Beschreibung der japanischen Verbmorphologie durch Vererbungshierarchien und Defaults

Studienarbeit

Aufgabensteller:
Prof. Dr. Bernd Mahr

Betreuer:
Dr. Manfred Stede

Dietrich Bollmann
Matrikelnummer: 154540
Stromstraße 40
10551 Berlin

14. März 1997

Inhaltsverzeichnis

Überblick

Die Morphologie von Verben zum Thema einer computerlinguistischen Arbeit zu machen, kann auf den ersten Blick zu Verwunderung führen: Der kombinatorischen Mächtigkeit von Syntax und Semantik entsprechend konzentrieren sich die meisten natürlichsprachlichen Systeme auf die Modellierung dieser Komponenten. Die Morphologie der Verben hingegen, die sich im Vergleich zu diesen Gebieten recht einfach zu gestalten scheint, wird meist durch Aufnahme der schon gebildeten Verbformen in ein sogenanntes Vollformlexikon behandelt.

Dieser Ansatz, der der Ersetzung algorithmischer Komplexität durch Speicherkomplexität entspricht, erklärt sich aus der Struktur der behandelten Sprachen: Fast immer handelt es sich um indogermanische Sprachen des sogenannten *flektierenden Types*, die nur eine relativ kleine, abgeschlossene Menge von Verbformen aufweisen.

Der Versuch jedoch, das Japanische mit diesem bewährten Mittel zu modellieren, scheitert schon bei einem kleinen Ausschnitt der Sprache. Dieses Mißlingen findet in der Struktur des Japanischen seine Erklärung: Wie auch das Türkische oder Ungarische rechnet sich das Japanische nicht zu den flektierenden, sondern zu den *agglutinierenden* Sprachen. Bezogen auf die Verbmorphologie bedeutet dieses, daß wir — anstatt einer Flexionstabelle, die alle Formen eines Verbes anschaulich beschreibt — zu jedem Verb durch rekursives Anfügen von Suffixen eine nahezu unerschöpfliche Menge möglicher Formen erhalten, deren vollständige Aufzählung nicht Ziel eines Lexikons sein kann.

Dank ihrer kombinatorischen Kraft erinnert die Morphologie der japanischen Verben damit weniger an die Verbmorphologie indoeuropäischer Sprachen, als vielmehr an die ebenfalls rekursiven Prozesse der Syntax.

Mit dieser Feststellung ist der Grundgedanke dieser Arbeit schon angesprochen: Durch die Ersetzung des Vollformlexikons durch ein morphosyntaktisches Lexikon und die Integration der morphologischen Prozesse in das generative Regelsystem kommen wir zu einer Morphosyntax des Japanischen. Die Darstellung der morphologischen Regeln und eines entsprechenden Lexikons bilden den Inhalt dieser Arbeit.

Analog zu den Tendenzen in der modernen Syntax lassen sich die morphologischen Regularitäten weitgehend durch das Zusammenspiel allgemeiner grammatischer Prinzipien und lexikalischer Information beschreiben. Um Redundanzen in der Darstellung der lexikalischen Information und den darauf aufsetzenden Prinzipien zu vermeiden, ergibt sich die Frage nach der Struktur des Lexikons. Diese Struktur läßt sich direkt aus der Natur der benötigten Information ableiten: Durch schrittweise Abstraktion und Klassifikation ergibt sich die Organisation der lexikalischen Einträge in natürlicher Weise als Vererbungshierarchie.

Die hierarchische, auf Vererbung von Information basierende Darstellung scheint jedoch in einer wesentlichen Eigenschaft natürlichen Sprachen zu widersprechen: Die Anordnug in Hierarchien ist durch die Klassifizierung gemäß sehr strenger Regeln gekennzeichnet; natürliche Sprachen hingegen zeichnen sich trotz stark regelhaften Verhaltens auch durch zahllose Abweichungen von Regeln aus. Dieser Widerspruch ist jedoch kein neuer, sondern für die Konfrontation logischer Systeme mit "wirklichen" Phänomenen typisch, und es wurden daher, unter dem Begriff *Defaultlogik* zusammengefaßt, verschiedene weniger strikte Ansätze zur Organisation von Information entwickelt.

Vererbungshierarchien und Defaults bieten damit eine angemessene Grundlage zur Darstellung unseres Untersuchungsgegenstandes: der Morphologie japanischer Verben.

Aus dem behandelten Gegenstand und den verwendeten Beschreibungsmitteln ergibt sich die Gliederung dieser Arbeit:

Kapitel 1 stellt eine Einführung in die linguistischen Grundlagen und Begriffe morphologischer Theorien dar. Anschließend werden diese Begriffe auf die japanische Sprache angewandt, so daß eine erste Skizze der behandelten Phänomene entsteht.

Kapitel 2 behandelt die formalen und linguistischen Annahmen, die unserer morphologischen Modellierung zugrundeliegen und deren Struktur wesentlich beeinflussen.

In *Kapitel 3* wird der Formalismus FLEX vorgestellt, den wir für unsere Modellierung verwenden. Auch wenn dieser Formalismus den linguistischen Anforderungen nicht in allen Punkten entgegenkommt, stellt er als Defaultbasierte Beschreibungslogik doch die wesentlichen Mittel unseres Lösungsansatzes zur Verfügung.

In *Kapitel 4* schließlich werden die bis dahin dargestellten Aspekte integriert: Die Morphologie japanischer Verben wird mit den Mitteln der Beschreibungslogik FLEX formalisiert.

In *Kapitel 5* werden die Ergebnisse dieser Arbeit noch einmal zusammengefaßt, und mögliche Ansätze für weitere Untersuchungen skizziert.

Mit Worten des Dankes möchte ich diesen Überblick beschließen. Das Thema dieser Arbeit wurde von Herrn Prof. Dr. Bernd Mahr gestellt, bei dem ich mich an dieser Stelle für die Aufgeschlossenheit meiner Interessenkombination gegenüber bedanken möchte. Herrn Dr. J. Joachim Quantz danke ich für die Unterstützung meiner Bemühungen, Linguistik, Japanologie und Informatik miteinander zu verbinden. Der allerherzlichste Dank gilt meinem Betreuer Dr. Manfred Stede: Ohne sein Interesse und seine Hilfe ist mir meine Arbeit — und insbesondere ihr Abschluß — nicht vorstellbar. Daß ich mein Studium meinen Interessen gemäß gestalten konnte, verdanke ich der Unterstützung meiner Eltern. Über ihren Langmut freue ich mich besonders. Ein weiteres großes Dankeschön möchte ich den Lehrern und Teilnehmern meiner Japanischkurse gegenüber ausdrücken. Aus ihrer Begleitung resultiert, daß mein Interesse an der japanischen Sprache und Kultur trotz aller Schwierigkeiten nur noch größer geworden ist.

1 Allgemeines zur Modifikation japanischer Verben

1.1 Einführende Bemerkungen zur Linguistik

In vielen Sprachen läßt sich ein Satz und die durch ihn ausgedrückte Bedeutung allein durch Änderung der Verbform in vielfältiger Weise variieren. Andersherum gesagt, nimmt die Veränderung der Verbform, traditionell als Konjugation oder Verbflexion[1] bezeichnet, eine zentrale Rolle in der Sprache und dementspechend auch in der Grammatik ein. So kann allein durch Veränderung der Verbform ein Geschehen als sich in der Vergangenheit, der Gegenwart oder der Zukunft abspielend dargestellt werden. Das Beschriebene kann als ein gerade beginnendes, als ein sich im Verlauf befindliches oder auch als schon abgeschlossenes Ereignis erscheinen. Es kann als Tatsache oder einfach als Möglichkeit beschrieben werden, als Wunsch des Sprechers, oder auch als etwas von ihm nur Vermutetes.

Die Beschreibung dieses Phänomenes bezogen auf die japanische Sprache und dessen anschließende computerlinguistische Modellierung bilden das Thema dieser Arbeit.

Um den Mechanismus dieser Modifikation in der japanischen Sprache besser beschreiben zu können, möchte ich im folgenden einige linguistische Grundbegriffe einführen, die im weiteren Vorgehen eine zentrale Rolle spielen werden. Anstelle genauer Definitionen, die den Rahmen dieser Arbeit sprengen würden, möchte ich mich dabei mit einer informellen Einführung begnügen.

1.1.1 Inhaltliche und formale Untersuchungen der Sprache, Morphologie und Syntax

Der komplexen Struktur sprachlicher Erscheinungen entsprechend unterteilt sich die Linguistik in verschiedene Teildisziplinen, die sich jeweils nur mit bestimmten Aspekten der Sprache befassen, zusammen aber diese in ihrer Gesamtheit zu erfassen versuchen. So unterscheidet man zwischen inhaltlichen, sich auf Bedeutung beziehenden Untersuchungen und solchen, die

1 Der Begriff der *Verbflexion* wird in dieser Arbeit in einer allgemeinen und einer speziellen Bedeutung benutzt: In seiner allgemeinen Verwendung ist er mit dem Begriff *Konjugation* vertauschbar; in seiner speziellen Verwendung hingegen steht der Terminus *Verbflexion* für die Suffigierung bestimmter ausgezeichneter Morpheme, die den morphologischen Prozeß abschließen, und das Ergebnis damit zu einem Bestandteil der nächsthöheren Beschreibungsebene — d.h. der Syntax — machen. In welchem Sinn der Begriff *Verbflexion* gebraucht wird, geht aus dem jeweiligen Kontext hervor.

sich mit der formalen Struktur von Sprache befassen. Die Letzteren wiederum untergliedern ihren Untersuchungsbereich in sich hierarchisch zueinander verhaltende Beschreibungsebenen: die Phonologie, die Morphologie und die Syntax. Während sich die Morphologie mit der inneren Struktur der Wörter befaßt, untersucht die Syntax die Fügung dieser zu komplexen Syntagmen und Sätzen. Die Phonologie auf der anderen Seite beschäftige sich mit der bedeutungsrelevanten Lautverteilung einer Sprache, und liefert damit die Bausteine der Morphologie.

Die hier behandelten Phänomene japanischer Verbmodifikation lassen sich größtenteils der Morphologie zuordnen, greifen aber — und dabei ist die Grenze zwischen Morphologie und Syntax oft schwer zu ziehen — in syntaktische Bereiche über. Da der wichtigere Teil meiner Arbeit sich innerhalb der Morphologie ansiedeln läßt, und die betrachteten syntaktischen Regularitäten von sehr einfacher Natur sind, kreisen meine Überlegungen vor allem um morphologische Fragestellungen, während die Syntax nur am Rande behandelt wird. Im folgenden möchte ich, um die anschließende Analyse der japanischen Verben vorzubereiten, kurz auf einige zentrale Begriffe der Morphologie und der Syntax eingehen.

1.1.2 Morphem, Morph und Allomorph

Wie eben angedeutet bedient sich das Japanische — wie auch die meisten anderen Sprachen — zur Modifikation der Verben zweier unterschiedlicher Mechanismen: Morphologische Veränderung und Einbettung in syntaktische Konstruktionen. Zum Verständnis der morphologischen Mechanismen sind die Begriffe *Morphem*, *Morph* und *Allomorph* unabdingbar und sollen nun anhand einfacher Beispiele eingeführt werden.

Versucht man, die konkreten Erscheinungen, also die orthographische oder phonologische Form von Wörtern, näher zu untersuchen, kann man durch den Vergleich ähnlicher Wortformen zu dem Ergebnis kommen, daß bestimmte Teile eines Wortes bestimmte Informationen ausdrücken:

(ex 1) *a.* *i.* kill
ii. form
b. *i.* killed
ii. formed

Der Vergleich der Vergangenheitsformen der Verben 'to kill' und 'to form' mit ihren Präsensformen, wie wir sie in (ex 1) vor uns haben, führt beispielsweise zu der Beobachtung, daß sich Vergangenheitsformen in zwei Bestandteile zerlegen lassen, von denen eines auf die lexikalische Bedeutung verweist, Während das andere dieser etwas wie "Vergangenheit" zufügt. Etwas formeller könnte man schreiben:

(ex 2) Verb + ed = Vergangenheitsform des Verbes

und im Falle unseres Beispiels:

(ex 3) *a.* kill + ed = killed *bzw.*
b. form + ed = formed

Solche durch Vergleich und Analyse gewonnenen Bestandteile der Wortformen, die im Idealfall nur eine grammatikalische oder lexikalische Information enthalten, nennt man *Morphe*.

Bei der Verallgemeinerung solcher Beobachtungen stößt man jedoch schnell auf Probleme. Wie soll man beispielsweise die offenbar analog sich verhaltende Wortform 'went' des Verbes 'to go', die ja der Gleichung 'go + ed = went' genügen müßte, unterteilen. Eine Lösung des Problems kann darin bestehen, die beobachteten Regularitäten nicht als direkt auf der Ebene der orthographischen oder phonologischen Oberfläche wirkend aufzufassen, sondern als solche eines abstrakten Systems, das sich auf — je nach Verb — unterschiedlich komplexen Regeln folgende Weise schließlich in den tatsächlichen Wortformen ausdrückt. Man kommt zu abstrakten Regularitäten, die etwa die folgende Gestalt annehmen könnten:

	a.	{kill}	+	{ed}	→	'killed'
(ex 4)	*b.*	{form}	+	{ed}	→	'formed'
	c.	{go}	+	{ed}	→	'went'

Zwei abstrakte Einheiten, die eine die lexikalische, die andere die temporale Information tragend, addieren sich, und realisieren sich auf mehr oder weniger komplexe Art in den tatsächlichen Wortformen. Die geschweiften Klammern sollen hierbei die eben bezeichnete Abstraktion, die einfachen Anführungszeichen hingegen die tatsächliche Realisierung verdeutlichen.

Durch das Vergleichen sich analog verhaltender Formen und das Untergliedern dieser auf möglichst ähnliche Art und Weise kommt man also zu einer Zerlegung der Wortformen in Bestandteile, die im Idealfall nur eine Information enthalten, und Morphe genannt werden. Da bei gleicher Information die Form solcher Morphe jedoch in verschiedenen Kontexten variieren kann, faßt man diese in Klassen zusammen. Solche Klassen gleichartige Information tragender Morphe bezeichnet man dann als *Morpheme*[2], verschiedene Morphe einer Klasse als *Allomorphe* relativ zum Morphem oder einem Element der gleichen Klasse.

Neben den Morphen und Morphemen auf der einen Seite erhalten wir bei der Analyse der Formen gleichzeitig eine Menge schematisierter, aus Morphemen zusammengesetzter "Wortbaupläne" auf der anderen Seite. In dem eben betrachteten Fall haben wir beispielsweise mit dem Schema

(ex 5) {Verbmorphem} + {Vergangenheitsmorphem}

eine Regel bzw. einen Bauplan zur Bildung der Vergangenheitsformen von Verben.

Bei der Bildung konkreter Wörter mit Hilfe eines solchen Systems aus Morphemen und Wortbauplänen werden zu den einzelnen Morphemen eines gegebenen Wortbauplans nach im weiteren noch zu untersuchenden Regeln Morphe — bzw. bestimmte Elemente der den Morphemen entsprechenden Klassen — ausgewählt und zu einem Wort verbunden. Später werden wir sehen, wie man den hier vorläufig eingeführten Begriff *Wortbauplan* durch ein rekursives Regelsystem ersetzen und auf diese Weise zu einem sehr viel ausdrucksfähigeren Formalismus gelangen kann.

1.1.3 Zum Verhältnis der Morpheme zu ihren Realisierungen: Flexion und Agglutination

Wir haben uns auf zwei Ebenen der grammatischen Betrachtung festgelegt: einer abstrakten, in der grammatische Regularitäten schematisch dargestellt werden können, und einer konkreten, die die tatsächliche Realisierung dieser abstrakten Entitäten zum Inhalt hat, und mit dieser entsprechend über Abstraktion bzw. Realisierung verbunden ist. Zu der ersteren gehören *Morpheme* als grammatische Einheiten, zu der letzteren die *Morphe* als deren Realisierung in einer konkreten Wortform.

Außerdem haben wir bemerkt, daß sich das Verhältnis von Oberflächenform zu Morph und Morphem nicht immer als einfache Abbildung gestaltet, sondern komplexer Natur sein kann. Zwei Extreme dieses Verhältnisses lassen sich — neben möglichen Kombinationen — vorstellen:

synthetische Form — Eine Menge von Morphemen drückt sich mittels einer komplexen Regel in einer Form aus, in der die den Morphemen entsprechenden einzelnen Morphe nicht mehr als Segmente erkennbar sind.

---

2 In unserem Beispiel wäre also mit {ed} das Vergangenheit ausdrückende Morphem gemeint

Die entstehenden Formen können also nicht durch lineare Segmentation in die einzelnen Morphe unterteilt werden. Man spricht in diesem Fall von *nicht-segmentalen Morphen*, bzw. *Morph-Kombinationen*. Die Präteritumform 'went' des Verbs 'to go' ist ein Beispiel für diese Art synthetischer Realisierung.

analytische Form — Morphem und Morph lassen sich einander direkt zuordnen; die Realisierung kommt folglich durch lineare Reihung der Morphe zu einer Morphkette zustande. Für diesen Fall sogenannter *segmentaler Morphe* haben wir in der Präteritumform 'killed' des Verbs 'to kill' ein gutes Beispiel.

Bezogen auf einzelne Wortklassen wie der der *Verben* oder *Nomen* wird der zuerst beschriebene, synthetische Typus in der Linguistik als *Flexion*[3] und der analytische Typus als *Agglutination* bezeichnet.

Sprachen neigen dazu, zumindest hinsichtlich einzelner Wortklassen, einen Bildungstyp vorzuziehen, und lassen sich dementsprechend in *eher agglutinierende* und *eher flektierende Sprachen* einteilen. Das Lateinische beispielsweise ist für seine verschliffenen, synthetischen Verb- bzw. Nominalformen bekannt, und wird dementsprechend zu den flektierenden Sprachen gerechnet. Das Türkische hingegen modifiziert Nomen bzw. Verben eher mit Mitteln der Agglutination und rechnet sich folglich zu den agglutinierenden Sprachen.

1.2 Zur Morphologie und Syntax japanischer Verben

Wir kommen nun zur Einführung in den Ausschnitt der japanischen Grammatik, der dieser Arbeit zugrunde liegt. Zuerst wollen wir dazu die Unterscheidung von Sprachen in agglutinierende und flektierende zur Kennzeichnung der japanischen Verbmorphologie verwenden.

1.2.1 Die japanische Verbmodifikation als Mischform von Flexion und Agglutination

Die japanische Sprache wird traditionell den agglutinierenden Sprachen zugeordnet. Verben und Adjektive, aber auch Nomina können auf vielerlei Weise

3 Der Begriff *Flexion* wird in der Linguistik in zwei verschiedenen Bedeutungen verwandt und kann sich neben der eben eingeführten allgemeinen Verwendung auch speziell auf die Modifikation von Nomina bzw. Verben beziehen. Wir haben diese Ausprägung des Begriffes *Flexion* schon in der Form *Verbflexion* kennengelernt. Die von mir benutzte Umschreibung "Modifikation japanischer Verben" ließe sich unter bestimmten Einschränkungen also durch "Flexion japanischer Verben" ersetzen. Um der Verwechselung vorzubeugen, und auch, weil "Modifikation" syntaktische Modifikationen mit einschließen kann, habe ich mich jedoch entschlossen, den Begriff *Modifikation* zu benutzen.

durch Anhängen von Suffixen modifiziert werden. Betrachten wir dazu, unter Vernachlässigung des jeweils letzten Morphes, das nur dem Abschluß des morphologischen Prozesses dient, die folgenden sich auseinander ableitenden Formen:

(ex 6)
a. 書く
kak-u
schreiben

b. 書かせる
kak-ase-ru
zum Schreiben veranlassen

c. 書かせられる
kak-ase-rare-ru
zum Schreiben veranlaßt werden

d. 書かせられます
kak-ase-rare-mas-u
zum Schreiben veranlaßt werden (höflich)

e. 書かせられました
kak-ase-rare-mas-ita-ϕ
zum Schreiben veranlaßt worden sein (höflich)

In dem einfachsten Falle (ex 6*a*), bezeichnet die Verbform einfach das Präsens des Verbes 'kaku'. Nach und nach wird dieses dann durch Suffigierung anderer Morpheme mehr und mehr modifiziert, so daß im letzten Falle (ex 6*e*) schließlich ein Ausdruck mit der Bedeutung 'zum Schreiben veranlaßt worden sein' entsteht, und zwar in einer Form, die im relativ höflichen Umgang zweier Gesprächspartner Verwendung finden könnte.

Wir haben es offensichtlich mit einer Modifikation durch Agglutinierung von Suffixen zu tun. Betrachtet man die sich auf diese Weise bildenden Formen jedoch genauer, ist die Verbindung der Morphe zu Ketten zwar offensichtlich, eine Teilung dieser in einzelne Morphe jedoch problematisch. Entsprechend findet man in der Tradition japanischer Sprachbeschreibungen eine Vielzahl unterschiedlicher Ansätze, solche komplexen Formen in einzelne Morphe zu zerlegen. Betrachten wir die folgenden, einfachen Beispiele[4]:

(ex 7) *a.* *i.* miru $\rightarrow$ mita

4 Das hier dargestellte Beispiel dient zur Illustration der Mechanismen der japanischen Verbmodifikation und unterscheidet sich von der endgültigen Modellierung insofern, als daß die hier dargestellten Potentialformen später als zwei unterschiedlichen Varianten dieser Modifikation angehörig analysiert werden.

ii. kaku → kaita

b. *i.* miru → mirareru

ii. kaku → kakeru

Beide Beispiele zeigen jeweils die Präsensform der Verben 'miru' (sehen) und 'kaku' (schreiben) und die diesen entsprechenden Präteritumformen in (ex 7*a*) (sah, schrieb) bzw. deren Potentialformen in (ex 7*b*) (sehen können, schreiben können). Zwar lassen sich in (ex 7*a*) 'mi' bzw. 'ka' als Stämme und 'ta' als Endung vermuten, jedoch kommt es bei (ex 7*a*-*ii*) zur Einfügung des Vokales 'i', und es ist schwer zu sagen, ob sich dieser zum Stamm oder zur Endung rechnen läßt. Im Beispiel (ex 7*b*) hingegen scheint es — nähme man 'mir' und 'kak' als jeweilige Stämme an — im Fall (ex 7*b*-*i*) zum Einschub der Folge 'ra' zu kommen, und wiederum ist es schwer zu sagen, ob diese eher dem Stamm oder der Endung zuzurechnen wäre. Noch schwieriger wird es, wollte man für beide Beispiele zusammen festlegen, welcher Teil der Formen den Stamm und welcher die Endung bildet.

Die sehr unterschiedliche Behandlung dieser Formen in japanischen Grammatiken läßt darauf schließen, daß auch von muttersprachlichen Sprechern keine klare Grenze zwischen den Morphen empfunden wird, sondern die Formen in etwas irritierender Weise als "Einheit" zweier ineinander verschmelzender "Teile" ohne feste Abgrenzung empfunden werden.

Diese Verschmelzung von Morphen aber ist typisch für den flektierenden Sprachbau, und man muß folglich die japanische Verbmodifikation als Mischform von Agglutination und Flexion betrachten, wobei der agglutinierende Typ der vorherrschende ist.

1.2.2 Syntaktische Mittel der Verbmodifikation

Im letzten Abschnitt haben wir gesehen, wie Verben im Japanischen auf morphologischer Ebene durch Agglutination von Suffixen modifiziert werden können. In diesem Abschnitt nun wollen wir uns mit der zweiten Möglichkeit der Verbmodifikation befassen, der Modifikation japanischer Verben durch Einbettung in Hilfsverbkonstruktionen.

Verben und Hilfsverben

Verben spielen eine zentrale Rolle in der Grammatik. Auf syntaktischer Ebene öffnen sie die Stellen für die anderen, zusammen mit ihnen den Satz bildenden Konstituenten, die sie dann, auf semantischer Ebene, in eine durch die lexikalische Bedeutung des Verbs festgelegte semantische Relation zueinander stellen.

In bestimmten Konstruktionen jedoch treten Verben nur in eingeschränkter, z.B. syntaktisch-strukturspendender Funktion auf, oder verbinden sich mit anderen Verben, um diesen einen bestimmten Bedeutungsaspekt hinzuzufügen. Solche, nur eingeschränkt lexikalische Bedeutung tragenden Verben, nennt man *Hilfsverben*.

Im Deutschen z.B. werden — ähnlich wie im Japanischen — Tempusformen außer durch morphologische Mittel auch mit bestimmten Hilfsverben gebildet, die ein anderes Vollverb in einer bestimmten Form als Argument nehmen. In den Sätzen

(ex 8) *a.* Er ist gelaufen.
b. Sie wird zurückkehren.
c. Er möchte essen.

beispielsweise, handelt es sich bei 'ist', 'wird' und 'möchte' um Hilfsverben, die dem Vollverb eine temporale bzw. modale Information hinzufügen.

Hilfsverben im Japanischen

Entgegen der üblichen Verwendung des Terminus *Hilfsverb* in der japanischen Grammatik sollen im folgenden hiermit nur solche Entitäten gemeint sein, die sich nicht auf morphologischer sondern auf syntaktischer Ebene mit anderen Verben verbinden. Natürlich ist durch solch eine definitorische Unterscheidung die schon vorher angesprochene Problematik der Abgrenzung der grammatikalischen Beschreibungsebenen angesprochen. Diese soll jedoch nicht in bezug auf die linguistischen und systematischen Motivationen, die zu den hier getroffenen Unterscheidungen geführt haben, weiter diskutiert, sondern nur als vorhanden zur Kenntnis genommen werden. Wann in der hier dargestellten Modellierung ein morphologischer Prozess als abgeschlossen bezeichnet und das Resultat folglich zum Objekt syntaktischer Vorgänge wird, soll erst später erklärt werden. Für den Augenblick mögen die folgenden Beispiele genügen:

(ex 9) *a.* 書きませんでした
kakimasen desita
(etw.) wurde nicht geschrieben (höflich)

b. 書くでしょう
kaku desyou
(etw.) wird vermutlich geschrieben (höflich, Vermutung)

c. 書かせられたくなかったです
kakaseraretaku nakatta desu
es nicht gerne mögen, zum Schreiben veranlaßt zu werden (höflich)

In allen Beispielen handelt es sich um Formen des Verbes 書く (*kaku*; schreiben): In (ex 9*a*) wird die negierte, höfliche Form 'kakimasen' durch 'desita', die höfliche Vergangenheitform des Partikelverbes 'da', in die Vergangenheit gesetzt. Im Beispiel (ex 9*b*) dient eben dieses Partikelverb 'da' in der höflichen volitional-Form 'desyou' zum Ausdruck einer Vermutung (Presumptiv) und schließt sich dazu an die höflichkeitsneutrale Form 'kaku' des Verbes 'kaku' an. Im Fall (ex 9*c*) schließlich wird die adverbialisierte, in den Voluntativ gesetzte Kausativ-Passiv-Form von 'kaku', d.h. die Form 'kakaseraretaku' zuerst durch die negierte Vergangenheitsform 'nakatta' des Hilfsverbes 'aru' ebenso negiert und in die Vergangenheit gesetzt; anschließend wird der so entstandene Komplex — also 'kakaseraretaku nakatta' — durch 'desu', der höflichen Variante von 'da', zusätzlich mit dem Aspekt der Höflichkeit ausgezeichnet.

In allen Beispielen dienen Hilfsverben dazu, bestimmten Verbformen bestimmte neue grammatikalische Merkmale hinzuzufügen, die diese selbst nicht bilden können. Dabei müssen bestimmte Kennzeichnungen — wie im ersten Beispiel die Höflichkeit — übrigens sowohl am Vollverb, als auch am Hilfsverb, ausgedrückt werden.

Literatur:
Eine sehr gute Einführung in die Grundlagen der Linguistik ist Lyons 1968 oder auch Grewendorf et al. 1996. Während die "Introduction to Theoretical Linguistics" von Lyons eher aus einer allgemeinen, nicht einer speziellen Sprachtheorie verpflichteten Perspektive einen Überblick über die moderne Linguistik verschafft, werden die linguistischen Problembereiche in dem Werk "Sprachliches Wissen" von Grewendorf, Hamm und Sternefeld aus der Sichtweise der sogenannten "Generativen Transformationsgrammatik", die von Noam Chomsky begründet wurde, dargestellt.

Für die Übertragung der dargestellten Begriffe auf die japanische Grammatik habe ich mich — neben verschiedenen eher didaktisch ausgerichteten Grammatiken wie Lewin 1990, McClain 1981 und Makino und Tsutsui 1986 und Darstellungen speziell zur Verbflexion wie Ishii et al. 1989 — vor allem auf die sehr ausführliche Darstellung in Rickmeyer 1995 (s.h. auch Rickmeyer 1983) gestützt.

Arbeiten zu verwandten Themen sind weiterhin: Iida et al. 1994, Kato 1985 und Udo 1982.

2 Prinzipien der Modellierung

Im ersten Kapitel haben wie uns mit den grundlegenden, linguistischen Aspekten der Modifikation japanischer Verben beschäftigt. Das Ziel dieser Arbeit ist jedoch, über eine solche intuitive Näherung hinaus die beschriebenen Aspekte der japanischen Grammatik exakt und für den Computer ausführbar zu formalisieren. Dem Interesse an computerlinguistischen Modellierungen entsprechend, gibt es eine große Anzahl möglicher, schon beschrittener Wege, zu einer solchen zu gelangen. Ich möchte deshalb nun die wesentlichen Überlegungen und Prinzipien, die dem hier beschriebenen Vorgehen zugrundeliegen, ausführen.

Ein Teil dieser Überlegungen und Prinzipien bezieht sich auf die *Form* in der wir die beschriebenen Sachverhalte darstellen; der andere Teil betrifft die *linguistischen* Grundannahmen unseres Modells. Wir unterscheiden daher in der folgenden Darstellung zwischen *formalen* und *linguistischen* Prinzipien.

2.1 Formale Prinzipien

2.1.1 *Form* und *linguistische Eigenschaften* als Aspekte *linguistischer Zeichen*

Zum Verständnis der folgenden Darstellungen ist der Begriff *linguistisches Zeichen* wesentlich. Mit ihm wird eine Informationsstruktur bezeichnet, die die für eine Sprachbeschreibung notwendigen Eigenschaften linguistischer Entitäten adäquat zu beschreiben erlaubt. Linguistische Entitäten haben eine doppelte Natur, die sich auch in der Gestalt der Zeichen spiegeln muß: Sie vereinen in sich den Aspekt der *Form* und den Aspekt der einer Form anhaftenden *linguistischen Eigenschaften*. Mit der *Form* des Morphes 'kaka' zum Verbmorphem {kaku} beispielsweise ist die Folge phonologischer bzw. orthographischer Zeichen 'k-a-k-a' gemeint. Die *linguistischen Eigenschaften* des Morphes 'kaka' hingegen sind unter anderem seine Bedeutung, die Angehörigkeit zum Morphem {kaku} und vor allem das Vermögen, sich nur mit bestimmten anderen Morphen verbinden zu können.

Beispiel 1:
Veranschaulichen wir uns die Unterscheidung von *Form* und *linguistischen Eigenschaften* und deren Zusammenwirken durch den Vergleich mit einem Puzzle, durch das die Kausativ-Perfekt Formen der Verben 書く (*kaku*; schreiben) und 見る (*miru*; sehen) modelliert werden soll:

(ex 10) Der Kausativ-Perfekt der Verben 書く (*kaku*) und 見る (*miru*)

a. 書かせた
kakaseta
zum Schreiben veranlaßt worden sein

b. 見させた
misaseta
zum Sehen veranlaßt worden sein

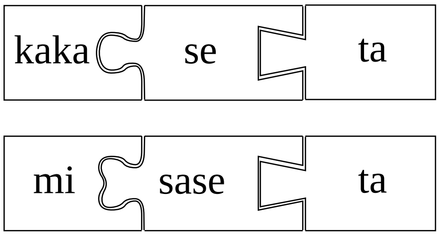
Wir legen zur Bildung dieser Formen die Verkettung der folgenden Morpheme zugrunde:

(ex 11) {Verb} + {Kausativ} + {Perfekt}

und analysieren die Formen als Morphketten der folgenden Gestalt:

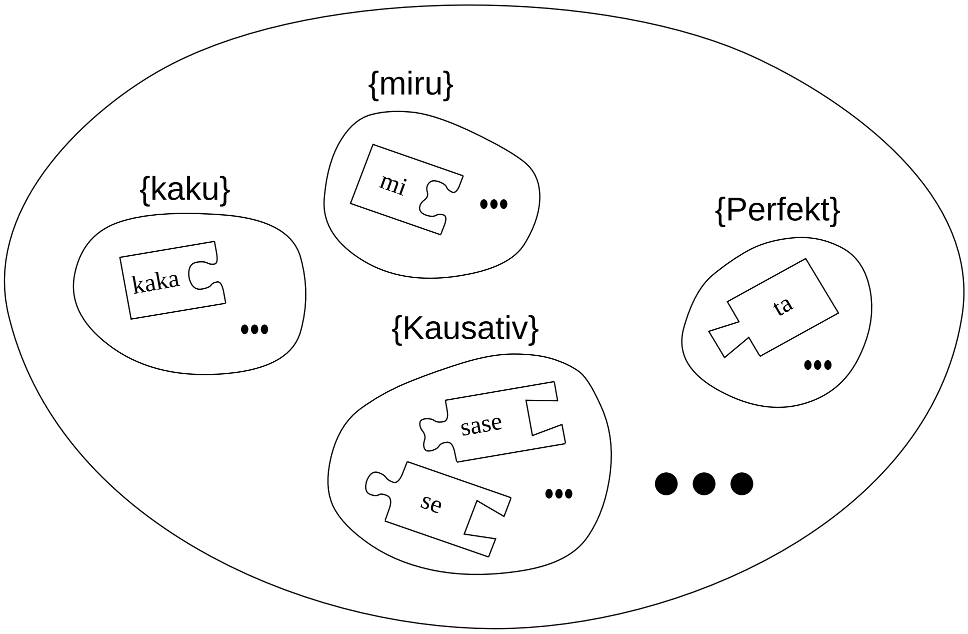
(ex 12) *a.* kaka + se + ta
b. mi + sase + ta

Wir wollen die morphologische Struktur dieser Formen folgendermaßen auf ein Puzzle abbilden:



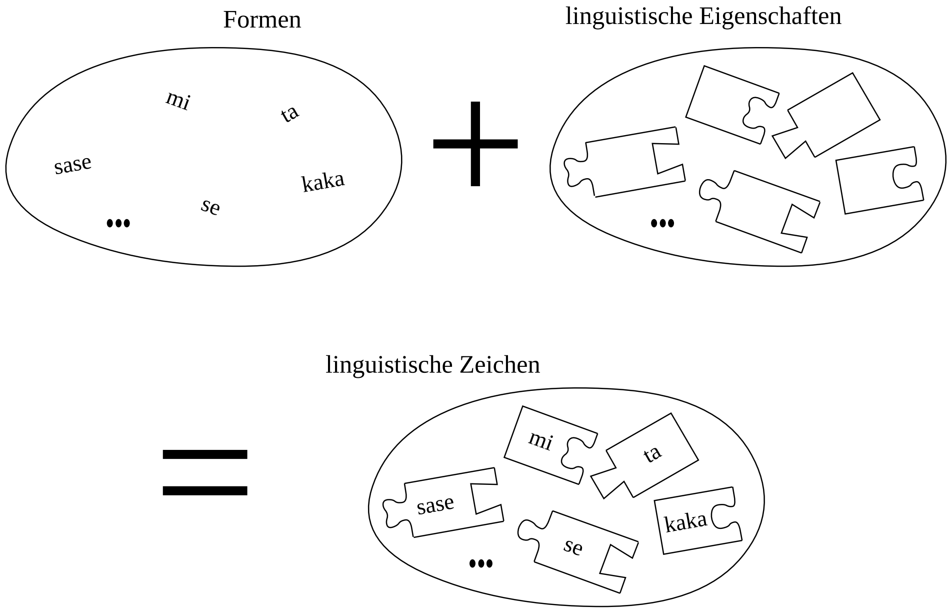
(fig 1) *Modellierung der morphologischen Struktur der Formen ‘kakaseta’ und ‘misaseta’ durch ein Puzzle*

Die an den betrachteten Verbformen beteiligten *Morphe* werden als Puzzleteile dargestellt; *Morpheme* als Untermengen der Menge aller Bausteine des Puzzles bzw. als Äquivalenzmengen relativ zur Relation “A ist *Allomorph* von B":



(fig 2) *Übertragung der Begriffe Morph, Morphem und Allomorph auf ein "morphologisches Puzzle"*

Die (atomaren) linguistischen Zeichen entsprechen in unserer Analogie den Teilen des Puzzles. Die Unterscheidung der Aspekte eines Zeichens in solche der *Form* und solche der (mit der Form assoziierten) *linguistischen Eigenschaften* läßt sich nun in der folgenden Weise an den Puzzleteilen wiederfinden:



(fig 3) *Formen und linguistische Eigenschaften als Aspekte linguistischer Zeichen*

Form — Jedes einzelne Teil des Puzzles trägt als "Bild" die dem Morph entsprechende Zeichenfolge. Diese Zeichenfolge beschreibt den Aspekt der *Form* einer linguistischen Entität, bzw. eines linguistischen Zeichens. Ohne die *Form* bzw. das Bild der einzelnen Puzzleteile ließen sich diese zwar in der gleichen Weise zusammensetzen, würden dabei aber keine Informationen über die entstehende Form des Gesamtkomplexes liefern.

linguistische Eigenschaften — Jedes einzelne Puzzleteil ist durch eine bestimmte "Außengestalt" ("Haken" und "Ösen") gekennzeichnet, dank derer es sich nur mit bestimmten anderen Puzzleteilen verbinden kann. Die Außengestalt steht für die *linguistischen Eigenschaften* eines Zeichens. Wären diese Eigenschaften nicht gegeben, ließe sich jedes Teil mit jedem beliebigen anderen verbinden, und man könnte neben den korrekten auch ungrammatische Formen wie z.B. 'kaka-sase-ta' ableiten.[5]

5 Natürlich haben die *linguistischen Eigenschaften* auch andere Funktionen, als alleine die kombinatorischen Möglichkeiten einzuschränken. So ordnen sie den Morphen beispielsweise eine *Bedeutung* zu, und liefern — in Ergänzung zu bestimmten globalen Prinzipien — die Informationen, die aus diesen Bedeutungsfragmenten eine Gesamtbedeutung zu konstruieren gestatten. Zu diesen (in unserem Modell allerdings weniger wichtigen) Eigenschaften — die übrigens ebenfalls, in diesem Falle aus semantischer Sicht, auf kombinatorischen Möglichkeiten zurückwirken können — werden wir jedoch erst später kommen.

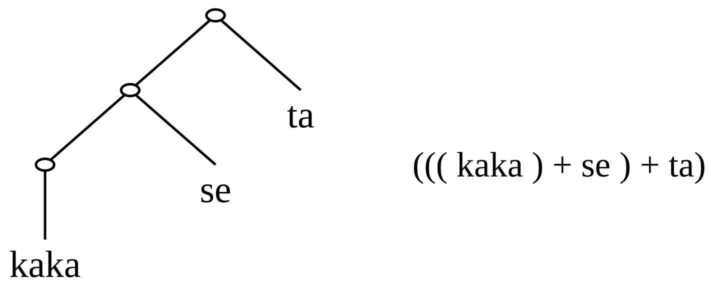
Erst wenn *Form* und *linguistische Eigenschaften* einander ergänzen, erhalten wir eine adäquate Modellierung der linguistischen Gesetzmäßigkeiten bei der Modifikation von Verben: Beide Aspekte zusammen beschreiben die Menge aller korrekten und nur der korrekten Kombinationen von Morphen zu Verbformen, gegeben eine bestimmte Kette von Morphemen.

