muß sowohl alleine als auch in Wechselwirkung mit anderem Wissen experimentell überprüft werden, indem man die Auswirkung auf das Kompositionsergebnis beobachtet. Bei Folgearbeiten, die sich an Komposition in komplizierteren Stilen oder größeren Formen wagen, wird das Musikwissen noch mehr zum limitierenden Faktor werden.

Anhang A

Weitere Kompositionsergebnisse

Notenbeispiel 20 Ergebnismenge mit Richtungsgleichheiten

Beim 1. Vierer der Melodie waren 20 Ergebnisse gefordert. Nur für 15 dieser ersten Vierer konnte der 2. Vierer der Melodie erstellt werden. Für eine Lösung wurde im 4. Vierer kein Baß erstellt, so daß insgesamt 14 komplette Menuette entstanden.

20/1

First system of musical notation for example 20/1. It consists of two staves: a treble clef staff and a bass clef staff. The key signature has one sharp (F#) and the time signature is 3/4. The melody in the treble staff begins with a quarter note G4, followed by quarter notes A4, B4, and C5. The bass staff begins with a half note G3.

Second system of musical notation for example 20/1, starting at measure 9. The treble staff continues the melody with quarter notes D5, E5, F#5, and G5. The bass staff continues with quarter notes G3, A3, B3, and C4.

20/2

First system of musical notation for example 20/2. It consists of two staves: a treble clef staff and a bass clef staff. The key signature has one sharp (F#) and the time signature is 3/4. The melody in the treble staff begins with a quarter note G4, followed by quarter notes A4, B4, and C5. The bass staff begins with a half note G3.

Second system of musical notation for example 20/2, starting at measure 9. The treble staff continues the melody with quarter notes D5, E5, F#5, and G5. The bass staff continues with quarter notes G3, A3, B3, and C4.

20/3

Musical score for exercise 20/3, measures 1-8 and 9-16. The score is in 3/4 time, key of D major (one sharp), and consists of two systems. Each system has a treble and bass staff. The first system (measures 1-8) shows a melodic line in the treble staff and a bass line in the bass staff. The second system (measures 9-16) continues the piece, with a measure rest at the beginning of the first staff.

20/4

Musical score for exercise 20/4, measures 1-8 and 9-16. The score is in 3/4 time, key of D major (one sharp), and consists of two systems. Each system has a treble and bass staff. The first system (measures 1-8) shows a melodic line in the treble staff and a bass line in the bass staff. The second system (measures 9-16) continues the piece, with a measure rest at the beginning of the first staff.

20/5

Musical score for exercise 20/5, measures 1-8 and 9-16. The score is in 3/4 time, key of D major (one sharp), and consists of two systems. Each system has a treble and bass staff. The first system (measures 1-8) shows a melodic line in the treble staff and a bass line in the bass staff. The second system (measures 9-16) continues the piece, with a measure rest at the beginning of the first staff.

20/6

Musical score for exercise 20/6, measures 1-8 and 9-16. The score is in 3/4 time, key of D major (one sharp), and consists of two systems. Each system has a treble and bass staff. The first system (measures 1-8) shows a melodic line in the treble staff and a bass line in the bass staff. The second system (measures 9-16) continues the piece, with a measure rest at the beginning of the first staff.

20/7

Musical score for 20/7, measures 1-8 and 9-16. The score is in 3/4 time, key of D major. The first system (measures 1-8) shows a melody in the treble clef and a bass line in the bass clef. The second system (measures 9-16) continues the melody and bass line, with a measure rest in the bass line at the beginning of the system.

20/8

Musical score for 20/8, measures 1-8 and 9-16. The score is in 3/4 time, key of D major. The first system (measures 1-8) shows a melody in the treble clef and a bass line in the bass clef. The second system (measures 9-16) continues the melody and bass line, with a measure rest in the bass line at the beginning of the system.

20/9

Musical score for 20/9, measures 1-8 and 9-16. The score is in 3/4 time, key of D major. The first system (measures 1-8) shows a melody in the treble clef and a bass line in the bass clef. The second system (measures 9-16) continues the melody and bass line, with a measure rest in the bass line at the beginning of the system.