2.1.2 Die Konkatenation als einzig erlaubte Grundoperation auf *Formen*

Der Begriff der *Konkatenation*, unter dem wir im folgenden das einfache Aneinanderketten zweier Formen zu einer neuen Form verstehen, steht für die wichtigste formale Einschränkung unseres Ansatzes: Alle Variationen der linguistischen Form müssen auf die Konkatenation, als einzig erlaubter Grundoperation auf orthographischen oder phonologischen Formen, zurückgeführt werden. Anders ausgedrückt besteht der Prozeß der Bildung bzw. Analyse von Formen einfach in dem Zusammenfügen kleinerer zu größeren bzw. der Trennung komplexerer in weniger komplexe:

(ex 13) kaka + se + ta = kakaseta

Der Hinweis darauf, daß eine komplexe Form aus zwei ebenfalls schon komplexen entstehen kann, verweist auf den rekursiven Charakter dieses Vorganges. Atomare Einheiten verbinden sich zu komplexen, komplexe wiederum zu noch komplexeren, usw. Man kommt zu ineinander verschachtelten, als Baum oder Klammerstruktur darstellbaren Gebilden:



(fig 4) *rekursive Bildung von Formen*

Beispiel 2:
Da die *Komposition* als Grundprinzip der Modellierung von ganz wesentlicher Bedeutung ist, möchte ich sie — späteren Überlegungen vorgreifend — in der Form eines linguistischen Beispiels genauer darstellen. Wir haben schon die folgenden Formen kennengelernt:

(ex 14) 書く → 書いた
kaku *kaita*
(schreiben) (geschrieben haben)

書く (*kaku*) ist die Präsensform des Verbes 書く (*kaku*; schreiben), 書いた (*kaita*) die diesem entsprechende Präteritumform. Traditionell wird die Ableitung dieser letzteren nun — unter Verweis auf eine obligatorische (phonologische) Verschleifung (音便 (*onbin*)) — folgendermaßen beschrieben:

(ex 15) {kaku} + {ta} → 'kaki.ta' → 'kaita'

Die Präteritumform des Verbes 'kaku' bildet sich also durch Anfügen des Präteritummorphemes {ta} an das Stammmorphem {kaku}. Da {kaku} sich durch verschiedene Allomorphe realisieren kann, die in der Konjugation traditionell als *Stämme* oder *Basisformen* bezeichnet werden, und da das immer nur durch das Morph 'ta' realisierte Präteritummorphem {ta} sich prinzipiell nur mit einer dieser Basisformen — die hier als *i-Basis*[6] bezeichnet werden soll — verbinden kann, und die in diesem Fall dem Allomorph 'kaki' entspricht, entsteht die Form 'kaki.ta'. Nun kommt es aber bei der Verbindung des Präteritummorphems {ta} und der i-Basis eines Verbes fast prinzipiell zu "Verschleifungen" zwischen den entsprechenden Morphen. Diese Verschleifungen, die zwar nicht allein durch phonologische Merkmale, sondern eigentlich durch das Präteritummorphem {ta} ausgelöst werden, sorgen nun — den phonologischen Regularitäten der japanischen Sprache entsprechend — für die Transformation von 'kaki.ta' in die Form 'kaita', die der tatsächlich ausgesprochenen entspricht.

Wir haben es also bei der traditionellen Ableitung formal gesehen mit einer *Konkatenation* und einer anschließenden *Transformation* zu tun:

(ex 16) kaki + ta $\xrightarrow{Konkatenation}$ kaki.ta $\xrightarrow{Transformation}$ kaita

Als grundlegendes Prinzip der hier eingeschlagenen Formalisierung haben wir aber gerade die Konkatenation als *einzige* Grundoperation gefordert. Es folgt, daß die eben angeführte, der japanischen Grammatiktradition folgende Ableitung nicht in unserem Formalismus dargestellt werden kann. Eine mögliche Ableitung könnte hingegen eine der folgenden sein:

(ex 17) *a.* kai + ta → kaita
b. ka + ita → kaita

6 Die in dieser Arbeit als *i-Basis* bezeichnete Basisform entspricht der 連用形-Form (*renyoukei-Form*) in der traditionellen japanischen Grammatik.

c. ka + i + ta → kaita

In jeder dieser drei Möglichkeiten wird nur die durch unser Grundprinzip erlaubte, durch Konkatenation realisierte Komposition zugrundegelegt. Tatsächlich werden wir die dritte dieser Ableitungen verwenden, und dafür das traditionelle System von Verbstämmen durch ein anderes ersetzen. Dazu aber kommen wir ausführlicher in dem Kapitel zu sprechen, das die endgültige Modellierung beschreibt.

Natürlich kann ein Formalismus nicht die Komplexität einer gestellten Aufgabe verändern, sondern bestenfalls eine — relativ zu anderen Formalismen — adäquatere und dadurch elegantere Darstellung ermöglichen. Es ist deshalb wichtig, noch einmal darauf hinzuweisen, daß das hier dargelegte Prinzip der Komposition durch Konkatenation nur die *orthographische* bzw. *phonologische Form* der sprachlichen Entitäten betrifft, nicht aber die ebenfalls im Formalismus darzustellenden, diesen anhaftenden linguistischen Eigenschaften. Diese sind zwar parallel zur Konkatenation der Formen zu kombinieren, doch sind dazu komplexere Operationen (z.B. die Unifikation) erforderlich. Die linguistischen Eigenschaften, zusammen mit einem System auf diesen aufsetzender Regeln, werden wir später — wie in unserem Puzzlebeispiel schon angedeutet — als die für die "Auswahl" der Allomorphe und die "Steuerung" der Komposition verantwortlichen Komponenten kennenlernen. Der Schwerpunkt der Modellierung verlagert sich damit auf eine andere Ebene, und ist — wie man vermuten wird — im Zusammenspiel von *linguistischen Eigenschaften* und *Regeln* zu suchen, auf das wir am Anfang verwiesen haben, und dem gehorchend die Komposition ausgeführt werden soll. Auf die Regeln werde ich später zu sprechen kommen; die diesen zugrundeliegenden, die morphologischen und syntaktischen Einheiten beschreibenden Informationsstrukturen jedoch sollen jetzt behandelt werden.

2.1.3 Linguistische Zeichen *oder* Morphologische und syntaktische Entitäten als Bündel ihrer Eigenschaften

Wenn ein so komplexer Prozeß wie die morphologische und syntaktische Modifikation japanischer Verben, einzig durch die Konkatenation einzelner Grundbausteine modelliert werden soll, müssen diese durch Beschreibungsmittel dargestellt werden, die mächtig genug sind, neben der *Form* der Bausteine alle zur Steuerung der Kokatenation notwendigen *linguistischen Eigenschaften* adäquat darzustellen. Nimmt man als Grundbausteine die im ersten Teil eingeführten *Morphe* an, und bezeichnet man die diesen inhärenten linguistischen Eigenschaften, die die zur Komposition nötigen Informationen bereitstellen, als *Merkmale* und deren konkrete Ausprägung als deren *Merkmalswerte*, kommt man zu einer Darstellung der Morphe durch ein Bündel von Merkmal-Wert-Paaren in etwa der folgenden Form:

$$\left[\begin{array}{lcl} <\boldsymbol{Morph\text{-}X}> & & \\ \textbf{Merkmal-1} & : & \text{Wert-1} \\ \textbf{Merkmal-2} & : & \text{Wert-2} \\ \quad\vdots & : & \quad\vdots \\ \textbf{Merkmal-n} & : & \text{Wert-n} \end{array}\right]$$

(fig 5) *Merkmal-Wert Matrix*

oder, unser letztes Beispiel zugrundelegend:

$$\left[\begin{array}{lcl} <\boldsymbol{kaki}> & & \\ \textbf{Base} & : & \text{i-base} \\ \textbf{Form} & : & \text{`kaki'} \\ \textbf{Morpheme} & : & \{\mathsf{kaku}\} \end{array}\right]$$

(fig 6) *Merkmal-Wert Matrix der Form ‘kaki’*[7]

Der Sinn der jeweils ersten Zeile der Matrizen, also ‘<*Morph-X*>’ bzw. ‘<*kaki*>’, besteht einfach in ihrer eindeutigen Benennung. Die Notwendigkeit dieser Namen wird erst an späterer Stelle aus der Modellierung der Merkmal-Wert-Bündel in FLEX ersichtlich.

Es wurde schon auf den rekursiven Charakter der Komposition verwiesen. Können also komplexe Formen aus atomaren oder selbst schon komplexen Formen nach den gleichen Prinzipien entstehen, folgt die Notwendigkeit einer einheitlichen Darstellung sowohl atomarer als auch abgeleiteter Formen. Gleich den Morphen werden deshalb auch morphologisch und syntaktisch komplexe Gebilde durch Merkmal-Wert-Bündel repräsentiert.

Ein wesentlicher Aspekt komplexer Formen ist eben die Tatsache, aus weniger komplexen abgeleitet worden zu sein. Dieser Sachverhalt muß sich auch in der Repräsentation komplexer Formen spiegeln. Es müssen folglich Merkmale existieren, die die Darstellungen der konstituierenden Elemente — also Merkmal-Wert-Bündel — als Werte enthalten. Das heißt, daß Merkmalwerte selbst komplexer Natur sein können, und Werte bzw. Merkmal-Wert-Bündel den gleichen theoretischen Status (oder *Typ*) haben müssen.

7 Dieses Beispiel soll nur der Illustration des bisher Gesagten dienen. Die wirkliche Modellierung hingegen wird eine etwas andere Form haben.

Beispiel 3:
Veranschaulichen wir uns diese Überlegung anhand einer vorläufigen Darstellung der Form 'kaita', für die wir hier die in (ex 17a) auf Seite 18 angeführte Ableitung 'kai + ta $\rightarrow$ kaita ' annehmen wollen:

$$\begin{bmatrix} \langle \mathit{kaita} \rangle & & \\ \textbf{Merkmal-X} & : & \text{x} \\ \textbf{Merkmal-Y} & : & \text{y} \\ \textbf{Merkmal-Z} & : & \text{z} \\ \textbf{Form} & : & \text{`kaita'} \\ \textbf{Morph 1} & : & \begin{bmatrix} \langle \mathit{kai} \rangle & & \\ \textbf{Base} & : & \text{onbin} \\ \textbf{Form} & : & \text{`kai'} \\ \textbf{Morpheme} & : & \{\mathsf{kaku}\} \end{bmatrix} \\ \textbf{Morph 2} & : & \begin{bmatrix} \langle \mathit{ta} \rangle & & \\ \textbf{Form} & : & \text{`ta'} \\ \textbf{Morpheme} & : & \{\mathsf{ta}\} \end{bmatrix} \end{bmatrix}$$

(fig 7) *Beispiel einer rekursiven Merkmal-Wert Matrix*

Beispiel 4:
Ein etwas komplexeres Beispiel einer Merkmal-Wert-Matrix findet sich in (fig 8). Dargestellt wird die Bildung der Form 書かせられた (*kakaserareta*), d.h. des Präteritums der Kausativ-Passiv Form des Verbes 書く (*kaku*; schreiben). Im Gegensatz zu Beispiel 3 wird in diesem Fall ein Ausschnitt des Zeichens dargestellt, durch welches diese Form in der endgültigen Modellierung repräsentiert wird.

Die Darstellung sprachlicher Einheiten erfolgt also durch Bündel von Merkmalen, deren Werte entweder atomar oder ebenfalls Bündel von Merkmalen sein können. Sie ist folglich, den Kompositionsprozeß spiegelnd, ebenso wie dieser rekursiver Natur. Im folgenden soll für solche komplexen, rekursiven Merkmal-Wert-Strukturen der Begriff *(linguistisches) Zeichen* verwendet werden.

Literatur:
Die Darstellung linguistischer Entitäten durch rekursive Merkmal-Wert Bündel kommt aus der Tradition der Unifikationsgrammatiken, und wird in Theorien wie der GPSG (Gazdar et al. 1985), der HPSG (Pollard und Sag 1987, Pollard und Sag 1994) oder der JPSG (Gunji 1987) verwandt. Das Zusammenspiel der Zeichen in diesen Theorien wird durch Regeln, für deren Funktionieren der Begriff *Unifikation* eine zentrale Rolle einnimmt, gesteuert. Eine gute Einführung in die Theorie der Unifikationsgrammatiken findet sich in Shieber 1986.

- *<kaku-causative-passive-past-flexive>* — *syntactical-sign*
  - **cat** : verb
  - **lexeme** : {kaku}
  - **j-lexeme** : {書く}
  - **meaning** : schreiben
  - **morphemes** : <{kaku}, {causative-1}, {passive}, {past-flexive}>
  - **level** : syntax
  - **phon** : '書(ka)kaserareta'
  - **base-sign** :
    - *<kaku-causative-passive-root>* — *composed-morphological-sign*
      - **cat** : vowel-verb
      - **lexeme** : {kaku}
      - **j-lexeme** : {書く}
      - **meaning** : schreiben
      - **morphemes** : <{kaku}, {causative-1}, {passive}>
      - **level** : morphology
      - **phon-base** : '書(ka)kaserare'
      - **phon-k-base** : '書(ka)kaserare'
      - **phon-i-base** : '書(ka)kaserare'
      - **phon-a-base** : '書(ka)kaserare'
      - **phon-t-base** : '書(ka)kaseraret'
      - **base-sign** :
        - *<kaku-causative-root>* — *composed-morphological-sign*
          - **cat** : vowel-verb
          - **lexeme** : {kaku}
          - **j-lexeme** : {書く}
          - **meaning** : schreiben
          - **morphemes** : <{kaku}, {causative-1}>
          - **level** : morphology
          - **phon-base** : '書(ka)kase'
          - **phon-k-base** : '書(ka)kase'
          - **phon-i-base** : '書(ka)kase'
          - **phon-a-base** : '書(ka)kase'
          - **phon-t-base** : '書(ka)kaset'
          - **base-sign** :
            - *<kaku-root>* — *root-sign*
              - **cat** : k-cons-verb
              - **lexeme** : {kaku}
              - **j-lexeme** : {書く}
              - **meaning** : schreiben
              - **morphemes** : <{kaku}>
              - **level** : morphology
              - **phon-base** : '書(ka)'
              - **phon-k-base** : '書(ka)k'
              - **phon-i-base** : '書(ka)ki'
              - **phon-a-base** : '書(ka)ka'
              - **phon-t-base** : '書(ka)it'
          - **suffix-sign** :
            - *<causative (cons-verb)>* — *suffix-sign*
              - **cat** : vowel-verb
              - **morphemes** : <{causative-1}>
              - **subcat-base** : a-base
              - **subcat-cat** : cons-verb
              - **phon** : 'se'
      - **suffix-sign** :
        - *<passive (vowel/kuru-verb)>* — *suffix-sign*
          - **cat** : vowel-verb
          - **morphemes** : <{passive}>
          - **subcat-base** : a-base
          - **subcat-cat** : vowel-verb or kuru-verb
          - **phon** : 'rare'
  - **suffix-sign** :
    - *<past-flexive (verb)>* — *flexive-sign*
      - **cat** : verb
      - **morphemes** : <{past-flexive}>
      - **subcat-base** : t-base
      - **subcat-cat** : verb
      - **phon** : 'a'

(fig 8) *Die Merkmal-Wert-Struktur der Form*
書かせられた *(kakaserareta)*

2.1.4 Grammatikalische Regularitäten als Bedingungen über linguistische Zeichen

Dank der bisher dargestellten Prinzipien haben wir eine Vorstellung von der Gestalt der Grundelemente unseres Systems und der Art und Weise, wie diese zur Bildung neuer Elemente rekursiv zusammengefügt werden können. Auf die Regularitäten aber, die diese Komposition einfacher Zeichen zu komplexeren steuern, sind wir bisher noch nicht eingegangen. Die einfache Natur der Zeichen und der Komposition im Gegensatz zu der darzustellenden Komplexität der Verbmodifikation verweisen jedoch auf diese Bedingungen — wie schon vorher angedeutet wurde — als wesentliches Element der Modellierung. Es kann also vermutet werden, daß die eigentlichen grammatikalischen Regularitäten auf dieser Ebene repräsentiert werden.

Die einzigen Möglichkeiten der Informationsdarstellung in den eingeführten linguistischen Zeichen sind Merkmal-Wert-Kombinationen. Stellt man sich unter "Regularitäten" eine Menge von *Implikationen* vor, und geht man weiter davon aus, daß Implikationen aus *Prämisse* und *Konklusion* in der Form "wenn etwas eine Prämisse erfüllt, so muß es auch die Konklusion erfüllen" zusammengesetzt sind, ergibt sich für die Formulierung grammatischer Regularitäten die folgende Form:

grammatikalische Regel (vorläufige Version) — Wenn das Merkmal-Wert-Muster A in einem sprachlichen Zeichen oder in den dieses konstituierenden Elementen vorhanden ist, so muß in diesem bzw. seinen Teilelementen auch das Merkmal-Wert-Muster B vorhanden sein.

Beispiel 5:
Betrachten wir die folgenden Formen:

(ex 18) *a.* mi + rareru → mirareru
b. kak + eru → kakeru

Beschrieben wird die Bildung der Potentialform der Verben 見る (*miru*; sehen) und 書く (*kaku*; schreiben). Die tatsächlichen Regularitäten vereinfachend wollen wir an dieser Stelle annehmen, daß die unterschiedliche Bildung der Formen durch die Zugehörigkeit zu unterschiedlichen Verbklassen verursacht wird: 'miru' sei Element der Verbklasse *vk1*, 'kaku' hingegen ordnen wir der Klasse *vk2* zu. Unter Zuhilfenahme verschiedener anderer Rollen wie 'Lexem' und 'Morpheme', deren genaue Bedeutung uns an dieser Stelle jedoch noch nicht interessieren soll, können wir damit die folgenden Zeichen zugrundelegen:

(ex 19) *a.* **Verbstämme**

i.

$$\left[\begin{array}{lll} \multicolumn{3}{l}{\textit{<miru>}} \\ \textbf{Lexem} & : & \{\mathsf{miru}\} \\ \textbf{Morpheme} & : & <\{\mathsf{miru}\}> \\ \textbf{Verbklasse} & : & \text{vk1} \\ \textbf{Form} & : & \text{`mi'} \end{array}\right]$$

ii.

$$\left[\begin{array}{lll} \multicolumn{3}{l}{\textit{<kaku>}} \\ \textbf{Lexem} & : & \{\mathsf{kaku}\} \\ \textbf{Morpheme} & : & <\{\mathsf{kaku}\}> \\ \textbf{Verbklasse} & : & \text{vk2} \\ \textbf{Form} & : & \text{`kak'} \end{array}\right]$$

b. **Potential-Morphem**

i.

$$\left[\begin{array}{lll} \multicolumn{3}{l}{\textit{<Potential (vk1)>}} \\ \textbf{Morpheme} & : & <\{\mathsf{Potential}\}> \\ \textbf{Verbklasse} & : & \text{vk1} \\ \textbf{Form} & : & \text{`rareru'} \end{array}\right]$$

ii.

$$\left[\begin{array}{lll} \multicolumn{3}{l}{\textit{<Potential (vk2)>}} \\ \textbf{Morpheme} & : & <\{\mathsf{Potential}\}> \\ \textbf{Verbklasse} & : & \text{vk2} \\ \textbf{Form} & : & \text{`eru'} \end{array}\right]$$

Die eben dargestellten Zeichen kombinierend, erhalten wir für die Potentialformen die folgenden komplexen Zeichen:

(ex 20) *a.* **Potentialform der Verben 'miru' und 'kaku'**

i.

$$\left[\begin{array}{lll} \multicolumn{3}{l}{\textit{<miru, Potential>}} \\ \textbf{Lexem} & : & \{\mathsf{miru}\} \\ \textbf{Morpheme} & : & <\{\mathsf{miru}\}, \{\mathsf{Potential}\}> \\ \textbf{Verbklasse} & : & \text{vk1} \\ \textbf{Form} & : & \text{`mirareru'} \\ \textbf{Basis} & : & \left[\begin{array}{lll} \multicolumn{3}{l}{\textit{<miru>}} \\ \textbf{Lexem} & : & \{\mathsf{miru}\} \\ \textbf{Morpheme} & : & <\{\mathsf{miru}\}> \\ \textbf{Verbklasse} & : & \text{vk1} \\ \textbf{Form} & : & \text{`mi'} \end{array}\right] \\ \textbf{Suffix} & : & \left[\begin{array}{lll} \multicolumn{3}{l}{\textit{<Potential (vk1)>}} \\ \textbf{Morpheme} & : & <\{\mathsf{Potential}\}> \\ \textbf{Verbklasse} & : & \text{vk1} \\ \textbf{Form} & : & \text{`rareru'} \end{array}\right] \end{array}\right]$$

ii.

$$\left[\begin{array}{lll} \multicolumn{3}{l}{\textit{<kaku, Potential>}} \\ \textbf{Lexem} & : & \{\mathsf{kaku}\} \\ \textbf{Morpheme} & : & <\{\mathsf{kaku}\}, \{\mathsf{Potential}\}> \\ \textbf{Verbklasse} & : & \text{vk2} \\ \textbf{Form} & : & \text{`kakeru'} \\ \textbf{Basis} & : & \left[\begin{array}{lll} \multicolumn{3}{l}{\textit{<kaku>}} \\ \textbf{Lexem} & : & \{\mathsf{kaku}\} \\ \textbf{Morpheme} & : & <\{\mathsf{kaku}\}> \\ \textbf{Verbklasse} & : & \text{vk2} \\ \textbf{Form} & : & \text{`kak'} \end{array}\right] \\ \textbf{Suffix} & : & \left[\begin{array}{lll} \multicolumn{3}{l}{\textit{<Potential (vk2)>}} \\ \textbf{Morpheme} & : & <\{\mathsf{Potential}\}> \\ \textbf{Verbklasse} & : & \text{vk2} \\ \textbf{Form} & : & \text{`eru'} \end{array}\right] \end{array}\right]$$

Um sicherzustellen, daß aus den Morphen unseres Beispiels nur korrekte Formen abgeleitet werden können, muß dafür gesorgt werden, daß nur Morphe der gleichen Verbklasse miteinander kombiniert werden können. Wir kommen damit zu unserem ersten Beispiel einer grammatikalischen Regel:

(ex 21) **Subkategorisierungsregel** — In einem komplexen Zeichen muß der Wert des Merkmals *Verbklasse* dem des entsprechenden Merkmals seiner Töchter entsprechen.

Ohne grammatische Regularitäten könnten sich beliebige Zeichen zu beliebigen neuen zusammensetzen. In dem eben dargestellten Fall beispielsweise verhindert nur die in (ex 21) dargestellte Regel die Bildung der ungrammatischen Formen *'mieru'[8] bzw. *'kakrareru'. Grammatikalische Regeln wirken also als Filter über die Menge aller kombinatorisch möglichen Zeichen, indem sie aus diesen nur solche als korrekt auszeichnen, die den durch sie gestellten grammatikalischen Anforderungen entsprechen.

Im letzten Abschnitt haben wir festgestellt, daß die Repräsentationen der Elemente, aus denen sich ein komplexes Zeichen bildet, als Werte in der Repräsentation des komplexen Zeichens dargestellt werden. Legt man diese Überlegungen zugrunde, kommt man zu dem Schluß, daß — gegeben ein beliebiges zusammengesetztes Zeichen — alle für eine grammatische Implikation nötigen Informationen schon in diesem vorhanden sind. Die eben aufgestellte Form einer grammatikalischen Regel läßt sich also, indem wir auf den expliziten Verweis auf die konstituierenden Elemente verzichten, zu der folgenden vereinfachen:

grammatikalische Regel — Wenn das Merkmal-Wert-Muster A in einem sprachlichen Zeichen vorhanden ist, so muß in diesem auch das Merkmal-Wert-Muster B vorhanden sein.

Wir können uns damit auf die folgende, formalisierte Schreibweise einigen:

grammatikalische Regel

$$\begin{bmatrix} \textit{Merkmal-Wert-Matrix A} \\ \ldots \end{bmatrix} \Longrightarrow \begin{bmatrix} \textit{Merkmal-Wert-Matrix B} \\ \ldots \end{bmatrix}$$

Durch die Verwendung von Variablen als Merkmalswerte wird es weiterhin möglich, Abhängigkeiten von Merkmalen untereinander auszudrücken:

8 'mieru' ist eine Variante der Potentialform 'mirareru', und deshalb als *Form* gesehen korrekt. Diese Variante wird in unserer Theorie jedoch durch Suffigierung eines anderen Morphems erklärt. Wenn 'mieru' an dieser Stelle also als *ungrammatisch* bezeichnet wird, bezieht sich dieses auf die zu realisierenden Morpheme <{miru}, {Potential}>.

grammatikalische Regel mit Merkmalsabhängigkeiten

$$\left[\begin{array}{l} \hfill {\scriptstyle \textit{Merkmal-Wert-Matrix A}} \\ \ldots \end{array}\right] \Longrightarrow \left[\begin{array}{l} \hfill {\scriptstyle \textit{Merkmal-Wert-Matrix B}} \\ \textbf{Merkmal-1}: X \\ \textbf{Merkmal-2}: f(X) \\ \ldots \end{array}\right]$$

Beispiel 6:
Wir wollen uns nun überlegen, wie wir die "Subkategorisierungsregel" aus (ex 21), Beispiel 5 in dieser neuen, formalen Schreibweise ausdrücken können.

Im Beispiel 5 treten die Merkmalswerte 'vk1' bzw. 'vk2' als Werte des Merkmals 'Verbklasse' auf, welches wiederum Teil der komplexen Werte der Merkmale 'Basis' und 'Suffix' des Gesamtzeichens ist. Benutzen wir die Schreibweise 'Basis | Verbklasse' bzw. 'Suffix | Verbklasse' zur Beschreibung solcher *Merkmalsketten*, können wir diesen Sachverhalt durch die folgenden Merkmal-Wert-Spezifikationen beschreiben:

(ex 22) *a.* *im Falle von (*ex 20*a-i):*

- *i.* ***Basis*** | ***Verbklasse*** : vk1
- *ii.* ***Suffix*** | ***Verbklasse*** : vk1

b. *im Falle von (*ex 20*a-ii):*

- *i.* ***Basis*** | ***Verbklasse*** : vk2
- *ii.* ***Suffix*** | ***Verbklasse*** : vk2

Benutzen wir 'VK', 'X' und 'Y' als Variablen für Merkmalswerte, läßt sich die in (ex 21) als *Subkategorisierungsregel* eingeführte Bedingung an komplexe Zeichen formal folgendermaßen schreiben:

(ex 23) **Subkategorisierungsregel**

$$\left[\begin{array}{ll} \multicolumn{2}{r}{\scriptstyle \textit{Zeichen-X}} \\ \textbf{Basis} & : Y \\ \textbf{Suffix} & : Z \end{array}\right] \Longrightarrow \left[\begin{array}{ll} \multicolumn{2}{r}{\scriptstyle \textit{Zeichen-X}} \\ \textbf{Verbklasse} & : VK \\ \textbf{Basis} \mid \textbf{Verbklasse} & : VK \\ \textbf{Suffix} \mid \textbf{Verbklasse} & : VK \end{array}\right]$$

Besitzt ein Zeichen die Merkmale 'Basis' und 'Suffix', muß es hinsichtlich des Merkmals 'Verbklasse' in Mutter- und Tochter-Zeichen gleich spezifiziert sein. 'mirareru' und 'kakeru' sind also korrekt, $*$'mieru' und $*$'kakrareru' hingegen — wie der ihnen vorgestellt Stern schon andeutet — ungrammatisch bezüglich dieser Regel.

Grammatische Regularitäten können also als eine Menge von Implikationen über linguistische Zeichen formuliert werden. Ein relativ zu einer solchen Grammatik korrekt geformtes Zeichen muß aus dieser Menge jede einzelne

Implikation erfüllen. Zusammen definieren damit Grammatik und die Menge aller kombinatorisch möglichen Zeichen eine Untermenge dieser letzteren, die nämlich aller grammatikalisch korrekt geformten Zeichen im Sinne der gegebenen linguistischen Formalisierung.

2.1.5 Unregelmäßigkeiten bei der Bildung von Verbformen

Ein weiteres sehr wichtiges Element unserer Modellierung ist die Behandlung der Unregelmäßigkeiten, die bei der Verbmodifikation auftreten. Es lassen sich zwei prinzipielle Wege vorstellen, diesem Phänomen gegenüberzutreten:

Beschreibung durch einen monotonen Formalismus — Unregelmäßigkeiten werden als extrem spezialisierte "Regelmäßigkeiten" in die grammatische Beschreibung aufgenommen.

Beschreibung mittels eines Defaultformalismus — Integration eines Verfahrens in den benutzten Formalismus, mit dem "Ausnahmen" relativ zu regelmäßigen Ableitungen dargestellt und behandelt werden können.

Die erste Alternative führt zur Eliminierung der Unregelmäßigkeiten durch Regeln, die sich nur auf das eigentlich unregelmäßige Verb oder die abweichende Form beziehen. "Allgemeine" Regeln hingegen müssen so eingeschränkt werden, daß sie nicht nur eine "Standartregel" angeben, von der in einzelnen Fällen (nämlich den "unregelmäßigen") abgewichen werden darf, sondern tatsächlich *nur auf solche* Verben oder Fälle bezogen werden können, auf die sie *vollkommen*, d.h. ohne Ausnahme, zutreffen. Man kann also sagen, daß bei dieser Strategie Ausnahmen zu Regelmäßigkeiten gemacht werden. Da diese Herangehensweise jedoch zu sehr komplexen, schlecht zu überblickenden Strukturen und Regularitäten führt, wurde sie hier zu Gunsten der zweiten aufgegeben.

Beispiel 7:
Das Verb 行く (*iku*; gehen) bildet nahezu alle seine Formen regelmäßig. Nur einige wenige — wie z.B. das *Partizip* (*Te-Form*) — werden unregelmäßig abgeleitet. Hat man nun einen Defaultformalismus zur Verfügung, kann man die bestehenden "Standartregeln" (Defaults) einfach in bezug auf die unregelmäßigen Formen dieses Verbes durch neue "Ausnahmeregeln" überschreiben:

(ex 24) Bildung des Partizips (Defaultformalismus):

a. Konsonantische Verben auf 'ku' bilden das Partizip durch Anfügen der Endung 'ite' *b.* Das Verb 'iku' hingegen bildet das Partizip mit der Endung 'tte'

Sollen die abweichenden Formen hingegen mit monotonen Mitteln beschrieben werden, müssen auch die "Standartregeln" derart eingeschränkt werden, daß sie wirklich nur die Verben betreffen, für die sie in allen Fällen zu richtigen Formen führen:

(ex 25) Bildung des Partizips (monotoner Formalismus):

a. Konsonantische Verben auf 'ku', bei denen es sich nicht um das Verb 'iku' handelt, bilden das Partizip durch Anfügen der Endung 'ite'.

b. Das Verb 'iku' bildet das Partizip mit der Endung 'tte'.

Man wird sich leicht vorstellen können, daß bei der Beschreibung komplexerer Sachverhalte durch monotone Regeln schnell ein extrem aufwendiges und daher schwer verständliches und wenig elegantes Regelsystem entsteht.

Zum genauen Verständnis des von uns gewählten Ansatzes ist die exakte Beschreibung des benutzten Formalismus nötig. Aus diesem Grund soll erst später auf die genaue Behandlung der Unregelmäßigkeiten eingegangen werden. Es sei jedoch schon einmal gesagt, daß anstatt *monotoner Regeln* ein *Defaultmechanismus* in der unterliegenden Beschreibungslogik FLEX diese Herangehensweise möglich macht.

2.2 Linguistische Prinzipien

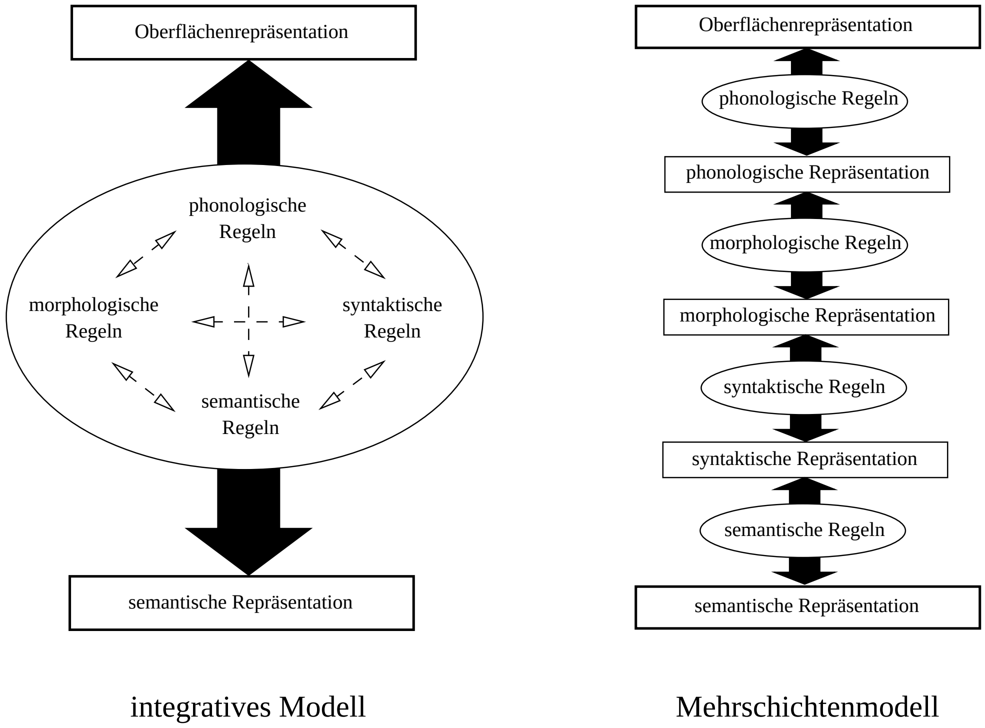
Nachdem wir die den Formalismus betreffenden Prinzipien umrissen haben, wollen wir uns nun den Prinzipien zuwenden, die die grammatikalische Formalisierung eingrenzen. Da die Struktur des Formalismus vor allem dadurch bestimmt wird, ob er eine adäquate linguistische Formalisierung erlaubt, und sich eine solche linguistische Formalisierung wiederum nur der Ausdrucksmittel bedienen kann, die ihr vom Formalismus bereitgestellt werden, und die sie ja indirekt selbst bestimmt hat, lassen sich formale und linguistische Prinzipien nicht wirklich voneinander trennen. Vielmehr wird auffallen, daß sich formale Prinzipien auch auf linguistische Fragestellungen beziehen lassen und umgekehrt. Teilweise gibt es deshalb Entsprechungen auf beiden Seiten. So folgt etwa aus der Konkatenation als einziger Grundoperation auf Formen die in diesem Abschnitt beschriebene Integration der Beschreibungsebenen und umgekehrt.

2.2.1 Die Interpretation der Beschreibungsebenen

Wir haben Morphologie und Syntax als sich hierarchisch zueinander verhaltende Beschreibungsebenen der Linguistik eingeführt. In vielen modernen Linguistischen Theorien — wie der "Theory of Government and Binding" (Chomsky 1981, von Stechow und Sternefeld 1988) — spiegeln sich diese Ebenen (zusätzlich zu anderen) im Prozeß der Ableitung oder Analyse sprachlicher Äußerungen wider. Eine Äußerung ensteht aus einer internen semantischen Struktur, indem sich diese nacheinander den linguistischen Beschreibungsebenen mehr oder weniger entsprechenden Prozessen unterwirft, und sich so von der ursprünglichen Darstellung durch viele Übergangsrepräsentationen hindurch zu ihrer endgültigen, geäußerten Form verwandelt. Der entsprechend umgekehrte Verwandlungsprozeß wird durchlaufen, wenn nicht das Hervorbringen einer Äußerung, sondern der Verstehensprozeß einer solchen modelliert werden soll.

Entgegen einer Interpretation der linguistischen Beschreibungsebenen als hintereinandergeschaltete Abbildungen verschiedener Darstellungsformen ineinander, sollen diese in dem hier verfolgten Ansatz als auf nur einer einzigen Repräsentation operierende, miteinander interagierende, gleichzeitig wirksame Prinzipien aufgefasst werden, die einen sie alle integrierenden und entsprechend sehr allgemeinen Prozeß steuern.

Die folgenden Schemata sollen die zwei Möglichkeiten, die Beschreibungsebenen in ein Computerlinguistisches Modell zu integrieren, verdeutlichen. Die rechte Skizze zeigt die Interpretation der Ebenen als zeitlich aufeinander folgende Verarbeitungsschichten; links hingegen treten sie als miteinander interagierende Prinzipien in Erscheinung:



(fig 9) *Zur Interpretation der Beschreibungsebenen in verschiedenen linguistischen Formalismen*

Morphologie und Syntax bedienen sich der gleichen Mechanismen, und unterscheiden sich nur durch die Verschiedenheit der Anwendungsbereiche ihrer Regularitäten: Morphologische Regeln beziehen sich auf morphologische Aspekte, syntaktische Regeln entsprechend auf syntaktische. Diese Aspekte wiederum, die ja nur als Merkmal-Wert-Paare dargestellt werden können, sind — abgesehen von ihrer Benennung — nur dadurch als syntaktische oder morphologische erkennbar, daß sie von entsprechend morphologischen oder syntaktischen Regeln betroffen werden. Die Zirkularität dieser Kriterien macht die Unmöglichkeit einer Abgrenzung auf formaler Ebene deutlich und verweist auf die metatheoretische Betrachtung als einzigem Ort der Unterscheidung.

Den formalen Kriterien folgend werden komplexe Zeichen durch die hierarchische Verschachtelung ihrer Konstituenten dargestellt. Da sich Wörter nach morphologischen Regeln aus Morphen bilden und syntaktische Phrasen nach syntaktischen Regularitäten aus Wörtern, wird man die Beschreibungsebenen jedoch in der Tiefe der Verschachtelung wiedererkennen können. Für den Formalismus jedoch ist diese Abgrenzung nicht sichtbar.

2.2.2 Zur Integration der Phonologie

Auch die Phonologie, die die in einer Sprache wirksamen Lautgesetze beschreibt, soll auf die im letzten Abschnitt dargestellte Weise in den Formalismus integriert werden. Wird sie in der vorhin dargestellten Modellierung der Beschreibungsebenen als einem geschichteten Modell oft als letzte Generierungsphase bzw. erste Analysephase beschrieben, die für bestimmte kontextabhängige Veränderungen und Verschleifungen der Formen verantwortlich ist, soll sie in dem hier eingeschlagenen Vorgehen derart mit den anderen Beschreibungsebenen verwoben werden, daß alle gebildeten Formen von vornherein nur phonologisch korrekt sein können.

Anstatt erst eine Struktur der vorkommenden Morphe und Wörter in der richtigen Reihenfolge zu bilden, die dann den phonologischen Modifikationen unterworfen wird, soll der Begriff des Morphes derart erweitert werden, daß er auch alle phonologischen Varianten mit einschließt. Phonologische Merkmale und Regeln beeinflussen die Komposition von Morphen und Wörtern dann derart, daß nur phonologisch erlaubte Varianten miteinander kombiniert werden können. Die Phonologie wird also gewissermaßen von der erweiterten Morphologie "verschluckt"[9].

9 Die hinter diesem Vorgehen verborgene Auffassung der Phonologie ist allerdings problematisch. Stellt man sich — wie dieses beispielsweise im amerikanischen Strukturalismus getan wird — die phonetischen Bestandteile eines Wortes (*Phone*) als Matrix phonetischer Merkmale vor, von denen so viele wie möglich durch sprachabhängige phonologische Regeln determiniert werden (*systematische Merkmale*), und nur der durch diese nicht erklärbare Teil das betreffende Wort wirklich kennzeichnet (*idiosynkratische Merkmale*), so wird man im Lexikoneintrag des betreffenden Wortes sicherlich nur die idiosynkratische Information repräsentieren. Die systematischen Merkmale hingegen können durch die Phonologie ergänzt werden. Beschränkt man also den Wirkungsbereich phonologischer Regeln ausschließlich auf das richtige Zusammenfügen schon vorgegebener Teile, wie das hier eingeschlagene Vorgehen dieses ja impliziert, müssen alle eigentlich vorhersagbaren Eigenschaften eines Wortes in den vorgegebenen Teilen schon repräsentiert werden. Warum trotzdem ein solches — auf den ersten Blick zu zahlreichen Redundanzen führendes — Vorgehen gewählt wurde, muß begründet werden:

1. **Orientierung an der orthograpischen Form**
   Entgegen der modernen deskriptiven Grammatiktradition befassen wir uns nicht mit der gesprochenen Sprache, sondern mit ihrer schriftsprachlichen Fixierung. — Der Schwerpunkt der Modellierung liegt also auf dem orthographischen Code. Phonetik und Phonologie werden nur insoweit behandelt, wie sie durch die Verschriftlichung hindurch zur Wirkung kommen.

2. **Konzentration auf die Morphologie**
   Das eigentliche Thema dieser Arbeit sind die *morphologischen Prozesse* der Verbmodifikation. Die Phonologie wird nur dann betrachtet, wenn sie zur Determinierung der schriftlichen Form wesentlich ist.

Eine besser ausgearbeitete phonologische Komponente, mit der die erwähnten Redundan-

Beispiel 8:
Anstatt diese Integration durch ein neues Beispiel zu verdeutlichen, möchte ich auf die schon im Abschnitt 2.1.2 ("die Konkatenation als einzige erlaubte Grundoperation auf *Formen*") beschriebene Ableitung der Präteritumform 'kaita' des Verbes 'kaku' verweisen. Wir hatten damals (unter anderem) zwei mögliche Wege diese zu generieren aufgezeigt:

(ex 26) *a.* kaki + ta → kaki.ta → kaita
b. ka + i + ta → kaita

Die Analyse (ex 26*a*) hatten wir verworfen, da die in ihr verwendete Transformation gegen die formalen Prinzipien verstößt, die nur die Konkatenation als Operation auf orthographischen oder phonologischen Formen erlauben. Die Ableitung (ex 26*b*) hingegen hatten wir als weiter zu verfolgende ausgewiesen, da sie genau diesen Prinzipien Folge leistet. Die erste Alternative entspricht nun der traditionellen Auffassung von Morphologie und Phonologie als einem Schichtenmodell aus zwei Komponenten, deren erstere die morphologische Komposition beschreibt, der die phonologische Transformation als zweite Komponente folgt. Die zweite Alternative aber veranschaulicht genau das in diesem Abschnitt geforderte Vorgehen, nämlich die Integration von Morphologie und Phonologie in einem gemeinsamen Prozeß, der — entsprechend des letzten Abschnitts — auf ganz ähnliche Weise ja auch die Syntax einschließen soll. Mit dieser Veranschaulichung der Beziehung zwischen Formalismus und lingustischer Modellierung haben wir übrigens auch ein schönes Beispiel für die angedeutete Entsprechung formaler und linguistischer Prinzipien.