20/10

Musical score for 20/10, measures 1-8 and 9-16. The score is in 3/4 time, key of D major. The first system (measures 1-8) shows a melody in the treble clef and a bass line in the bass clef. The second system (measures 9-16) continues the melody and bass line, with a measure rest in the bass line at the beginning of the system.

20/11

Musical score for exercise 20/11, consisting of two systems of two staves each. The first system shows the beginning of the piece in G major, 3/4 time. The second system starts at measure 9, marked with a '9' above the treble clef. The notation includes treble and bass clefs, a key signature of one sharp (F#), and a 3/4 time signature.

20/12

Musical score for exercise 20/12, consisting of two systems of two staves each. The first system shows the beginning of the piece in G major, 3/4 time. The second system starts at measure 9, marked with a '9' above the treble clef. The notation includes treble and bass clefs, a key signature of one sharp (F#), and a 3/4 time signature.

20/13

Musical score for exercise 20/13, consisting of two systems of two staves each. The first system shows the beginning of the piece in G major, 3/4 time. The second system starts at measure 9, marked with a '9' above the treble clef. The notation includes treble and bass clefs, a key signature of one sharp (F#), and a 3/4 time signature.

20/14

Musical score for exercise 20/14, consisting of two systems of two staves each. The first system shows the beginning of the piece in G major, 3/4 time. The second system starts at measure 9, marked with a '9' above the treble clef. The notation includes treble and bass clefs, a key signature of one sharp (F#), and a 3/4 time signature.

Notenbeispiel 21 Ergebnismenge mit Gegenrichtung

Beim 1. Vierer der Melodie waren 20 Ergebnisse gefordert. Nur für 9 dieser ersten Vierer konnte der 2. Vierer der Melodie erstellt werden. Für eine Lösung wurde im 4. Vierer kein Baß erstellt, so daß insgesamt 8 komplette Menuette entstanden.

21/1

Musical notation for 21/1, showing two systems of treble and bass staves in 3/4 time with a key signature of one sharp (F#). The first system shows a melodic line in the treble and a bass line in the bass. The second system continues the melody and bass line.

21/2

Musical notation for 21/2, showing two systems of treble and bass staves in 3/4 time with a key signature of one sharp (F#). The first system shows a melodic line in the treble and a bass line in the bass. The second system continues the melody and bass line.

21/3

Musical notation for 21/3, showing two systems of treble and bass staves in 3/4 time with a key signature of one sharp (F#). The first system shows a melodic line in the treble and a bass line in the bass. The second system continues the melody and bass line.

21/4

Two systems of musical notation for 21/4. Each system consists of a treble and bass staff. The key signature is one sharp (F#) and the time signature is 3/4. The first system shows a melodic line in the treble and a bass line in the bass. The second system starts with a measure marked with a '9' in the treble staff, indicating a measure rest or a specific starting point.

21/5

Two systems of musical notation for 21/5. Each system consists of a treble and bass staff. The key signature is one sharp (F#) and the time signature is 3/4. The first system shows a melodic line in the treble and a bass line in the bass. The second system starts with a measure marked with a '9' in the treble staff.

21/6

Two systems of musical notation for 21/6. Each system consists of a treble and bass staff. The key signature is one sharp (F#) and the time signature is 3/4. The first system shows a melodic line in the treble and a bass line in the bass. The second system starts with a measure marked with a '9' in the treble staff.

21/7

Two systems of musical notation for 21/7. Each system consists of a treble and bass staff. The key signature is one sharp (F#) and the time signature is 3/4. The first system shows a melodic line in the treble and a bass line in the bass. The second system starts with a measure marked with a '9' in the treble staff.

21/8

Musical score for exercise 21/8, measures 1-8 and 9-16. The score is in 3/4 time, key of D major. The first system (measures 1-8) shows a melodic line in the treble clef and a bass line in the bass clef. The second system (measures 9-16) continues the piece, starting with a measure rest in the treble clef.

Notenbeispiel 22 Ergebnismenge mit Strukturvorgabe

22/1

Musical score for exercise 22/1, measures 1-8 and 9-16. The score is in 3/4 time, key of D major. The first system (measures 1-8) shows a melodic line in the treble clef and a bass line in the bass clef. The second system (measures 9-16) continues the piece, starting with a measure rest in the treble clef.

22/2

Musical score for exercise 22/2, measures 1-8 and 9-16. The score is in 3/4 time, key of D major. The first system (measures 1-8) shows a melodic line in the treble clef and a bass line in the bass clef. The second system (measures 9-16) continues the piece, starting with a measure rest in the treble clef.

22/3

Musical score for exercise 22/3, measures 1-8 and 9-16. The score is in 3/4 time, key of D major. The first system (measures 1-8) shows a melodic line in the treble clef and a bass line in the bass clef. The second system (measures 9-16) continues the piece, starting with a measure rest in the treble clef.

Appendix B

English Summary

B.1 Introduction

This thesis describes a knowledge based approach to composition of minuets¹ for piano in early classical style (that is the period of the young W.A. Mozart) by a computer.

- The goal is, to approximate the style of this period as closely as possible. The system should only generate *stylistically correct* minuets - pieces an expert would accept as a classical minuet. On the other side, the system is allowed to undergenerate. That means, the system is not designed to produce every correct minuet, but to produce only correct minuets.
- The minuet composition system combines different aspects of composition such as melody, harmony, structure, and voice leading.

In the past, approaches to simulating tonal composition have focussed on a single aspect of composition. Examples are the counterpoint [Sch89], which considers linear composition and voice lead, chorale harmonization which considers harmony and voice lead [Ebc86], and composition of sonatas by means of building blocks [Ber95] which focusses on structural composition.

- The system works with explicit representations of knowledge. This allows experiments with different combinations of knowledge and opens the way for experimental musicological research.

¹a genre of dance music from the 17th and 18th century in triple meter

A musicological hypothesis can be represented in the system and verified/falsified by computer experiments. A similar approach has been used by experimental archeology.

- Contemporary literature² provides a correct view of the style.

There were several important reasons to choose classical minuets as genre for this approach:

- Minuet composition covers the most important aspects and phenomena of classical composition: melody, harmony, rhythm, form and voice-leading. In spite of this completeness, minuets are short and built regularly. For this reason, contemporary literature had already used minuets as a model genre for composition and minuets will be a good genre for computer experiments with the different phenomena.
- The second reason is, that style and genre are well researched. Extensive modern and contemporary literature and well-defined musical terms and criteria to judge results are available. This is a prerequisite for the computer science side of the research, because this project does not have the goal to discover some novel musical aesthetics, but to compose minuets, which are accepted as stylistically correct.

The author of a knowledge-based AI system has to acquire the domain knowledge, has to formalize it and to provide an algorithm which the computer can use to find a solution. Often, it is helpful to look at the ways human experts codify their knowledge, e.g. their literature. Explicit knowledge about composition is needed for teaching composition students or for scientific work – e.g. for comparisons between styles or for classification of borderline cases of works. Human composers learn some subtasks of composition by imitation “*It should sound like this*”, so not all knowledge is available in an explicit form.

For the field of classical minuets, the following knowledge sources are considered:

1. The usage of period literature like [Rie52] and [Koc82] or modern literature with an historic approach [Bud83] ensures a correct view of the style.
2. Ahistoric systematic theories such as the usual theory of harmony [KD84] describe properties universal to all tonal music.

²In musicology, *period* or *contemporary* literature (zeitgenössische Literatur) is literature written in the same time period as the musical period or style it describes.

Systematic theories are usually more formalized and therefore easier to implement than knowledge from period literature.

3. Asking experts (musicologists, composition teachers, and composers) in the style yields explicit knowledge, which might not have been written down.
4. Manual analysis can either give an abstract verbal description of a minuet or it can search minuets for evidence in favour or against a specific hypothetical rule.
5. Computer analysis of works can help with the second form of analysis described above: the search for evidence for or against a rule.
6. Computer composition experiments: The composition algorithm is run with a rule set including a hypothetic rule and then with another rule set without that particular rule. The results are compared (preferably by an expert).
7. Cognitive introspection. Ask someone, who has just composed a minuet, how and why he made certain decisions.