Zuletzt soll darauf hingewiesen werden, daß ein solches integrierendes Vorgehen aus linguistischer Sicht natürlich problematisch ist. Zumindest wäre

---

zen vermieden werden könnten, soll späteren Modellierungen überlassen werden. Wie eine solche allerdings aussehen könnte, sei hier kurz angedeutet:

Wir werden später sehen, daß die Allomorphe der behandelten Morpheme in einem systematischen Verhältnis zueinander stehen. Anstatt die Allomorphe für jedes Morphem aufzuzählen, können sie im Normalfall durch ein Regelsystem aus einer einzigen Form abgeleitet werden. Unserem "Morphpuzzle" — um auf die benutzte Analogie zurückzukommen — wird also eine Komponente vorgeschaltet, die aus einem Minimum an Information (zugrundeliegende Form und Regelsystem) erst die konkreten Puzzlesteine konstruiert. Dieses Vorgehen betrifft dabei teilweise auch jetzt schon phonologische Gesetzmäßigkeiten.

In einer späteren Modellierung, die eher auf die phonologischen Regularitäten fokussiert, ließe sich dieses Vorgehen zu einer relativ eigenständigen phonologischen Komponente ausweiten. Würde man dabei nicht alle Allophone generieren, wie dieses dem augenblicklichen Vorgehen in der morphologischen Komponente entspricht, sondern — durch bestimmte Merkmale getriggert — nur die tatsächlich benötigten Formen, wären sowohl deklarative als auch algorithmische Redundanzen eleminiert.

die Verwendung des Begriffes *Morph* auch für phonologische Varianten zu überdenken. Da jedoch erstens die Abgrenzung von Morphologie und Phonologie ohnehin problematisch ist und von der verwandten Theorie abhängt, und zweitens der Schwerpunkt dieser Arbeit eher auf die Vollständigkeit der beschriebenen Formen, deren Korrektheit und die Einfachheit der Beschreibung gelegt werden soll, als auf das Einhalten traditioneller linguistischer Vorgehensweisen, wurde auf die Verwendung anderer Begriffe verzichtet.

2.2.3 Zur Modellierung der Verbmodifikation durch Kategorien, Basisformen und Subkategorisierung

Die japanischen Verben lassen sich — ihrem systematischen Verhalten bei der Verbmodifikation entsprechend — verschiedenen *Flexionsklassen* oder *Kategorien* zuordnen. So besteht zwischen den Allomorphen eines Verbes — die auch als *Basisformen* bezeichnet werden sollen — ein bestimmter systematischer Zusammenhang, der sich durch ein Regelsystem beschreiben läßt, das von der Kategorie oder Flexionsklasse des Verbes abhängig ist. Ebenso ist die Auswahl eines Allomorphes zur Realisierung eines modifizierenden Morphems von der jeweiligen Kategorie des zu modifizierenden Verbes abhängig.

Bestimmte Eigenschaften der modifizierenden Morpheme wiederum bestimmen, welches Allomorph des Verbes, an das das Morphem sich anschließt, zur Realisierung kommen soll. In "traditioneller" Sprechweise ausgedrückt, verbinden sich die modifizierenden Suffixe nur mit bestimmten *Stamm-* oder *Basisformen* der Verben. Wir haben es also mit einer komplizierten, wechselseitigen Abhängigkeit der Morphe voneinander zu tun. Betrachten wir dazu ein weiteres Beipiel, für das wir diesmal die Konditionalformen der uns schon bekannten Verben 'kaku' und 'miru' heranziehen wollen:

Beispiel 9:

(ex 27) *a.* *i.* kaku
ii. kakeba
wenn (er) schreibt, dann ...
b. *i.* miru
ii. mireba
wenn (er) sieht, dann ...

Den Morphemen {kaku}, {miru} und {Konditional} sollen die folgenden Allomorphe entsprechen:

	{kaku}	=	{*ka, kak, kaki, kaka, kait*}
(ex 28)	{miru}	=	{*mi*}
	{Konditional}	=	{*reba, eba, ureba*}

Ohne uns weiter um die Bedeutung und Verwendung der Formen zu kümmern, können wir die bei ihrer Bildung wirksamen, gegenseitigen Abhängigkeiten folgendermaßen formulieren:

1. Das Konditionalmorphem {Konditional} fordert von dem Verb, an das es sich anschließt, die Realisierung durch die (in unserer Modellierung) *k-base* genannte Basisform (bzw. das dieser entsprechende Allomorph).

2. (a) Die Verbklasse der sogenannten *vokalischen Verben*, denen das Verb 'miru' angehört, verbinden sich nur mit dem Allomorph 'reba' des Konditionalmorphems.

   (b) Die sogenannten *konsonantischen Verben* hingegen, zu denen sich 'kaku' rechnet, wählen 'eba' als Realisierung des Konditionalmorphems.

Das erste Kriterium, das von dem modifizierenden Suffix abhängig ist, führt zur Auswahl des Allomorphs 'mi' für {miru}. Der Bedingung 2.a zufolge, die man sich am elegantesten als Eigenschaft des Allomorphes 'reba' vorstellen kann, verbindet sich dieses dann mit der Variante 'reba' des Konditionalmorphemes zu der Form 'mireba'. Entsprechenden Überlegungen für das Verb 'kaku' zufolge, kommen wir für dieses zu der Form 'kakeba' synthetisiert aus den Morphen 'kak' und 'eba'.

Stellt man sich vor, daß jedes Morph als konkrete Ausprägung des Morphemes, dem es angehört, dessen Eigenschaften erbt, kann man sagen, daß die modifizierenden Morphe die Basisform und die Kategorie der Verben bestimmen, an die sie sich anschließen. Für diesen Sachverhalt soll im folgenden der Begriff *Subkategorisierung* verwendet werden, und wir wollen sagen, daß ein bestimmtes modifizierendes Morphem für eine bestimmte Kategorie und Basisform der zu modifizierenden Verben *subkategorisiert*.

2.3 Zusammenfassung

Wir haben nun die Begriffe *Kategorie*, *Basisform* und *Subkategorisierung* und die Phänomene, für die diese stehen, kennengelernt. Ebenso haben wir gesehen, wie bestimmte Regularitäten von diesen Gebrauch machen, um die korrekte Wahl der Allomorphe zu steuern, und damit zu korrekten Formen zu kommen. Ziel der Modellierung ist es, diese Regularitäten in allgemeiner Weise für alle Verbklassen und Modifikationen darzustellen. Dieses soll durch ein System von Zeichen aus Merkmal-Wert-Paaren und grammatischen Regeln geschehen, in denen die gerade beispielhaft dargestellten Abhängigkeiten formalisiert werden. Verben entsprechende Morpheme sollen dazu durch komplexe Verbzeichen realisiert werden, in denen sowohl die

Kategorien als auch alle Allomorphe oder Basisformen als Merkmale dargestellt werden können. Modifizierende Suffixe hingegen werden durch Zeichen dargestellt, deren Merkmale angeben, mit Verben welcher Kategorie sie sich verbinden können und an welche Basisformen dieser sie sich anschließen. Das Zusammenspiel der Morpheme wird dann durch eine Menge von grammatischen Regeln formalisiert, die auf diese Merkmale zugreifen.

Damit haben wir eine ungefähre Vorstellung der Mechanismen, deren sich die japanische Verbmodifikation bedient, sowie der Art und Weise, wie diese formal beschrieben werden kann. Diese erste Skizze soll nun zu einer formal exakten Beschreibung ausgearbeitet werden. Ohne auf das bisher Erarbeitete weiter Bezug zu nehmen soll deshalb im nächsten Kapitel der Formalismus der *Beschreibungslogiken* eingeführt werden. Ist diese zweite Grundlage der angestrebten Modellierung beschrieben, kann schließlich erklärt werden, wie die im ersten Kapitel aufgestellten Prinzipien und Mechanismen in einem hybriden System aus Prolog und der Beschreibungslogik FLEX integriert werden können. Schließlich werden dann die linguistischen Details der japanischen Verbmodifikation beschrieben und formalisiert.

3 Beschreibungslogiken

3.1 Wissensrepräsentation

Die künstliche Intelligenz (KI) — und mit ihr die Computerlinguistik — bildet einen relativ neuen Wissenschaftsbereich, über dessen genaue Definition noch kein Konsens gebildet werden konnte. Sie läßt sich deshalb eher durch Aufzählung der sich zu ihr rechnenden Forschungsunternehmen charakterisieren, als durch eine genaue Eingrenzung von Gegenstandsbereich und verwendeter Methodik. Wollte man aber, in bewußt vager Formulierung, die Erklärung und Modellierung "intelligenten" Verhaltens als ihre Grundmotivation anerkennen, stieße man, den unterschiedlichen Perspektiven der Erklärungsversuche entsprechend, auf sehr verschiedenartige Grundannahmen und "genauere" Formulierungen dieser Grundproblematik. Einigkeit besteht jedoch trotz aller Unterschiede darüber, daß zu intelligentem Verhalten "Wissen" erforderlich ist — was immer man unter diesem Begriff auch genauer verstehen möchte.

Zur Methode der Informatik, und damit auch der künstlichen Intelligenz, gehört es, Hypothesen über ein zu erforschendes Objektgebiet durch Integration in ausführbare Computermodelle zu validieren. Dabei sich ergebende Probleme dienen als Ausgangsbasis für neue Hypothesen und genauere Modellierungen. Von Computermodell zu Computermodell fortschreitend hofft man so, durch immer bessere Beschreibung, das zu erklärende Objekt schließlich durchdringen zu können.

Diese operationale Vorgehensweise, zusammen mit der Schlüsselposition des Begriffes "Wissen" in der KI, machen einen zumindest vorläufigen Wissensbegriff erforderlich. Es ergibt sich folglich die Frage, was genau in dieser Disziplin unter "Wissen" verstanden wird, und wie sich dieses darstellen läßt.

Eine mögliche Näherung kann darin bestehen, Wissen als ein System von Begrifflichkeiten zu einem bestimmten, abgrenzbaren "Weltausschnitt" aufzufassen. Durch die Analyse der Strukturen innerhalb eines solchen Systems wird versucht, sich auf eine möglichst kleine Menge grundlegender Begriffe und Begriffsbeziehungen zu einigen, deren Elemente in dem Sinne minimal und voneinander unabhängig sind, als daß sie sich selbst nicht auseinander definieren lassen, alle anderen Begriffe und Begriffsbeziehungen des Systems aber aus ihnen abgeleitet werden können. Von diesen "Primitiven" ausgehend, wird das Begriffssystem auf eine mathematische Struktur[10], das heißt

10 Der Begriff *Struktur* wird in dieser Arbeit in zwei verschiedenen Bedeutungen verwendet: Zum einen ist mit *Struktur*— entsprechend der umgangssprachlichen Verwendung dieses Terminus — einfach etwas auf irgend eine Weise Strukturiertes gemeint; wird jedoch

eine Menge von Objekten und diese organisierende Relationen abgebildet. Modelliert man Begriffe als *Objekte* und deren Beziehungen als *Relationen*, bedeutet dieses Vorgehen die Festlegung einer Menge *primitiver Begriffsobjekte* und *-relationen* zu dem zu modellierenden Gegenstandsgebiet. Alle übrigen Begriffe und Begriffsbeziehungen hingegen werden mit dem Mittel der Definition als "Abkürzungen" durch Komposition gebildeter *komplexer Objekte* und *Relationen* dargestellt. Um die so erhaltene "Wissensstruktur" für die Lösung von Problemen oder die Ableitung von Folgerungen verwenden zu können, werden schließlich Mechanismen definiert, mit denen diese interpretiert und damit nutzbar gemacht werden kann.

Beispiel 10:
Wollte man beispielsweise mit Mitteln der KI — das heißt unter Voraussetzung dieses Wissensbegriffes — einen kleinen Ausschnitt seiner privaten Bibliothek modellieren, und ginge man dabei von den primitiven Objekten 'Pipi Langstrumpf', 'Buch' und 'Abenteuer' und von den primitiven Relationen 'handelt von' und 'Hauptperson' aus, ließe sich ein 'Abenteuerroman' als 'Buch und handelt von: Abenteuer' definieren, und der Begriff 'Lieblingsbuch' dann als 'Abenteuerroman und Hauptperson: Pipi Langstrumpf' — vorausgesetzt natürlich, daß jedes Buch über Pipis Abenteuer ein Lieblingsbuch ist, und andere Lieblingsbücher nicht existieren.

Sicherlich ist die Darstellung von "Wissen" als einem System aus Begriffen und Begriffsrelationen sehr abstrakt, und für manche Anwendung deshalb für eine adäquate Modellierung des Gegenstandsbereiches nicht ausreichend. Für die "traditionelle" KI ist sie jedoch — im Gegensatz z.B. zu konnektionistischen Ansätzen — kennzeichnend, und soll deshalb im folgenden zugrundegelegt werden.

3.1.1 Semantische Netze und Frames

Zwei im Bereich der Künstlichen Intelligenz sehr verbreitete Wissensrepräsentationsformalismen, denen dieser Wissensbegriff zugrundeliegt, sind *Semantische Netze* und *Frames*. Beide definieren "Wissen", unter Annahme bestimmter grundlegender Objekte und Relationen, als aus diesen gebildete Struktur, und stellen Mechanismen bereit, die zu deren Interpretation und zur Ableitung bestimmter Schlußfolgerungen dienen können. Da beide Formalismen jedoch aus konkreten Problemstellungen abgeleitet wurden, entbehren sie der genauen Festlegung der verwendbaren Darstellungsmittel

von einer *mathematischen Struktur* gesprochen, ist von der mathematischen Präzisierung dieses Begriffes die Rede. Eine *Struktur* in diesem Sinne ist ein Tripel aus einer nichtleeren Grundmenge und jeweils einer Menge auf dieser definierter Relationen und Funktionen. Da wir vor allem mit einstelligen und zweistelligen Relationen zu tun haben werden, werden wir oft die ersteren als *Teilmengen* und die letzteren allgemein als *Relationen* ansprechen.

und deren Bedeutung. Zwar ist es möglich, kleine, einfach strukturierte Wissensbereiche darzustellen, und für diese begrenzte Folgerungsmechanismen bereitzustellen; werden die zu modellierenden Probleme jedoch komplexer, entstehen dank der formalen Unklarheit leicht unüberschaubare und widersprüchliche Situationen. Vor allem aber ist es nahezu unmöglich, allgemeine Folgerungsmechanismen zu definieren, bzw. deren Konsequenzen zu überschauen. Theoretische Betrachtungen hinsichtlich der Ausdrucksmächtigkeit oder der Zulässigkeit bestimmter Folgerungen aber lassen sich überhaupt nicht durchführen.

Die entstehenden Probleme verweisen auf die Notwendigkeit, Repräsentationsformalismen zu entwerfen, die eindeutig definiert, und dank weniger aber mächtiger Ausdrucksmittel sowohl leicht zu untersuchen, als auch in verschiedensten Gebieten anwendbar sind.

3.1.2 Logik

Wesentlich für unsere bisherigen Betrachtungen sind die Begriffe *Objekt* und *Relation*. Wissen wird aufgefaßt, als aus diesen konstruierte Struktur. Versucht man eine Disziplin zu finden, die sich formal mit solchen Begriffen auseinandersetzt, bietet sich die mathematische Logik an. Unter ähnlichen Grundannahmen befaßt sich diese mit ebenso ähnlichen Fragestellungen, wie sie bei der Definition der dargestellten Repräsentationsformalismen im Rahmen der KI auftreten: Auch ihre Problematik besteht in der Darstellung und Strukturierung von "Wissen" und der Untersuchung der Möglichkeiten, aus solchen Darstellungen Folgerungen abzuleiten. Im Unterschied zu der relativ unformalen Herangehensweise, durch die sich die Anfänge der KI auszeichnen, ist die Logik — als Teildisziplinen der Mathematik — sehr streng strukturiert, und kann deshalb selber Objekt theoretischer Untersuchungen werden.

Tatsächlich gibt es zahlreiche Versuche, Wissen in KI-Systemen mit Hilfe logischer Formalismen darzustellen, bzw. die vorhin angesprochenen Darstellungssysteme in die Logik zu übersetzen. Aber auch wenn die Logik — dank ihrer einfachen, durch Mittel der Kombination jedoch gleichzeitig sehr mächtigen Struktur — in Aussagekraft und theoretischer Fundierung die gestellten Anforderungen erfüllt, hat sie doch für Computermodellierungen sehr erhebliche Nachteile: Zwar lassen sich viele Probleme in ihr formalisieren, und ein Großteil der jeweils interessanten Folgerungen ableiten; die Formalisierungen jedoch sind für den Menschen, aufgrund ihrer kombinatorischen Komplexität, kaum nachvollziehbar, und die Folgerungen, wenn auch theoretisch möglich, zur tatsächlichen Berechnung oft zu aufwendig.

Ausgehend von der intuitiven Einfachheit der traditionellen KI-Repräsentationen und der formalen Strenge der mathematischen Logik kam es daher zur Entwicklung neuer Repräsentationsformalismen, zu denen sich auch die im folgenden beschriebenen *Beschreibungslogiken* und mit diesen auch das in dieser Arbeit verwendete Berliner System FLEX rechnet.

3.1.3 Beschreibungslogiken

Die folgenden zwei Punkte sollen, die bisherigen Betrachtungen zusammenfassend, der weiteren Darstellung zugrundegelegt werden:

1. Die von den traditionellen Wissensrepräsentationsformalismen bereitgestellten Ausdrucksmittel reichen — zumindest für eine bestimmte Klasse von Problemen — zur Darstellung von "Wissen" aus.[11] Wissen kann also als eine Struktur aus Konzepten und Konzeptrelationen modelliert werden.

2. Für Wissensrepräsentationen ist der Folgerungsbegriff essentiell. Zu seiner genauen Spezifizierung ist eine mathematische Fundierung des Formalismus, wie sie z.B. in der formalen Logik durchgeführt wird, unabdingbar.

Beschreibungslogiken stehen für den Versuch, von diesen zwei Forderungen ausgehend, einen neuen Formalismus zu entwickeln — d.h. die relativ zum Vorgehen in der KI bestehenden Vorteile der schon dargestellten Ansätze zur Wissensdarstellung in einem neuen System zu vereinen. Entsprechend Punkt eins werden die wesentlichen Ausdrucksmittel der traditionellen Wissensrepräsentationsformalismen zugrundegelegt, jedoch als formale Sprache strenger als ursprünglich gegliedert. Punkt zwei Folge leistend, wird — analog zum Vorgehen in der formalen Logik — zwischen der *Form* der Sprache und ihrer *Bedeutung* unterschieden. Wie in der Logik ist mit der *Bedeutung* eines Ausdruckes die Klasse der mathematischer Strukturen gemeint, in denen seine *Auswertung* oder *Interpretation* den Wert *wahr* ergibt. Und wiederum steht der Begriff *Struktur* für ein Tupel aus einer Grundmenge und einer Menge sich auf diese beziehender Teilmengen und Relationen. Nur für relativ wenige, vorgegebene Grundausdrücke wird die Interpretation durch den Formalismus festgelegt. Alle anderen für eine konkrete Modellierung benötigten Ausdrücke für Elemente, Teilmengen und Relationen werden hingegen mit Hilfe dieser Grundausdrücke neu eingeführt, und zueinander in

11 Diese Forderung ist als Arbeitshypothese für die weiteren Überlegungen zu betrachten. Es ist nicht gemeint, daß das Phänomen "Wissen" tatsächlich durch diese Ausdrucksmittel adäquat charakterisiert wäre.

Beziehung gesetzt. Durch diese Trennung in vordefinierte und relativ zu einer Anwendung aus diesen definierbare Begriffe kann eine große Flexibilität der Beschreibungsmöglichkeiten erreicht werden. Dank der formalen Spezifizierung der benutzten Sprache und der klaren, mengentheoretischen Definition der Grundausdrücke lassen sich die Eigenschaften und die Mächtigkeit des Formalismus mit mathematischen Mitteln untersuchen, so daß trotz starker Ähnlichkeit mit den anfangs angeführten KI-Formalismen die bei diesen auftretenden Schwierigkeiten vermieden werden können.

3.2 Die Beschreibungslogik FLEX

In den vorangehenden Überlegungen haben wir uns allgemein mit dem Wissensbegriff in der KI und Repräsentationsformalismen, die diesen zugrundelegen, befaßt. Die Ausdrucksmittel traditioneller KI-Wissensrepräsentationen mit dem mathematischen Vorgehen formaler Logiken kombinierend, soll der Formalismus der Beschreibungslogiken für das weitere Vorgehen ausgewählt, und im folgenden näher beschrieben werden. Da insbesondere die formale Strenge der Definition von Beschreibungslogiken uns zu dieser Entscheidung bewegt, wollen wir uns unter diesem Gesichtspunkt mit dem auf Beschreibunslogiken basierenden Berliner System FLEX genauer befassen. Dazu sollen die wichtigsten der von FLEX bereitgestellten Ausdrucksmöglichkeiten dargestellt und in ihrer Bedeutung beschrieben werden. Um der Entscheidung für "formales" Vorgehen gerecht zu werden, werden wir jedoch zuerst die Begriffe *Syntax* und *Semantik* und darauf aufbauend den *Folgerungsbegriff* einführen, und einige Gründe für ihre zentrale Stellung in einer formalen Beschreibung darstellen.

3.2.1 Syntax, Semantik und Folgerungsbegriff

Ein möglicher Weg, einen "Weltausschnitt" zu modellieren, ist es, alle über ihn bekannten Aussagen in einer Liste aufzureihen, und das zu einem Zeitpunkt relevante Wissen in Form einer solchen Aussage mit einem Suchalgorithmus aus dieser Liste zu extrahieren. Wollte man beispielsweise einen Familienstammbaum modellieren, ließen sich alle Familienmitglieder aufzählen und alle bekannten Verhältnisse zwischen ihnen, wie 'Vater von', 'Mutter von', 'Großvater von', 'Großmutter von', 'Urgroßvater von', 'Urgroßmutter von', 'Ururgroßvater von', ... , 'abstammend von' usw. ebenfalls durch eine lange Liste angeben. Statt eines Stamm*baumes* erhielte man einen komplexen "Stamm*graphen*", in dem alle verwandschaftlichen Relationen explizit gekennzeichnet wären.

Im allgemeinen Fall — und so auch in dem eben angedeuteten — wird man jedoch versuchen, nur grundsätzliche Beziehungen aufzulisten, und anschließend ein mechanisches Verfahren angeben, mit dem man alle übrigen, impliziten Relationen automatisch erschließen kann. So läßt sich aus 'A ist Vater von B' und 'B ist Vater von C' mit einem solchen Verfahren auf 'A ist Großvater von C' schließen. Die Bezeichnung *Stammbaum* deutet auf eben diese Abstraktion hin: explizit dargestellt werden nur Eltern-Kind- und Paar-Beziehungen, alle übrigen aber lassen sich aus diesen auf einfache Weise folgern.

Die Regel zum Erschließen der Großvaterbeziehung läßt sich zur Folgerung einer konkreten Beziehung verwenden. Wie aber läßt sich das Folgern für allgemeine Situationen "mechanisieren", und woher weiß man, daß ein solches mechanisches Verfahren tatsächlich zu den Folgerungen führt, die man wünscht, das heißt zu allen richtigen und nur zu diesen? — Mit diesen Fragen sind wir bei der Kernproblematik der Untersuchung logischer Kalküle angelangt, der nämlich, die sich rund um den Folgerungsbegriff ansiedelt.

Der in der modernen Logik eingeschlagene Weg, sich über den jeweils verwendeten Folgerungsmechanismus Klarheit zu verschaffen, besteht darin, möglichst genau — das heißt in Form mathematischer Strukturen — die "Verhältnisse" zu charakterisieren, über deren Beziehungen nachgedacht werden soll. Es wird auf diese Weise möglich — zum Beispiel durch das Führen mathematischer Beweise — eindeutig nachzuweisen, wann etwas aus einem Sachverhalt folgt, und wann nicht.

Anschließend wird mit formalen Mitteln, d.h. durch die Angabe einer *formalen Syntax*, eine eindeutige Sprache definiert, in der man alle für das Vorgehen interessanten Beziehungen innerhalb einer solchen Struktur ausdrücken kann. Da nun über eine formale Sprache zu einem formalen Gegenstandsbereich verfügt wird, liegt es nahe, das Verhältnis zwischen beiden, das heißt die "Bedeutung" sprachlicher Ausdrücke relativ zu einer Struktur ebenfalls formal zu charakterisieren. Dieses geschieht durch eine als *Interpretation* bezeichnete Abbildung der syntaktischen Ausdrücke in Konstrukte einer Struktur.

Mit dem so erhaltenen System läßt sich die Folgerungsbeziehung zwischen Ausdrücken der formalen Sprache präzisieren:

Folgerungsbeziehung (vorläufige Formulierung) — Ein Ausdruck A folgt aus einem Ausdruck B, wenn man zeigen kann, daß in allen Strukturen, in denen B gilt, auch A gelten muß.

Es ist also möglich, "Beweise" zu führen, indem man die zu betrachtenden Ausdrücke in Strukturen übersetzt, und über die dabei erzwungenen Verhältnisse nachdenkt.

Da die verwendete Sprache formal definiert ist, lassen sich *Kalküle*, das heißt "mechanische" Verfahren angeben, mit deren Hilfe man aus einer Menge bestimmter Ausdrücke andere ableiten kann. Die eigentliche Motivation für das bisherige Vorgehen liegt nun darin, daß sich von solch einem mechanischen Verfahren untersuchen läßt, ob die durch dieses abgeleiteten Aussagen wirklich denen entsprechen, die man auch durch logische Überlegungen folgern kann; das heißt, ob es die folgenden zwei Bedingungen erfüllt:

Korrektheit — Ein Kalkül ist *korrekt*, wenn es sich mathematisch nachweisen läßt, daß alle Aussagen, die man mit seiner Hilfe aus einer gegebenen Menge ebensolcher Aussagen ableiten kann, auch "tatsächlich", d.h. im Sinne des oben eingeführten Folgerungsbegriffes, aus diesen folgen.

Vollständigkeit — Kann man beweisen, daß *alle* im Sinne der Folgerungsbeziehung folgerbaren Aussagen auch durch das Kalkül ableitbar sind, wird dieses als *vollständig* bezeichnet.

Ist ein Kalkül sowohl korrekt, als auch vollständig, kann es als *adäquat* im Sinne des Folgerungsbegriffes bezeichnet werden.

Weitere Eigenschaften eines Kalküles, wie beispielsweise die *Entscheidbarkeit* logischer Folgerungen, lassen sich auf ähnliche Weise untersuchen. Da das System FLEX, das wir in unserer Modellierung verwenden wollen, jedoch weder vollständig, noch entscheidbar — natürlich aber korrekt — ist, die im Rahmen unserer Modellierung wichtigen Folgerungen jedoch alle zu erschließen erlaubt, wollen wir die eben angedeuteten Überlegungen allerdings nicht weiter ausführen. Als wesentlich jedoch soll die Möglichkeit der mathematischen Klärung des Folgerungsbegriffes zurückbehalten werden, und damit die dieser zugrundeliegenden folgenden zwei Begriffe:

Syntax — d.h. die Definition einer eindeutigen Sprache,

Semantik — womit wir von nun an zu einer gegebenen Sprache eine *Struktur* zusammen mit einer *Inerpretation*, die zwischen Sprache und Struktur vermittelt, bezeichnen wollen.

Die genaue Form des Folgerungsmechanismus des Systems FLEX samt dessen Eigenschaften sollen uns jedoch nicht weiter interessieren.

Syntax

Die Syntax der Sprache, die zur Formulierung von Sachverhalten durch das System FLEX benutzt wird, soll rekursiv durch eine kontextfreie Grammatik definiert werden.

Da die Ausdrucksmittel von FLEX, von den grundlegensten ausgehend, nach und nach eingeführt werden, wollen wir die Syntax ebenfalls nach und nach spezifizieren. Die natürlichsprachliche Beschreibung der Konstrukte von FLEX und die formale Charakterisierung von Syntax und Semantik werden also parallel zueinander erfolgen.

Literatur:
Bei der Darstellung der Syntax durch kontextfreie Regeln lehne ich mich an die KL-One-Einführung von Luck und Owsnicki-Klewe 1990 und den KIT-Report 100 (Hoppe et al. 1993) an. Der KIT-Report 100 führt in die Beschreibungslogik BACK ein, die FLEX als Vorgängermodell vorausging.

Semantik und Folgerungsbegriff

Interpretation

Wir haben uns schon auf eine Darstellung der Bedeutung durch mathematische Strukturen, d.h. Teilmengen und Relationen relativ zu einem nicht leeren Grundbereich geeinigt. Begriffe oder *Konzepte*, wie wir diese auch nennen wollen, werden extensional durch Teilmengen des Domains interpretiert. *Instanzen* bestimmter Konzepte, d.h. sprachliche Ausdrücke für einzelne "Objekte", die unter einen Begriff fallen, werden auf Elemente des Grundbereiches abgebildet. Beziehungen von Instanzen zueinander entsprechen zweistelligen Relationen über die Elemente des Grundbereiches, und werden auf syntaktischer Ebene durch sogenannte *Rollen* dargestellt. Das Verhältnis zwischen Ausdrücken und Strukturen werden wir durch eine *Interpretationsfunktion* präzisieren. Diese wird zuerst für atomare Ausdrücke der Sprache angegeben, und dann über den Aufbau der Sprache kompositionell erweitert. Bezeichnen wir den Gegenstandsbereich durch das Symbol $\mathcal{D}$ (*Domain*), die Menge möglicher Ausdrücke der Beschreibungslogik als $\mathcal{T}$ und die Interpretationsfunktion durch das Symbol $[\![\ ]\!]^{\Im}$, können wir sagen:

Eine **Interpretation** $[\![\ ]\!]^{\Im}$ ist eine Abbildung von $\mathcal{T}$ auf Elemente, Teilmengen und zweistellige Relationen relativ zu einem Grundbereich $\mathcal{D}$ (bzw. in $\mathcal{D} \cup \mathcal{D}^* \cup (\mathcal{D} \times \mathcal{D})^*$) für die *unter anderem* gilt:

1. $[\![\mathsf{i}]\!]^{\Im} \in \mathcal{D}$ **falls** i eine Instanz ist.
2. $[\![\mathsf{k}]\!]^{\Im} \subseteq \mathcal{D}$ **falls** k ein Konzept ist.
3. $[\![\mathsf{r}]\!]^{\Im} \subseteq \mathcal{D} \times \mathcal{D}$ **falls** r eine Rolle ist.

Eine Interpretation ordnet einer Menge von Ausdrücken, oder einer *Terminologie*, wie wir eine solche in Zukunft bezeichnen wollen, also eine konkrete Struktur zu. Dabei legt sie für die meisten der vorgegebenen Ausdrücke die

Bedeutung fest. Wie haben bei der Aufzählung der Eigenschaften einer Interpretation "unter anderem" geschrieben, da wir die von FLEX vorgegebenen Ausdrücke zusammen mit den weiteren Eigenschaften einer Interpretation — nämlich für diese eine bestimmte Bedeutung festzulegen — erst nach und nach kennenlernen werden.

Terme und Aussagen

Den bisher betrachteten Ausdrücken einer Terminologie werden durch eine Interpretation Extensionen in Form von Elementen, Teilmengen und Relationen relativ zu einem Domain $\mathcal{D}$ zugewiesen. Sie lassen sich durch vordefinierte Verknüpfungsmittel, wie z.B. **and** zu komplexen Ausdrücken verbinden, denen semantisch ebenfalls eine Extension entspricht. Wir wollen Ausdrücke, die für solche Extensionen stehen, in Zukunft als *Terme* bezeichnen.

Später werden wir sehen, wie sich Konzepte als *Unterkonzept* oder *Spezialisierung* schon bekannter Konzepte *einführen* oder *definieren* lassen. Wir werden dabei erstmals Ausdrücken begegnen, die nicht für Elemente, Teilmengen oder Relationen stehen, sondern Aussagen über deren Verhältnisse zueinander treffen, die *wahr* oder *falsch* sein können. Ausdrücke dieser Art sollen im folgenden als *Aussagen* bezeichnet werden.

Modell

Bildet eine Interpretation eine gegebene Terminologie so ab, daß alle *Aussagen* der Terminologie *wahr* sind, d.h. werden dabei die Konzepte und Instanzen so abgebildet, daß *unter anderem*

1. alle als *Spezialisierung* zu einem Term eingeführten Konzepte als Teilmengen der Interpretation dieses Termes erscheinen,
2. die Extensionen *definierter Konzepte* sich mit denen der Terme, durch die sie definiert wurden, überdecken und
3. alle *Instanzen* bestimmter Konzeptterme als Elemente der Extensionen dieser Terme abgebildet werden,

bezeichnet man eine solche Interpretation als *Modell* der entsprechenden Terminologie. Verwendet man das Symbol :: zur Einführung von Instanzen, das Symbol :< zur Einführung von Konzepten und das Symbol ':=' zu deren Definition, läßt sich das gleiche formal in der folgenden Weise ausdrücken:

Modell (vorläufige Formulierung) — Eine Interpretation $[\![\ ]\!]^{\Im}$ einer Terminologie $\mathbf{T} \subseteq \mathcal{T}$ wird genau dann als *Modell* bezeichnet, wenn *unter anderem* gilt:

$$
\begin{array}{llllll}
1. & [\![\mathsf{k}]\!]^{\Im} & \subseteq & [\![\mathsf{kt}]\!]^{\Im} & \textbf{falls} & \text{`}\mathsf{k} \;:<\; \mathsf{kt}\text{'} \in \mathbf{T} \\
2. & [\![\mathsf{k}]\!]^{\Im} & = & [\![\mathsf{kt}]\!]^{\Im} & \textbf{falls} & \text{`}\mathsf{k} \;:=\; \mathsf{kt}\text{'} \in \mathbf{T} \\
3. & [\![\mathsf{i}]\!]^{\Im} & \in & [\![\mathsf{kt}]\!]^{\Im} & \textbf{falls} & \text{`}\mathsf{i} \;::\; \mathsf{kt}\text{'} \in \mathbf{T}
\end{array}
$$

Dabei stehen die Variablen k, kt und i respektiv für Konzepte, Konzeptterme und Instanzen. Wie bei der Erklärung des Begriffes *Interpretation*, verweist der Ausdruck "unter anderem" auf die Vorläufigkeit des Gesagten: Neben den Bedingungen 1., 2. und 3. werden wir später noch andere Anforderungen an Modelle kennenlernen und zu berücksichtigen haben.

Folgerung

Mit dem Begriff *Modell*, können wir die Definition der Folgerungsbeziehung einfacher formulieren. Wir haben bisher gesagt, daß "ein Ausdruck A aus einem Ausdruck B folgt, wenn man zeigen kann, daß in allen Strukturen, in denen B gilt, auch A gelten muß." Dank des Begriffes *Modell* können wir nun einfacher sagen:

Folgerungsbeziehung (Neuformulierung) — *Ein Ausdruck* **A** *folgt aus einem Ausdruck* **B**, *wenn alle Modelle von* **B** *auch Modelle von* **A** *sind.*

Entscheident also ist bei der Charakterisierung der Semantik nicht die Angabe einer konkreten Interpretation oder eines konkreten Modells, sondern:

1. die Angabe aller möglichen Interpretationen, bzw.
   der Bedingungen, die unter allen denkbaren Abbildung einer Terminologie in eine Struktur genau die Menge der möglichen Interpretationen auszeichnet.

2. die Angabe der Bedingungen, unter denen eine Semantik einer Terminologie (daß heißt eine Interpretation mit der zu ihr gehörenden Struktur) ein *Modell* dieser Terminologie darstellt.

Bei der Darstellung der Semantik von FLEX werden wir uns auf die Angabe dieser Bedingungen beschränken.

Literatur:
Der hier beschriebenen Behandlung der Folgerungsbeziehung liegt die KL-ONE Einführung Luck und Owsnicki-Klewe 1990 und das BACK-Tutorium Hoppe et al. 1993 zugrunde. Über das modelltheoretische Vorgehen und die Präzisierung des Folgerungsbegriffes lassen sich genauere Darstellungen in der Literatur zur mathematischen Logik finden. Eine gute Einführung in dieses Gebiet ist Ebbinghaus et al. 1992.

3.2.2 Ausdrucksmittel des Systems FLEX

Es soll nun der in dieser Arbeit verwendete Sprachausschnitt des Systemes FLEX dargestellt werden. Die in unserer Modellierung verwendeten Ausdrücke werden dazu nach und nach umgangssprachlich eingeführt und in bezug auf Syntax und Semantik präzisiert. Bedingungen an Interpretationen werden dabei mit (int n) nummeriert, Bedingungen an Modelle mit (mod n) und syntaktische Regeln mit (syn n). (ex n) bezeichnet Beispiele zur Verwendung der Ausdrücke von FLEX.

Konzepte[12]

Sollen Konzepte relativ zu einer Grundmenge verwendet werden, bietet sich die Interpretation dieser als Teilmengen des Domains an. Sei also $\mathcal{K}$ die Menge der in einer Modellierung verwendeten Konzeptnamen, und $\mathcal{D}$ der Domain, so gilt:

(int 1) $\quad [\![k]\!]^{\Im} \quad \subseteq \quad \mathcal{D} \qquad$ für alle $k \in \mathcal{K}$

Werden beispielsweise Menschen — neben anderen "Dingen" des beschriebenen Universums — durch Elemente des Grundbereichs modelliert, bezeichnet der Begriff 'Mensch' die Menge all der Elemente des Domains, die für Menschen stehen. Wird das Konzept als *Primitive* eingeführt, daß heißt, charakterisiert man es unter Zuhilfenahme schon bekannter anderer Konzepte nur durch die Angabe der notwendigen, nicht aber der hinreichenden Bedingungen, läßt sich natürlich nicht erschließen, welche Objekte in diese Teilmenge fallen. Bekannt ist nur, daß "Menschen" eine Teilmenge des Domains darstellen.

Es gibt ausgezeichnete Begriffe 'ctop' und 'cbot', die durch die Semantik der Sprache auf den Gegenstandsbereich selber bzw. auf die leere Menge abgebildet werden:

(int 2) $\quad [\![\mathbf{ctop}]\!]^{\Im} \quad = \quad \mathcal{D}$

(int 3) $\quad [\![\mathbf{cbot}]\!]^{\Im} \quad = \quad \emptyset$

Die syntaktische Regel, durch die Konzepte eingeführt werden, hat die folgende Form (Wie auch in (int 1) bezeichnet $\mathcal{K}$ die Menge der in einer Modellierung verwendeten Konzeptnamen):

(syn 1) $\quad \text{KonzeptName} \quad \longrightarrow \quad k \quad$ für alle $\quad k \in \mathcal{K} \cup \{\mathbf{ctop},\mathbf{cbot}\}$

12 *Begriff* und *Konzept* werden im folgenden synonym verwendet.

Spezialisierung von Konzepten

Durch das Symbol ':<' kann die *Unterkonzeptbeziehung* zwischen zwei Konzepten spezifiziert werden. Ein Konzept A ist *Unterkonzept* oder *Spezialisierung* eines Konzeptes B, wenn alle Elemente, die unter A fallen auch unter B fallen — oder anders gesagt, die Extension von A Teilmenge der Extension von B ist. Es gilt also:

(syn 2)

KonzeptEinführung	$\longrightarrow$	KonzeptName	:<	KonzeptTerm.[13]
KonzeptTerm	$\longrightarrow$	KonzeptName.		

(mod 1) $[\![\mathsf{k}]\!]^{\Im} \subseteq [\![\mathsf{kt}]\!]^{\Im}$ **falls** 'k :< kt' $\in \mathbf{T}$ [14]

(mod 1) ist keine notwendige Bedingung einer Interpretation; vielmehr ist sie — wie die Numerierung durch *mod* schon andeutet — eine der Bedingungen, die die Interpretation einer Terminologie als Modell auszeichnet, wenn sie für alle Aussagen dieser Art erfüllt wird.

Unter Zuhilfenahme des primitiven Konzeptes 'ctop' und des Symbols ':<', läßt sich der Begriff 'Mensch' nun folgendermaßen einführen:

(ex 1) Mensch :< **ctop**.

'ctop' ist also ein allgemeineres Konzept als 'Mensch', und 'Mensch' eine Spezialisierung von 'ctop'. Alle Objekte, auf die sich der Begriff 'Mensch' anwenden läßt, lassen sich auch durch 'ctop' beschreiben. Über den Begriff 'Mensch' ist also zu diesem Zeipunkt nichts weiter bekannt, als daß seine Extension eine Teilmenge der Extension von 'ctop' und damit des Domains ist.

Andere Begriffe lassen sich in der gleichen Weise angeben:

(ex 2)
Sache :< **ctop**.
Farbe :< **ctop**.

Schon engeführte Konzepte lassen sich weiter unterteilen:

13 'KonzeptTerme' werden im nächsten Absatz genauer beschrieben. Vorläufig wollen wir uns unter ihnen — wie in der nächsten Regel beschrieben — ihre einfachste Ausprägung, nämlich 'KonzeptNamen' vorstellen.

14 Wir haben diese Bedingung schon als erste Bedingung der Modellbeziehung kennengelernt.

(ex 3)

Mann	:<	Mensch.
Frau	:<	Mensch.
Lesender	:<	Mensch.
Buch	:<	Sache.