The last four knowledge sources should give access to implicit knowledge, which is not available in literature.

B.2 Knowledge Based Minuet Composition

The structure of minuets is thoroughly discussed in period literature. The basic unit of a composition is the *sentence*³. Each sentence is closed by an end formula. The composition process is split into two levels: the macro and the micro level. The *linear composition* works on the micro level. It composes note by note one sentence of one voice. Its strategy is discussed later.

On the macro level, the composition strategy is quite simple: The composition plan is executed sequentially by calling the linear composition for each voice and each sentence. The following dependencies must be considered: The composition of the melody depends on the preceding sentences of the melody. The composition of the bass depends on the preceding sentences of the bass and the melody up to the same sentence.

The musical score produced by the composition process is stored in a *score chart*. It represents the information classical composers wrote down. Notes

³Original: *Satz*. Another more common english term is *phrase*.

are stored with pitch and duration, while key and meter are fixed for the whole chart.⁴ On top of this, the score chart also stores abstract information for and about the composition process in form of *structural elements*. One example is harmonic information, which is stored by degree (as in degree theory) relative to a current key. The current key is either the base key or a modulation of the base key.

B.2.1 Voice Generation

Voice generation composes one voice for one sentence note by note. For the voice generation, two sequences of rules are defined. The first sequence describes the composition of the first note of the sentence, the other describes the following intervals. Composing one note starts with a set of possible alternatives. Each rule of the rule sequence is applied on the current set of alternatives. The rule can add or remove alternatives from the alternative set or it can modify alternatives. After the last rule has been applied, the set of correct alternatives is determined. The set may be empty or may contain a single alternative or several alternatives.

The set of correct alternatives defines a search space. The usual search strategies known in artificial intelligence (e.g. depth-first-search [RN95]) can be used to search for a solution. With the current rule set, the random-shot strategy proved to be the most suitable. The random shot strategy selects the successor state out of the set of possible successors by random. In case of a fail or an empty successor set, the next attempt starts with the start state (the beginning of the sentence) again.

The musical knowledge about the composition of a melody is coded in the rules. The rules differ for the two basic types of melodies: Scale-based melodies, which consist mainly of seconds, and chord based melodies, which consists of broken chords. The bass is composed with the same algorithm as the melody, but it uses a different rule set.

The rules belong to the following categories:

- General interval rules, which define the basic properties of either the scale based or the chord based melodies.
- Leading note rules
- Rules for metric stress and durations
- Rules for bottom-up structural elements

⁴Sometimes, classical composers also wrote down interpretational information such as articulation, dynamics and intended instrument. This is neglected here.

- Rules for the vertical direction: harmony check (melody and bass) and consonance check (bass only).
- Soft rules - preferences. They do not remove or add alternatives but modify the alternatives' weight. The random selection of the successor state takes the relative weights of the alternatives into account.
- Rules which call other components: correspondence checker, voicelead checker
- Wrap-rules are used to switch rules on or off for certain sections of the minuet, e.g. for chord-based sections, for scale-based sections or for sections governed by an end formula.

A full list of the rules can be found in chapter 3.6.

B.2.2 Structure

Structural relations are an important element of classical style. When the linear composition rules determine the set of possible following notes, the rules consider all structural elements, which govern the note to be composed. Structural elements differ in their origin, the computations, which are necessary to obey them and whether they govern a stretch of metrical time (e.g. two bars) or a given number of notes.

Top-down, absolute The structural elements in the structure plan are defined before the linear composition starts - thus the term *top-down*. Most of them are absolute, that means they don't depend on results of preceding linear composition.

Top-down elements govern a stretch of metrical time.

Top-Down, relative Relative top-down elements make a reference to earlier composition results.

Most correspondences between sections of the minuet belong to this category of structural elements. A correspondence requires similarity of its governed region with some music material, called the *model*. If the model is a preceding part of the composition, the correspondence is a relative structural element. The existence of such a correspondence is known before starting the composition process. Its exact requirements are determined during the composition process, when the model is composed.