Konzeptterme

Möchte man ein Konzept als Unterkonzept zweier schon eingeführter Begriffe angeben, lassen sich diese durch den vordefinierten Ausdruck 'and' verknüpfen. Diese Verknüpfung entspricht der Schnittmengenbildung der Interpretationen beider Begriffe. Wir nennen einen solchen Ausdruck *Konzeptterm*:

(syn 3) KonzeptTerm $\longrightarrow$ KonzeptTerm **and** KonzeptTerm.

(int 4) $$[\![\mathsf{kt}_1 \quad \textbf{and} \quad \mathsf{kt}_2]\!]^{\Im} \quad = \quad [\![\mathsf{kt}_1]\!]^{\Im} \quad \cap \quad [\![\mathsf{kt}_2]\!]^{\Im}$$

Mit den schon eingeführten Konzepten 'Mann', 'Frau' und 'Lesender' können wir die Konzepte 'Leser' und 'Leserin' z.B. folgendermaßen einführen:

(ex 4)

Leser	:<	Lesender	**and**	Mann.
Leserin	:<	Lesender	**and**	Frau.

Wir spezifizieren durch diese Angaben eine Menge von Konzepten, deren Extensionen in Teilmengen- und Schnittmengenrelation zueinander stehen. Aus dem bisher explizit Angegebenen lassen sich schon jetzt einige implizite Sachverhalte folgern. Beispielsweise folgt daraus, daß 'Leser' 'Männer' und 'Männer' 'Menschen' sind, daß 'Leser' auch 'Menschen' sein müssen.

Instanzen

Individuelle Konzepte oder *Instanzen* sind Ausdrücke, die semantisch genau ein Element des Domains bezeichnen. Ähnlich den Konzepten, werden sie relativ zu einem Konzeptterm, d.h. als Bezeichner eines Elementes der durch den Term beschriebenen Extension eingeführt. Sei $\mathcal{I}$ die Menge der verwendeten Instanzennamen, so können wir schreiben:

(syn 4)

InstanzenName	$\longrightarrow$	i	für alle $i \in \mathcal{I}$
InstanzenEinführung	$\longrightarrow$	InstanzenName :: KonzeptTerm.	

(mod 2) $[\![\mathsf{i}]\!]^{\Im} \in [\![\mathsf{kt}]\!]^{\Im}$ **falls** '$\mathsf{i} :: \mathsf{kt}$' $\in \mathbf{T}$

Wir haben nun die Möglichkeit, Elemente zu kennzeichnen, die unter ein Konzept fallen. Wir können beispielsweise sagen:

(ex 5) Marianne :: Frau.

Durch diese Einführung weisen wir 'Marianne' als Instanz des Konzeptes 'Frau' aus. Wir fordern damit die Existens eines Elementes, das zur Teilmenge der 'Frauen' gehört, und durch den Namen 'Marianne' bezeichnet werden soll. Da 'Marianne' eine 'Frau' ist, und 'Frauen' 'Menschen' sind, ist damit implizit auch gesagt, daß 'Marianne' unter das Konzept 'Mensch' fällt.

In der gleichen Weise können wir die Konzepte 'rot', 'gelb' und 'blau' als Instanzen des Konzeptes 'Farbe' spezifizieren, und modellieren Farben damit als Elemente des Universums und nicht extensional durch die Menge aller z.B. roten Dinge:

(ex 6)

rot :: Farbe.
gelb :: Farbe.
blau :: Farbe.

Subsumption

Die Spezifizierung von Wissen in Form von Mengen und Relationen ist nur dann sinnvoll, wenn sie zur Beantwortung bestimmter Fragen bzw. zur Folgerung bestimmter Aussagen benutzt werden kann. In diesem Zusammenhang ist offensichtlich die "Zugehörigkeit" bestimmter Konzepte und Instanzen zu anderen Konzepten von besonderer Wichtigkeit. Wir wollen diese "Zugehörigkeit" nun genauer betrachten, und durch den Begriff *Subsumption* präzisieren:

Subsumption einer Instanz durch einen Konzeptterm — Ein Konzeptterm kt *subsumiert* eine Instanz i, wenn sich aufgrund der Terminologie beweisen läßt, daß die Interpretation der Instanz Element der Interpretation des Konzepttermes ist, bzw. daß gilt:

$$[\![\mathsf{i}]\!]^{\Im} \in [\![\mathsf{kt}]\!]^{\Im}$$

Subsumptionsbeziehung zwischen Konzept und Konzeptterm — Ein Konzeptterm kt *subsumiert* ein Konzept k, wenn sich aufgrund der Terminologie beweisen läßt, daß alle Instanzen von k auch Instanzen von kt sind, bzw. daß gilt:

$$[\![\mathsf{k}]\!]^{\Im} \subseteq [\![\mathsf{kt}]\!]^{\Im}$$

Definierte und primitive Konzepte

In den bisher angeführten Beispielen haben wir Konzepte mit Hilfe des Prädikates ':<' als Unterkonzepte von Konzepttermen eingeführt. Wir wissen von den Instanzen eines solchen Konzeptes deshalb nur, daß sie in einem Modell als Elemente der Extension des Konzepttermes erscheinen. Wir haben hingegen kein Kriterium, um umgekehrt von Instanzen des Konzepttermes allgemein aussagen zu können, ob diese auch durch das Konzept subsumiert werden. Anders ausgedrückt wissen wir zwar, welche Eigenschaften Instanzen eines Konzeptes *notwendigerweise* erfüllen müssen, nicht aber, welche Eigenschaften *hinreichend* sind, um beliebige Elemente als unter das Konzept fallend erkennen zu können — kurz: wir kennen zwar die *notwendigen*, nicht aber die *hinreichenden* Eigenschaften des Konzeptes.

Die durch den Begriff *Subsumption* beschriebene Zugehörigkeit bestimmter Konzepte und Instanzen zu anderen Konzepten, läßt sich deshalb nur dann erschließen, wenn sie schon explizit in der Terminologie angegeben wurde.

Soll es hingegen prinzipiell möglich sein, die Subsumptionsbeziehung bezüglich bestimmter Konzepte zu erschließen, müssen wir sowohl die *notwendigen*, als auch die *hinreichenden* Bedingungen dieser Konzepte kennen. Wird ein Konzept k in diesem Sinne unter Verwendung eines Konzepttermes kt eingeführt, erhalten wir die folgende Situation:

1. Der Konzeptterm kt gibt die *notwendigen* Bedingungen des Konzeptes k an. Instanzen von k sind also auch Instanzen von kt:

   $$[\![\mathsf{k}]\!]^{\Im} \quad \subseteq \quad [\![\mathsf{kt}]\!]^{\Im}$$

2. Die *hinreichenden* Bedingungen werden gleichfalls durch den Konzeptterm kt beschrieben. Es gilt also auch das umgekehrte Verhältnis:

   $$[\![\mathsf{kt}]\!]^{\Im} \quad \subseteq \quad [\![\mathsf{k}]\!]^{\Im}$$

Die Extensionen von Konzeptterm und Konzept sind also gleich, d.h., das Konzept stellt sozusagen eine "Abkürzung" des Konzepttermes dar, bzw.:

$$[\![\mathsf{kt}]\!]^{\Im} \quad = \quad [\![\mathsf{k}]\!]^{\Im}$$

Wir wollen im folgenden für Konzepte, die nur unter Angabe der notwendigen Bedingungen eingeführt wurden, den Begriff *primitives Konzept*, und für solche, für die sowohl die notwendigen als auch die hinreichenden Bedingungen angegeben wurden, den Begriff *definiertes Konzept* benutzen. Verwenden wir zur Einführung definierter Konzepte das Symbol ':=', kommen wir zu den folgenden Regeln:[15]

15 Vergleiche auch (syn 2) und (mod 1).

(syn 5)

$$\text{KonzeptEinführung} \longrightarrow \text{KonzeptName} := \text{KonzeptTerm}.$$

(mod 3) $[\![\mathsf{k}]\!]^{\Im} = [\![\mathsf{kt}]\!]^{\Im}$ **falls** '$\mathsf{k} := \mathsf{kt}$' $\in \mathbf{T}$

Die folgenden Beispiele mögen die Unterscheidung primitiver von definierten Konzepten verdeutlichen:

(ex 7)

Erdbeere	:<	Frucht
	and	Rot[16].

Primitives Konzept:
Alle Erdbeeren sind rote Früchte,
aber nicht jede rote Frucht muß eine Erbeere sein.

(ex 8)

Erdbeere	:=	Frucht
	and	Rot.

Definiertes Konzept:
Alle Erdbeeren sind rote Früchte,
und alle roten Früchte sind auch Erdbeeren.
Ist also eine Frucht rot, erfüllt sie die notwendigen und hinreichenden Bedingungen einer Erdbeere.

Automatische Klassifizierung

Ist über ein Konzept bekannt, daß es die hinreichenden Bedingungen eines definierten Konzeptes erfüllt, kann es automatisch als Unterkonzept des letzteren klassifiziert werden. Ebenso lassen sich Instanzen auf diese Weise automatisch als Instanzen eines definierten Konzeptes klassifizieren.

Führen wir beispielsweise das Konzept "Walderdbeere" auf die folgende Weise ein:

(ex 9)

Walderdbeere	:<	Frucht
	and	Rot
	and	Klein.

16 Die Eigenschaft 'rot' wird hier — anders als in (ex 6) — durch das Konzept 'rot', d.h. extensional modelliert. Wie sich Eigenschaften durch ein Zusammenspiel von *Rollen* und *Instanzen* darstellen lassen, wie dieses in (ex 6) schon angedeutet wurde, werden wir im nächsten Abschnitt sehen.

so kann — entsprechend der Definition des Konzeptes 'Erdbeere' — auf

(ex 10) Walderdbeere :< [17] Erdbeere.

geschlossen werden. Die durch ein Folgerungskalkül automatisch vollzogene Einordnung von Konzepten und Instanzen wird *automatische Klassifizierung* genannt.

Attribute *oder* Rollen

Wir haben bisher die *Spezialisierung* als Beziehung zwischen Konzepten und Konzepttermen und die *Instanziierung* als Beziehung zwischen Instanzen und Konzepttermen kennengelernt. Um einem Konzept oder einer Instanz mit diesen Mitteln die Eigenschaft 'rot' zuzuweisen, haben wir auf ein entsprechendes Prädikat 'rot' zurückgegriffen, und das Konzept bzw. die Instanz als dessen Spezialisierung bzw. Instanziierung charakterisiert.

Schwieriger wird es, wenn mit diesen Mitteln eine Beziehung zwischen Instanzen ausgedrückt werden soll. Um beispielsweise von den Instanzen 'Bertrand' und 'Kate' ausgehend 'Bertrand' als 'Vater' von 'Kate' zu kennzeichnen, könnte ein Konzept 'Kinder-von-Bertrand' eingeführt, und 'Kate' als dessen Instanz ausgewiesen werden. Eine echte Relation zwischen den Instanzen 'Bertrand' und 'Kate' aber wäre dadurch nicht beschrieben.

Auch die Darstellung von Relationen als Mengen von Tupeln, und die Rückführung der Tupel auf Mengen von Mengen ist mit den bisher eingeführten Mitteln nicht möglich, und würde eine Darstellung ohnehin nur unnötig verkomplizieren.

Es liegt also nahe, ein neues Konstrukt einzuführen, mit dem sich Relationen direkt ausdrücken lassen. Wir gehen dazu von der Menge $\mathcal{R}$ der *Relations-* bzw. *Rollennamen* aus, und benutzen die folgende Schreibweise:

(syn 6)

$$\begin{array}{rcll} \text{KonzeptTerm} & \longrightarrow & \text{RollenKette : InstanzenName.} & \\ \text{RollenKette}^{18} & \longrightarrow & \text{RollenTerm.} & \\ \text{RollenTerm} & \longrightarrow & \text{RollenName.} & \\ \text{RollenName} & \longrightarrow & r. & \text{für alle } r \in \mathcal{R} \end{array}$$

17 Das Symbol ':<' wird in diesem Zusammenhang nicht deklarativ zur Forderung der Subsumption bei der Einführung von Konzepten verwendet, sondern als Prädikat, das das Verhältnis der Subsumption zweier Konzepte zueinander charakterisiert, und das folglich auch erschlossen werden kann.

18 'RollenTerme' und 'RollenKetten' werden später ausführlicher dargestellt.

Diese Schreibweise impliziert, daß Ausdrücke der Form 'RollenKette : InstanzenName' ebenso wie andere Konzeptterme als Mengen interpretiert werden. Tatsächlich bezeichnet der Ausdruck 'Farbe:rot' beispielsweise die Menge all der Elemente, die mit der Instanz 'rot' über die Rolle 'Farbe' in Relation stehen:

$$\text{(int 5)} \qquad [\![\mathsf{r{:}i}]\!]^{\Im} \quad = \quad \{\ x : [\![\mathsf{i}]\!]^{\Im} \in [\![\mathsf{r}]\!]^{\Im}(x)\ \}$$

Die Schreibweise '$[\![\mathsf{r}]\!]^{\Im}(x)$' ist dabei folgendermaßen zu verstehen:

$$\text{(int 6)} \qquad [\![\mathsf{r}]\!]^{\Im}(x) \quad := \quad \{y :< x,y > \in [\![\mathsf{r}]\!]^{\Im}\}$$

Das Rollenkonstrukt anwendent, können wir die Vaterrelation zwischen 'Kate' und 'Bertrand' nun folgendermaßen darstellen:

(ex 11)	Kate	::	Frau
		and	Vater:Bertrand.

In gleicher Weise können wir die Eigenschaft, 'rot' zu sein — die in (ex 6) eingeführte Instanz 'rot' des Konzeptes 'Farbe' voraussetzend — nun durch eine Rolle anstatt durch Verwendung eines entsprechenden Konzeptes kennzeichnen:

(ex 12)	Erdbeere	:<	Frucht
		and	Farbe:rot.

Rollen werden als Relationen interpretiert. Sie können deshalb auf mehrere Instanzen bzw. Rollenfüller verweisen. Eine Instanz wie die folgende:

(ex 13)	Grimms-Märchen	::	Buch
		and	Farbe:rot
		and	Farbe:blau
		and	Farbe:gelb.

ist also nicht widersprüchlich, sondern besagt nur, daß es sich bei dem betrachteten Buch um ein mehrfarbiges[19] handeln muß. Die erlaubte Anzahl

19 Genaugenommen besagt die Existenz verschiedener Füller der Rolle 'Farbe' natürlich nichts über "Mehrfarbigkeit", sondern einfach nur, daß die Instanz 'Grimms-Märchen' mit den Instanzen 'rot', 'blau' und 'gelb' über eine als 'Farbe' bezeichnete Relation in Beziehung steht. — Die Beantwortung der Frage, wie diese strukturelle Information letztendlich relativ zu dem modellierten Weltausschnitt *interpretiert* werden soll, ist hingegen durch die der Modellierung unterliegende Abstraktion bestimmt.

von Rollenfüllern kann durch andere vordefinierte Konstrukte eingeschränkt werden. Auf diese Möglichkeit soll im nächsten Abschnitt genauer eingegangen werden.

Eigenschaften von Konzepten vererben sich auf Spezialisierungen und Instanzen. Sind beispielsweise 'Taschenbücher' 'blau' und 'Handtaschenbücher' 'Taschenbücher' und ist weiter 'Jim-Knopf' ein 'Handtaschenbuch', so folgt, daß 'Jim-Knopf', ebenso wie alle anderen 'Handtaschenbücher', ebenfalls 'blau' sein muß.

Einführung und Definition von Rollen

Im letzten Abschnitt wurde die Verwendung von Rollen beschrieben, ohne genauer auf ihre Einführung einzugehen; dieses soll nun nachgeholt werden.

Rollen werden als Relationen interpretiert, und erhalten damit eine Teilmenge des Kreuzproduktes $\mathcal{D} \times \mathcal{D}$ als Extension. Sei $\mathcal{R}$ die Menge der in einer Terminologie verwendeten Rollennamen, so gilt:

(int 7) $\quad [\![r]\!]^{\Im} \subseteq \mathcal{D} \times \mathcal{D} \qquad$ für alle $r \in \mathcal{R}$

Es liegt nahe, Rollen ähnlich wie Konzepte explizit einzuführen. Dazu kann von der vordefinierten "allgemeinsten" Rolle 'rtop' ausgegangen werden.

(syn 7) $\quad$ RollenName $\longrightarrow$ **rtop**

(int 8) $\quad [\![\mathbf{rtop}]\!]^{\Im} = \mathcal{D} \times \mathcal{D}$

Wie bei der Einführung bzw. Definition von Konzepten verwenden wir die Symbole ':<' und ':=' auch zur Einführung bzw. Definition von Rollen. Wir erhalten die folgende Syntax und Semantik:

(syn 8)	RollenEinführung	$\longrightarrow$	RollenName	:<	RollenTerm
		|	RollenName	:=	RollenTerm.

(mod 4) $\quad \begin{array}{lllll} [\![\mathsf{r}]\!]^{\Im} & \subseteq & [\![\mathsf{rt}]\!]^{\Im} & \mathbf{falls} & \text{'}\mathsf{r} \;:<\; \mathsf{rt}\text{'} \in \mathbf{T} \\ [\![\mathsf{r}]\!]^{\Im} & = & [\![\mathsf{rt}]\!]^{\Im} & \mathbf{falls} & \text{'}\mathsf{r} \;:=\; \mathsf{rt}\text{'} \in \mathbf{T} \end{array}$

Mit den dargestellten Mitteln können wir die Rolle 'Farbe' beispielsweise folgendermaßen einführen:

(ex 14) $\quad$ Farbe :< **rtop**.

Da 'rtop' dem Kreuzprodukt $\mathcal{R} \times \mathcal{R}$ entspricht, ist damit nichts weiter ausgesagt, als daß die Rolle 'Farbe' eine beliebige zweistellige Relation über den Domain $\mathcal{D}$ darstellt.

Rollenterme

Ähnlich den Konzepten können Rollen durch 'and' zu *Rollentermen* verknüpft werden:

(syn 9)
$$\begin{array}{lcl} \text{RollenTerm} & \longrightarrow & \text{RollenTerm} \\ & | & \text{RollenTerm } \mathbf{and} \text{ RollenTerm.} \end{array}$$

(int 9)
$$[\![\mathsf{rt}_1 \quad \mathbf{and} \quad \mathsf{rt}_2]\!]^{\Im} \quad = \quad [\![\mathsf{rt}_1]\!]^{\Im} \quad \cap \quad [\![\mathsf{rt}_2]\!]^{\Im}$$

Die Verbindung von Konzepttermen und Rollentermen ergibt sich durch die Möglichkeit, *Domain* und *Range* einer Rolle zu spezifizieren:

(syn 10)
$$\begin{array}{lcl} \text{RollenTerm} & \longrightarrow & \mathbf{domain}\,(\text{KonzeptTerm}) \\ & | & \mathbf{range}\,(\text{KonzeptTerm}). \end{array}$$

(int 10)
$$\begin{array}{rcl} [\![\mathbf{domain}\,(\mathsf{kt})]\!]^{\Im} & = & [\![\mathsf{kt}]\!]^{\Im} \times \mathcal{D} \\ [\![\mathbf{range}\,(\mathsf{kt})]\!]^{\Im} & = & \mathcal{D} \times [\![\mathsf{kt}]\!]^{\Im} \end{array}$$

Die Rolle 'Vater' beispielsweise läßt sich damit als Relation über 'Menschen' und 'Männer' einführen:

(ex 15)
$$\begin{array}{lll} \text{Vater} & :< & \mathbf{rtop} \\ & \mathbf{and} & \mathbf{domain}\,(\text{Menschen}) \\ & \mathbf{and} & \mathbf{range}\,(\text{Männer}). \end{array}$$

Anzahlrestriktionen auf Rollen

In bestimmten Fällen ist es sinnvoll, die Anzahl möglicher Rollenfüller einzuschränken. Im Falle unserer 'Vater'-Rolle ist es zum Beispiel sicher nicht adäquat, von mehr als einem möglichen Vater pro Person auszugehen. Ebenso lassen sich Konzepte vorstellen, die als Spezialisierung anderer Konzepte eingeführt werden, im Gegensatz zu diesen jedoch in bezug auf bestimmte Rollen weniger oder keine Füller zulassen: Ein Hocker z.B. kann mit diesem Mittel als ein Stuhl ohne Lehne definiert werden. Beschränkungen dieser Art sollen künftig als *Anzahlrestriktionen* bezeichnet werden.

Anzahlrestriktionen lassen sich teilweise direkt bei der Einführung von Rollen angeben, und teilweise in Form eines Konzepttermes bei der Einführung von Konzepten und Instanzen. Wir wollen im folgenden die Möglichkeiten solcher Beschränkungen betrachten:

Anzahlrestriktionen bei der Einführung von Rollen:

Die einzige Anzahlrestriktion, die sich in FLEX direkt bei der Einführung von Rollen spezifizieren läßt, besteht darin, diese als *Features* auszuweisen. Features sind funktionale Rollen, und zeichnen sich damit durch die folgende Eigenschaft aus:

$$feature(R) \overset{def}{\Longleftrightarrow} \forall_{x,y,z\in\mathcal{D}}[(<x,y>\in R \wedge <x,z>\in R) \rightarrow y = z]$$

Eine dieser Eigenschaft entsprechende Beschränkung kann nicht allgemein in Form einer Extension $E \subseteq \mathcal{D} \times \mathcal{D}$ formuliert werden. Die Kennzeichnung einer Rolle als Feature läßt sich folglich nicht durch einen Rollenterm, und damit auch nicht in Form einer Rolleneinführung der bisher betrachteten Form ausdrücken. Features werden deshalb in FLEX durch eine andere Form der Einführung prinzipiell von Rollen unterschieden, so daß ihr funktionales Verhalten durch eine besondere Behandlung innerhalb der Semantik sichergestellt werden kann. Unter Annahme einer Menge von Feature-Namen $\mathcal{F}$ kommen wir zu den folgenden syntaktischen Regeln:

(syn 11)

FeatureName	$\longrightarrow$	f.	für alle $f \in \mathcal{F}$
FeatureEinführung	$\longrightarrow$	FeatureName :< RollenTerm **and** [20] **feature**	
	$\mid$	FeatureName := RollenTerm **and feature**.	

Die Abbildung von Features in Funktionen kann entweder über die Interpretationsfunktion sichergestellt, oder als zusätzliche Bedingung an Modelle einer Terminologie formuliert werden. In unserem Falle wird der zweite Weg beschritten:

(mod 5) $\forall_{x,y,z\in\mathcal{D}}[(<x,y>\in [\![f]\!]^{\Im} \wedge <x,z>\in [\![f]\!]^{\Im}) \rightarrow y = z]$
für alle $f \in \mathcal{F}$

Anzahlrestriktionen innerhalb von Konzepttermen:

Einfacher ist die Formulierung von Anzahlrestriktionen innerhalb von Konzepttermen. Angenommen, wir hätten ein Konzept 'Kind' mit der Rolle 'Geschwister', so ließe sich das Konzept 'Einzelkind' als 'Kind and no(Geschwister)' definieren. Die Interpretation des Konzepttermes 'no(Geschwister)' ist dabei genau die Menge der Elementen des Universums, die hinsichtlich der Rolle 'Geschwister' mit keinem anderen Element in Beziehung stehen:

20 Dieses **and** ist nicht mit dem bisher verwendeten **and** zu verwechseln: Die Kombination '**and feature**' bildet vielmehr eine Einheit und unterscheidet Rolleneinführungen syntaktisch von Featureeinführungen.

$$[\![\mathbf{no}(Geschwister)]\!]^{\Im} \quad = \quad \{d : [\![Geschwister]\!]^{\Im}(d) = \emptyset\}$$

In dieser Weise formulierte Restriktionen lassen sich also extensional als Menge aller sie erfüllenden Elemente interpretieren, und durch **and**— bzw. die Schnittmengenbildeung — mir anderen Konzepttermen verknüpfen:

(syn 12)

$$\begin{array}{lcll} KonzeptTerm & \longrightarrow & \mathbf{atmost}(n, RollenName) & \textit{für} \quad n \in \mathbb{N} \\ & | & \mathbf{atleast}(n, RollenName) & \textit{für} \quad n \in \mathbb{N} \\ & | & \mathbf{some}(RollenName) & \\ & | & \mathbf{no}(RollenName). & \end{array}$$

(int 11)

$$\begin{array}{rcl} [\![\mathbf{atmost}(n, \mathsf{r})]\!]^{\Im} & = & \{d : \ | [\![\mathsf{r}]\!]^{\Im}(d) | \ \leq \ n\} \\ [\![\mathbf{atleast}(n, \mathsf{r})]\!]^{\Im} & = & \{d : \ | [\![\mathsf{r}]\!]^{\Im}(d) | \ \geq \ n\} \\ [\![\mathbf{some}(\mathsf{r})]\!]^{\Im} & = & \{d : \ | [\![\mathsf{r}]\!]^{\Im}(d) | \ > \ 0\} \\ [\![\mathbf{no}(\mathsf{r})]\!]^{\Im} & = & \{d : \ [\![\mathsf{r}]\!]^{\Im}(d) \ = \ \emptyset\} \end{array}$$

'atmost' und 'atleast' reichen aus, um alle anderen Restriktionen aus ihnen zu definieren. Da eine Modellierung durch Verwendung auch der letzten beiden Ausdrücke jedoch übersichtlicher wird, haben wir auch diese eingeführt.

Rollenkomposition

Bei unseren Überlegungen zu Syntax, Semantik und Folgerungsbegriff haben wir darüber nachgedacht, die Großvaterbeziehung auf die Komposition der Vaterbeziehung zurückzuführen. Die systematische Behandlung solcher Rückführungen hatten wir als wesentliche Motivation für die formale Definition von Wissensrepräsentationsformalismen dargestellt. Wie also läßt sich eine solche Abhängigkeit zwischen Rollen in FLEX ausdrücken? Gehen wir zur Beantwortung dieser Frage von dem folgenden Beispiel aus:

(ex 16)

Kate	::	Vater:Bertrand.
Bertrand	::	Vater:John.

Offensichtlich ist 'John' der Vater des Vaters von 'Kate', und damit ihr Großvater. Um diese Verknüpfungen von Rollen beschreiben zu können, stellt FLEX den Ausdruck 'comp' zur Verfügung, der folgendermaßen definiert ist:

(syn 13)

$$\begin{array}{lcl} \text{RollenTerm} & \longrightarrow & \text{RollenKette.}^{21} \\ \text{RollenKette} & \longrightarrow & \text{RollenTerm } \mathbf{comp} \text{ RollenKette} \\ & | & \text{RollenTerm.} \end{array}$$

21 'RollenKetten' sind Namensvarianten von 'RollenTermen', und deshalb prinzipiell nicht notwendig. Wenn ich trotzdem nicht auf sie verzichte, geschieht dieses, um sie im Text besser von anderen Rollentermen unterscheiden zu können.

(int 12) $[\![\mathsf{rt}\ \mathbf{comp}\ \mathsf{rk}]\!]^{\Im} :=$
$$\{<x,z>: \exists_y[<x,y>\in [\![\mathsf{rt}]\!]^{\Im} \wedge\ <y,z>\in [\![\mathsf{rk}]\!]^{\Im}\ ]\}$$

Wir können nun die Rolle 'Großvater' folgendermaßen definieren:

(ex 17) Großvater **:=** Vater **comp** Vater.

Role Value Maps

Mit Hilfe des von FLEX vordefinierten Ausdruckes 'rvm_equal' läßt sich die Menge der Elemente spezifizieren, die bezüglich zweier Rollenketten mit den gleichen Instanzen in Beziehung stehen:

(syn 14)

KonzeptTerm $\longrightarrow$ **rvm_equal**(RollenKette, RollenKette).

(int 13) $[\![\ \mathbf{rvm\_equal}(\mathsf{rk}_1,\ \mathsf{rk}_2)\ ]\!]^{\Im} := \{x : [\![\mathsf{rk}_1]\!]^{\Im}(x) = [\![\mathsf{rk}_2]\!]^{\Im}(x)\}$

Ist bei einer Instanz oder einem Konzept für nur eine dieser Rollenketten der Wert spezifiziert, kann durch 'rvm_equal' der Wert der nicht spezifizierten Rollenkette durch den Wert der spezifizierten definiert werden. Auf diese Weise können wir die 'Großvater'-Rolle auch folgendermaßen modellieren:

(ex 18)	Mensch	:<	...
		and	**rvm_equal**(Großvater, Vater **comp** Vater).

Regeln

Bestimmte Beziehungen zwischen Konzepten (bzw. Konzepten und Instanzen) lassen sich nur in Form von Regularitäten angeben. Betrachten wir z.B. das Konzept 'Erdbeere' wie es in (ex 12) eingeführt wurde:

Erdbeere	:<	Frucht
	and	Farbe:rot.

Stellen wir uns vor, daß in dem zu modellierenden Kontext alles, was lecker schmeckt und die Farbe 'Rot' hat eine Erdbeere ist. Sei nun ein beliebiger roter und leckerer Gegenstand gegeben, läßt sich folgern, daß es sich um eine Erdbeere handeln muß.

Ein solcher begrifflicher Zusammenhang kann mit den bisher eingeführten Sprachkonstrukten nicht formalisiert werden. Zwar ließe sich eine Erdbeere einfach als Äquivalent zu einem leckeren, roten Gegenstand definieren; die bisherige Kennzeichnung einer solchen als einer roten Frucht ginge dadurch aber verloren. Außerdem würde folgen, daß nicht nur alles, was rot und lecker ist, eine Erdbeere sein muß, sondern auch umgekehrt jede Erdbeere rot ist und lecker schmeckt, was ja nicht gefordert werden soll.

Wollen wir Begriffszusammenhänge der eben beschriebenen Art in einer Beschreibungslogik aufnehmen, müssen wir zu ihrer Darstellung ein neues Konstrukt einführen. Umgangssprachlich formuliert soll eine Beziehung der folgenden Art ausgedrückt werden:

> *Wenn etwas von der Art* A *ist, so folgt, daß es auch von der Art* B *sein muß.*

Unter Verwendung der Begriffe Subsumption und Konzeptterm entspricht dieses der Forderung:

> *Wenn eine Instanz oder ein Konzept von dem Konzeptterm* A *subsumiert wird, so soll er auch vom Konzeptterm* B *subsumiert werden.*

Wir wollen das Symbol '$\Rightarrow$' von nun an zur Darstellung solcher Forderung verwenden, und für zwei Konzeptterme A und B schreiben:

$$\mathsf{A} \Rightarrow \mathsf{B}.$$

Es ergibt sich die folgende syntaktische Regel:

(syn 15) Regel $\longrightarrow$ KonzeptTerm $\Rightarrow$ KonzeptTerm.

Die Semantik eines solchen Ausdruckes bezüglich der Klasse der Modelle können wir folgendermaßen angeben:

(mod 6) $[\![\mathsf{kt}_1]\!]^{\mathfrak{I}} \subseteq [\![\mathsf{kt}_2]\!]^{\mathfrak{I}}$ **falls** '$\mathsf{kt}_1 \Rightarrow \mathsf{kt}_2$' $\in \mathbf{T}$

Die Modellbedingung (mod 6) ähnelt der, die wir bei der Einführung primitiver Konzepte in (mod 1) gefordert haben. Der Unterschied besteht allerdings darin, daß in dem hier dargestellten Fall für kt_1 ein beliebiger Konzeptterm stehen kann, während bei einer Konzepteinführung an dieser Stelle nur ein Konzeptname stehen darf. Etwas wie (ex 19) Frisur:struppig **and** Beine:4 $\Rightarrow$ Köter.

ließe sich folglich in der Form einer Konzepteinführung genausowenig ausdrücken, wie unsere Behauptung bezüglich leckerer, roter Dinge: Diese läßt sich nun — unter Verwendung der Rolle *schmeckt* und der Instanz *lecker* — in der folgenden Weise formalisieren:

(ex 20) schmeckt:lecker **and** Farbe:rot $\Rightarrow$ Erdbeere.

Defaults

Legt man nur die bisher dargestellten Ausdrucksmöglichkeiten zugrunde, rechnet sich die durch diese definierte Beschreibungslogik zu den sogenannten *monotonen Logiken*. Eine monotone Logik zeichnet sich dadurch aus, daß alle aus einer Menge von Voraussetzungen folgerbaren Aussagen auch dann ihre Gültigkeit bewahren, wenn zu dieser Menge beliebige neue Voraussetzungen hinzugefügt werden. Die gefolgerten Sachverhalte gelten also unbedingt, und lassen sich nicht durch später hinzukommendes "Wissen" revidieren.

In unserem letzten Beispiel sind wir davon ausgegangen, daß alles, was von roter Farbe ist und lecker schmeckt, eine Erdbeere sein muß. Würde sich nachträglich herausstellen, daß auch rote Gummibärchen zu den leckeren Dingen unseres Universums gehören, ließe sich dieser neue Sachverhalt nicht in die bisherige Darstellung integrieren. Nur eine Neuformulierung der gesamten Formalisierung, und damit eine Zurückname aller bisherigen Folgerungen könnte dieser neuen Situation gerecht werden.

Durch die Integration von Defaultregeln — und damit den Verzicht auf die Monotonität des Formalismus — wird eine solche nachträgliche Einschränkung bisheriger "Annahmen" möglich. Einigt man sich auf eine Darstellung der Defaultregeln durch das Symbol '$\overset{n}{\leadsto}$', wobei die Ziffer n die Bewertung der "Wichtigkeit" eines Defaults darstellen soll, kann ein Default, das leckere, rote Dinge als Erdbeeren auszeichnet, in der folgenden Weise angegeben werden:

(ex 21) schmeckt:lecker **and** Farbe:rot $\overset{10}{\leadsto}$ Erdbeere.

Durch ein zweite, höher bewertete Defaultregel, läßt sich die erstere revidieren:

(ex 22) schmeckt:lecker **and** Süßigkeit $\overset{20}{\leadsto}$ Gummibärchen.

Keine Aussage verbietet bisher, daß es Dinge geben könnte, die zugleich Erdbeeren und Gummibärchen sind. Da die beiden angegebenen Defaults sich also nicht widersprechen, kommen sie bei leckeren, roten Süßigkeiten beide zur Anwendung. Erst eine Regel wie die folgende, die die Extensionen beider Konzepte als disjunkt erklärt, und damit Erdbeeren und Gummibärchen als miteinander unvereinbare Konzepte auszeichnet, erzwingt das erwünschte Verhalten.

(ex 23) **disjoint** (Erdbeere,Gummibärchen).

(syn 16) Aussage $\longrightarrow$ **disjoint** (KonzeptName,KonzeptName).

(mod 7) $[\![\mathsf{k}_1]\!]^{\Im} \cap [\![\mathsf{k}_2]\!]^{\Im} = \emptyset$ **falls** ‘**disjoint** (k_1,k_2)’ $\in \mathbf{T}$

Syntaktisch werden Defaultregeln analog zu den “normalen” Regeln eingeführt:

(syn 17) Default $\longrightarrow$ KonzeptTerm $\stackrel{n}{\leadsto}$ KonzeptTerm für $n \in \mathbb{N}$

Die semantische Behandlung der Defaultregeln hingegen ist schwieriger, da sie einen anderen Folgerungsbegriff voraussetzt. Versuchen wir uns von diesem eine Vorstellung zu machen:

3.2.3 Defaultfolgerung

Der bisher verwendete Folgerungsbegriff basiert auf der Notion des Modells. Ihm entsprechend folgt eine Aussage B genau dann aus einer Aussage A, wenn alle Modelle von A auch Modelle von B sind. Über den Modellbegriff läßt sich die Menge der Interpretationen einer Terminologie in die Menge der Modelle und die Menge der “Nicht-Modelle” zerlegen. Modelle unterscheiden sich von den “Nicht-Modellen” dadurch, daß in ihnen alle Aussagen der betrachteten Terminologie war sind, während in den “Nicht-Modellen” zumindest gegen eine Aussage der Terminologie verstoßen wird. Da in allen Modellen alle Aussagen gelten, ist jedes Modell zu jedem anderen prinzipiell gleichwertig.

Anders verhält es sich bei den Modellen von Defaultlogiken. Unterteilt man die Aussagen einer Terminologie mit Defaultregeln in die Menge der Defaultaussagen $\mathcal{A}_{\mathcal{D}}$ und die Menge aller übrigen Aussagen $\mathcal{A}_{\mathcal{M}}$ und abstrahiert man von der Menge der Defaultaussagen $\mathcal{A}_{\mathcal{D}}$, unterscheiden sich Defaultmodelle nicht von den bisher betrachteten: Defaultmodelle hinsichtlich $\mathcal{A}_{\mathcal{M}}$ sind genau die Interpretationen, in denen alle Aussagen von $\mathcal{A}_{\mathcal{M}}$ erfüllt werden.

Ob zusätzlich Defaultregeln aus $\mathcal{A}_\mathcal{D}$ erfüllt werden oder nicht, ist für die Eigenschaft, Modell zu sein, nicht von Bedeutung. In einem zweiten Schritt aber lassen sich Defaultmodelle hinsichtlich ihrer "Verstöße" gegen Defaultregeln bewerten und über diese Bewertung ordnen. Diese Ordnung schließlich erlaubt die Kennzeichnung der "günstigsten" Modelle hinsichtlich der Menge $\mathcal{A}_\mathcal{D}$, mit deren Hilfe schließlich die Defaultfolgerung definiert werden kann.

Die Eigenschaft, ein Defaultmodell zu sein, ist also von der Menge der Defaultregeln $\mathcal{A}_\mathcal{D}$ nicht abhängig; wohl aber die Ordnung über die Defaultmodelle. Im Gegensatz zum herkömmlichen Modellbegriff müssen nicht alle Aussagen erfüllt werden, sondern — was die Menge der Defaultregeln $\mathcal{A}_\mathcal{D}$ betrifft — eine (hinsichtlich der Bewertung) möglichst "günstige"[22] Kombination von Aussagen.

Sehen wir uns im folgenden die Bewertungsfunktion und die Ordnung über Defaultmodelle genauer an[23]:

Bewertungsfunktion von Modellen

Zuerst müssen wir präzisieren, wann gegen ein Default verstoßen wird, und wie man die Anzahl solcher Verstöße formal fassen kann:

Menge der Verstöße gegen ein Default hinsichtlich eines Modells $\mathcal{M}$ und einer Terminologie T

Sei $\mathcal{A}_\mathcal{D} = \{\delta_1, \ldots, \delta_n\}$ die Menge der Defaults von $\mathbf{T}$, $\delta \in \mathcal{A}_\mathcal{D}$ mit $\delta = \delta_p \overset{\delta_w}{\leadsto} \delta_c$ ein Element dieser Menge, und $\mathcal{I}$ die Menge der in $\mathbf{T}$ vorkommenden Instanzennamen so berechnet sich die Menge $E_M(\delta)$ der *Ausnahmen* von δ hinsichtlich $\mathcal{M}$ und $\mathbf{T}$ auf die folgende Weise:

$$E_M(\delta) \quad \overset{def}{=} \quad \{i \in \mathcal{I} : [\![i]\!]^\Im \in [\![\delta_p]\!]^\Im \wedge [\![i]\!]^\Im \notin [\![\delta_c]\!]^\Im\}$$

Jedes Objekt also, das die Prämisse eines Defaults erfüllt, gegen dessen Konklusion aber verstößt, gehört der Menge der Verstöße dieses Defaults an.

Mit Hilfe dieser von Terminologie und Modell abhängigen Menge und den Gewichtungen der Defaults lassen sich die Modelle bewerten: $\mid E_M(\delta) \mid$ bezeichnet die Anzahl der relativ zu dem Default δ falsch klassifizierten Objekte. Das Produkt dieser Anzahl und des Gewichtes δ_w des Defaults δ ergibt eine negative Bewertung der Modelle hinsichtlich der Defaultregel δ.

22 Was genau unter dem Terminus "günstig" verstanden werden soll — und das damit nicht unbedingt eine möglichst *große* Menge von Aussagen gemeint ist — ergibt sich aus der Definition der Bewertungsfunktion im nächsten Abschnitt.

23 Die dargestellte Behandlung der Defaults ist Suska 1994 entnommen, und kann dort genauer eingesehen werden.

Betrachtet man die Summe dieser Bewertungen über die Menge der Defaults erhält man eine Bewertung des gesamten Modells:

$$\mathbf{score}^{-}(M) \quad \stackrel{def}{=} \quad \sum_{\delta \in \mathcal{A}_{\mathcal{D}}} (\mid E_M(\delta) \mid \cdot \delta_w)$$

Dem Modell wird also für jeden Verstoß gegen eine Defaultregel durch eine diesem nicht entsprechend klassifizierte Instanz das Gewicht dieses Defaults als Negativbewertung angelastet.