The definition of a correspondence specifies the governed region, the location of the model, a function to compare the start note and one to compare the following intervals of the governed region with the model. Examples for correspondences are the sequence or the inversion.

Correspondences with an external model - musical material, which is given beforehand - are also possible. They are absolute structural elements.

Bottom-up Bottom-up structural elements are generated during linear composition by a composition rule rather than being defined beforehand by the composition plan. They are used to define melodic figures, which are allowed under the condition, that they are continued in a specific way.

Examples are the transition and the fourth suspension. The transition allows a non-harmonic note under the condition, that the transition is continued correctly. The (non-harmonic) transition note must be reached by a second and continued by a second in the same direction.

According to the composition levels, structure levels are defined. The macrostructure defines the length of the sentences and the type of their end-formulas. The microstructure defines all kind of details inside the sentences.

B.2.3 Voice Leading Rules

The voice leading rules restrict the relative movement of two voices. They combine linear (horizontal) aspects with simultaneous (vertical) aspects. A well-known example for a voice lead rule is *parallel fifth are forbidden*.

Voice lead rules are applied to a quadruple of four notes (two from each voice). The quadruples are generated from each pair of voices. The rules check the quadruple for forbidden phenomena. Some of the phenomena are only audible, when the notes are directly adjacent. Other phenomena are also forbidden, when the notes of the quadruple are separated by short notes or rests, e.g. accent parallels. This fact is modeled by the *reach* of rules. When generating the quadruples for the voice lead check, the system uses a simple model with the following reach classes: *direct*, *accent half beat*, *accent beat*, *accent bar*. Each class includes the preceding classes.

B.2.4 Pattern Matching of End Formulas

Patterns describe a set of data elements by similarity to a pattern. The pattern is an abstract intensional description of the data set. A special pattern

language for tonal music in a voice-oriented format has been developed in [Bec00].

End formula patterns as described in [Bud83] combine harmony with metric weight, chord position, and melodic movement. The exact octave and spacing of the harmony is not specified. Patterns for end formulas are represented in the pattern language mentioned above.

The macro structure describes the length of the sentences and the types and keys of their end formulas. The end formula is either a full or weak cadence⁵ and both can be either in the base key of the piece or on the fifth degree. A solution for a sentence is only accepted, if its end matches an end formula of the correct type.

B.2.5 Technical Aspects

The program is implemented in Common Lisp. It uses CLOS⁶, multimethods and multiple inheritance. The program follows the CLtL2-Standard with the exception of the macro `cp-make`, which uses the Meta Object Protocol, and which is therefore not guaranteed to be portable.

The system contains explicit knowledge: linear composition rules, end formula patterns, voice lead rules, and structural plans. For linear composition and for voice leading the inference machines which work on this knowledge are implemented as a compiler, which generates Lisp code. The Lisp code is then compiled with the Lisp-Compiler. The other representations are interpreted.

The program is called directly from the Lisp Read-Eval-Print-Loop (Command Line). This also provides an API⁷ for further research. Musical output is written as a file for the Lilypond [NN99] music typesetter. Lilypond generates the printed music (via \TeX and PostScript) and MIDI-files for listening online.

B.3 Results

The first two examples are typical results of the composition system. The melody contains both chord based and scale base bars. In the third sentence, there is a Fonte⁸-sequence. It consists of a two-bar motive with dominant-tonic harmony and its tonal sequence one note lower.

⁵Orig.: Kadenz, Absatz

⁶Common Lisp Object System

⁷Application Programmers Interface

⁸Fonte: ital/lat. into the well. Original term by J.Riepel[Rie52].

Minuet example 23 Minuet with Rhythmical Model

This minuet has a mixed melody with chord-based and scale-based segments. All bars not governed by an end formula use the same rhythmical model.

Minuet example 24 Minuet with Melodic Relations

The composition of this minuet considers interior structural relations:

- The bars 3 and 5 correspond with bar 1 by interval similarity
- Bars 6 and 14 correspond with bar 2 by equal rhythm
- Bar 13 corresponds with bar 1 by inverted interval similarity.

The interval similarity (and its inversion) governs a chord based section of the melody. The result is a chord analogon: The same number of steps inside a chord. This is frequently used in classical composition. In bar 7 on beat 1 is a fourth suspension, which is resolved on beat 2.

Minuet example 25 Bass to a given melody

The image displays a musical score for a minuet. It consists of two systems of two staves each. The top staff of each system is in treble clef and contains a given melody. The bottom staff is in bass clef and contains a composed bass line. The music is in 3/4 time. The first system covers measures 1 through 8. The second system starts at measure 9 and ends with a triplet in the bass line. The key signature has one sharp (F#).

Due to the system’s modular architecture, it can be used to compose a bass to a given minuet melody. In this example, only the bass is composed by the system. The melody is a teaching example from [Rie52]. The structure description and harmony base has been constructed by hand by analyzing Riepel’s melody. The melody contains a Monte⁹-Sequence.

B.4 Summary

The work opens opportunities for further research. The composition system can be extended to integrate more advanced musical elements, to add middle voices and to compose in larger forms. Due to the system’s modular architecture, parts can be reused for other musicological research, e.g. the voice lead checker can be used for analysis of existing compositions.

It is possible for a computer to compose music in a well-defined tonal style and genre. Preceding work demonstrated it for single subtasks of composition, this work demonstrates the integration of several subtasks. No mystical “*artists inspiration*” is necessary.

In the recent decades, symbolic AI has developed a broad range of techniques for knowledge representation and inference. They are sufficient for tonal composition. *The limiting factor is the musical knowledge.* Every single knowledge element has to be acquired, tested and integrated with the existing knowledge.

⁹ital./lat. up the hill, see Fonte

Bibliography

- [Bec00] BECKER, INA: *Computerunterstützte Analyse von tonaler Musik*. Diplomarbeit Nr. 1813, Stuttgart, Universität, Fakultät Informatik, Stuttgart, 2000. 104 Bl.
- [Ben60] BENARY, PETER: *Die deutsche Kompositionslehre des 18. Jhds.* Jenaer Beiträge zur Musikforschung. VEB Breitkopf und Härtel, Leipzig, 1960.
- [Ber95] BERGGREN, ULF: *Ars combinatoria – Algorithmic Construction of Sonata Movements by Means of Building Blocks Derived from W.A.Mozart’s Piano Sonatas*. Dissertation, Uppsala University, 1995. 174 S.
- [Blu97] BLUME, FRIEDRICH (editor): *Musik in Geschichte und Gegenwart*. Verwendete Auflagen: 1958,1997. Musiklexikon, 17 Bände.
- [Bud83] BUDDAY, WOLFGANG: *Grundlagen musikalischer Formen der Wiener Klassik: an Hand der zeitgenössischen Theorie von Joseph Riepel u. Heinrich Christoph Koch dargest. an Menuetten u. Sonatensätzen (1750 - 1790)*. Bärenreiter Verlag, Kassel, 1983.
- [Cop91] COPE, DAVID: *Computers and Musical Style*. Computer Music and Digital Audio Series. Oxford University Press, 1991.
- [Cop95] COPE, DAVID: *Experiments in Musical Intelligence*. The computer music and digital audio series ; 12. A-R Editions, Madison, Wisc., 1995. Enthält: 1 CD-ROM, XIV, 263 S.
- [Ebc86] EBCIOGLU, KEMAL: *An Expert System for Harmonization of Chorales in the Style of J.S.Bach*. Technical Report 86-09, Department of Computer Science, University of Buffalo - State University of New York, 1986.