Ordnung über die Modelle und maximale Modelle

Mit Hilfe der Bewertungsfunktion $\mathbf{score}^{-}(M)$ können wir jedem Modell ein Gewicht in Form einer natürlichen Zahl zuordnen. Wir erhalten damit eine quasilineare Ordnung der Modelle. Die Ordnung ist nicht wirklich linear, da Modelle, in denen Defaultregeln verschieden ausgelegt werden, die gleiche Bewertung erhalten können: Seien beispielsweise zwei gleich gewichtete Defaults gegeben, deren Prämissen identisch sind, die sich aber in bezug auf die Konklusionen widersprechen, lassen sich zwei unterschiedliche Modelle konstruieren, die sich — bei gleicher Bewertung — nur hinsichtlich der Klassifizierung eines Objektes relativ zu diesen Defaults unterscheiden. Entsprechend gibt es zu einer Terminologie nicht unbedingt nur ein relativ zu den Defaults optimales Modell, sondern eine Menge sich unter Umständen widersprechender maximaler Modelle. Mit Hilfe dieser Menge maximaler Modelle läßt sich die Defaultfolgerung nun folgendermaßen definieren:

Defaultfolgerung

Eine Aussage der Form 'i :: kt' folgt dann aus einer Terminologie, wenn alle im Sinne der Modellbewertung maximalen Modelle auch Modelle dieser Aussage sind.

Dieser Defaultfolgerungsbegriff läßt sich auf die Default-Subsumption von Konzepten ausdehnen. Da für unsere Modellierung jedoch nur Folgerungen im vorher beschriebenen Sinne interessant sind, soll an dieser Stelle nicht weiter auf diese Möglichkeit eingegangen werden. Genauere Untersuchungen der hier verwendeten Defaultlogik lassen sich jedoch in Quantz und Royer 1992, Quantz und Ryan 1993 und Suska 1994 finden.

3.2.4 Externe Funktionen

Beschreibungslogiken wurden konzipiert, um begriffliche Zusammenhänge elegant modellieren zu können. Diese Eleganz kann aber nur auf Kosten der Ausdrucksmächtigkeit des Formalismus erreicht werden. Bei konkreten Modellierungen wird man deshalb immer wieder an die Grenzen der Ausdrucksmöglichkeiten von Beschreibungslogiken stoßen.[24]

Wollte man beispielsweise eine Rolle 'Bekannter' einführen, die sich aus der Vereinigung der Füller zweier Rollen 'Freund' und 'Nachbar' definiert, käme man in einen solchen Konflikt mit der Mächtigkeit von FLEX: Es gibt keine Möglichkeit, die Vereinigung der Füller zweier Rollen auszudrücken.

Um diesen Schwierigkeiten gerecht zu werden, wird die Möglichkeit bereitgestellt, FLEX durch externe Prädikate zu erweitern. Der vordefinierte Ausdruck 'compute' vermittelt dabei zwischen FLEX und externen Prädikaten:

(syn 18)

KonzeptTerm	$\longrightarrow$	**compute**(Rollenname, Prädikatname(RollenKettenListe)).
RollenKettenListe	$\longrightarrow$	RollenKette
	|	RollenKette, RollenKettenListe.

Unser Beispiel können wir nun folgendermaßen darstellen:

(ex 24)

Mensch	:<	...
	and	**compute**(Bekannter, Vereinigung(Freund, Nachbar)).

Die Interpretation von 'Vereinigung' läßt sich explizit durch ein Prologprädikat definieren. Angenommen, das Prädikat 'Vereinigung' spezifiziert tatsächlich die Vereinigung der Füller zweier Rollen, so ist die Interpretation des Ausdruckes '**compute**(Bekannter, Vereinigung(Freund, Nachbar))' die Menge der Objekte des Domains, deren Füllermenge bezüglich der Rolle 'Bekannter' identisch ist mit der Vereinigung der Füllermengen bezüglich der Rollen 'Freund' und 'Nachbar'.

Allgemein läßt sich also sagen:

(int 14)
$$[\![\mathbf{compute}(\mathsf{r}_0,\ \mathsf{P}(\mathsf{r}_1,\ldots,\mathsf{r}_n))]\!]^{\Im} = \{x : [\![\mathsf{r}_0]\!]^{\Im}(x)=[\![\mathsf{P}]\!]^{\Im}([\![\mathsf{r}_1]\!]^{\Im}(x),\ldots,[\![\mathsf{r}_n]\!]^{\Im}(x))\}$$

Für die n-stellige Funktion P wird die Interpretation durch das externe Prolog-Prädikat $\mathsf{P}' : (\mathcal{D}^*)^n \longrightarrow \mathcal{D}^*$ definiert:

24 Beschreibunslogiken ließen sich relativ einfach auf Turingmächtigkeit erweitern; die von FLEX erzwungene, klare und deklarative Form der Wissensdarstellung würde dadurch aber verloren gehen.

(int 15) $[\![\mathsf{P}]\!]^{\Im} \quad = \quad \mathsf{P'}$

Mit dem Konstrukt 'compute' haben wir die Möglichkeit, die Ausdrucksmächtigkeit von FLEX in Abhängigkeit der Anforderungen durch eine Modellierung beliebig zu erweitern, ohne die Eleganz des Formalismus zu gefährden. Im Extremfall ließe sich die gesamte Modellierung durch ein Prologprogramm realisieren, und FLEX nur als "Interface" zu diesem verwenden:

(ex 25)

Berechnung	:<	**ctop**
	and	**compute**(Ergebnis, modellierung(arg_1, ...,arg_n)).
Anwendung	::	Berechnung
	and	arg_1
	⋮	⋮
	and	arg_n.

Es bleibt darauf hinzuweisen, daß externe Prädikate nur sehr begrenzt durch den Folgerungskalkül von FLEX berücksichtigt werden können. Nur konkrete Resultate eines externen Prädikates — wie die Belegung einer Rolle mit bestimmten Füllern als Folge der Anwendung eines Compute-Ausdruckes — können von dem Folgerungskalkül berücksichtigt werden. Die eigentliche Semantik eines externen Prädikates jedoch bleibt dem Kalkül verschlossen.

3.2.5 Anfragen an FLEX

Eine Modellierung dient der Beantwortung von Fragen hinsichtlich der modellierten Sachverhalte. In diesem Absatz sollen deshalb die von FLEX bereitgestellten Möglichkeiten, Anfragen zu formulieren, beschrieben werden.

Anfragen an FLEX werden mit Hilfe eines Folgerungskalküls beantwortet. Dieses Kalkül ist hinsichtlich des semantischen Folgerungsbegriffes von Beschreibungslogiken nicht vollständig, so daß nicht alle Anfragen immer richtig beantwortet werden können. Für die in dieser Arbeit dargestellte Formalisierung allerdings ist das verwendete Kalkül ausreichend, so daß uns diese Unvollständigkeit nicht weiter interessieren soll.

Wir wollen für eine Terminologie $\mathbf{T}$ und eine Aussage φ im folgenden $\mathbf{T} \vdash_{Flex} \varphi$ schreiben, wenn FLEX die Ableitung von φ aus der Terminologie $\mathbf{T}$ ohne Berücksichtigung der Defaults erlaubt. Werden die Defaults ebenfalls verwendet, schreiben wir $\mathbf{T} \mathrel{|\!\!\sim}_{Flex} \varphi$.

Anfragen bezüglich der Subsumption von Konzepttermen

Seien kt_1 und kt_2 zwei Konzeptterme, und $\mathbf{T}$ eine Terminologie, so wird die Anfrage 'kt_1 ?< kt_2' positiv beantwortet, falls sich mit dem Ableitungskalkül von FLEX aus der Terminologie $\mathbf{T}$ die Subsumption von kt_1 durch kt_2 beweisen läßt. In allen anderen Fällen — d.h. auch wenn das Kalkül nicht mächtig genug ist, um eine vorhandene Subsumption nachzuweisen — wird die Anfrage negativ beantwortet. Formal können wir schreiben:

$$\mathsf{kt}_1 \ ?< \ \mathsf{kt}_2 = \begin{cases} yes & \text{falls } \mathbf{T} \ \vdash_{Flex} \mathsf{kt}_1 \ :< \ \mathsf{kt}_2, \\ no & \text{sonst} \end{cases}$$

Anfragen bezüglich der Subsumption von Instanzen durch Konzeptterme

Die Subsumption einer Instanz i durch einen Konzeptterm kt läßt sich durch die Anfrage i ?: kt überprüfen. Kann die Subsumption von FLEX nachgewiesen werden, ergibt sich eine positive Antwort; besteht hingegen keine Subsumptionsbeziehung, oder kann eine solche nicht nachgewiesen werden, fällt die Antwort negativ aus:

$$\mathsf{i} \ ?: \ \mathsf{kt} = \begin{cases} yes & \text{falls } \mathbf{T} \ \vdash_{Flex} \mathsf{i} :: \mathsf{kt}, \\ no & \text{sonst} \end{cases}$$

Ermitteln der Füller einer Rolle

Die Anfrage 'obj_get_role_fillers' erlaubt die Berechnung aller bekannten — d.h. explizit in einer Terminologie eingeführten — Objekte hinsichtlich einer Instanz i und einer Rolle r ohne Berücksichtigung von Defaults:

$$\textbf{obj\_get\_role\_fillers}(\mathsf{i}, \mathsf{r}) = \{p : \mathbf{T} \vdash_{Flex} \mathsf{i}{::}\mathsf{r}{:}p\}$$

Sollen Defaults ebenfalls betrachtet werden, kann der Ausdruck 'obj_dbox_get_role_fillers' verwendet werden, und wir erhalten:

$$\textbf{obj\_dbox\_get\_role\_fillers}(\mathsf{i}, \mathsf{r}) = \{p : \mathbf{T} \mid\!\sim_{Flex} \mathsf{i}{::}\mathsf{r}{:}p\}$$

Wir haben damit die für unsere Modellierung wesentlichen Konstrukte von FLEX einführend beschrieben. Es verbleibt, das Gesagte formal zu präzisieren, und insbesondere auf den verwendeten Folgerungskalkül näher einzugehen. Da das Thema dieser Arbeit jedoch die Modellierung japanischer Verben und nicht der dazu zugrundegelegte Formalismus FLEX ist, soll das bisher Gesagte an dieser Stelle genügen. Es sei jedoch auf die angegebene Literatur verwiesen, wenn ein genaueres Verständnis einzelner Punkte erwünscht ist.

Zum Schluß werden die bisher beschriebenen Regeln und Bedingungen noch einmal zusammenfassend aufgeführt. Gleichzeitig sollen dabei einige verbleibende Lücken — wie beispielsweise die Startregel der beschriebenen Syntax — gefüllt werden.

3.2.6 Zusammenfassung von Syntax, Interpretation und Modellbedingungen

Syntax

Terminologie	$\longrightarrow$	Ausdruck *.
Ausdruck	$\longrightarrow$	Einführung
	$\mid$	Implikation
	$\mid$	Aussage.
Einführung	$\longrightarrow$	InstanzenEinführung
	$\mid$	KonzeptEinführung
	$\mid$	RollenEinführung
	$\mid$	FeatureEinführung.
InstanzenEinführung	$\longrightarrow$	InstanzenName :: KonzeptTerm.
InstanzenName	$\longrightarrow$	i. für alle $i \in \mathcal{I}$
KonzeptEinführung	$\longrightarrow$	KonzeptName :< KonzeptTerm
	$\mid$	KonzeptName := KonzeptTerm.
KonzeptName	$\longrightarrow$	k. für alle $k \in \mathcal{K} \cup \{\mathbf{ctop}, \mathbf{cbot}\}$
KonzeptTerm	$\longrightarrow$	KonzeptName
	$\mid$	KonzeptTerm **and** KonzeptTerm
	$\mid$	RollenKette : InstanzenName
	$\mid$	**atmost** (n,RollenName) für $n \in \mathbb{N}$
	$\mid$	**atleast** (n,RollenName) für $n \in \mathbb{N}$
	$\mid$	**some** (RollenName)
	$\mid$	**no** (RollenName)
	$\mid$	**rvm_equal** (RollenKette, RollenKette)
	$\mid$	**compute** (Rollenname, Prädikatname(RollenKettenListe)).
RollenEinführung	$\longrightarrow$	RollenName :< RollenTerm
	$\mid$	RollenName := RollenTerm.
RollenName	$\longrightarrow$	r. für alle $r \in \mathcal{R} \cup \{\mathbf{rtop}, \mathbf{rbot}\}$
RollenTerm	$\longrightarrow$	RollenName
	$\mid$	RollenTerm **and** RollenTerm
	$\mid$	RollenKette
	$\mid$	**domain** (KonzeptTerm)
	$\mid$	**range** (KonzeptTerm).

RollenKette	$\longrightarrow$	RollenTerm
	$\mid$	RollenTerm **comp** RollenKette.
RollenKettenListe	$\longrightarrow$	RollenKette
	$\mid$	RollenKette, RollenKettenListe.
FeatureEinführung	$\longrightarrow$	FeatureName :< RollenTerm **and feature**
	$\mid$	FeatureName := RollenTerm **and feature**.
FeatureName	$\longrightarrow$	f. für alle f $\in \mathcal{F}$
Implikation	$\longrightarrow$	Regel
	$\mid$	Default.
Regel	$\longrightarrow$	KonzeptTerm $\Rightarrow$ KonzeptTerm.
Default	$\longrightarrow$	KonzeptTerm $\overset{n}{\leadsto}$ KonzeptTerm. für $n \in \mathbb{N}$
Aussage	$\longrightarrow$	**disjoint** (KonzeptName,KonzeptName).

Interpretation

Es werden die folgenden Variablen verwendet:

k	für Konzepte
kt	für Konzeptterme
i	für Instanzen
r	für Rollen
rt	für Rollenterme
rk	für Rollenketten
P	für Prolog-Prädikate
n	für Natürliche Zahlen

$$
\begin{array}{rcll}
[\![k]\!]^{\Im} & \subseteq & \mathcal{D} & \text{für alle } k \in \mathcal{K} \\
[\![\mathbf{ctop}]\!]^{\Im} & := & \mathcal{D} & \\
[\![\mathbf{cbot}]\!]^{\Im} & := & \emptyset & \\
[\![\mathsf{kt}_1 \ \mathbf{and}\ \mathsf{kt}_2]\!]^{\Im} & := & [\![\mathsf{kt}_1]\!]^{\Im} \cap [\![\mathsf{kt}_2]\!]^{\Im} & \\
[\![\mathsf{r{:}i}]\!]^{\Im} & := & \{\ x : [\![\mathsf{i}]\!]^{\Im} \in [\![\mathsf{r}]\!]^{\Im}(x)\ \} & \\
[\![\mathsf{r}]\!]^{\Im}(x) & := & \{y :< x,y > \in [\![\mathsf{r}]\!]^{\Im}\} & \\
[\![r]\!]^{\Im} & \subseteq & \mathcal{D} \times \mathcal{D} & \text{für alle } r \in \mathcal{R} \\
[\![\mathbf{rtop}]\!]^{\Im} & := & \mathcal{D} \times \mathcal{D} & \\
[\![\mathsf{rt}_1 \ \mathbf{and}\ \mathsf{rt}_2]\!]^{\Im} & := & [\![\mathsf{rt}_1]\!]^{\Im} \cap [\![\mathsf{rt}_2]\!]^{\Im} & \\
[\![\mathbf{domain}\,(\mathsf{kt})]\!]^{\Im} & := & [\![\mathsf{kt}]\!]^{\Im} \times \mathcal{D} & \\
[\![\mathbf{range}\,(\mathsf{kt})]\!]^{\Im} & := & \mathcal{D} \times [\![\mathsf{kt}]\!]^{\Im} & \\
[\![\mathbf{atmost}\,(n,\mathsf{r})]\!]^{\Im} & := & \{d :\mid [\![\mathsf{r}]\!]^{\Im}(d) \mid \leq n\} & \\
[\![\mathbf{atleast}\,(n,\mathsf{r})]\!]^{\Im} & := & \{d :\mid [\![\mathsf{r}]\!]^{\Im}(d) \mid \geq n\} & \\
[\![\mathbf{some}\,(\mathsf{r})]\!]^{\Im} & := & \{d :\mid [\![\mathsf{r}]\!]^{\Im}(d) \mid > 0\} & \\
[\![\mathbf{no}\,(\mathsf{r})]\!]^{\Im} & := & \{d : [\![\mathsf{r}]\!]^{\Im}(d) = \emptyset\} & \\
[\![\mathsf{rt}\ \mathbf{comp}\ \mathsf{rk}]\!]^{\Im} & := & \{< x, z >: \exists_y [< x, y > \in [\![\mathsf{rt}]\!]^{\Im} \wedge < y, z > \in [\![\mathsf{rk}]\!]^{\Im}]\} & \\
[\![\ \mathbf{rvm\_equal}\,(\mathsf{rk}_1,\ \mathsf{rk}_2)\ ]\!]^{\Im} & := & \{x : [\![\mathsf{rk}_1]\!]^{\Im}(x) = [\![\mathsf{rk}_2]\!]^{\Im}(x)\} & \\
[\![\mathbf{compute}\,(\mathsf{r}_0,\ \mathsf{P}(\mathsf{r}_1,\ldots,\mathsf{r}_n))]\!]^{\Im} & := & \{x : [\![\mathsf{r}_0]\!]^{\Im}(x) = [\![\mathsf{P}]\!]^{\Im}([\![\mathsf{r}_1]\!]^{\Im}(x), \ldots, [\![\mathsf{r}_n]\!]^{\Im}(x))\} & \\
[\![\mathsf{P}]\!]^{\Im} & := & \mathsf{P'} &
\end{array}
$$

Modell

$$
\begin{array}{rclcrcl}
[\![\mathsf{i}\ ]\!]^{\Im} & \in & [\![\mathsf{kt}\ ]\!]^{\Im} & \textbf{falls} & \text{`}\mathsf{i}\ ::\ \mathsf{kt}\text{'} & \in & \mathbf{T} \\
[\![\mathsf{k}\ ]\!]^{\Im} & \subseteq & [\![\mathsf{kt}\ ]\!]^{\Im} & \textbf{falls} & \text{`}\mathsf{k} :< \mathsf{kt}\text{'} & \in & \mathbf{T} \\
[\![\mathsf{r}]\!]^{\Im} & \subseteq & [\![\mathsf{rt}]\!]^{\Im} & \textbf{falls} & \text{`}\mathsf{r} :< \mathsf{rt}\text{'} & \in & \mathbf{T} \\
[\![\mathsf{k}]\!]^{\Im} & = & [\![\mathsf{kt}]\!]^{\Im} & \textbf{falls} & \text{`}\mathsf{k} := \mathsf{kt}\text{'} & \in & \mathbf{T} \\
[\![\mathsf{r}]\!]^{\Im} & = & [\![\mathsf{rt}]\!]^{\Im} & \textbf{falls} & \text{`}\mathsf{r} := \mathsf{rt}\text{'} & \in & \mathbf{T} \\
[\![\mathsf{k}_1]\!]^{\Im} \cap [\![\mathsf{k}_2]\!]^{\Im} & = & \emptyset & \textbf{falls} & \text{`}\mathbf{disjoint}\,(\mathsf{k}_1,\mathsf{k}_2)\text{'} & \in & \mathbf{T} \\
[\![\mathsf{kt}_1]\!]^{\Im} & \subseteq & [\![\mathsf{kt}_2]\!]^{\Im} & \textbf{falls} & \text{`}\mathsf{kt}_1 \Rightarrow \mathsf{kt}_2\text{'} & \in & \mathbf{T}
\end{array}
$$

$$\forall_{x,y,z \in \mathcal{D}}[(< x, y > \in [\![f]\!]^{\Im} \wedge < x, z > \in [\![f]\!]^{\Im}) \rightarrow y = z] \qquad f\ddot{u}rallef \in \mathcal{F}$$

4 Beschreibung der japanischen Verbmorphologie

Nachdem die Grundlagen der japanischen Verbmorphologie, die Prinzipien der Modellierung und die Beschreibungslogik FLEX vorgestellt wurden, soll nun überlegt werden, wie sich diese drei Komponenten zu einem System vereinen lassen. Wir kommen damit zu dem zentralen Abschnitt der Arbeit: der Modellierung eines hinsichtlich der auftretenden morphologischen Probleme repräsentativen Ausschnittes der japanischen Grammatik in der Beschreibunslogik FLEX.

FLEX ist ein System zur Darstellung konzeptuellen Wissens, und kein computerlinguistischer Formalismus. Es ist daher nicht verwunderlich, daß es nicht zur Darstellung aller zur Wirkung kommenden Mechanismen adäquat sein kann, sondern an verschiedenen Stellen durch Prolog-Prädikate erweitert werden muß — sei es intern durch in Prolog formulierte compute-Prädikate oder extern durch Einbettung der FLEX-Modellierung in ein Prolog-Programm. Das im folgenden skizzierte System gleicht in seiner Gesamtstruktur daher einem Netz interdependenter Regularitäten, aus dem nur schwer ein Teil beschrieben werden kann, ohne das Verständnis der übrigen vorauszusetzen. Von Anfang an werden deshalb bei der Erklärung der einzelnen Aspekte der Beschreibung Ergebnisse erst später beschriebener Komponenten als gegeben vorausgesetzt. Aus einem anfangs recht sperrigen Gebilde entsteht dabei — durch schrittweises "Auflösen" vorausgesetzter, statischer Bausteine durch dynamische Regelsysteme — am Ende das Bild der interagierenden Gesamtstruktur mit all ihren internen Verknüpfungen und Abhängigkeiten.

4.1 Morphologische Zeichen

4.1.1 Linguistische Zeichen als Instanzen

In 2.1.3 (S. 19) haben wir linguistische Zeichen als rekursive Merkmal-Wert-Matrizen definiert. An dieser Stelle wollen wir uns nun überlegen, wie sich die Objekte unserer Beschreibung — d.h. morphologische Formen und ihre Eigenschaften — durch solche Zeichen darstellen lassen, und wie diese in FLEX modelliert werden können.

Erinnern wir uns an die in (ex 6*b*) (S. 9) dargestellte Kausativform des Verbes 書く (*kaku*; schreiben):

(ex 29) 書かせる
kakaseru
zum Schreiben veranlassen

Diese Form läßt sich folgendermaßen in Morphe bzw. Morpheme untergliedern:

(ex 30)	Morphe:	*kaka*	*se*	*ru*
	Morpheme:	{kaku}	{Kausativ}	{Flexiv}

In einer ersten Näherung lassen sich Form und linguistische Eigenschaften[25] durch die folgenden Attribute charakterisieren:

1. 'phon' — Die *Form*, die durch das Zeichen dargestellt wird.
2. 'lexeme' — Attribut zur Kennzeichnung des Verbes (bzw. des lexikalischen Stammorphems), von dem die durch das Zeichen beschriebene Form abgeleitet wurde.
3. 'morphemes' — Attribut zur Beschreibung der Morpheme aus denen sich eine Form zusammensetzt .

Wir kommen damit zu dem folgenden Zeichen:

(ex 31) *Der Kausativ des Verbes 書く (kaku; schreiben)*
als Beispiel zur Darstellung von Verbformen als linguistische Zeichen:

<kaku, causative, flexive[26]>		*sign*[26]
phon	:	'kakaseru'
lexeme	:	'kaku'
morphemes	:	<{kaku}, {causative}, {flexive}>

Diese Darstellung läßt sich nahezu direkt in FLEX übertragen:

(flex 1) *Der Kausativ des Verbes 書く (kaku; schreiben)*
als Beispiel zur Darstellung linguistischer Zeichen in FLEX

```
kaku_causative_flexive :: sign
        and     phon            : 'kakaseru'
        and     lexeme          : 'kaku'
        and     morphemes       : '[kaku,causative,flexive]'27.
```

25 Vergleiche die Unterscheidung der Aspekte linguistischer Zeichen in solche der *Form* und solche der *linguistischen Eigenschaften* in 2.1.1, Seite 13.

26 Die erste Zeile linguistischer Zeichen dient — soweit dieses im jeweiligen Fall erforderlich bzw. möglich ist — der Vergabe eines *Namens*, mit dem sich das Zeichen eindeutig kennzeichnen läßt, und der Angabe des *Zeichentyps*. Der Name erscheint in spitzen Klammern links in der ersten Zeile; der Zeichentyp wird rechts vom Zeichennamen angegeben.

Die Motivation zu dieser zusätzlichen Information ergibt sich direkt aus der Umsetzung der Zeichen in FLEX: vergleiche (ex 31) und (flex 1).

Die Definition bzw. Einführung des Konzeptes ‘sign’, als dessen Instanz wir unser Zeichen dargestellt haben, und der Attribute bzw. Rollen, aus denen es sich zusammensetzt, sind für uns vorläufig nicht von Interesse.

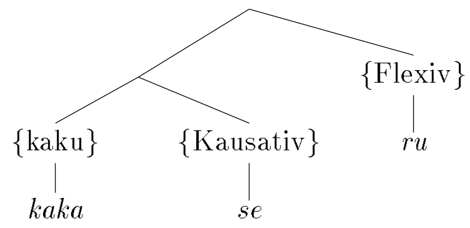
Mit diesen einfachen Mitteln haben wir bereits eine erste Näherung an die Lösung unseres Problems gefunden: Wir können morphologische Formen als linguistische Zeichen innerhalb eines Vollformlexikons darstellen. Unser eigentliches Interesse gilt aber der inneren morphologischen Struktur dieser Zeichen, und damit der *Komposition* von Morphemen zu komplexen Formen, der wir uns im nächsten Abschnitt zuwenden wollen.

4.2 Die Komposition von Zeichen

4.2.1 Komplexe Zeichen

Das Thema dieser Arbeit ist die *Morphologie japanischer Verben.* Die im letzten Abschnitt betrachtete Form ist morphologisch komplex, denn sie setzt sich aus drei *Morphen* bzw. *Morphemen* zusammen:

(ex 32)



Wir müssen unsere bisherige Darstellung also durch ein komplexeres Zeichen ersetzen, das dieser rekursiven Struktur Rechnung trägt:

(ex 33) *Der Kausativ des Verbes 書く (kaku; schreiben), dargestellt als rekursives bzw. komplexes Zeichen:*

27 Listen sind in FLEX als Rollenfüller nicht zugelassen. Sie werden daher als Atome modelliert, die für evtl. Operationen auf ihnen explizit in Listen konvertiert werden müssen. (s.h. auch Fußnote 29 S. 77)

$$
\begin{bmatrix}
\langle\textit{kaku, causative, flexive}\rangle & & \textit{composed\_morphological\_sign}\\
\textbf{phon} & : & \text{‘kakaseru’}\\
\textbf{lexeme} & : & \{\mathsf{kaku}\}\\
\textbf{morphemes} & : & \langle\{\mathsf{kaku}\}, \{\mathsf{causative}\}, \{\mathsf{flexive}\}\rangle\\
\textbf{daughter\_base} & : & \begin{bmatrix}
\langle\textit{kaku, causative}\rangle & & \textit{composed\_morphological\_sign}\\
\textbf{phon} & : & \text{‘kakase’}\\
\textbf{lexeme} & : & \{\mathsf{kaku}\}\\
\textbf{morphemes} & : & \langle\{\mathsf{kaku}\}, \{\mathsf{causative}\}\rangle\\
\textbf{daughter\_base} & : & \begin{bmatrix}
\langle\textit{kaku}\rangle & & \textit{root\_sign}\\
\textbf{phon} & : & \text{‘kaka’}\\
\textbf{lexeme} & : & \{\mathsf{kaku}\}\\
\textbf{morpheme} & : & \{\mathsf{kaku}\}\\
\textbf{morphemes} & : & \langle\{\mathsf{kaku}\}\rangle
\end{bmatrix}\\
\textbf{daughter\_suffix} & : & \begin{bmatrix}
\langle\textit{causative}\rangle & & \textit{suffix\_sign}\\
\textbf{phon} & : & \text{‘se’}\\
\textbf{morpheme} & : & \{\mathsf{causative}\}\\
\textbf{morphemes} & : & \langle\{\mathsf{causative}\}\rangle
\end{bmatrix}
\end{bmatrix}\\
\textbf{daughter\_suffix} & : & \begin{bmatrix}
\langle\textit{flexive}\rangle & & \textit{flexive\_sign}\\
\textbf{phon} & : & \text{‘ru’}\\
\textbf{morpheme} & : & \{\mathsf{flexive}\}\\
\textbf{morphemes} & : & \langle\{\mathsf{flexive}\}\rangle
\end{bmatrix}
\end{bmatrix}
$$

Die Attribute ‘daughter_base’ und ‘daughter_suffix’ haben linguistische Zeichen als Werte. In FLEX können wir diese rekursive Einbettung von Zeichen in Zeichen darstellen, indem wir jedes Teilzeichen als Instanz modellieren und den Namen der Instanzen, die den eingebetteten Zeichen entsprechen, als Füller der Rollen ‘daughter_base’ bzw. ‘daughter_suffix’ verwenden. Wir kommen damit zu folgender Darstellung:

(flex 2) *Der Kausativ des Verbes 書く (kaku; schreiben) als Beispiel zur Darstellung rekursiver linguistischer Zeichen in* FLEX

```
kaku_causative_flexive :: composed_morphological_sign
        and     phon            : 'kakaseru'
        and     lexeme          : 'kaku'
        and     morphemes       : '[kaku,causative,flexive]'
        and     daughter_base   : kaku_causative
        and     daughter_suffix : flexive.

kaku_causative :: composed_morphological_sign
        and     phon            : 'kakase'
        and     lexeme          : 'kaku'
        and     morphemes       : '[kaku,causative]'
        and     daughter_base   : kaku
        and     daughter_suffix : causative.

kaku :: root_sign
        and     phon            : 'kaka'
        and     lexeme          : 'kaku'
        and     morpheme        : 'kaku'
        and     morphemes       : '[kaku]'.
```

```
causative :: suffix_sign
        and     phon            : 'se'
        and     morpheme        : 'causative'
        and     morphemes       : '[causative]'.

flexive :: flexive_sign
        and     phon            : 'ru'
        and     morpheme        : 'flexive'
        and     morphemes       : '[flexive]'.
```

Die dargestellten Zeichen zerfallen in die folgenden Kategorien:

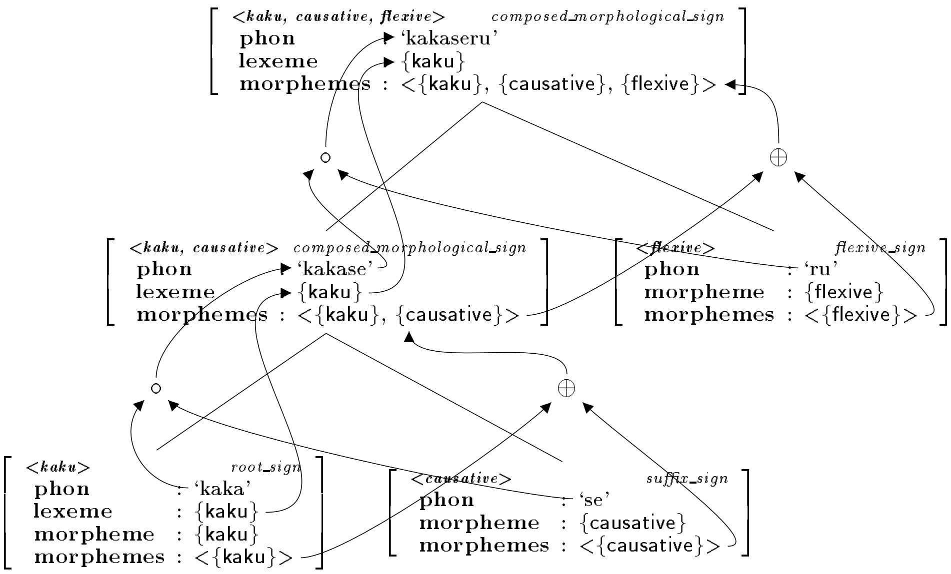
1. *Atomare Zeichen* bzw. Zeichen für *Morpheme* — Wir unterscheiden zwischen drei verschiedenen Morphemklassen:
   (a) *Stämme* ('root_sign') — Jede morphologische Form hat als Ausgangspunkt den *Stamm* bzw. das *Stammorphem* eines Verbes. In unserem Beispiel ist {kaku} das Stammorphem des Verbes 'kaku'.
   (b) *Suffixe* ('suffix_sign') — Morpheme, die sich an ein Stammorphem bzw. eine morphologisch komplexe Form anschließen, werden als *Suffixe* bezeichnet.
   (c) *Flexive* ('flexive_sign') — *Flexive* sind spezielle Suffixe, die den morphologischen Bildungsprozeß abschließen. Die durch ihre Suffigierung erhaltene Form wird damit Objekt der nächsthöheren linguistischen Beschreibungsebene, d.h. der Syntax.
2. *komplexe Zeichen* ('composed_morphological_sign') bzw. Zeichen für *Morphemkompositionen* — Formen, die sich aus mehreren Morphemen zusammensetzen, bezeichnen wir als *komplexe Zeichen*.

Diese Unterteilung der linguistischen Zeichen motiviert sich dadurch, daß Morpheme in Abhängigkeit der Kategorie, der sie angehören, verschiedene linguistische Eigenschaften haben können, und sich unter Umständen dadurch bedingt unterschiedlich verhalten.

Wir modellieren diese Kategorien als unterschiedliche Zeichenarten bzw. Konzepte. Die einzelnen Zeichen für Morpheme und Morphemketten werden dann — wie in (flex 2) schon dargestellt — als Instanzen der jeweiligen Zeichenarten definiert.

4.2.2 Komposition und grammatische Regeln

Bis zu diesem Zeitpunkt haben wir Form, Eigenschaften und Struktur unserer Zeichen explizit angegeben. Dank der neuen, geschachtelten Darstellungsform ist aber offensichtlich, daß die Attribute komplexer Zeichen aus denen ihrer Töchter vorhersagbar sind, und folglich mit Hilfe grammatischer Regeln, wie wir sie im Abschnitt "Grammatikalische Regularitäten als Bedingungen über linguistische Zeichen" (2.1.4, S. 23) eingeführt haben, ermittelt werden können. Verwenden wir das Symbol '$\circ$ ' zur Kennzeichnung der Komposition von Formen und das Symbol '$\oplus$ ' zur Darstellung der Verkettung von Listen, können wir diese Zusammenhänge folgendermaßen veranschaulichen:[28]



(fig 10) *Attribut-Abhängigkeiten in komplexen Zeichen*

Es ergeben sich also die folgenden Abhängigkeiten:

1. der Wert des Attributes 'phon' komplexer Zeichen ergibt sich aus der Komposition der 'phon'-Attribute seiner Tochterzeichen;

2. das Attribut 'lexeme' ergibt sich entlang des Attributes 'daughter_base' aus dem Verbstamm;

28 Die Attribute 'daughter_base' und 'daughter_suffix' wurden der besseren Darstellbarkeit halber in diesem Diagramm aus den Klammern herausgezogen, so daß sich die rekursive Einbettung von Zeichen als Baum darstellt.

3. die Liste der Morpheme 'morphemes' schließlich ergibt sich, indem an die Liste der Morpheme des Basiszeichens die entsprechende Liste der Suffixtochter angehängt wird.

Damit kommen wir zu den folgenden Regeln:

(ex 34) *a.* $$\begin{bmatrix} \textit{composed\_} \\ \textit{morphological\_} \\ \textit{sign} \end{bmatrix} \Longrightarrow \begin{bmatrix} \textbf{phon} : P1 \circ P2 \\ \textbf{daughter\_base} \mid \textbf{phon} : P1 \\ \textbf{daughter\_suffix} \mid \textbf{phon} : P2 \end{bmatrix}$$

b. $$\begin{bmatrix} \textit{composed\_} \\ \textit{morphological\_} \\ \textit{sign} \end{bmatrix} \Longrightarrow \begin{bmatrix} \textbf{lexeme} : L \\ \textbf{daughter\_base} \mid \textbf{lexeme} : L \end{bmatrix}$$

c. $$\begin{bmatrix} \textit{composed\_} \\ \textit{morphological\_} \\ \textit{sign} \end{bmatrix} \Longrightarrow \begin{bmatrix} \textbf{morphemes} : M1 \oplus M2 \\ \textbf{daughter\_base} \mid \textbf{morphemes} : M1 \\ \textbf{daughter\_suffix} \mid \textbf{morphemes} : M2 \end{bmatrix}$$

Zusammengefaßt ergibt sich die folgende Regel:

(ex 35) $$\begin{bmatrix} \textit{composed\_} \\ \textit{morphological\_} \\ \textit{sign} \end{bmatrix} \Longrightarrow \begin{bmatrix} \textbf{phon} & : P1 \circ P2 \\ \textbf{lexeme} & : L \\ \textbf{morphemes} & : M1 \oplus M2 \\ \textbf{daughter\_base} & : \begin{bmatrix} \textbf{phon} & : P1 \\ \textbf{lexeme} & : L \\ \textbf{morphemes} & : M1 \end{bmatrix} \\ \textbf{daughter\_suffix} & : \begin{bmatrix} \textbf{phon} & : P2 \\ \textbf{morphemes} & : M2 \end{bmatrix} \end{bmatrix}$$

Dank der Unterscheidung verschiedener Zeichenarten mittels entsprechender Konzepte können wir diese Regel als FLEX-Regel in unsere Modellierung übernehmen. Die Komposition von Formen und die Listenverkettung modellieren wir mit entsprechenden compute-Prädikaten[29]:

(flex 3) *Berechnung der Merkmale* 'phon', 'lexeme' *und* 'morphemes' *komplexer Zeichen aus den entsprechenden Merkmalen ihrer Töchter:*

```
composed_morphological_sign  =>
       compute(  phon,      concat( daughter_base   comp phon,
```

29 FLEX erlaubt keine Rollenketten in compute-Prädikaten. In einer tatsächlichen Modellierung müssen Rollenkomposition und Konkatenation bzw. Rollenverkettung deshalb in einem Prolog-Prädikat kombiniert, und dieses innerhalb des compute-Ausdruckes verwendet werden. (Bezüglich der Morphemlisten als Füller des Attributes 'morphemes' vergleiche auch die Fußnote 27 auf Seite 73)

Im folgenden werden wir auf Anmerkungen wie diese verzichten, wenn die Realisierung zwar aus gegebenen Beschränkungen von FLEX umständlicher erfolgen muß, sich aber an dem prinzipiellen Vorgehen nichts ändert.

```
                                     daughter_suffix comp phon ))
     and rvm_equal( lexeme,          daughter_base comp lexeme )
     and compute(   morphemes, append( daughter_base   comp morphemes,
                                     daughter_suffix comp morphemes )).
```

Versteht man diese morphologischen Regeln in der Form, wie wir sie im Abschnitt 2.1.4 "Grammatikalische Regularitäten als Bedingungen über linguistische Zeichen" (S. 23) eingeführt haben, d.h. als "Filter" über alle kombinatorisch möglichen Zeichen, genügt diese deklarative Form der Darstellung, um einen mächtigen Teil der Morphologie des Japanischen zu beschreiben. Unser Thema beinhaltet jedoch die Umsetzung der Regeln in die durch die compute-Erweiterungen verursachte, bedingt prozedurale Realität des Systems FLEX, aus der sich, wie im nächsten Kapitel beschrieben, einige Einschränkungen ergeben.

4.2.3 Grenzen der Modellierung in FLEX

Die Beschreibungslogik FLEX wurde konzipiert, um Konzeptstrukturen angeben zu können, und diese zur Beschreibung von Situationen und Objekten zu verwenden. Die sich aus dieser Aufgabenstellung ergebenden Beschreibungsmittel sind denen, die in der Computerlinguistik zur formalen Beschreibung von Sprache verwandt werden, so ähnlich, daß sie — wie es hier geschehen soll — auch auf diesem Gebiet angewandt werden können. Trotzdem ergibt sich aus der linguistischen Verwendung eine Verschiebung der Anforderungen, so daß die Möglichkeiten, die durch eine Beschreibungslogik gegeben sind, teilweise über das Benötigte hinausgehen, und damit die Verarbeitung unnötig komplex machen, teilweise aber auch den durch die sprachliche Struktur gestellten Anforderungen nicht genügen.

FLEX wurde der Zielsetzung unserer Arbeit entsprechend trotzdem einem existierenden linguistischen Formalismus vorgezogen, da es erlaubt, Vererbungshierarchien und Defaults, die ja Thema dieser Arbeit sind, miteinander kombiniert zu verwenden. Aus der Auswertung der in der linguistischen Beschreibung wirklich benötigten Darstellungsmittel ergibt sich als ein Resultat dieser Arbeit die Spezifikation eines Formalismus, der den Ansprüchen der linguistischen Formalisierung entgegenkommt — und damit eine Anregung für weitere Untersuchungen.

Um die Diskrepanz zwischen den durch FLEX bereitgestellten Beschreibungsmöglichkeiten und denen eines aus linguistischer Sicht adäquaten Systems deutlich zu machen, wollen wir uns nun anhand eines Beispiels überlegen, was ein im Sinne unserer Modellierung "ideales" System, das die in 4.2.2 angegebenen Regeln als Filter bzw. Constraint auffaßt, aus dem bisher Bekannten und einigen wenigen zusätzlichen Regeln zu folgern in der Lage sein könnte.

Berechnung der Form aus einer Liste von Morphemen

Gegeben sei das folgende, teilspezifizierte Zeichen:

(ex 36) $$\left[\begin{array}{lr} & \textit{sign} \\ \textbf{morphemes} : <\{\mathsf{kaku}\}, \{\mathsf{causative}\}, \{\mathsf{flexive}\}> & \end{array}\right]$$

Zu dem bisher Bekannten wollen wir die folgenden Regeln als gültig annehmen:

(ex 37)

a. $\textit{sign} \Longrightarrow \textit{suffix\_sign} \vee \textit{flexive\_sign} \vee \textit{root\_sign} \vee \textit{composed\_morphological\_sign}.$

b. $$\textit{root\_sign} \Longrightarrow \left[\begin{array}{lll} <\textit{kaku}> & & \textit{root\_sign} \\ \textbf{phon} & : \text{‘kaka’} & \\ \textbf{lexeme} & : \{\mathsf{kaku}\} & \\ \textbf{morpheme} & : \{\mathsf{kaku}\} & \\ \textbf{morphemes} & : <\{\mathsf{kaku}\}> & \end{array}\right] \vee \quad \ldots$$

c. $$\textit{suffix\_sign} \Longrightarrow \left[\begin{array}{lll} <\textit{causative}> & & \textit{suffix\_sign} \\ \textbf{phon} & : \text{‘se’} & \\ \textbf{morpheme} & : \{\mathsf{causative}\} & \\ \textbf{morphemes} & : <\{\mathsf{causative}\}> & \end{array}\right] \vee \quad \ldots$$

d. $$\textit{flexive\_sign} \Longrightarrow \left[\begin{array}{lll} <\textit{flexive}> & & \textit{flexive\_sign} \\ \textbf{phon} & : \text{‘ru’} & \\ \textbf{morpheme} & : \{\mathsf{flexive}\} & \\ \textbf{morphemes} & : <\{\mathsf{flexive}\}> & \end{array}\right] \vee \quad \ldots$$

Das Zeichen (ex 38) entspricht keinem der in (ex 37) angegebenen Morphemzeichen (‘root_sign’, ‘suffix_sign’ bzw. ‘flexive_sign’) und muß nach (ex 37a) folglich vom Typ ‘composed_morphological_sign’ sein. Damit folgt aber, daß es alle in (ex 35), Abschnitt (4.2.2, S. 77) aufgestellten Bedingungen für komplexe Zeichen erfüllen muß. Da bisher nur das Merkmal ‘morphemes’ bekannt ist, und dieses mehrere Morpheme enthält, folgt also, daß es Tochterzeichen geben muß, die hinsichtlich eben dieses Merkmals so spezifiziert sein müssen, daß die Verknüpfung ihrer Werte die Liste

$$<\{\mathsf{kaku}\}, \{\mathsf{causative}\}, \{\mathsf{flexive}\}>$$

ergibt. Von den möglichen Zerlegungen soll an dieser Stelle nur eine weiterverfolgt werden, da sie die einzige ist, die auch die übrigen Bedingungen erfüllt:

(ex 38) $$<\{\mathsf{kaku}\}, \{\mathsf{causative}\}, \{\mathsf{flexive}\}> \;=\; <\{\mathsf{kaku}\}, \{\mathsf{causative}\}> \oplus <\{\mathsf{flexive}\}>.$$

Die gleiche Argumentation führt schließlich zur Zerlegung der Kette

$\langle\{\mathsf{kaku}\}, \{\mathsf{causative}\}\rangle$

so daß wir schließlich zu folgendem Zeichen kommen:

$$\text{(ex 39)}\quad \left[\begin{array}{lll} & & \hfill\textit{composed\_morphological\_sign} \\ \textbf{morphemes} & : & \langle\{\mathsf{kaku}\}, \{\mathsf{causative}\}, \{\mathsf{flexive}\}\rangle \\ \textbf{daughter\_base} & : & \left[\begin{array}{lll} & & \hfill\textit{composed\_morphological\_sign} \\ \textbf{morphemes} & : & \langle\{\mathsf{kaku}\}, \{\mathsf{causative}\}\rangle \\ \textbf{daughter\_base} & : & \left[\ \textbf{morphemes} : \langle\{\mathsf{kaku}\}\rangle\ \right] \\ \textbf{daughter\_suffix} & : & \left[\ \textbf{morphemes} : \langle\{\mathsf{causative}\}\rangle\ \right] \end{array}\right] \\ \textbf{daughter\_suffix} & : & \left[\ \textbf{morphemes} : \langle\{\mathsf{flexive}\}\rangle\ \right] \end{array}\right]$$

Die Werte des Attributes 'morphemes' der innersten Zeichen enthalten jeweils nur ein Morphem, können also weiter nicht zerlegt werden. Sie widersprechen damit den Bedingungen komplexer Zeichen und müssen nach Regel (ex 37a) folglich vom Typ 'root_sign', 'suffix_sign' oder 'flexive_sign' sein. Aus den Werten des Merkmals 'morphemes' und den Regeln für Wurzel-, Suffix- und Flexiv-Zeichen (ex 37) ergibt sich ihre genaue Gestalt und damit das folgende Zeichen:

$$\text{(ex 40)}\quad \left[\begin{array}{lll} & & \hfill\textit{composed\_morphological\_sign} \\ \textbf{morphemes} & : & \langle\{\mathsf{kaku}\}, \{\mathsf{causative}\}, \{\mathsf{flexive}\}\rangle \\ \textbf{daughter\_base} & : & \left[\begin{array}{lll} & & \hfill\textit{composed\_morphological\_sign} \\ \textbf{morphemes} & : & \langle\{\mathsf{kaku}\}, \{\mathsf{causative}\}\rangle \\ \textbf{daughter\_base} & : & \left[\begin{array}{lll} \langle\textit{kaku}\rangle & & \hfill\textit{root\_sign} \\ \textbf{phon} & : & \text{`kaka'} \\ \textbf{lexeme} & : & \{\mathsf{kaku}\} \\ \textbf{morpheme} & : & \{\mathsf{kaku}\} \\ \textbf{morphemes} & : & \langle\{\mathsf{kaku}\}\rangle \end{array}\right] \\ \textbf{daughter\_suffix} & : & \left[\begin{array}{lll} \langle\textit{causative}\rangle & & \hfill\textit{suffix\_sign} \\ \textbf{phon} & : & \text{`se'} \\ \textbf{morpheme} & : & \{\mathsf{causative}\} \\ \textbf{morphemes} & : & \langle\{\mathsf{causative}\}\rangle \end{array}\right] \end{array}\right] \\ \textbf{daughter\_suffix} & : & \left[\begin{array}{lll} \langle\textit{flexive}\rangle & & \hfill\textit{flexive\_sign} \\ \textbf{phon} & : & \text{`ru'} \\ \textbf{morpheme} & : & \{\mathsf{flexive}\} \\ \textbf{morphemes} & : & \langle\{\mathsf{flexive}\}\rangle \end{array}\right] \end{array}\right]$$

Aus den Regeln (ex 35) ergeben sich schließlich die Werte der Merkmale 'phon' und 'lexeme' und damit die folgende Zeichengestalt:

(ex 41)

$$
\left[\begin{array}{lll}
 & & \textit{composed\_morphological\_sign} \\
\textbf{phon} & : & \text{`kakaseru'} \\
\textbf{lexeme} & : & \{\mathsf{kaku}\} \\
\textbf{morphemes} & : & <\{\mathsf{kaku}\}, \{\mathsf{causative}\}, \{\mathsf{flexive}\}> \\
\textbf{daughter\_base} & : & \left[\begin{array}{lll}
 & & \textit{composed\_morphological\_sign} \\
\textbf{phon} & : & \text{`kakase'} \\
\textbf{lexeme} & : & \{\mathsf{kaku}\} \\
\textbf{morphemes} & : & <\{\mathsf{kaku}\}, \{\mathsf{causative}\}> \\
\textbf{daughter\_base} & : & \left[\begin{array}{lll}
\textit{<kaku>} & & \textit{root\_sign} \\
\textbf{phon} & : & \text{`kaka'} \\
\textbf{lexeme} & : & \{\mathsf{kaku}\} \\
\textbf{morpheme} & : & \{\mathsf{kaku}\} \\
\textbf{morphemes} & : & <\{\mathsf{kaku}\}>
\end{array}\right] \\
\textbf{daughter\_suffix} & : & \left[\begin{array}{lll}
\textit{<causative>} & & \textit{suffix\_sign} \\
\textbf{phon} & : & \text{`se'} \\
\textbf{morpheme} & : & \{\mathsf{causative}\} \\
\textbf{morphemes} & : & <\{\mathsf{causative}\}>
\end{array}\right]
\end{array}\right] \\
\textbf{daughter\_suffix} & : & \left[\begin{array}{lll}
\textit{<flexive>} & & \textit{flexive\_sign} \\
\textbf{phon} & : & \text{`ru'} \\
\textbf{morpheme} & : & \{\mathsf{flexive}\} \\
\textbf{morphemes} & : & <\{\mathsf{flexive}\}>
\end{array}\right]
\end{array}\right]
$$

Ein Vergleich mit (ex 33) (S. 73) zeigt, daß dieses Zeichen dem gesuchten Ergebnis entspricht.

Von einer Liste von Morphemen ausgehend können wir also allein durch Anwendung der morphologischen Regeln auf das vollständige Zeichen schließen. Insbesondere haben wir damit auch ein Verfahren, um zu einer beliebigen Morphem-Liste die entsprechende morphologische Realisierung zu berechnen.

Analyse einer Form in die konstituierenden Morpheme

In ganz ähnlicher Weise, wie im letzten Abschnitt von den Morphemen über das vollständig spezifizierte Zeichen auf die Form geschlossen wurde, läßt sich auch aus der Form über das vollständige Zeichen auf die Liste der konstituierenden Morpheme schließen.

Sehen wir uns das folgende Zeichen an:

(ex 42) $\left[\ \textbf{phon} : \text{`kakaseru'}\ \right]$

Kein Zeichen vom Typ 'root_sign', 'suffix_sign' bzw. 'flexive_sign' besitzt 'kakaseru' als Wert des Merkmals 'phon', so daß dieses Zeichen nach (ex 37*a*) nur vom Typ 'composed_morphological_sign' sein kann. Aus den möglichen Zerlegungen von 'kakaseru' und den morphologischen Constraints ergibt sich über eine ähnliche Argumentationskette wie der im letzten Abschnitt dargestellten wiederum das in (ex 41) dargestellte Zeichen und damit auch die gesuchte Kette von Morphemen.

Entscheidung über die Korrektheit eines Zeichens

Betrachten wir die folgenden Zeichen:

(ex 43) *a.*

$$\begin{bmatrix} \textbf{phon} & : \text{‘kakaseru’} \\ \textbf{morphemes} & : \langle\{\mathsf{flexive}\}, \{\mathsf{causative}\},\{\mathsf{kaku}\}\rangle \end{bmatrix}$$

b.

$$\begin{bmatrix} \textbf{lexeme} & : \{\mathsf{kaku}\} \\ \textbf{daughter\_base} & : \begin{bmatrix} \textbf{daughter\_suffix} : \begin{bmatrix} \textbf{phon} : \text{‘se’} \end{bmatrix} \end{bmatrix} \\ \textbf{daughter\_suffix} & : \begin{bmatrix} \textbf{morpheme} : \{\mathsf{flexive}\} \end{bmatrix} \end{bmatrix}$$

Die Merkmale des Zeichens (ex 43*a*) führen entlang der in den letzten beiden Abschnitten dargestellten Argumentation zu einem Widerspruch, und können folglich nicht im Sinne der gegebenen morphologischen Spezifikation korrekt sein.

Das Zeichen (ex 43*b*) hingegen entpuppt sich als möglich und sogar eindeutig bestimmt, denn es führt zu dem gleichen Zeichen wie die Argumentationen aus den letzten beiden Abschnitten.

Vergleich des postulierten Systems mit der Beschreibungslogik FLEX

Bei der Umsetzung des soeben dargestellten morphologischen Regelsystems in die Beschreibungslogik FLEX stoßen wir auf Schwierigkeiten, die mit ihren Konsequenzen an dieser Stelle aufgezeigt werden sollen:

1. *Listen als Merkmalswerte* — Listen sind in FLEX als Füller von Rollen nicht zugelassen. Wir können den Wert des Attributes ‘morphemes’ also nicht als Prolog-Liste darstellen, sondern sind gezwungen, das Verhalten einer Liste durch ein Prolog-Atom zu simulieren.

2. *Listenverkettung* — Die Listenverkettung ist in FLEX nicht vorgesehen, und muß durch ein compute-Prädikat realisiert werden. Compute-Prädikate können aber durch den FLEX-Folgerungskalkül nicht direkt berücksichtigt werden, stoßen diesen jedoch durch ihre Resultate erneut an. Damit kann die Verkettung von Listen nicht als echtes Prädikat bzw. Constraint im Sinne unserer in Kapitel 1 vorgestellten, deklarativen Grammatikregeln dargestellt werden, sondern nur als gerichtete, d.h. in nur einer Richtung anwendbare Funktion bzw. Prozedur.

3. *Komposition von Formen* — Auch die Komposition von Rollenfüllern ist in FLEX nicht vorgesehen, und wird daher durch ein compute-Prädikat simuliert. Es ergeben sich die gleichen Konsequenzen wie bei der Listenverkettung.

4. *Constraints* — Während Constraints wie die in (ex 37) angegebenen einfach bestimmte Bedingungen an Zeichen stellen, und somit keine vorgegebene Anwendungsrichtung haben, lassen sich die compute-Prädikate nur in einer, schon bei ihrer Definition festgelegten Richtung benutzen.

   Die Regel (ex 34*c*) wurde in unserem Beispiel für die folgenden Folgerungen angewandt:

   (ex 44) *a.* *Generierung:*
   $<\{\mathsf{kaku}\}, \{\mathsf{causative}\}, \{\mathsf{flexive}\}>$
   $\Longrightarrow \quad <\{\mathsf{kaku}\}, \{\mathsf{causative}\}> \oplus <\{\mathsf{flexive}\}>.$
   $\Longrightarrow \quad <\{\mathsf{kaku}\}> \oplus <\{\mathsf{causative}\}> \oplus <\{\mathsf{flexive}\}>.$

   b. *Analyse:*
   $<\{\mathsf{kaku}\}> \oplus <\{\mathsf{causative}\}> \oplus <\{\mathsf{flexive}\}>.$
   $\Longrightarrow \quad <\{\mathsf{kaku}\}, \{\mathsf{causative}\}> \oplus <\{\mathsf{flexive}\}>.$
   $\Longrightarrow \quad <\{\mathsf{kaku}\}, \{\mathsf{causative}\}, \{\mathsf{flexive}\}>$

   Wir können also die gleiche Regel sowohl zur *Generierung* als auch zur *Analyse* von Formen verwenden. Geben wir den Regeln hingegen eine Richtung, d.h. interpretieren wir sie nicht als Filter für korrekte Zeichen, sondern als Prozeduren zur Berechnung bestimmter Rollenfüller, müssen wir uns auf eine der beiden Anwendungen beschränken.

5. *Disjunktionen* — Zusätzlich zu den bisher angeführten Einschränkungen besitzen wir in FLEX nicht die Möglichkeit, Konklusionen und Disjunktionen innerhalb der morphologischen Spezifikation so zu verwenden, daß sie zur Errechnung des Gesamtzeichens in der eben dargestellten Weise benutzt werden könnten (vergleiche beispielsweise die Regel (ex 37*a*), S. 79).

Aus den genannten Punkten folgt, daß wir uns bei der Modellierung der Morphologie in FLEX zwischen der *Analyse* und der *Generierung* von Formen entscheiden müssen. Wenn sich daraus aber auch Einschränkungen bezüglich der angegebenen, konkreten Umsetzung der morphologischen Beschreibung in FLEX ergeben, bleibt diese selbst dank ihrer abstrakten, deklarativen Form natürlich für beide Richtungen gültig und anwendbar[30].

30 In Carpenter 1992 wird ein Constraint-basierter Formalismus vorgestellt, der — ähnlich, wie wir es im Abschnitt (4.2.3, S. 78) von einem "idealen" Formalismus gefordert hatten, — keine Festlegung auf eine Verarbeitungsrichtung notwendig macht. Wenn wir hier kein vergleichbares System verwenden, liegt das an unserer Entscheidung, den Schwerpunkt der Untersuchung auf die Anwendbarkeit von Defaults in Kombination mit Vererbungshierarchien zu legen. Defaults aber wurden bisher in kein solches System integriert.

4.2.4 Die Komposition in FLEX

Wie im letzten Abschnitt diskutiert, müssen wir den morphologischen Regeln, die wir in unserer Modellierung benutzen wollen, eine Richtung geben, d.h. uns zwischen der *Analyse* und der *Generierung* von Formen entscheiden. Da bei der Errechnung der Form zu einer Liste von Morphemen alle wesentlichen morphologischen Regularitäten zur Anwendung kommen, bei der Analyse einer vorgegebenen Form hingegen auf viele dieser Regeln verzichtet werden kann, da also die Modellierung der Generierung in gewissem Sinne auch die Analyse enthält, soll im Weiteren, soweit wir uns über die Modellierung in FLEX Gedanken machen, die Generierung von Formen zum Schwerpunkt unserer Überlegungen werden. Die parallel entwickelte morphologische Spezifikation bleibt von dieser Einschränkung allerdings unberührt.

Kommen wir also zurück zu den Regeln (flex 3) und überlegen uns, welche Konsequenzen sich aus der eingeschränkten Adäquatheit von FLEX zur Beschreibung morphologischer Sachverhalte für unsere Modellierung ergeben. Die in (flex 3) formulierten Regeln hatten die folgende Form:

(flex 4) *Berechnung der Merkmale* 'phon', 'lexeme' *und* 'morphemes' *komplexer Zeichen aus den entsprechenden Merkmalen ihrer Töchter*

```
composed_morphological_sign  =>
       compute(   phon,      concat( daughter_base   comp phon,
                                     daughter_suffix comp phon ))
   and rvm_equal( lexeme,            daughter_base comp lexeme )
   and compute(   morphemes, append( daughter_base   comp morphemes,
                                     daughter_suffix comp morphemes )).
```

Die Instantiierung der Attribute 'phon', 'lexeme' und 'morphemes' in den Tochterzeichen lösen entsprechend der Semantik der compute-Prädikate die Berechnung der entsprechenden Attribute des Mutterzeichens aus. Die Regel läßt sich also nur nutzen, wenn die Gesamtstruktur des Zeichens und die Attributwerte der terminalen Zeichen bekannt sind. Da wir jedoch aus der Morphemliste auf die Form schließen wollen, müssen wir zu einer Umsetzung der Regel (ex 34*c*) in FLEX kommen, die die Liste der Morpheme 'morphemes' des Mutterzeichens auf die Töchter verteilt:

(flex 5) *Das Ermitteln der Tochterzeichen gemäß der Liste spezifizierter Morpheme und die Berechnung der Merkmale* 'phon' *und* 'lexeme' *aus den entsprechenden Merkmalen dieser Töchter:*

```
composed_morphological_sign  =>
       compute(   daughter_base,     base(morphemes))
   and compute(   daughter_suffix,   suffix(morphemes))
```

```
          compute(   phon,      concat( daughter_base   comp phon,
                                        daughter_suffix comp phon ))
      and rvm_equal( lexeme,            daughter_base comp lexeme ).
```

Die Prädikate 'base' und 'suffix' haben den Nebeneffekt, die entsprechenden Tochterzeichen einzuführen, bzw. — falls diese wie im Falle von Suffixen oder den Wurzeln von Verben terminal sind — nach ihnen im morphologischen Lexikon zu suchen. Als Füller der Rollen 'daughter_base' bzw. 'daughter_suffix' werden die Namen der eingeführten bzw. gefundenen Tochterinstanzen zurückgegeben. Die Attribute 'phon' und 'lexeme' des Mutterzeichens können wie vorher aus den Tochterzeichen ermittelt werden.[31]

Wir haben uns in Kapitel 1 mit linguistischen Zeichen und der Rolle der Konkatenation für unsere Modellierung befaßt. In diesem Abschnitt wurden die Resultate dieser Überlegungen auf eine konkrete Modellierung mittels der Beschreibungslogik FLEX übertragen. Es wurde gezeigt, wie sich linguistische Zeichen in FLEX darstellen lassen, und wie die Komposition einfacher Zeichen zu komplexen modelliert werden kann.

Die Grundstruktur unseres morphologischen Modells wurde damit vorgestellt. Um diese jedoch klar hervortreten zu lassen, haben wir die morphologischen Regularitäten des Japanischen stark vereinfacht dargestellt. Das folgende Kapitel befaßt sich deshalb eingehender mit der linguistischen Seite unseres Themas, nämlich dem Problem der Allomorphie bei der Modifikation japanischer Verben und den Beschreibungsmitteln, durch die das Zusammenspiel der Allomorphe beschrieben werden kann.

4.3 Allomorphie und Subkategorisierung

Bis zum jetzigen Zeitpunkt sind wir in unseren Überlegungen von einer direkten Entsprechung zwischen Morphem und Morph ausgegangen. Die Umsetzung einer Liste von Morphemen konnte deshalb vereinfachend durch die folgenden Schritte behandelt werden:

1. Auflösung der Morphemliste in die konstituierenden Morpheme

2. Umsetzung der Morpheme in die jeweils entsprechenden Morphe

3. Komposition der Morphe entsprechend der in der Morphemliste vogegebenen Reihenfolge

31 In dieser Arbeit soll die *Möglichkeit* und *Struktur* einer Modellierung der Verbmorphologie untersucht werden, nicht deren detaillierte Realisierung. Um dieser Anforderung nachzukommen, wird an dieser Stelle auf die Angabe der genauen Implementierung der compute-Prädikate verzichtet.

Das tatsächliche Verhältnis zwischen Morphem und Morph gestaltet sich allerdings im Normalfall nicht so eindeutig, da den meisten Morphemen mehrere Allomorphe entsprechen können. Die Wahl des jeweils korrekten Allomorphes wird dabei, wie im Kapitel 2.2.3, S. 33 beschrieben, durch die Abhängigkeiten zwischen Kategorien, Basisformen und Subkategorisierung bestimmt.

Das Zusammenspiel dieser Eigenschaften soll nun beschrieben werden: Der nächste Abschnitt behandelt die Abhängigkeiten zwischen Kategorien und Subkategorisierung; die Beziehungen zwischen Verbstamm und Subkategorisierung bilden das Thema der sich anschließenden Überlegungen.

4.3.1 Kategorien und Subkategorisierung

Japanische Verben bilden einander entsprechende Formen teilweise in unterschiedlicher Art und Weise. Der Vergleich der Formen einer hinreichend großen Menge von Verben führt jedoch zu einer relativ kleinen Menge von Bildungsschemata, denen sich die einzelnen Verben zuordnen lassen. Erklärt man die Formunterschiedlichkeiten durch Allomorphie und bezeichnet die Verbgruppen als Verbkategorien, kann gesagt werden, daß die Kategorie eines Verbes die Wahl des zu suffigierenden Allomorphes bestimmt, oder umgekehrt die Allomorphe für bestimmte Verbkategorien subkategorisieren[32].

Betrachten wir das folgende Beispiel:

(ex 45) *a.* *i.* 書く
kaku
ii. 書かせる
kakaseru
b. *i.* 食べる
taberu
ii. 食べさせる
tabesaseru

Unter den schon betrachteten Präsens- und Kausativformen des Verbes 書く (*kaku*; schreiben) sind die entsprechenden Formen des Verbes 食べる (*taberu*; essen) gezeigt. Die Analyse der Formen in die konstituierenden Morphe ergibt das folgende Schema[33]:

32 Vergleiche die Einführung dieses Begriffes in Abschnitt 2.2.3, Seite 34.

33 Die dargestellte Untergliederung der Formen in Morphe weicht ebenso wie die von uns vorgeschlagene Kategorisierung der Verben von der traditionellen Beschreibung der japanischen Verbmorphologie ab. Die Motivation ergibt sich aus dem formalen Prinzip der Konkatenation als einzig erlaubter Operation auf Formen (vergleiche Abschnitt 2.1.2,

(ex 46)

Verb	Kategorie	Stamm	Kausativ	Flexiv
kaku	konsonantisches Verb	*kaka*	*se*	*ru*
taberu	vokalisches Verb	*tabe*	*sase*	*ru*

Wir haben das Verb 'kaku' der Kategorie der *konsonantischen Verben*, das Verb 'taberu' hingegen der Kategorie der *vokalischen Verben* zugeordnet. Das Kausativ-Morphem 'se' subklassifiziert folglich für die konsonantischen Verben, sein Allomorph 'sase' hingegen für die vokalischen Verben. Unter Verwendung der Rollen 'subcat_cat' und 'cat' kommen wir damit zu den folgenden, revidierten Grundzeichen:

(ex 47) a. *Verbstämme:*

i. *Beispiel eines konsonantischen Verbes:*

$$\left[\begin{array}{lll} \langle \textit{kaku} \rangle & & \textit{root\_sign} \\ \textbf{phon} & : & \text{'kaka'} \\ \textbf{lexeme} & : & \{\mathsf{kaku}\} \\ \textbf{morpheme} & : & \{\mathsf{kaku}\} \\ \textbf{morphemes} & : & \langle\{\mathsf{kaku}\}\rangle \\ \textbf{cat} & : & \text{cons\_verb} \end{array}\right]$$

ii. *Beispiel eines vokalischen Verbes:*

$$\left[\begin{array}{lll} \langle \textit{taberu} \rangle & & \textit{root\_sign} \\ \textbf{phon} & : & \text{'tabe'} \\ \textbf{lexeme} & : & \{\mathsf{taberu}\} \\ \textbf{morpheme} & : & \{\mathsf{taberu}\} \\ \textbf{morphemes} & : & \langle\{\mathsf{taberu}\}\rangle \\ \textbf{cat} & : & \text{vowel\_verb} \end{array}\right]$$

b. *Kausativ-Morphe:*

i. *konsonantische Verben:*

$$\left[\begin{array}{lll} \langle \textit{causative, cons\_verb} \rangle & & \textit{suffix\_sign} \\ \textbf{phon} & : & \text{'se'} \\ \textbf{morpheme} & : & \{\mathsf{causative}\} \\ \textbf{morphemes} & : & \langle\{\mathsf{causative}\}\rangle \\ \textbf{cat} & : & \text{vowel\_verb} \\ \textbf{subcat\_cat} & : & \text{cons\_verb} \end{array}\right]$$

ii. *vokalische Verben:*

$$\left[\begin{array}{lll} \langle \textit{causative, vowel\_verb} \rangle & & \textit{suffix\_sign} \\ \textbf{phon} & : & \text{'sase'} \\ \textbf{morpheme} & : & \{\mathsf{causative}\} \\ \textbf{morphemes} & : & \langle\{\mathsf{causative}\}\rangle \\ \textbf{cat} & : & \text{vowel\_verb} \\ \textbf{subcat\_cat} & : & \text{vowel\_verb} \end{array}\right]$$

Den Zusammenhang zwischen den Merkmalen 'subcat_cat' und 'cat' können wir durch folgendes Prinzip sicherstellen:

Seite 17) und dem dadurch bedingten Fehlen von Transformationen, als auch aus der Nutzung von Defaultmechanismen und Vererbungshierarchien in die Beschreibung der morphologischen Zusammenhänge.

(ex 48) *das Subkategorisierungsprinzip für Suffix-Allomorphe:*

$$\left[\ {}^{\textit{composed\_}}_{\textit{morphological\_}}\ \textit{sign}\ \right] \Longrightarrow \left[\begin{array}{l}\textbf{daughter\_base} \mid \textbf{cat}: C \\ \textbf{daughter\_suffix} \mid \textbf{subcat\_cat}: C\end{array}\right]$$

Neben der Rolle 'subcat_cat' haben wir Suffixen mittels der Rolle 'cat' auch eine Kategorie zugeordnet. Die Motivation ergibt sich aus der Betrachtung der Flexive in (ex 46): Trotz verschiedener Kausativ-Morpheme für vokalische und konsonantische Verben schließen sich an die gebildeten Kausativ-Formen von 'kaku' und 'taberu' die gleichen Flexive an. Dieses Verhalten rechtfertigt sich aber nicht durch ein Zusammenfallen der Präsens-Flexive für vokalische und konsonantische Verben, wie das folgende Beispiel zeigt:

(ex 49) *a.* *Beispiele zur Flektierung konsonantischer und vokalischer Verben:*

i. 書く
kaku

ii. 食べる
taberu

b. *morphologische Analyse der Beispiele:*

Verb	**Kategorie**	**Stamm**	**Flexiv**
kaku	konsonantisches Verb	*kak*	*u*
taberu	vokalisches Verb	*tabe*	*ru*

Die Kausativformen der Verben 'kaku' und 'taberu' verhalten sich bei der Bildung der flektierten Form also beide wie vokalische Verben. Dieses Verhalten entsteht dadurch, daß bei der Suffigierung von Morphemen die entstehende Form nicht die Kategorie der Basis übernimmt, sondern die durch das Suffixmorphem bestimmte Kategorie. Sowohl 'kakase' — d.h. die Kausativform des konsonantischen Verbes 'kaku' — als auch 'tabesase' — d.h. die Kausativform des vokalischen Verbes 'taberu' — übernehmen also vom Kausativsuffix die Kategorie der vokalischen Verben.

Die Vererbung des Merkmals 'cat' vom Suffix auf die komplexe Form läßt sich damit durch folgende Regel beschreiben:

(ex 50) *die Vererbung der Verb-Kategorie bei komplexen Zeichen:*

$$\left[\ {}^{\textit{composed\_}}_{\textit{morphological\_}}\ \textit{sign}\ \right] \Longrightarrow \left[\begin{array}{l}\textbf{cat}: C \\ \textbf{daughter\_suffix} \mid \textbf{cat}: C\end{array}\right]$$

Subkategorisierungsprinzip und die Vererbungsregel der Verb-Kategorie lassen sich mit den schon dargestellten Regeln zusammenfassen:

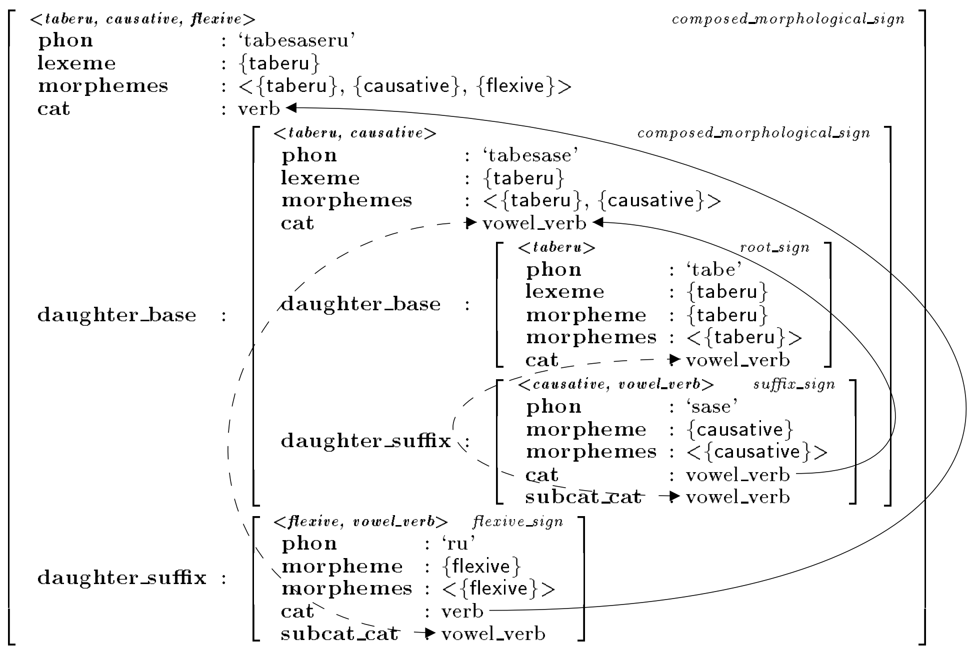
(ex 51)
$$\begin{bmatrix} {}^{\textit{composed\_}}_{\substack{\textit{morphological\_}\\ \textit{sign}}} \end{bmatrix} \Longrightarrow \begin{bmatrix} \textbf{phon} & : P1 \circ P2 \\ \textbf{lexeme} & : L \\ \textbf{morphemes} & : M1 \oplus M2 \\ \textbf{cat} & : C \\ \textbf{daughter\_base} & : \begin{bmatrix} \textbf{phon} & : P1 \\ \textbf{lexeme} & : L \\ \textbf{morphemes} & : M1 \\ \textbf{cat} & : SC \end{bmatrix} \\ \textbf{daughter\_suffix} & : \begin{bmatrix} \textbf{phon} & : P2 \\ \textbf{morphemes} & : M2 \\ \textbf{cat} & : C \\ \textbf{subcat\_cat} & : SC \end{bmatrix} \end{bmatrix}$$

Die morphologischen Prinzipien führen damit zusammen mit den Morph-Zeichen zu den folgenden komplexen Zeichen:

(ex 52) *a.* *kakaseru:*

$$\begin{bmatrix} \multicolumn{2}{l}{\textit{<kaku, causative, flexive>} \hfill \textit{composed\_morphological\_sign}} \\ \textbf{phon} & : \text{`kakaseru'} \\ \textbf{lexeme} & : \{\text{kaku}\} \\ \textbf{morphemes} & : <\{\text{kaku}\}, \{\text{causative}\}, \{\text{flexive}\}> \\ \textbf{cat} & : \text{verb} \\ \textbf{daughter\_base} & : \begin{bmatrix} \multicolumn{2}{l}{\textit{<kaku, causative>} \hfill \textit{composed\_morphological\_sign}} \\ \textbf{phon} & : \text{`kakase'} \\ \textbf{lexeme} & : \{\text{kaku}\} \\ \textbf{morphemes} & : <\{\text{kaku}\}, \{\text{causative}\}> \\ \textbf{cat} & : \text{vowel\_verb} \\ \textbf{daughter\_base} & : \begin{bmatrix} \multicolumn{2}{l}{\textit{<kaku>} \hfill \textit{root\_sign}} \\ \textbf{phon} & : \text{`kaka'} \\ \textbf{lexeme} & : \{\text{kaku}\} \\ \textbf{morpheme} & : \{\text{kaku}\} \\ \textbf{morphemes} & : <\{\text{kaku}\}> \\ \textbf{cat} & : \text{cons\_verb} \end{bmatrix} \\ \textbf{daughter\_suffix} & : \begin{bmatrix} \multicolumn{2}{l}{\textit{<causative, cons\_verb>} \hfill \textit{suffix\_sign}} \\ \textbf{phon} & : \text{`se'} \\ \textbf{morpheme} & : \{\text{causative}\} \\ \textbf{morphemes} & : <\{\text{causative}\}> \\ \textbf{cat} & : \text{vowel\_verb} \\ \textbf{subcat\_cat} & : \text{cons\_verb} \end{bmatrix} \end{bmatrix} \\ \textbf{daughter\_suffix} & : \begin{bmatrix} \multicolumn{2}{l}{\textit{<flexive, vowel\_verb>} \quad \textit{flexive\_sign}} \\ \textbf{phon} & : \text{`ru'} \\ \textbf{morpheme} & : \{\text{flexive}\} \\ \textbf{morphemes} & : <\{\text{flexive}\}> \\ \textbf{cat} & : \text{verb} \\ \textbf{subcat\_cat} & : \text{vowel\_verb} \end{bmatrix} \end{bmatrix}$$

b. *tabesaseru:*



Die Umsetzung unseres Regelsystems in die Beschreibungslogik FLEX gestaltet sich aufwendiger:

Wir haben das Merkmal 'morphemes' bisher auf die Töchter komplexer Zeichen aufgeteilt. Bei einer sich ergebenden Morphemliste mit nur einem Morphem wurde nach einem entsprechenden Lexikoneintrag gesucht, und der Namen des gefundenen Zeichens als Tochter eingetragen. Bei längeren Morphemketten hingegen wurde ein neues komplexes Zeichen mit entsprechender 'morphemes'-Rolle eingeführt, um damit den Bildungsprozeß in rekursiver Weise erneut anzustoßen. Um das Wechselspiel zwischen dem Merkmal 'cat' der Basis und dem Merkmal 'subcat_cat' des Suffixes beschreiben zu können, müssen wir nun mit der Berechnung der Basistochter beginnen, und die Instantiierung des Attributes 'cat' der Basistochter benutzen, um die Wahl der korrekten Suffixtochter anzustoßen.

Die FLEX-Modellierung betreffend ergibt sich folglich als einzige Veränderung die Parametrisierung der 'suffix()'-Regel mit der Rollenkette 'daughter_base com cat':

(flex 6) *Berechnung der korrekten Suffixtochter komplexer Zeichen entsprechend der zu realisierenden Morpheme und der Kategorie der Basistochter:*

```
composed_morphological_sign  =>
   . . .
   and compute(daughter_suffix,suffix(morphemes,daughter_base comp cat))
   . . .
```

Das Prolog-Prädikat hingegen, durch welches das compute-Prädikat 'suffix()' realisiert wird, muß bei der Auswahl des Suffixzeichens neben der Rolle 'morpheme' nun auch den Wert des Attributes 'subcat_cat' berücksichtigen. Es könnte die folgende Form haben:

(flex 7) *Realisierung des compute-Prädikates* 'suffix' *als in einer Prolog-Klausel integrierte* FLEX-*Anfrage:*

```
suffix(Instanz,Morpheme,Cat) :-
   Instanz ::  suffix_sign
           and morpheme   : Morpheme
           and subcat_cat : Cat.
```

4.3.2 Verbstämme und Subkategorisierung

Im letzten Absatz haben wir die Abhängigkeiten zwischen der Kategorie der Basisformen und der Realisierung der Suffixmorpheme behandelt. Basisform und Suffix treten aber — wie schon im Kapitel 2.2.3 beschrieben — noch auf eine weitere Weise in Beziehung. Sehen wir uns dazu die folgenden Beispiele an:

(ex 53)

Verb	Kategorie	Präsens	Konditional	Perfekt	Partizip
書く *kaku*	kons. Verb	書く *kaku*	書けば *kakeba*	書いた *kaita*	書いて *kaite*
飲む *nomu*	kons. Verb	飲む *nomu*	飲めば *nomeba*	飲んだ *nonda*	飲んで *nonde*
食べる *taberu*	vok. Verb	食べる *taberu*	食べれば *tabereba*	食べた *tabeta*	食べて *tabete*

Dargestellt werden Präsens-, Konditional-, Perfekt- und das Partizipialform der uns schon bekannten Verben 書く (*kaku*; schreiben) und 食べる (*taberu*; essen), sowie des Verbes 飲む (*nomu*; trinken), das wir zur Veranschaulichung der Funktion der Verbstämme hinzugenommen haben. Die Teilung der Formen in Morphe soll in unserem Modell in folgender Weise vorgenommen werden:[34]

34 Wie im Abschnitt 1.2, Seite 8 schon beschrieben, ist es schwierig, in diesen Formen die konstituierenden Morphe zu erkennen. Erst der Vergleich einer großen Anzahl verschiedener Formen, gebildet von ebenso vielen verschiedenen Verben, führt letztlich zu Unterteilungen und Regeln, die die zur Wirkung kommenden morphologischen Gesetzmäßigkeiten elegant und effizient zu beschreiben gestatten.

Unter dem Aspekt der Konkatenation als einziger Grundoperation auf Formen (vergleiche 2.1.2, S. 17) wurde für die hier behandelte Modellierung der Verbmorphologie eine Unterteilung der Formen in Morphe gewählt, die durch Abwägen der Anzahl von Allomorphen

(ex 54)

Verb	Kategorie	Präsens	Konditional	Perfekt	Partizip
kaku	kons. Verb	*kak-u*	*kak-eba*	*kait-a*	*kait-e*
nomu	kons. Verb	*nom-u*	*nom-eba*	*nond-a*	*nond-e*
taberu	vok. Verb	*tabe-ru*	*tabe-reba*	*tabet-a*	*tabet-e*

Neben der Allomorphie der Präsens- bzw. Perfekt Flexive, die Thema des letzten Abschnittes waren, ergibt sich als neues Problem auch eine Abweichung in der Realisierung der Verb-Morpheme:

(ex 55) *a.* *Verbmorpheme:*

{kaku} = {*kak, kait*}
{nomu} = {*nom, nond*}
{taberu} = {*tabe, tabet*}

b. *Suffixmorpheme:*

{Flexiv} = {*u, ru*}
{Perfekt-Flexiv} = {*a*}
{Konditional} = {*eba, reba*}
{Partizip} = {*e*}

Die Betrachtung der Morphemfolgen in der Tabelle (ex 54) führt aber schließlich neben der Beobachtung der schon beschriebenen Abhängigkeiten zur Aufdeckung eines weiteren Zusammenhanges zwischen den Realisierungen von Suffix- und Basismorphem: Präsens- und Konditional-Suffix fordern systematisch die Realisierung durch die in (ex 55*a*) zuerst aufgeführten Allomorphe 'kak', 'nom' bzw. 'tabe'; Perfekt- und Partizip-Morphem hingegen verlangen die Realisierung durch das jeweils zweite Allomorph, d.h. 'kait', 'nond' bzw. 'tabet'. die Suffixmorpheme fordern bzw. *subkategorisieren* also eine bestimmte Realisierung bzw. — wie wir sie von nun an nennen wollen — *Stammform* der Verben.

Unter Zuhilfenahme der Rollen 'subcat_base' und 'base' läßt sich dieser Zusammenhang folgendermaßen darstellen:

(ex 56) *Das Subkategorisierungsprinzip für Stammformen* (*vorläufige Version*):

$$\left[\ \begin{array}{r}\scriptstyle composed\_\\ \scriptstyle morphological\_\\ \scriptstyle sign\end{array}\right] \Longrightarrow \left[\begin{array}{l}\textbf{daughter\_base} \mid \textbf{base}: B\\ \textbf{daughter\_suffix} \mid \textbf{subcat\_base}: B\end{array}\right]$$

---

und benötigten morphologischen Regeln den Beschreibungsaufwand möglichst gering zu halten sucht.