- [HI59] HILLER, LEJAREN and LEONARD ISAACSON: *Experimental Music*. Greenwood Press, Westport CT, 1959.
- [Hö00] HÖNLE, NICOLA: *Objektorientiertes Patternmatching auf harmonisch analysierten Musikdaten*. Studienarbeit Nr. 1793, Stuttgart, Universität, Fakultät Informatik, Stuttgart, 2000.
- [KD84] KOSTKA, STEFAN and PAYNE DOROTHY: *Tonal Harmony*. McGraw-Hill, 1984.
- [KdRB91] KICZALES, GREGOR, JIM DES RIVIERES and DANIEL G. BOBROW (editors): *The Art of the Metaobject Protocol*. 1991.
- [Kee89] KEENE, SONYA E.: *Object-oriented Programming in Common Lisp: a Programmer's Guide to CLOS*. Addison-Wesley, Reading [u.a.], 1989. XIX, 266 S.
- [Koc82] KOCH, HEINRICH CHRISTOPH: *Versuch einer Anleitung zur Composition, 3 Bände entstanden 1782-1793*. Reprint 1969 Georg Olms Verlag Hildesheim, 1782.
- [Lan98] LANGNICKEL, JOACHIM: *Kantional-Harmonik, ein System zur Harmonisierung von Choralmelodien im Stil des Kantionalsatzes (17. Jhd)*. Universität Karlsruhe. Zu finden unter <http://i11www.ira.uka.de/musik/index.en.html>, 1998?
- [Lar92] LARNESTAM, ULF (alias ULF BERGGREN): *Simulation of Keyboard Sonata Movements on the Style of Mozart*. Computing in Musicology, 1992.
- [LJ83] LEHRDAHL, FRED and RAY JACKENDOFF: *A Generative Theory of Music*. MIT Press, Cambridge Massachusetts, 1983.
- [Lob44] LOBE, JOHANN CHRISTIAN: *Compositions-Lehre: oder umfassende Theorie von der thematischen Arbeit und den modernen Instrumentalformen*. Reprint 1988 Olms, Hildesheim, 1844.
- [Loy89] LOY, GARETH: *Composing with Computers – a Survey of Some Compositional Formalisms and Music Programming Languages*. In MATHEWS, MAX V. and JOHN R. PIERCE (editors): *Current Directions in Computer Music Research*, System Development Foundation Benchmark. MIT Press, 1989.

- [Mah87] MAHLING, ANDREAS: *Begriffsorientierte Darstellung musischen Wissens als Grundlage für wissensbasierte musikalische Systeme*. Studienarbeit Nr. 636, Universität Stuttgart, Inst. für Informatik, Stuttgart, 1987. 93 Bl.
- [Mat39] MATTHESON, JOHANN: *Der vollkommene Capellmeister*. Hamburg, 1739.
- [Max84] MAXWELL, HENRY JOHN: *An Artificial Intelligence Approach to Computer-Implemented Analysis of Harmony in Tonal Music*. Dissertation, Indiana University, 1984. 298 S.
- [Max92] MAXWELL, HENRY JOHN: *An Expert System for Harmonic Analysis of Tonal Music*. In BALABAN, MIRA, KEMAL EBCIOGLU and OTTO LASKE (editors): *Understanding Music with AI: Perspectives on Music Cognition*, System Development Foundation Benchmark. MIT Press, 1992.
- [Moz91] MOZART, WOLFGANG AMADEUS: *Klavierstücke 1762-1791*. Henle Verlag, München, 1791. Ausg. 1955/1983.
- [NN99] NIENHUYS, HAN-WEN and JAN NIEUWENHUIZEN: *Lilypond – The Gnu Music Typesetter – Documentation*. Technical Report, Available at <http://www.cs.uu.nl/people/hanwen/lilypond/index.html>, 1999.
- [Nor92] NORVIG, PETER: *Paradigms of Artificial Intelligence Programming: case studies in Common Lisp*. Morgan Kaufmann, San Mateo, 1992. 946 S.
- [Rie52] RIEPEL, JOSEPH: *Sämtliche Schriften zur Musiktheorie (entstanden 1752-1782)*, Ed. Thomas Emmerich. Reprint 1996 Böhlau Verlag, Wien, Köln, Weimar, 1752.
- [Rit69] RITZEL, FRED: *Die Entwicklung der Sonatenform im musiktheoretischen Schrifttum des 18. und 19. Jahrhunderts*. Neue Musikgeschichtliche Forschungen. Breitkopf und Härtel, Wiesbaden, 1969.
- [RK91] RICH, ELAINE and KEVIN KNIGHT: *Artificial intelligence*. McCraw-Hill, New York, NY [u.a.], 2nd ed. edition, 1991. XVII, 621 S.