Die durch dieses Vorgehen entstehende morphologische Analyse weicht von den mir bekannten Systemen ab, und wäre damit sicherlich einer genaueren Betrachtung wert. Da wir uns in dieser Arbeit jedoch mit der *formalen Seite* einer morphologischen Modellierung beschäftigen, wird auf die Darstellung der ihr zugrundeliegenden Überlegungen verzichtet.

Die Präsens- und Perfekt-Formen des Verbes 'kaku' beispielsweise würden sich mittels dieser Regel in folgender Form ergeben:

(ex 57) *a.* *'kaku', Präsens:*

```
[ <kaku, flexive>                          composed_morphological_sign ]
[ phon             : 'kaku'                                            ]
[ lexeme           : {kaku}                                            ]
[ morphemes        : <{kaku}, {flexive}>                               ]
[ cat              : verb                                              ]
[                    [ <kaku>                    root_sign ]           ]
[                    [ phon         : 'kak'                ]           ]
[                    [ lexeme       : {kaku}               ]           ]
[ daughter_base    : [ morpheme     : {kaku}               ]           ]
[                    [ morphemes    : <{kaku}>             ]           ]
[                    [ cat          : cons_verb            ]           ]
[                    [ base         : k_base               ]           ]
[                    [ <flexive, cons_verb>   flexive_sign ]           ]
[                    [ phon         : 'u'                  ]           ]
[                    [ morpheme     : {flexive}            ]           ]
[ daughter_suffix  : [ morphemes    : <{flexive}>          ]           ]
[                    [ cat          : verb                 ]           ]
[                    [ subcat_cat   : cons_verb            ]           ]
[                    [ subcat_base  : k_base               ]           ]
```

b. *'kaku', Perfekt:*

```
[ <kaku, flexive>                          composed_morphological_sign ]
[ phon             : 'kaita'                                           ]
[ lexeme           : {kaku}                                            ]
[ morphemes        : <{kaku}, {past_flexive}>                          ]
[ cat              : verb                                              ]
[                    [ <kaku>                    root_sign ]           ]
[                    [ phon         : 'kait'               ]           ]
[                    [ lexeme       : {kaku}               ]           ]
[ daughter_base    : [ morpheme     : {kaku}               ]           ]
[                    [ morphemes    : <{kaku}>             ]           ]
[                    [ cat          : cons_verb            ]           ]
[                    [ base         : t_base               ]           ]
[                    [ <past_flexive, cons_verb>  flexive_sign ]       ]
[                    [ phon         : 'a'                      ]       ]
[                    [ morpheme     : {past_flexive}           ]       ]
[ daughter_suffix  : [ morphemes    : <{past_flexive}>         ]       ]
[                    [ cat          : verb                     ]       ]
[                    [ subcat_cat   : cons_verb                ]       ]
[                    [ subcat_base  : t_base                   ]       ]
```

Bei diesem Vorgehen entspricht weiterhin jedes Morph genau einem atomaren Zeichen in unserem Formalismus. Im Abschnitt 4.4 (S. 102) werden wir versuchen, die Beziehungen zwischen den Basisformen von Morphemen durch ein Regelsystem zu beschreiben. Dieses Regelsystem macht zur effizienten Darstellung in FLEX das Zusammenfassen bestimmter Morphe innerhalb eines Zeichens notwendig. Von dem eben dargestellten Vorgehen abweichend wollen wir deshalb die Entsprechung zwischen elementaren Zeichen und Morphen aufgeben, und die Basismorphe der Verben in folgender Weise zusammenfassen:

(ex 58) *Zusammenfassung der Basismorphe des Verbes 'kaku'*

<kaku>		*root_sign*
phon_k_base	:	'kak'
phon_t_base	:	'kait'
lexeme	:	{kaku}
morpheme	:	{kaku}
morphemes	:	<{kaku}>
cat	:	cons_verb

Analog zu (ex 57) kommen wir damit zu folgenden Zeichen:

(ex 59) *a.* *'kaku', Präsens:*

<kaku, flexive>		*composed_morphological_sign*
phon	:	'kaku'
lexeme	:	{kaku}
morphemes	:	<{kaku}, {flexive}>
cat	:	verb
daughter_base	:	<kaku> *root_sign* **phon_k_base** : 'kak' **phon_t_base** : 'kait' **lexeme** : {kaku} **morpheme** : {kaku} **morphemes** : <{kaku}> **cat** : cons_verb
daughter_suffix	:	<flexive, cons_verb> *flexive_sign* **phon** : 'u' **morpheme** : {flexive} **morphemes** : <{flexive}> **cat** : verb **subcat_cat** : cons_verb **subcat_base** : k_base

b. *'kaku', Perfekt:*

<kaku, flexive>		*composed_morphological_sign*
phon	:	'kaita'
lexeme	:	{kaku}
morphemes	:	<{kaku}, {past_flexive}>
cat	:	verb
daughter_base	:	<kaku> *root_sign* **phon_k_base** : 'kak' **phon_t_base** : 'kait' **lexeme** : {kaku} **morpheme** : {kaku} **morphemes** : <{kaku}> **cat** : cons_verb
daughter_suffix	:	<past_flexive, cons_verb> *flexive_sign* **phon** : 'a' **morpheme** : {past_flexive} **morphemes** : <{past_flexive}> **cat** : verb **subcat_cat** : cons_verb **subcat_base** : t_base

Die Regel (ex 56) kann zur Erklärung dieser Formen nicht mehr verwandt werden. Vielmehr müssen wir das Subkategorisierungsprinzip für Stammformen (ex 56) und die Regel zur Bestimmung der Form in komplexen Zeichen (ex 34) (S. 77) zu einer neuen Regel zusamenfassen, um mit der neuen Gestalt unserer Basiszeichen umgehen zu können. An der Stelle einer einzigen Regel für komplexe Zeichen erhalten wir das folgende Regelsystem:

(ex 60) *Das Subkategorisierungsprinzip für Stammformen:*

$$a.\quad \begin{bmatrix} & \textit{composed\_morphological\_sign} \\ \textbf{daughter\_suffix} \mid \textbf{subcat\_base} & : \text{k\_base} \end{bmatrix}$$

$$\Longrightarrow \begin{bmatrix} \textbf{phon} : P1 \circ P2 & \\ \textbf{daughter\_base} \mid \textbf{phon\_k\_base} & : P1 \\ \textbf{daughter\_suffix} \mid \textbf{phon} & : P2 \end{bmatrix}$$

$$b.\quad \begin{bmatrix} & \textit{composed\_morphological\_sign} \\ \textbf{daughter\_suffix} \mid \textbf{subcat\_base} & : \text{t\_base} \end{bmatrix}$$

$$\Longrightarrow \begin{bmatrix} \textbf{phon} : P1 \circ P2 & \\ \textbf{daughter\_base} \mid \textbf{phon\_t\_base} & : P1 \\ \textbf{daughter\_suffix} \mid \textbf{phon} & : P2 \end{bmatrix}$$

Welche der Regeln für ein komplexes Zeichen Gültigkeit hat — und damit welche 'phon'-Rolle der Basistochter Verwendung findet — wird also durch den Wert des Attributes 'subcat_base' der Suffixtochter bestimmt.

4.3.3 Die Hierarchie verbaler Kategorien

In unseren bisherigen Ausführungen haben wir die Verben in die Gruppen der vokalischen und konsonantischen Verben unterteilt. Da nach (ex 48) (S. 88) in komplexen Zeichen die Kategorie des Basiszeichens mit der Kategorie übereinstimmen muß, für die das Suffix subkategorisiert, muß die Anzahl der Allomorphe aller Suffixmorpheme mit der Anzahl der Kategorien übereinstimmen.[35] In unserem Fall müßte jedes Suffixmorphem folglich durch zwei Zeichen dargestellt werden. Eine detailliertere Beschreibung der japanischen Verbmorphologie jedoch müßte von einer weitaus größeren Zahl verbaler Kategorien ausgehen, und damit von einer entsprechend größeren Anzahl linguistischer Zeichen für Morphe. Abgesehen davon, daß eine solche Modellierung unnötig komplex werden müßte, entspräche sie sicherlich nicht der linguistischen Analyse der Verbmorphologie, wie das folgende Beispiel aufzeigen soll. Erinnern wir uns dazu an die in (ex 54) (S. 92) dargestellten Verbformen:

(ex 61)

Verb	**Kategorie**	**Präsens**	**Konditional**	**Perfekt**	**Partizip**
kaku	kons. Verb	*kak-u*	*kak-eba*	*kait-a*	*kait-e*
nomu	kons. Verb	*nom-u*	*nom-eba*	*nond-a*	*nond-e*
taberu	vok. Verb	*tabe-ru*	*tabe-reba*	*tabet-a*	*tabet-e*

Im Gegensatz zum Präsens und Konditional werden Perfekt und Partizip bei allen Verben trotz unterschiedlicher Kategorien durch Suffigierung eines

35 Natürlich bezieht sich diese Beschränkung eigentlich nicht auf die Anzahl der Morphe, sondern auf die Anzahl der Morphzeichen: Zeichen für verschiedene Kategorien können natürlich den gleichen Wert bezüglich des Merkmals 'phon' aufweisen, und damit auf nur ein existierendes Morph verweisen.

einzigen Morphes realisiert. Überträgt man diese Beobachtung auf ein Modell mit einer größeren Anzahl von Kategorien, ergibt sich eine beträchtliche Redundanz in der Beschreibung der Morphe durch linguistische Zeichen.

FLEX erlaubt die Verwendung der Disjunktion bei der Angabe von Rollenfüllern. Ein Ausweg aus dem eben beschriebenen Problem könnte also die Disjunktive Verknüpfung der Kategorien bei der Darstellung solcher Morphe sein:

(ex 62) *Perfekt-Flexiv:*
(konsonantische und vokalische Verben)

$$\left[\begin{array}{lll} \textit{<past\_flexive, cons\_verb>} & & \textit{flexive\_sign} \\ \textbf{phon} & : & \text{'a'} \\ \textbf{morpheme} & : & \{\mathsf{past\_flexive}\} \\ \textbf{morphemes} & : & <\{\mathsf{past\_flexive}\}> \\ \textbf{cat} & : & \text{verb} \\ \textbf{subcat\_cat} & : & \text{cons\_verb } \textbf{or} \text{ vowel\_verb} \\ \textbf{subcat\_base} & : & \text{t\_base} \end{array}\right]$$

Anstelle der *Identität* der Füller, wie wir sie bisher in unserem Subkategorisierungsprinzip gefordert haben, müßten wir nun die *Unifizierbarkeit* der Kategorienfüller fordern, die durch das Symbol '$\overset{?}{\sqcup}$' dargestellt werden soll:

(ex 63) *das Subkategorisierungsprinzip für Suffix-Allomorphe:*

$$\left[\begin{array}{r} \textit{composed\_} \\ \textit{morphological\_} \\ \textit{sign} \end{array}\right] \Longrightarrow \left[\begin{array}{l} \textbf{daughter\_base} \mid \textbf{cat}: C1 \\ \textbf{daughter\_suffix} \mid \textbf{subcat\_cat}: C2 \end{array}\right] \quad \wedge \quad C1 \overset{?}{\sqcup} C2$$

Durch die Notwendigkeit, alle subkategorisierten Kategorien aufzuzählen, wird jedoch auch bei diesem Vorgehen eine unnötig komplexe Darstellungsweise der morphologischen Gesetzmäßigkeiten erzwungen: Eine eingehendere Analyse des morphologischen Verbverhaltens führt nämlich zu der Beobachtung, daß sich die Verben bestimmter Kategorien in verschiedenen Kontexten regelmäßig ähnlich verhalten, und damit hinsichtlich solcher Kontexte besser wie als gemeinsamen Kategorie zugehörig beschrieben werden sollten; in anderen Kontexten jedoch zerfallen die Verben ihrem morphologischen Verhalten entsprechend in noch feinere *Unterklassen*.

Die japanischen Verben lassen sich damit nicht eindeutig in eine bestimmte Menge von Unterklassen zerlegen, sondern deuten durch ihr Verhalten vielmehr auf ein mehrstufiges bzw. hierarchisches Kategoriensystem. Der extensionalen Semantik von Konzepten und Konzeptsubsumption in Beschreibungslogiken entsprechend, können wir damit in der Sprache dieser Systeme sagen, daß die Kategorien japanischer Verben eine *Konzepthierarchie* bilden.

Überlegen wir uns also, wie diese Kategorien-Hierarchie aussehen könnte: Die Untersuchung der bisher betrachteten Beispielformen führt zu den folgenden Ergebnissen:

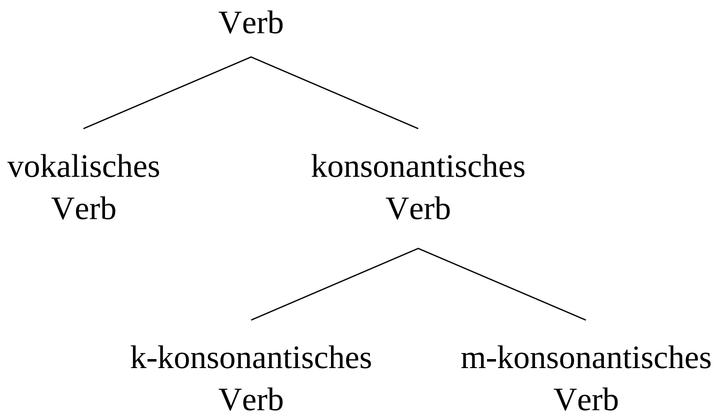
1. Hinsichtlich der Syntax ist es sicherlich sinnvoll, alle Verben unabhängig von ihrem morphologischen Verhalten unter einer gemeinsamen Kategorie zusammenzufassen. Wir wollen diese einfach 'verb' nennen.

   Diese Kategorie erweist sich jedoch auch hinsichtlich morphologischer Prozesse als motiviert: das Vergangenheitsflexiv beispielsweise hat für alle Verben die gleiche Realisierung, und kann folglich hinsichtlich der Rolle 'subcat_cat' durch 'verb' spezifiziert werden.

2. Das Präsensflexiv erscheint bei vokalischen Verben als 'ru', bei konsonantischen Verben hingegen als 'u'. Die nächst feinere Untergliederung wird daher die Zweiteilung der Verben in *vokalische* und *konsonantische* sein.

3. Eine weitere Aufteilung wird sich an späterer Stelle (Abschnitt 4.4, Seite 102) aus der Ableitung der 't-base' ergeben, und betrifft nur die konsonantischen Verben: diese werden hinsichtlich ihres Stammkonsonanten in 'k-konsonantische', 'm-konsonantische', etc. unterteilt.

Wir kommen damit zu folgender Hierarchie verbaler Kategorien, die wir natürlich für eine vollständigere Modellierung noch erweitern müssen:



(fig 11) *Die Hierarchie verbaler Kategorien (Auszug)*

Bei der Beschreibung der Subkategorisierung können wir mittels der Subsumptionsbeziehung diese Kategorienhierarchie ausnutzen. Anstatt die einheitliche Realisierung des Perfektmorphemes durch das Morph 'a' durch eine Menge von Zeichen zu beschreiben, die sich — bei gleichem Wert der 'phon'

Rolle — jeweils nur auf eine Kategorie beziehen, bzw. durch eine lange Disjunktion aller Kategorien innerhalb nur eines Zeichens, spezifizieren wir einfach ‘verb’ als Füller der Rolle ‘subcat_cat’ und verlangen — wie schon in (ex 63) (S. 96) dargestellt — nicht die Gleichheit, sondern die Unifizierbarkeit der Füller.

Wir können damit natürlich genauso

- für alle Kategorien verschiedene Realisierungen eines Morphems angeben, indem wir diese mit ‘vowel_verb’, ‘k_cons_verb’ etc. spezifizieren;
- mittels der Kategorien ‘vowel_verb’ bzw. ‘cons_verb’ nur zwei verschiedene Realisierungen für vokalische und konsonantische Verben fordern;
- wie eben geschildert, unabhängig von der Kategorie eines Verbes die gleiche Realisierung zulassen.

Wollte man bei der Realisierung von Morphemen für bestimmte Verbklassen Ausnahmen von allgemeinen Regeln zulassen, könnte man die Realisierung durch das bezüglich einer Verbkategorie “speziellste” Morph fordern. Beispielsweise könnte man neben einem Morphem, welches für alle Verben subkategorisiert, ein weiteres angeben, daß einzig für m-kons. Verben Gültigkeit hat. Die Suche nach der “speziellsten” Realisierung würde anstatt der generellen Form bei m-kons. Verben folglich das “speziellere”, nur für m-kons. Verben subkategorisierende Morph ergeben. In unserem Ansatz können wir allerdings dank des gewählten Systems von Verbstämmen auf diesen zusätzlichen Defaultmechanismus verzichten. Bei der Ableitung der Basisformen in Abschnitt 4.4 (S. 102) jedoch werden wir auf einen ähnlichen Mechanismus zurückkommen.

Diesen Abschnitt zusammenfassend soll anhand einiger Beispiele die Form unserer Zeichen und Regeln und ihre Umsetzung in die Beschreibungslogik FLEX dargestellt werden. Diese Umsetzung gestaltet sich diesmal sehr einfach, da Regeln und Zeichen so formuliert wurden, daß sie sich direkt in FLEX übersetzten lassen.

Zeichen für Verbmorpheme haben die folgende Gestalt:

(ex 64) *Verbmorpheme: ‘kaku’:*

$$\begin{bmatrix} \langle \textit{kaku} \rangle & & \textit{root\_sign} \\ \textbf{phon\_k\_base} & : & \text{‘kak’} \\ \textbf{phon\_a\_base} & : & \text{‘kaka’} \\ \textbf{phon\_t\_base} & : & \text{‘kait’} \\ \textbf{lexeme} & : & \{\mathsf{kaku}\} \\ \textbf{morpheme} & : & \{\mathsf{kaku}\} \\ \textbf{morphemes} & : & <\{\mathsf{kaku}\}> \\ \textbf{cat} & : & \text{cons\_verb} \end{bmatrix}$$

Das entsprechende Zeichen in FLEX ist zu diesem isomorph:

(flex 8) *Verbmorpheme:*

```
kaku    ::    root_sign
        and   phon_k_base  : 'kak'
        and   phon_a_base  : 'kaka'
        and   phon_t_base  : 'kait'
        and   lexeme       :  kaku
        and   morpheme     :  kaku
        and   morphemes    : '[kaku]'
        and   cat          :  cons_verb.
```

Sehr ähnlich lassen sich Flexive in unserem abstrakten Formalismus darstellen. Nur die verwendeten Rollen unterscheiden sich teilweise von denen der Verbmorpheme:

(ex 65) *Flexive:*

a. *Flexiv:*

$$\begin{bmatrix} \langle \textit{flexive, cons\_verb} \rangle & & \textit{flexive\_sign} \\ \textbf{phon} & : & \text{`u'} \\ \textbf{morpheme} & : & \{\text{flexive}\} \\ \textbf{morphemes} & : & \langle\{\text{flexive}\}\rangle \\ \textbf{cat} & : & \text{verb} \\ \textbf{subcat\_cat} & : & \text{cons\_verb} \\ \textbf{subcat\_base} & : & \text{k\_base} \end{bmatrix} , \begin{bmatrix} \langle \textit{flexive, cons\_verb} \rangle & & \textit{flexive\_sign} \\ \textbf{phon} & : & \text{`ru'} \\ \textbf{morpheme} & : & \{\text{flexive}\} \\ \textbf{morphemes} & : & \langle\{\text{flexive}\}\rangle \\ \textbf{cat} & : & \text{verb} \\ \textbf{subcat\_cat} & : & \text{vowel\_verb} \\ \textbf{subcat\_base} & : & \text{k\_base} \end{bmatrix}$$

b. *Perfekt-Flexiv:*

$$\begin{bmatrix} \langle \textit{past\_flexive, cons\_verb} \rangle & & \textit{flexive\_sign} \\ \textbf{phon} & : & \text{`a'} \\ \textbf{morpheme} & : & \{\text{past\_flexive}\} \\ \textbf{morphemes} & : & \langle\{\text{past\_flexive}\}\rangle \\ \textbf{cat} & : & \text{verb} \\ \textbf{subcat\_cat} & : & \text{verb} \\ \textbf{subcat\_base} & : & \text{t\_base} \end{bmatrix}$$

Wiederum ergibt sich die FLEX-Darstellung in direkter Weise:

(flex 9) *Flexive:*

```
flexive_cons_verb :: flexive_sign
        and   phon         : 'u'
        and   morpheme     : 'flexive'
        and   morphemes    : '[flexive]'
        and   cat          :  verb
        and   subcat_cat   :  cons_verb
        and   subcat_base  :  k_base.

flexive_vowel_verb :: flexive_sign
        and   phon         : 'ru'
```

```
        and   morpheme     : 'flexive'
        and   morphemes    : '[flexive]'
        and   cat          :  verb
        and   subcat_cat   :  vowel_verb
        and   subcat_base  :  k_base.

past_flexive_verb :: flexive_sign
        and   phon         : 'a'
        and   morpheme     : 'past_flexive'
        and   morphemes    : '[past_flexive]'
        and   cat          :  verb
        and   subcat_cat   :  verb
        and   subcat_base  :  t_base.
```

Suffixe — hier anhand des Kausatives dargestellt — teilen mit den Flexiven auch die Rollen:

(ex 66) *Suffixe:*

a. *Kausativ-Suffix, konsonantische Verben:*

$$\begin{bmatrix} \textit{<causative, cons\_verb>} & & \textit{suffix\_sign} \\ \textbf{phon} & : & \text{`se'} \\ \textbf{morpheme} & : & \{\text{causative}\} \\ \textbf{morphemes} & : & <\{\text{causative}\}> \\ \textbf{cat} & : & \text{vowel\_verb} \\ \textbf{subcat\_cat} & : & \text{cons\_verb} \\ \textbf{subcat\_base} & : & \text{a\_base} \end{bmatrix} , \begin{bmatrix} \textit{<causative, vowel\_verb>} & & \textit{suffix\_sign} \\ \textbf{phon} & : & \text{`sase'} \\ \textbf{morpheme} & : & \{\text{causative}\} \\ \textbf{morphemes} & : & <\{\text{causative}\}> \\ \textbf{cat} & : & \text{vowel\_verb} \\ \textbf{subcat\_cat} & : & \text{vowel\_verb} \\ \textbf{subcat\_base} & : & \text{a\_base} \end{bmatrix}$$

In FLEX haben sie damit folgende Form:

(flex 10) *Kausativ-Suffix:*

```
causative_cons_verb :: suffix_sign
        and   phon         : 'se'
        and   morpheme     : 'causative'
        and   morphemes    : '[causative]'
        and   cat          :  vowel_verb
        and   subcat_cat   :  cons_verb
        and   subcat_base  :  a_base.

causative_vowel_verb :: suffix_sign
        and   phon         : 'sase'
        and   morpheme     : 'causative'
        and   morphemes    : '[causative]'
        and   cat          :  vowel_verb
        and   subcat_cat   :  vowel_verb
        and   subcat_base  :  a_base.
```

Die revidierten grammatischen Regeln haben die folgende Gestalt:

(ex 67) a. $$\begin{bmatrix} \textit{composed\_morphological\_sign} \end{bmatrix} \Longrightarrow \begin{bmatrix} \textbf{lexeme} & : L \\ \textbf{morphemes} & : M1 \oplus M2 \\ \textbf{cat} & : C \\ \textbf{daughter\_base} & : \begin{bmatrix} \textbf{lexeme} & : L \\ \textbf{morphemes} & : M1 \\ \textbf{cat} & : SC \end{bmatrix} \\ \textbf{daughter\_suffix} & : \begin{bmatrix} \textbf{morphemes} & : M2 \\ \textbf{cat} & : C \\ \textbf{subcat\_cat} & : SC \end{bmatrix} \end{bmatrix}$$

b. $$\begin{bmatrix} \textit{composed\_morphological\_sign} \\ \textbf{daughter\_suffix} \mid \textbf{subcat\_base} : \text{k\_base} \end{bmatrix} \Longrightarrow \begin{bmatrix} \textbf{phon} : P1 \circ P2 \\ \textbf{daughter\_base} \mid \textbf{phon\_k\_base} : P1 \\ \textbf{daughter\_suffix} \mid \textbf{phon} \quad : P2 \end{bmatrix}$$

c. $$\begin{bmatrix} \textit{composed\_morphological\_sign} \\ \textbf{daughter\_suffix} \mid \textbf{subcat\_base} : \text{t\_base} \end{bmatrix} \Longrightarrow \begin{bmatrix} \textbf{phon} : P1 \circ P2 \\ \textbf{daughter\_base} \mid \textbf{phon\_t\_base} : P1 \\ \textbf{daughter\_suffix} \mid \textbf{phon} \quad : P2 \end{bmatrix}$$

Ihre FLEX-Darstellungen unterscheiden sich — wie schon in Abschnitt 4.2.4, Seite 84 diskutiert — durch die vorgegebene Anwendungsrichtung und das explizite Berechnen der Töchter mittels der Prädikate 'base()' und 'suffix()':

(flex 11) *Das Ermitteln der Tochterzeichen gemäß der Liste spezifizierter Morpheme und die Berechnung der Merkmale* 'phon' *und* 'lexeme' *aus den entsprechenden Merkmalen dieser Töchter:*

```
composed_morphological_sign
   =>   compute(   daughter_base,    base(morphemes))
    and compute(   daughter_suffix, suffix( morphemes,
                                            daughter_base comp cat))
    and rvm_equal( lexeme,           daughter_base comp lexeme )
    and rvm_equal( cat,              daughter_suffix comp cat  ).

     composed_morphological_sign
and (daughter_suffix comp subcat_base):k_base
   =>   compute( phon, concat( daughter_base   comp phon_k_base,
                               daughter_suffix comp phon        )).

     composed_morphological_sign
and (daughter_suffix comp subcat_base):t_base
   =>   compute( phon, concat( daughter_base   comp phon_t_base,
                               daughter_suffix comp phon        )).
```

Durch die Beschreibung des Zusammenspiels von Zeichen und Regeln wurden damit die zentralen Mechanismen unseres morphologischen Modells vorgestellt. Um allerdings die Verbmorphologie des Japanischen vollständiger zu beschreiben, müßten wir — wie schon vorher angedeutet — eine größere Zahl von Verbkategorien und Stammformen voraussetzen. Thema ist jedoch die Frage nach der *Möglichkeit* und *prinzipiellen Struktur* einer morphologischen Beschreibung der japanischen Verben, nicht die konkrete Modellierung der Gesamtheit möglicher Formen. Die Ausformulierung einer vollständigeren Beschreibung der japanischen Verbmorphologie auf der Grundlage der hier dargestellten Ergebnisse soll deshalb einer späteren Arbeit vorbehalten sein.

4.4 Morphologische Regeln und Basisformen

Die Basisformen der Verben wurden bisher als Teil der lexikalischen Einträge im morphologischen Lexikon vorausgesetzt. Ihre Bildung ist jedoch nicht von Verb zu Verb verschieden und läßt sich — genauso wie die bisherigen Subsysteme der Morphologie — durch Bildung von Verbgruppen und auf diesen aufsetzende Regularitäten systematisieren, d.h. in Form von Regelsystemen beschreiben. Anhand der Präsens-, Kausativ- und Partizipialformen der bisher eingeführten Verbkategorien wollen wir nun versuchen, ein solches Regelsystem zu formulieren:

(ex 68) *Präsens, Kausativ und Partizip der Verben 食べる (taberu; essen), 書く (kaku; schreiben) und 飲む (nomu; trinken):*

Verb	Kategorie	Präsens	Kausativ	Partizip
食べる *taberu*	vok. Verb	食べる *taberu*	食べさせる *tabesaseru*	食べて *tabete*
書く *kaku*	k-kons. Verb	書く *kaku*	書かせる *kakaseru*	書いて *kaite*
飲む *nomu*	m-kons. Verb	飲む *nomu*	飲ませる *nomaseru*	飲んで *nonde*

Die dargestellten Formen sollen in folgender Weise in Morphe zerlegt werden:

(ex 69) *Analyse der Präsens-, Kausativ- und Partizipialformen der Verben 'taberu', 'kaku' und 'nomu' in die konstituierenden Morphe:*

Verb	Kategorie	Präsens	Kausativ	Partizip
taberu	vok. Verb	*tabe-ru*	*tabe-sase-ru*	*tabet-e*
kaku	k-kons. Verb	*kak-u*	*kaka-se-ru*	*kait-e*
nomu	m-kons. Verb	*nom-u*	*noma-se-ru*	*nond-e*

Gehen wir davon aus, daß sich

1. das Präsens-Flexiv an die k-basis,
2. das Kausativ-Suffix an die a-Basis und
3. das Flexiv zur Bildung des Partizips an die t-Basis

anschließen, müssen wir folglich zur Bildung dieser Formen die folgenden Zeichen voraussetzen:

(ex 70) *Verbmorpheme:*

a. *'taberu':*

$$\begin{bmatrix} \langle \textit{taberu} \rangle & & \textit{root\_sign} \\ \textbf{phon\_k\_base} & : \text{'tabe'} & \\ \textbf{phon\_a\_base} & : \text{'tabe'} & \\ \textbf{phon\_t\_base} & : \text{'tabet'} & \\ \textbf{lexeme} & : \{\mathsf{taberu}\} & \\ \textbf{morpheme} & : \{\mathsf{taberu}\} & \\ \textbf{morphemes} & : \langle\{\mathsf{taberu}\}\rangle & \\ \textbf{cat} & : \text{vowel\_verb} & \end{bmatrix}$$

b. *'kaku':*

$$\begin{bmatrix} \langle \textit{kaku} \rangle & & \textit{root\_sign} \\ \textbf{phon\_k\_base} & : \text{'kak'} & \\ \textbf{phon\_a\_base} & : \text{'kaka'} & \\ \textbf{phon\_t\_base} & : \text{'kait'} & \\ \textbf{lexeme} & : \{\mathsf{kaku}\} & \\ \textbf{morpheme} & : \{\mathsf{kaku}\} & \\ \textbf{morphemes} & : \langle\{\mathsf{kaku}\}\rangle & \\ \textbf{cat} & : \text{k\_cons\_verb} & \end{bmatrix}$$

c. *'nomu':*

$$\begin{bmatrix} \langle \textit{nomu} \rangle & & \textit{root\_sign} \\ \textbf{phon\_k\_base} & : \text{'nom'} & \\ \textbf{phon\_a\_base} & : \text{'noma'} & \\ \textbf{phon\_t\_base} & : \text{'nond'} & \\ \textbf{lexeme} & : \{\mathsf{nomu}\} & \\ \textbf{morpheme} & : \{\mathsf{nomu}\} & \\ \textbf{morphemes} & : \langle\{\mathsf{nomu}\}\rangle & \\ \textbf{cat} & : \text{m\_cons\_verb} & \end{bmatrix}$$

Wir erhalten damit folgendes Schema von Basisformen:

(ex 71) *k-, a- und t-Basis der Verben 'taberu', 'kaku' und 'nomu':*

Verb	**Kategorie**	**k-base**	**a-base**	**t-base**
taberu	vok. Verb	*tabe*	*tabe*	*tabet*
kaku	k-kons. Verb	*kak*	*kaka*	*kait*
nomu	m-kons. Verb	*nom*	*noma*	*nond*

Als erste Näherung an ein Regelsystem zur Ableitung der Basisformen wollen wir diese in ein allen Basen eines Verbes gemeinsames Präfix und ein die entsprechende Basis kennzeichnendes Suffix zerlegen:

(ex 72) *Analyse von k-, a- und t-Basis bezüglich gemeinsamer Präfixe:*

Verb (Kategorie)	taberu (vok. Verb)	kaku (k-kons. Verb)	nomu (m-kons. Verb)
k-base	ϕ	*k*	*m*
a-base	*tabe* ϕ	*ka* *ka*	*no* *ma*
t-base	*t*	*it*	*nd*

Bei genauerer Betrachtung können wir ein gemeinsames Infix bei der Bildung der k- und a-Basis erkennen:

(ex 73) *Die Basisformen als Resultat der Verknüpfung von Verbpräfix, Infix und Suffix:*

Verb (Kategorie)	taberu (vok. Verb)	kaku (k-kons. Verb)	nomu (m-kons. Verb)
k-base	ϕ ϕ	*k* ϕ	*m* ϕ
a-base	*tabe* ϕ	*ka* *a*	*no* *a*
t-base	*t*	*it*	*nd*

Die in diesem Schema verwandten Bausteine sollen in der folgenden Weise benannt werden:

(ex 74) *Bezeichnungen der Bausteine:*

Basis	**Bausteine**
k-base	stem_cons_infix k_base_infix
a-base	phon_base a_base_infix
t-base	t_base_infix

An dieser Stelle sei auf die Unterschiede in der Ableitung der Basisformen bei vokalischen (z.B. 'taberu') und konsonantischen Verben (z.B. 'kaku', 'nomu') eingegangen. Gemäß unseres Prinzips, bei der Bildung von Formen mit der Konkatenation als einziger Operation auszukommen,[36] wurden die beobachteten Formen so systematisiert, daß ihre Bildung durch ein möglichst einfaches Regelsystem beschrieben werden kann. Wir haben uns dabei nicht auf die Lautgestalt, sondern eine für unsere Zwecke geeignete Umschrift (Nipponsiki-Umschrift) der orthographischen Darstellung gestützt, da unsere Untersuchungen auf die Beschreibung des geschriebenen Japanisch ausgerichtet sind.

Aus diesem Vorgehen folgt, daß durch das Regelsystem zwar die vorkommenden Formen korrekt beschrieben werden sollen, dabei jedoch nicht der

36 Vergleiche Abschnitt 2.1.2, Seite 17.

Anspruch erhoben wird, tatsächlich ablaufende psychologische Prozesse zu beschreiben. Unser Vorgehen ist also deskriptiver (korrekte Beschreibung der Formen) und formaler Art (Beschreibungsökonomie).

Um diesen Vorgaben gemäß die beobachteten Sachverhalte beschreiben zu können, bleibt uns bezüglich der Unterschiedlichkeit der Bildung von Basisformen bei vokalischen und konsonantischen Verben die Wahl zwischen zwei verschiedenen Vorgehensweisen:

1. Wir können für beide Verbkategorien unterschiedliche Formprimitive und damit auch unterschiedliche Regeln zur Bildung der Basisformen annehmen. Im Extremfall könnten wir sogar die Existenz gleicher Basisformen bei vokalischen und konsonantischen Verben in Frage stellen und damit die Bildung der Verbformen durch vollkommen unterschiedliche Regelsysteme erklären.

   (ex 75) *Ableitung von Präsens, Kausativ und Partizip bei Annahme unterschiedlicher Basisformen für vokalische und konsonantische Verben:*

vokalische Verben				
Form	**Schema**		**Beispiel**	
Präsens		*ru*		*ru*
Kausativ	vowel_base	*sase-ru*	*tabe*	*sase-ru*
Partizip		*ta*		*ta*

konsonantische Verben				
Form	**Schema**		**Beispiel**	
Präsens	k_cons_base	*u*	*kak*	*u*
Kausativ	a_cons_base	*se-ru*	*kaka*	*se-ru*
Partizip	t_cons_base	*a*	*kait*	*a*

2. Durch die Annahme des gleichen Regelsystems für die Ableitung der Konjugationsformen bei vokalischen und konsonantischen Verben kommen wir für beide Verbgruppen auch auf das gleiche System von Basisformen. In diesem Fall liegt es daher nahe, auch ein einheitliches Bildungsschema zur Ableitung dieser Basisformen anzunehmen.

In der dargestellten Analyse der Verbbasen wurde der zweite Standpunkt angenommen. Obwohl aus diesem Ansatz eine auf den ersten Blick redundant anmutende Ableitung der Basisformen bei vokalischen Verben resultiert, die durch das mehrfache Suffigieren von ϕ-Formen (vergleiche (ex 76)) gekennzeichnet ist, wirkt sich dieser Ansatz dank der Einheitlichkeit der Behandlung insgesamt vereinfachend aus.

(ex 76) *Die Bildung der Basisformen vokalischer Verben durch wiederholtes Suffigieren von ϕ-Formen:*

$$tabe \begin{cases} \phi \begin{cases} \phi \\ \phi \end{cases} \\ t \end{cases}$$

Es lassen sich verschiedene andere Systeme vorstellen, die sich zwischen den dargestellten Extremen ansiedeln. Ein "richtiges" System kann entsprechend der pragmatisch/formalen Vorgehensweise jedoch nicht existieren. Letztendlich ist es daher ein Abwägen zwischen der Eleganz und der Effizienz der Beschreibung, das die Entscheidung zugunsten des einen oder anderen Systems motiviert. Daß wir uns bei dem hier vorgestellten System im Sinne einer möglichst systematischen und auf Einfachheit ausgerichteten Beschreibung entschieden haben, zeigt die Existens eines 'k_base_infix', das bei allen Verben durch eine ϕ-Form realisiert wird, und auf welches man folglich ebensogut hätte verzichten können.

4.4.1 Morphologische Regeln zur Bestimmung der Infixe

Die in (ex 73) dargestellte Tabelle kann direkt in eine Menge morphologischer Regeln überführt werden. Vor der Aufstellung dieser Regeln müssen wir allerdings noch klären, welche Zeichentypen von ihnen betroffen sind.

Wie in 1.2, Seite 8 beschrieben, können japanische Verben durch rekursives Affigieren von Suffixen mehrfach modifiziert werden. Da bei jeder dieser Modifikationen die Stammformen der schon gebildeten Form benötigt werden, existieren diese nicht nur für das Verbmorphem, sondern ebenso für die schon gebildeten, komplexen Zeichen. Bei der Ableitung der letzteren kann nun auf zweierlei Weise verfahren werden:

1. Wir können die Stammformen für die Suffixe berechnen, und anschließend auf die komplexen Zeichen übertragen;

2. Wird in den Suffixzeichen nur die Basisform spezifiziert, kann diese zur Berechnung der Basisform der komplexen Zeichen benutzt werden. Aus diesen lassen sich dann mittels der gleichen Regeln, die auch bei der Ableitung der Stämme bei Verbmorphemen Anwendung finden, die Stammformen der komplexen Zeichen ermitteln.

Der erste Ansatz resultiert in einem sehr aufwendigen Regelsystem zur Ableitung der Formen komplexer Zeichen; der zweite Weg hingegen ist einfacher, da aus den Tochterzeichen nur eine Form ermittelt werden muß, und wird daher von uns verfolgt.

Zur Realisierung des zweiten Ansatzes müssen die Basisregeln so formuliert werden, daß sie sowohl für atomare als auch für komplexe Zeichen Gültigkeit erhalten. Wir erreichen dieses mittels des Konzeptes 'base_sign' welches sowohl komplexe Zeichen als auch Zeichen für Verbmorpheme unter sich subsumiert:

(ex 77) *Basis-Zeichen:*

$$\begin{tikzcd}[arrows=-] & \text{base\_sign} \arrow[dl] \arrow[dr] & \\ \text{root\_sign} & & \text{composed\_morphological\_sign} \end{tikzcd}$$

Überlegen wir uns nun, wie sich die Bildung des 'stem_cons_infix' mit diesen Mitteln in Regeln fassen läßt. Wir haben die bisher betrachteten konsonantischen Verben in k-kons. Verben und m-kons. Verben unterteilt. Der erste Buchstabe dieser Kategorien wurde dabei so gewählt, daß er dem Stammkonsonanten der entsprechenden Kategorie entspricht. Während das 'stem_cons_infix' für vokalische Verben leer ist, haben m-kons. Verben folglich 'm' und k-kons. Verben 'k' als Stammkonsonanten. Diese Abhängigkeit kann durch Parametrisierung der Regeln mittels der Kategorienbezeichnung dargestellt werden:

(ex 78) *Regeln zur Ableitung des Stammkonsonanten:*

$$a. \quad \begin{bmatrix} \hfill {\scriptstyle \textit{base\_sign}} \\ \textbf{cat} : \text{vowel\_verb} \end{bmatrix} \Longrightarrow [\ \textbf{stem\_cons\_infix} : \phi\ ]$$

$$b. \quad \begin{bmatrix} \hfill {\scriptstyle \textit{base\_sign}} \\ \textbf{cat} : \text{k\_cons\_verb} \end{bmatrix} \Longrightarrow [\ \textbf{stem\_cons\_infix} : \text{`k'}\ ]$$

$$c. \quad \begin{bmatrix} \hfill {\scriptstyle \textit{base\_sign}} \\ \textbf{cat} : \text{m\_cons\_verb} \end{bmatrix} \Longrightarrow [\ \textbf{stem\_cons\_infix} : \text{`m'}\ ]$$

Die Regeln für die übrigen Infixe ergeben sich direkt aus (ex 73) und (ex 74):

(ex 79) *Regeln zur Ableitung des k-base-Infixes:*

$$\begin{bmatrix} \hfill {\scriptstyle \textit{base\_sign}} \\ \textbf{cat} : \text{verb} \end{bmatrix} \Longrightarrow [\ \textbf{k\_base\_infix} : \phi\ ]$$

(ex 80) *Regeln zur Ableitung des a-base-Infixes:*

$$a. \quad \begin{bmatrix} \hfill {\scriptstyle \textit{base\_sign}} \\ \textbf{cat} : \text{vowel\_verb} \end{bmatrix} \Longrightarrow [\ \textbf{k\_base\_infix} : \phi\ ]$$

$$b.\quad \begin{bmatrix} & {\scriptstyle base\_sign} \\ \textbf{cat} & : \text{cons\_verb} \end{bmatrix} \Longrightarrow \begin{bmatrix} \textbf{k\_base\_infix}: \text{`a'} \end{bmatrix}$$

(ex 81) *Regeln zur Ableitung des t-base-Infixes:*

$$a.\quad \begin{bmatrix} & {\scriptstyle base\_sign} \\ \textbf{cat} & : \text{vowel\_verb} \end{bmatrix} \Longrightarrow \begin{bmatrix} \textbf{t\_base\_infix}: \text{`t'} \end{bmatrix}$$

$$b.\quad \begin{bmatrix} & {\scriptstyle base\_sign} \\ \textbf{cat} & : \text{k\_cons\_verb} \end{bmatrix} \Longrightarrow \begin{bmatrix} \textbf{t\_base\_infix}: \text{`it'} \end{bmatrix}$$

$$c.\quad \begin{bmatrix} & {\scriptstyle base\_sign} \\ \textbf{cat} & : \text{m\_cons\_verb} \end{bmatrix} \Longrightarrow \begin{bmatrix} \textbf{t\_base\_infix}: \text{`nd'} \end{bmatrix}$$

Nachdem wir damit die benötigten Infixe hergeleitet haben, müssen wir nun die Regeln formulieren, denen entsprechend Verbbasis und Infixe zu Basisformen verbunden werden können.