- [RN95] RUSSELL, STUART J. and PETER NORVIG: *Artificial Intelligence: a modern approach*. Prentice Hall series in artificial intelligence. Prentice Hall, Englewood Cliffs, NJ, 1995. XXVIII, 932 S.
- [Sch89] SCHOTTSTAEDT, WILLIAM: *Automatic Counterpoint*. In MATH-
EWS, MAX V. and JOHN R. PIERCE (editors): *Current Direc-
tions in Computer Music Research*, System Development Foun-
dation Benchmark. MIT Press, 1989.
- [Ste90] STEELE, G.L. JR.: *Common LISP: The Language*. Digital Press,
second edition, 1990.
- [Tau94] TAUBE, HEINRICH: *Common Music*. Technical Report, Zentrum
für Kunst und Medientechnologie, Karlsruhe, Germany, 1994.
available at `ccrma-ftp.stanford.edu:/pub/Lisp/cm.tar.Z`.
- [Tau99] TAUBE, HEINRICH: *Automatic Tonal Analysis: Toward the Im-
plementation of a Music Theory Workbench*. Handout bei der
AISB'99 in Edinburgh, 1999.
- [uGM87] GÜNTHER MASSENKEIL, MARC HONEGGER UND: *Das große
Lexikon der Musik*. Herder Verlag, Freiburg, 1987.
- [Wö95] WÖTZER, GEORG: *Von Bach-Schemelli zur Bach-Maschine*. In
VÖLKL, HELMUT (editor): *Kirchenmusik als Erbe und Auftrag*,
pages 225–285. Carus Verlag, 1995.
- [WPA99] WIGGINS, GERAINT and SOMNUK PHON-AMNUASIUK: *The
Four-Part Harmonization Problem: A Comparison between Ge-
netic Algorithms and a Rule-Based System*. In WIGGINS,
GERAINT (editor): *Proceedings of the AISB'99 Symposium on
Musical Creativity*, pages 28–34. Edinburgh College of Art & Di-
vision of Informatics, University of Edinburgh, 1999.
- [WS97] WIGGINS, GERAINT and ALAN SMAILL: *Musical Knowledge:
what can Artificial Intelligence bring to the musician*. Techni-
cal Report 903, Department of Artificial Intelligence, University
of Edinburgh, 1997.

Lebenslauf Mathis Löthe

Geboren am 12.11.1970	in Berlin
1976-1980	Grundschule Kernen im Remstal
1980-1989	Friedrich-Schiller-Gymnasium Fellbach
3.5.1989	Allgemeine Hochschulreife (Abitur)
1.10.1989 - 30.9.1990	Wehrdienst in Horb und Osnabrück
15.10.1990 - 12.10 1995	Studium der Informatik
	an der Universität Stuttgart
Nebenfach	Mathematik
Studienschwerpunkt	Mensch-Maschine-Kommunikation
Studienarbeit	Referenzierung auf Objekte in terminologischen Beschreibungen
Diplomarbeit	Modellbasierte Feinbewegung eines Roboters auf Basis von Ultraschall
Tätigkeiten als wissenschaftliche Hilfskraft:	
	Tutor für theoretische und praktische Informatik
	Programmierung in Common Lisp
15.11.1995 - 31.3.2001	Wissenschaftlicher Mitarbeiter
	Institut für Informatik der Universität Stuttgart
	Abteilung Intelligente Systeme
Forschung	Computermusik (Dissertationsthema) Parsing natürlicher Sprache
Lehrtätigkeit	Künstliche Intelligenz Praktische Informatik
Nebentätigkeiten	Lehrauftrag für Systementwicklung/Expertensysteme an der Berufsakademie Stuttgart Trainer für Common Lisp
Seit 1.7.2001	Softwareentwickler und Trainer im Bereich Java-Technologie