Wiederum ergibt sich das Schema direkt aus (ex 73) und (ex 74):

(ex 82) *die Bildung der Stammformen:*

Basis	**Ableitung**
k-Basis	phon_base ∘ stem_cons_infix ∘ k_base_infix
a-Basis	phon_base ∘ stem_cons_infix ∘ a_base_infix
t-Basis	phon_base ∘ t_base_infix

Ähnlich der schon betrachteten Regeln zur Ableitung der Formen in komplexen Zeichen kommen wir zu den folgenden drei Regeln:

(ex 83) *Regeln zur Ableitung der Verbbasen:*

$$a.\quad \begin{bmatrix} {\scriptstyle base\_sign} \end{bmatrix} \Longrightarrow \begin{bmatrix} \textbf{phon\_k\_base} & : \text{P1} \circ \text{P2} \circ \text{P3} \\ \textbf{phon\_base} & : \text{P1} \\ \textbf{stem\_cons\_infix} & : \text{P2} \\ \textbf{k\_base\_infix} & : \text{P3} \end{bmatrix}$$

$$b.\quad \begin{bmatrix} {\scriptstyle base\_sign} \end{bmatrix} \Longrightarrow \begin{bmatrix} \textbf{phon\_a\_base} & : \text{P1} \circ \text{P2} \circ \text{P3} \\ \textbf{phon\_base} & : \text{P1} \\ \textbf{stem\_cons\_infix} & : \text{P2} \\ \textbf{a\_base\_infix} & : \text{P3} \end{bmatrix}$$

$$c.\quad \begin{bmatrix} {\scriptstyle base\_sign} \end{bmatrix} \Longrightarrow \begin{bmatrix} \textbf{phon\_t\_base} & : \text{P1} \circ \text{P2} \\ \textbf{phon\_base} & : \text{P1} \\ \textbf{t\_base\_infix} & : \text{P2} \end{bmatrix}$$

Damit haben wir die Bildung der Basisformen in Abhängigkeit der Kategorie und eines allen Basen gemeinsamen Präfixes in unser morphologisches System integriert. Von nun an können wir folglich bei den lexikalischen Einträgen der

Verben auf die Angabe der einzelnen Basen verzichten, und müssen nur noch das allen Basen gemeinsame "Verbpräfix" — das wir von nun im Unterschied zu den k-, a- bzw. t-Basen einfach als *Verbbasis* bzw. *Basis* bezeichnen wollen — angeben. Am Beispiel der Verben 'taberu', 'kaku' und 'nomu' betrachtet, haben die lexikalischen Einträge damit die folgende Form:

(ex 84) *Lexikalische Einträge für Verben:*

a. *'taberu':*

$$\left[\begin{array}{lll} \langle\textit{taberu}\rangle & & \textit{root\_sign} \\ \textbf{phon\_base} & : & \text{'tabe'} \\ \textbf{lexeme} & : & \{\mathsf{taberu}\} \\ \textbf{morpheme} & : & \{\mathsf{taberu}\} \\ \textbf{morphemes} & : & \langle\{\mathsf{taberu}\}\rangle \\ \textbf{cat} & : & \text{vowel\_verb} \end{array}\right]$$

b. *'kaku':*

$$\left[\begin{array}{lll} \langle\textit{kaku}\rangle & & \textit{root\_sign} \\ \textbf{phon\_base} & : & \text{'ka'} \\ \textbf{lexeme} & : & \{\mathsf{kaku}\} \\ \textbf{morpheme} & : & \{\mathsf{kaku}\} \\ \textbf{morphemes} & : & \langle\{\mathsf{kaku}\}\rangle \\ \textbf{cat} & : & \text{k\_cons\_verb} \end{array}\right]$$

c. *'nomu':*

$$\left[\begin{array}{lll} \langle\textit{nomu}\rangle & & \textit{root\_sign} \\ \textbf{phon\_base} & : & \text{'no'} \\ \textbf{lexeme} & : & \{\mathsf{nomu}\} \\ \textbf{morpheme} & : & \{\mathsf{nomu}\} \\ \textbf{morphemes} & : & \langle\{\mathsf{nomu}\}\rangle \\ \textbf{cat} & : & \text{m\_cons\_verb} \end{array}\right]$$

Es verbleibt, die beschriebenen Regeln und Zeichen in den Formalismus FLEX zu überführen. Diese Umsetzung geschieht in diesem und den folgenden Fällen jedoch durch das direkte Übertragen der dargestellten Regeln und Zeichen in die Syntax von FLEX, und ist damit so einfach, daß wir von nun an auf ihre Darstellung verzichten.

4.5 Unregelmäßige Formen

Die letzten Kapitel hatten die Bildung der regelmäßigen Verbformen zum Inhalt. Wie in allen Sprachen gibt es jedoch auch im Japanischen — vor allem bei sehr oft benutzten Formen — Abweichungen von diesen Regeln. Diese unregelmäßigen Formen bilden das Thema der folgenden Beobachtungen.

Den bisher betrachteten Mechanismen entsprechend, kann es auf zweierlei Weise zu Unregelmäßigkeiten kommen:

1. *Unregelmäßige Verbbasen* — einzelne Verben oder Verbgruppen bilden bestimmte Basisformen anders als es die für sie gültigen, allgemeinen Regeln vermuten lassen;

2. *Unregelmäßige Morphemkombinationen* — manche Verben realisieren bestimmte Morphemkombinationen mittels spezieller Formen, die nicht der Komposition der beteiligten, regelmäßig gebildeten Morphe entsprechen, bzw. sich nicht einmal in einzelne Morphe unterteilen lassen.

Da ihrer verschiedenen Natur entsprechend unterschiedliche Mechanismen zur Behandlung dieser Unregelmäßigkeiten benutzt werden müssen, sollen diese unabhängig voneinander in getrennten Abschnitten beschrieben werden.

4.5.1 Unregelmäßige Stammformen

Im Abschnitt "Verbstämme und Subkategorisierung" (4.3.2, S. 91) haben wir uns die Präsens-, Kausativ- und Partizipialformen verschiedener Verbklassen angesehen. Für die Gruppe der k-kons. Verben sind wir dabei zu dem folgenden Schema gekommen:

(ex 85) *Ableitung der Präsens-, Kausativ- und Partizipialformen bei k-kons. Verben:*

Basis	**k_cons_verb**	**Beispiele**	**Form**
k-base	*k* ϕ	*ka-k - u*	Präsens
a-base	*ka* *a*	*ka-k-a - se - ru*	Kausativ
t-base	*it*	*ka-it - e*	Partizip

Vergleichen wir diese Ableitung mit den entsprechenden Formen des Verbes 行く (*iku*; gehen, fahren), einem weiteren Beispiel der k-kons. Verben:

(ex 86) *Ableitung der Präsens-, Kausativ- und Partizipialformen des Verbes 行く (iku; gehen, fahren):*

Beispiele	**Form**
i-k - u	Präsens
i-k-a - se - ru	Kausativ
*i-**tt** - e*	Partizip

Offensichtlich entsprechen k- und a-Stamm dieses Verbs den angegebenen Regeln zur Ableitung der Stämme von k-kons. Verben; der t-Stamm hingegen müßte bei regelmäßiger Bildung die Form ∗'iita' haben, wird aber tatsächlich durch die Form 'itta' realisiert. Wir haben es also mit einer unregelmäßigen Form zu tun.

Zur Behandlung solcher unregelmäßiger Formen haben wir verschiedene Möglichkeiten:

1. Durch Schaffung einer neuen Kategorie und Differenzierung der schon aufgestellten Regeln können wir die beobachtete Abweichung als neue "regelmäßige" Bildung in unser System integrieren;

2. Durch Spezifizierung der unregelmäßigen Basisform im Lexikoneintrag des entsprechenden Verbes können wir die regelmäßig gebildete Form "überschreiben".

Bei Abweichungen, die bei mehreren Verben auftreten, wird man den ersten Weg wählen; hat man es jedoch mit einer wirklichen Ausnahme zu tun — d.h. mit dem einmaligen Auftreten einer unregelmäßigen Form bei nur einem einzigen Verb — scheint die Lösung des Problems durch Lexikoneintrag angemessener.

In unserem Fall ist die Abweichung einmalig, und wir beschreiben sie deshalb durch die Aufnahme der abweichenden t-Basis in den Lexikoneintrag:

(ex 87) *Behandlung der unregelmäßigen t-Basis des Verbes 'iku' durch Eintrag im Lexikon: 'iku':*

```
<iku>                      root_sign
  phon_base    : 'i'
  phon_t_base  : 'itt'
  lexeme       : {iku}
  morpheme     : {iku}
  morphemes    : <{iku}>
  cat          : k_cons_verb
```

Betrachten wir ein anderes Beispiel:

(ex 88) *Die regelmäßige, aber falsche Ableitung der Präsens- und Kausativform des Verbes 買う (kau; kaufen):*

Form	書く (schreiben) *kaku*	買う (kaufen) *kau*
Präsens	*kaku*	*kau*
Kausativ	*kakaseru*	*kawaseru*

Zur Ableitung der Formen wurde im Falle der Präsensform die k-Basis verwendet, und im Falle der Kausativform die a-Basis. Bei regelmäßiger Bildung dieser Basen müßten wir folglich zu folgenden Formen kommen:

(ex 89) *Ableitung der Präsens-, Kausativ- und Partizipialformen des Verbes 買う (kau; kaufen):*

Form	買う (kaufen) *kau*
Präsens	**kawu*
Kausativ	*kawaseru*

Die Präsensform ‘kau’ dieses Verbes widerspricht jedoch der regelmäßigen Ableitung *‘kawu’ und ist somit unregelmäßig.

Im Gegensatz zur Abweichung der t-Basis von 行く (*iku*) haben wir es in diesem Fall jedoch nicht mit einer einmalig auftretenden Unregelmäßigkeit zu tun, sondern mit der Abweichung einer ganzen Gruppe von Verben bezüglich der gegebenen Regeln zur Bildung des k-Stammes. In diesem Fall werden wir folglich durch Bildung einer eigenen Unterklasse (den w-konsonantischen Verben) und eines neuen Regelsystems den ersteren der vorgeschlagenen Wege zur Behandlung von Unregelmäßigkeiten wählen. Von dem allgemeinen Schema zur Bildung der k-Basis abweichend (vergleiche (ex 83*a*), S. 108), kommen wir damit zu folgender Regel:

(ex 90) *Regel zur Ableitung der k-Basis bei w-kons. Verben:*

$$\begin{bmatrix} & \textit{base\_sign} \\ \textbf{cat} & : \text{w\_cons\_verb} \end{bmatrix} \Longrightarrow \begin{bmatrix} \textbf{phon\_k\_base} & : \text{P1} \circ \text{P2} \\ \textbf{phon\_base} & : \text{P1} \\ \textbf{k\_base\_infix} & : \text{P2} \end{bmatrix}$$

Beide Strategien zur Behandlung von unregelmäßigen Formen — d.h. a) die Aufnahme der Form in das Lexikon und b) die Differenzierung des bisherigen Systems von Regularitäten — führen zu Widersprüchen zwischen den abweichenden und den regelmäßig sich bildenden Formen. Bei dem Verb 行く (*iku*) beispielsweise kommen wir zu folgender Situation:

1. die t-Basis muß eindeutig bestimmt sein (die Rolle ‘phon_t_base’ ist funktional);

2. Die Regeln zur Bildung der t-Basis führen im Falle von ‘iku’ zu der Form ‘iit’;

3. Im Lexikoneintrag des Verbes ‘iku’ wurde ‘itt’ als t-Basis spezifiziert.

Da alle drei Punkte Gültigkeit haben, erhalten wir einen Widerspruch. Mit dem System FLEX haben wir jedoch die Möglichkeit, monotone Regeln durch Defaultregeln zu ergänzen.[37] Ändern wir also das im letzten Kapitel vorgestellte Regelsystem derart, daß es — wie es ja auch der grammatischen Intuition entspricht — nicht Anspruch auf absolute Gültigkeit erhebt, sondern nur den Standardfall beschreibt, formulieren wir unsere morphologischen Gesetze also als *Defaults*, können wir den auftretenden Widerspruch vermeiden. Durch höher bewertete (bzw. monotone) Regeln für die Unregelmäßigkeiten können wir in diesem Fall die Defaultformen überschreiben.

Für die zuerst dargestellte Unregelmäßigkeit kommen wir damit zu folgender Regel:

37 vergleiche die entsprechenden Abschnitte aus 3: ”Defaults”(3.2.2, S. 60) und ”Defaultfolgerung”(3.2.3, S. 61).

(ex 91) *Die Defaultregel zur Herleitung der t-Basis und die Behandlung der Abweichung bei dem Verb 'iku' durch Lexikoneintrag:*

a. $$\begin{bmatrix} \textit{base\_sign} \end{bmatrix} \overset{10}{\leadsto} \begin{bmatrix} \textbf{phon\_t\_base} & : \mathrm{P1} \circ \mathrm{P2} \\ \textbf{phon\_base} & : \mathrm{P1} \\ \textbf{t\_base\_infix} & : \mathrm{P2} \end{bmatrix}$$

b. $$\begin{bmatrix} <\textit{\textbf{iku}}> & & \textit{root\_sign} \\ \textbf{phon\_base} & : \text{'i'} & \\ \textbf{phon\_t\_base} & : \text{'itt'} & \\ \textbf{lexeme} & : \{\mathsf{iku}\} & \\ \textbf{morpheme} & : \{\mathsf{iku}\} & \\ \textbf{morphemes} & : <\{\mathsf{iku}\}> & \\ \textbf{cat} & : \text{k\_cons\_verb} & \end{bmatrix}$$

Das Verb 'kaku' beispielsweise bildet seine t-Basis mittels Regel (ex 91*a*). Da es keine Alternative zu der sich ergebenden Form gibt, erhalten wir die t-Basis 'kait'. Im Falle des Verbes 'iku' hingegen kommen wir zu den Formen 'iit' und 'itt'. Die erste wurde mittels der Defaultregel (ex 91*a*) gebildet, die zweite hingegen mit monotonen Mitteln, d.h. durch direkte Angabe des Wertes im Lexikoneintrag. Der monotone Füller überschreibt den durch das Default abgeleiteten, und wir kommen zur Form 'itt'.

Das zweite Beispiel soll durch das folgende Regelsystem beschrieben werden:

(ex 92) *Die Defaultregel zur Herleitung der k-Basis und die Behandlung der Abweichung bei w-kons. Verben durch eine höher bewertete Defaultregel:*

a. $$\begin{bmatrix} \textit{base\_sign} \end{bmatrix} \overset{10}{\leadsto} \begin{bmatrix} \textbf{phon\_k\_base} & : \mathrm{P1} \circ \mathrm{P2} \circ \mathrm{P3} \\ \textbf{phon\_base} & : \mathrm{P1} \\ \textbf{stem\_cons\_infix} & : \mathrm{P2} \\ \textbf{k\_base\_infix} & : \mathrm{P3} \end{bmatrix}$$

b. $$\begin{bmatrix} & \textit{base\_sign} \\ \textbf{cat} & : \text{w\_cons\_verb} \end{bmatrix} \overset{100}{\leadsto} \begin{bmatrix} \textbf{phon\_k\_base} & : \mathrm{P1} \circ \mathrm{P2} \\ \textbf{phon\_base} & : \mathrm{P1} \\ \textbf{k\_base\_infix} & : \mathrm{P2} \end{bmatrix}$$

'kaku' ist ein 'verb', aber kein 'w_cons_verb'. Die k-Basis kann also einzig durch Regel (ex 92*a*) abgeleitet werden und führt damit auf die Form 'kak'. Für das w-kons. Verb 'kau' hingegen sind beide Regeln anwendbar. Wir kommen damit zu den sich widersprechenden Formen 'ka' und 'kaw'. Die erstere jedoch ist hinsichtlich der Defaults höher bewertet und überschreibt folglich die zweite. Wir erhalten also die Form 'ka' als gültige k-Basis des Verbes 'kau'.

4.5.2 Unregelmäßige Morphemkompositionen

Nicht nur bei der Ableitung von Morphen bzw. Basisformen gibt es Unregelmäßigkeiten, sondern auch bei der Verbindung bestimmter Morpheme zu komplexen Formen. Solche als *synthetisch* bezeichneten unregelmäßigen Formen entsprechen nicht der Komposition einzelner Morphe, sondern bilden Einheiten, die sich nicht sinnvoll in Segmente zerlegen lassen. Betrachten wir dazu die folgenden Formen der r-kons. Verben 乗る (*noru*; einsteigen, fahren) und ある (*aru*; sein):

(ex 93) *Präsens und negiertes Präsens der Verben 乗る (noru; einsteigen, fahren) und ある (aru; sein):*

Verb	Präsens	Negation
乗る *noru*	*noru*	*noranai*
ある *aru*	*aru*	*nai*

Die negierte Präsensform entsteht im regelmäßigen Fall durch die Verkettung der Morpheme:

(ex 94) {verbmorphem} ∘ {negation} ∘ {flexiv}

Im Falle des Verbes 'noru' können wir die negierte Präsensform diesem Schema entsprechend in Morphe segmentieren; das Verb 'aru' hingegen realisiert Verb- und Negationsmorphem in der synthetischen Form 'na' und nicht — wie nach den Regeln zur Ableitung der a-Basis, an die sich das Negationsmorphem im regelmäßigen Fall anschließt, zu erwarten — durch *'ara-na-i':

(ex 95) *Die synthetische Realisierung der Morphemkette* <{Verbmorphem}, {Negation}> *beim Verb ある (aru; sein) im Vergleich zur regelmäßigen Bildung der Negation beim Verb 乗る (noru; einsteigen, fahren):*

Verb	**Morphe**		
	Verbmorph	Negation	Flexiv
乗る *noru*	*nora* -	*na* -	*i*
ある *aru*	*na*	-	*i*

Da wir es bei dieser Form mit einer einmaligen Abweichung zu tun haben, müssen wir die Regularitäten zur Ableitung der regelmäßigen, aber hinsichtlich des Sprachgebrauches für das Verb 'aru' falschen Form als Default umformulieren, und dieses durch den Lexikoneintrag der negierten Präsensform von 'aru' überschreiben.

Sehen wir uns dazu die regelmäßige Ableitung des Zeichens für $\langle$\{noru\}, \{negation\}$\rangle$ an:

Ausgehend von den Zeichen für die Morpheme \{noru\} und \{negation\}

(ex 96) *a.* *Lexikoneintrag des Verbes 乗る (noru; einsteigen, fahren):*

$$\left[\begin{array}{lll} \langle \textit{noru} \rangle & & \textit{root\_sign} \\ \textbf{phon\_base} & : & \text{`no'} \\ \textbf{lexeme} & : & \{\text{noru}\} \\ \textbf{morpheme} & : & \{\text{noru}\} \\ \textbf{morphemes} & : & \langle\{\text{noru}\}\rangle \\ \textbf{cat} & : & \text{r\_cons\_verb} \end{array}\right]$$

b. *Lexikoneintrag des Negationssuffixes:*

$$\left[\begin{array}{lll} \langle \textit{negation} \rangle & & \textit{suffix\_sign} \\ \textbf{phon} & : & \text{`na'} \\ \textbf{morpheme} & : & \{\text{negation}\} \\ \textbf{morphemes} & : & \langle\{\text{negation}\}\rangle \\ \textbf{cat} & : & \text{adjective}^{38} \\ \textbf{subcat\_cat} & : & \text{verb} \\ \textbf{subcat\_base} & : & \text{a\_base} \end{array}\right]$$

kommen wir mittels der Regeln zur Ableitung der Stammformen und der Regeln für komplexe Zeichen zu dem folgenden Zeichen für die Morphemfolge $\langle$\{noru\}, \{negation\}$\rangle$:

(ex 97)

$$\left[\begin{array}{lll} \langle \textit{noru, negation} \rangle & & \textit{composed\_morphological\_sign} \\ \textbf{phon} & : & \text{`norana'} \\ \textbf{lexeme} & : & \{\text{noru}\} \\ \textbf{morphemes} & : & \langle\{\text{noru}\}, \{\text{negation}\}\rangle \\ \textbf{cat} & : & \text{adjective} \\ \textbf{daughter\_base} & : & \left[\begin{array}{lll} \langle \textit{noru} \rangle & & \textit{root\_sign} \\ \textbf{phon\_base} & : & \text{`no'} \\ \textbf{phon\_a\_base} & : & \text{`nora'} \\ \textbf{lexeme} & : & \{\text{noru}\} \\ \textbf{morpheme} & : & \{\text{noru}\} \\ \textbf{morphemes} & : & \langle\{\text{noru}\}\rangle \\ \textbf{cat} & : & \text{r\_cons\_verb} \end{array}\right] \\ \textbf{daughter\_suffix} & : & \left[\begin{array}{lll} \langle \textit{negation} \rangle & & \textit{suffix\_sign} \\ \textbf{phon} & : & \text{`na'} \\ \textbf{morpheme} & : & \{\text{negation}\} \\ \textbf{morphemes} & : & \langle\{\text{negation}\}\rangle \\ \textbf{cat} & : & \text{adjective} \\ \textbf{subcat\_cat} & : & \text{verb} \\ \textbf{subcat\_base} & : & \text{a\_base} \end{array}\right] \end{array}\right]$$

Wollen wir dieses Zeichen für das Verb 'aru' direkt im Lexikon spezifizieren, müssen wir nur die für die weiteren morphologischen Prozesse relevanten Attribute angeben. Wir erhalten damit folgendes Zeichen:

(ex 98)

$$\left[\begin{array}{lll} \langle \textit{aru, negation} \rangle & & \textit{composed\_morphological\_sign} \\ \textbf{phon} & : & \text{`na'} \\ \textbf{lexeme} & : & \{\text{aru}\} \\ \textbf{morphemes} & : & \langle\{\text{aru}\}, \{\text{negation}\}\rangle \\ \textbf{cat} & : & \text{adjective} \end{array}\right]$$

38 Adjektive und Verben verhalten sich in der Morphologie und Syntax des Japanischen sehr ähnlich, und können — wie hier gezeigt — ineinander überführt werden: Wie das Beispiel veranschaulicht, verhält sich die Negation von Verben morphologisch wie ein Adjektiv.

Wie für die Ausnahmen bezüglich der Basisformen schon gezeigt, müssen wir den sich ergebenden Widerspruch zwischen direkt im Lexikon spezifizierten und regelmäßig abgeleiteten Formen durch Deklaration der morphologischen Regeln als Defaults begegnen. Stellen wir uns die in (ex 67), Seite 101 angeführten Regeln für komplexe Zeichen also als Defaults vor. Wir erhalten damit zwei unterschiedliche Ableitungen für die Folge
<{aru}, {negation}, {flexiv}>:

1. die erste Ableitung führt auf das unregelmäßige, im Lexikon spezifizierte Zeichen mit dem Phon-Wert 'na';

2. die zweite Ableitung hingegen ergibt das falsche, aber den allgemeinen Defaultegeln entsprechende Zeichen mit dem Phon-Wert 'aranai'.

Nehmen wir eine zusätzliche Regel an, die das Vorhandensein zweier verschiedener Zeichen für eine Morphemfolge verbietet, kommen wir damit zu einem Widerspruch. Das erste Zeichen jedoch wurde im Lexikon spezifiziert, entstammt also einem "monotonen" Ableitungsprozeß; das zweite Zeichen hingegen folgt aus der Anwendung der Defaultregeln, hat also den Status einer "Standartlösung" für den Fall des Ausbleibens anderer höher bewerteter bzw. monotoner Ableitungen. Der entstandene Konflikt führt folglich zur Wahl des vorgegebenen Lexikoneintrages und damit zur korrekten Form.

Bei der Umsetzung dieser Vorgehensweise in FLEX stoßen wir jedoch auf das schon aus den letzten Abschnitten bekannte Problem:

Das Zusammenspiel der Attribute in komplexen Zeichen läßt sich nicht durch FLEX-Regeln beschreiben, sondern erfordert eine Erweiterung des Formalismus durch compute-Prädikate. Diese simulieren die grammatischen Constraints durch folgende Strategie:

Gegeben sei ein Zeichen des Typs 'composed_morphological_sign', dessen Attribut 'morphemes' die zu realisierenden Morpheme enthält. Zur Berechnung der Tochterzeichen werden diese Morpheme mittels der compute-Prädikate 'base(morphemes)' und 'suffix(morphemes)' auf Basis- und Suffix-Tochter verteilt. Die so spezifizierten Töchter stoßen wiederum grammatische Regeln an, die auf gleiche Weise, d.h. rekursiv, zur Berechnung ihrer Attribute führen. Nur im Falle einelementiger Morphemlisten werden Zeichen nicht in dieser Weise berechnet, sondern aus dem Lexikon ermittelt und in die entsprechenden Attribute des Mutterzeichens eingetragen. Für die Berechnung der negierten Präsensform des Verbes 'aru' kommen wir damit zu folgender Abfolge teilspezifizierter Zeichen:

(ex 99) *a.* $\left[\begin{array}{ll} & \textit{composed\_morphological\_sign} \\ \textbf{morphemes} & : \ <\{\mathsf{aru}\},\ \{\mathsf{negation}\},\ \{\mathsf{flexiv}\}> \end{array}\right]$

$$
b.\quad \left[\begin{array}{lll}
 & & \textit{composed\_morphological\_sign} \\
\textbf{morphemes} & : & \langle\{\mathsf{aru}\}, \{\mathsf{negation}\}, \{\mathsf{flexiv}\}\rangle \\
\textbf{daughter\_base} & : & \left[\begin{array}{lll} & & \textit{composed\_morphological\_sign} \\ \textbf{morphemes} & : & \langle\{\mathsf{aru}\}, \{\mathsf{negation}\}\rangle \end{array}\right] \\
\textbf{daughter\_suffix} & : & \left[\begin{array}{lll} & & \textit{suffix\_sign} \\ \textbf{morphemes} & : & \langle\{\mathsf{flexiv}\}\rangle \end{array}\right]
\end{array}\right]
$$

$$
c.\quad \left[\begin{array}{lll}
 & & \textit{composed\_morphological\_sign} \\
\textbf{morphemes} & : & \langle\{\mathsf{aru}\}, \{\mathsf{negation}\}, \{\mathsf{flexiv}\}\rangle \\
\textbf{daughter\_base} & : & \left[\begin{array}{lll} & & \textit{composed\_morphological\_sign} \\ \textbf{morphemes} & : & \langle\{\mathsf{aru}\}, \{\mathsf{negation}\}\rangle \\ \textbf{daughter\_base} & : & \left[\begin{array}{lll} & & \textit{root\_sign} \\ \textbf{morphemes} & : & \langle\{\mathsf{aru}\}\rangle \end{array}\right] \\ \textbf{daughter\_suffix} & : & \left[\begin{array}{lll} & & \textit{suffix\_sign} \\ \textbf{morphemes} & : & \langle\{\mathsf{negation}\}\rangle \end{array}\right] \end{array}\right] \\
\textbf{daughter\_suffix} & : & \left[\begin{array}{lll} & & \textit{suffix\_sign} \\ \textbf{morphemes} & : & \langle\{\mathsf{flexiv}\}\rangle \end{array}\right]
\end{array}\right]
$$

Erst im Stadium (ex 99*c*) wird für Zeichen des Types 'suffix_sign', 'flexive_sign' bzw. 'root_sign' — d.h. bei Zeichen, die für nur ein Morphem stehen — nach den lexikalischen Einträgen gesucht. Für das Morphem {aru}, heißt das, daß die Form nicht gemäß des Eintrages für die synthetische Realisierung der Morpheme <{aru}, {negation}> gebildet wird, sondern entsprechend der "regelmäßigen" Ableitung — also der im Sinne des japanischen Sprachgebrauches falschen Art und Weise.

Wollen wir das Zeichen richtig, d.h. dem unregelmäßigen Lexikoneintrag entsprechend berechnen, müssen wir das Prädikat 'base(morphemes)' so modifizieren, daß schon beim Übergang von (ex 99*a*) nach (ex 99*b*) das im Lexikon spezifizierte, synthetische Zeichen Verwendung findet, und nicht durch Bildung eines neuen Zeichens vom Typ 'composed_morphological_sign' der "regelmäßige" Ableitungsprozeß erneut angestoßen wird.

Das Prädikat 'base(morphemes)' muß folglich so umgeschrieben werden, daß es zuerst nach eventuell im Lexikon schon spezifizierten, "unregelmäßigen" Zeichen sucht, und nur bei deren Nichtvorhandensein den regelmäßigen Ableitungsprozeß veranlaßt.

Wie schon in Abschnitt (4.2.3, S. 78) und (4.2.4, S. 84) müssen wir zur Realisierung unserer Beschreibung in FLEX Zugeständnisse in bezug auf Deklarativität und Eleganz der Modellierung machen. Bezüglich der unserer Realisierung zugrundeliegenden, auf Vererbungshierarchien und Defaultfolgerung basierenden mathematischen Spezifizierung der morphologischen Sachverhalte jedoch bleibt die Deklarativität erhalten.

Ein wesentliches Ergebnis dieses Kapitels ist damit neben der Angabe der morphologischen Spezifikation der Nachweis der Existenz eines Systems, welches diese Spezifikation ohne Verlust der Deklarativität umzusetzen gestattet.

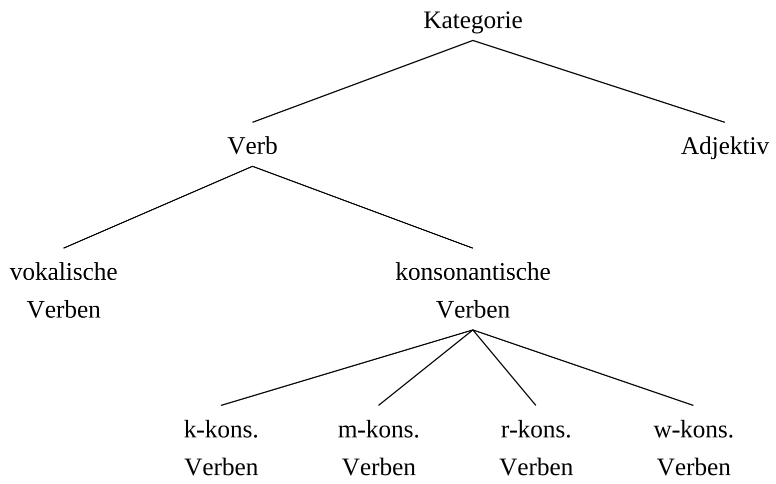
4.6 Die Organisation des Lexikons

Bisher haben wir alle Merkmale morphologischer Zeichen, soweit sie sich nicht aus den Regeln für Basisformen bzw. komplexe Zeichen ergeben haben, explizit im Lexikon angegeben. Bei der Spezifizierung von Zeichentypen wie beispielsweise den 'root_sign', 'suffix_sign' oder 'flexive_sign', bei der Verwendung bestimmter, nur bei manchen Zeichentypen auftretender Merkmale als auch bei der Spezifizierung der Kategorienhierarchie und der Subsumption, die bei der Aufstellung von Regeln ausgenutzt wurde, haben wir dabei schon Mechanismen von Vererbungshierarchien ausgenutzt, ohne explizit auf diese hinzuweisen. In diesem Abschnitt wollen wir uns nun genauer mit diesen Mechanismen auseinandersetzen, und insbesondere versuchen, mit ihrer Hilfe die Lexikoneinträge soweit als möglich aus allgemeinen Prinzipien vorhersagbar zu machen, und damit zu vereinfachen.

Konstatieren wir zuerst einige der schon gemachten Annahmen bezüglich der Hierarchie von Typen:

4.6.1 Die Hierarchie der Verbkategorien

Das einfachste Beispiel einer Konzepthierarchie haben wir in Abschnitt (4.3.3, S. 95) bereits eingeführt: Durch das Anordnen der Verben entsprechend sich hierarchisch zueinander verhaltender Kategorien konnten wir morphologische Regeln so ausdrücken, daß sie die *allgemeinste* Klasse von Verben, für die sie Gültigkeit haben, betreffen. Mittels verschieden bewerteter Defaultregeln waren wir — wie in 4.5.1 bereits geschildert — darüber hinaus in der Lage, Ausnahmen in der Bildung von Formen bezüglich bestimmter Untergruppen allgemeinerer Kategorien zu beschreiben. Zur Erinnerung sei die dabei verwendete Hierarchie, um inzwischen hinzugekommene Kategorien erweitert, an dieser Stelle nocheinmal angeführt:



(fig 12) *Die Hierarchie verbaler Kategorien*

Für eine vollständigere Beschreibung der Formen japanischer Verben kann diese Hierarchie leicht ausgebaut werden.

4.6.2 Die Typenhierarchie morphologischer Zeichen

Bis zum jetzigen Zeitpunkt sind wir fünf unterschiedlichen Zeichentypen begegnet:

1. *Wurzel-Zeichen* ('root_sign') — zur Angabe der Verbmorphe;
2. *Suffix-Zeichen* ('suffix_sign') — für alle Morpheme, die sich direkt oder mittelbar an Verbmorpheme anschließen können;
3. *Flexiv-Zeichen* ('flexive_sign') — zur Darstellung spezieller Suffixe, die den morphologischen Bildungsprozeß abschließen;
4. *komplexe morphologische Zeichen* ('composed_morphological_sign') — zur Darstellung der bei der Morphemkomposition entstehenden komplexen Formen;
5. *Basis-Zeichen* ('base_sign') — dieser Typ subsumiert komplexe Zeichen und Wurzelzeichen, um sie gemeinsam durch die Regeln zur Ableitung der Basisformen ansprechen zu können.

Den verschiedenen Zeichenarten haben wir bestimmte Eigenschaften in Form von Attributen zugeschrieben:

1. Die Rolle 'phon', die neben den *linguistischen Eigenschaften* einen der zwei wesentlichen Aspekte linguistischer Zeichen — nämlich den der *Form* — beschreibt, findet sich bei allen Zeichenarten. Die *linguistischen Eigenschaften* werden durch die übrigen Attribute beschrieben.
2. Das Attribut 'cat' findet sich ebenfalls bei allen morphologischen Zeichen. Es ist darüber hinaus auch für die Syntax elementar, die allerdings als Resultat der dargestellten Prozesse in dieser Arbeit nur am Rande behandelt wird.
3. Die Liste der Morpheme 'morphemes' ist nur bei komplexen Zeichen wirklich motiviert. Zur Vereinfachung der morphologischen Regeln wurde sie jedoch allen morphologischen Zeichen zugewiesen.
4. Die Rolle 'morpheme', die bei atomaren morphologischen Zeichen auf das entsprechende Morphem verweist, finden wir nicht bei komplexen Zeichen, wohl aber bei solchen für Wurzeln, Suffixe und Flexive.
5. Die weiteren Rollen lassen sich aus der Tabelle (ex 100) entnehmen, und wurden schon eingeführt.

Die Auflistung aller Attribute im Zusammenhang mit der Zeichenart, der sie angehören, führt zu folgender Tabelle:

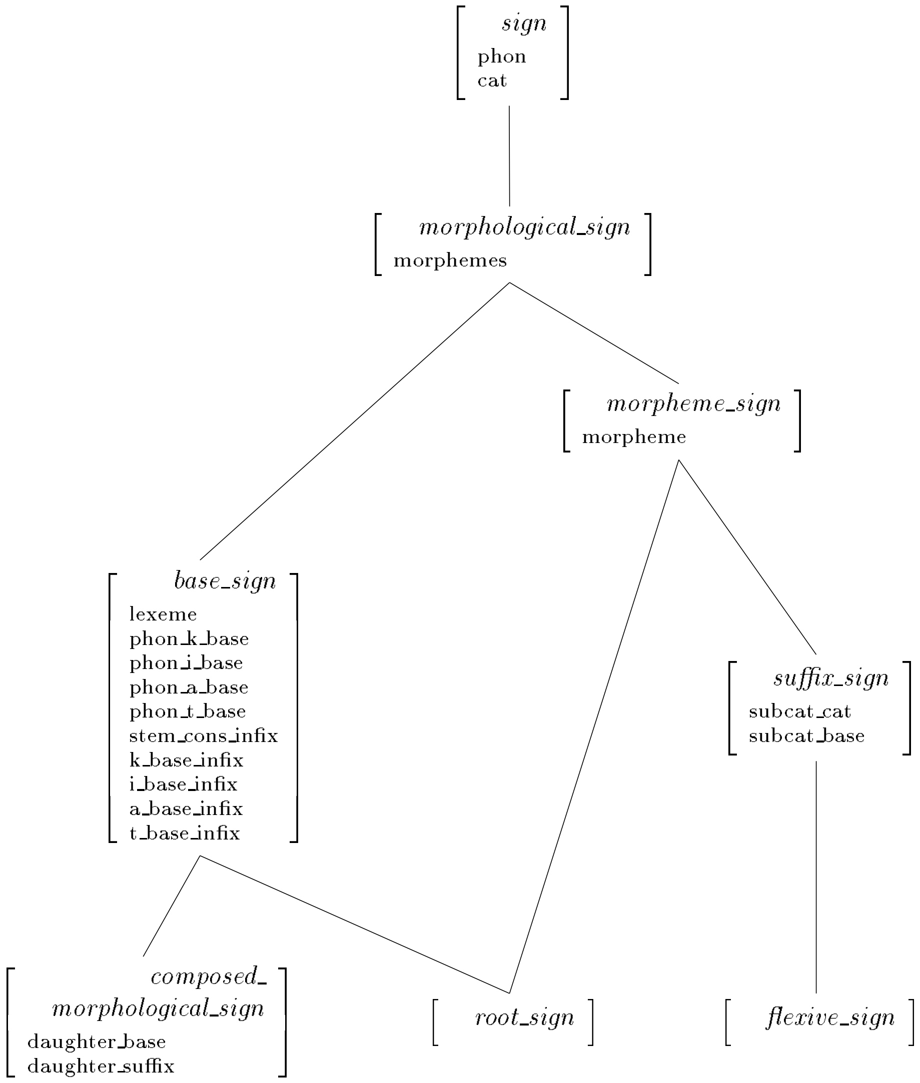
(ex 100) *Typen morphologischer Zeichen und ihre Attribute:*

	suffix_ sign	flexive_ sign	base_ sign	root_ sign	composed_ morph._sign
phon	•	•	•	•	•
cat	•	•	•	•	•
morphemes	•	•	•	•	•
morpheme	•	•	○	•	○
subcat_cat	•	•	○	○	○
subcat_base	•	•	○	○	○
lexeme	○	○	•	•	•
phon_k_base	○	○	•	•	•
phon_i_base	○	○	•	•	•
phon_a_base	○	○	•	•	•
phon_t_base	○	○	•	•	•
stem_cons_infix	○	○	•	•	•
k_base_infix	○	○	•	•	•
i_base_infix	○	○	•	•	•
a_base_infix	○	○	•	•	•
t_base_infix	○	○	•	•	•
daughter_base	○	○	○	○	•
daughter_suffix	○	○	○	○	•

Attribute stehen für *Eigenschaften* bestimmter Zeichen. Zeichen wiederum werden als Instanzen bestimmter *Zeichentypen* oder — wie solche in der Terminologie der Beschreibungslogiken genannt werden — *Konzepte* eingeführt. Es liegt also nahe, Konzepte und Eigenschaften miteinander zu identifizieren oder anders gesagt, Konzepte mittels bestimmter notwendiger und hinreichender Attribute zu definieren. Definieren wir aber allgemeine Zeichen beispielsweise durch das Auftreten der Attribute 'phon' und 'cat', Suffix-Zeichen hingegen durch das *zusätzliche* Auftreten der Rollen 'subcat_cat' und 'subcat_base', wie dieses die Tabelle (ex 100) nahelegt, ergibt sich das Subsumptionsverhältnis zwischen diesen beiden Zeichentypen. Wollen wir also Redundanzen in der Beschreibung von Zeichen vermeiden, können wir diese Subsumption umgekehrt ausnutzen, und speziellere Zeichen explizit als Spezialisierung allgemeinerer kennzeichnen. Wir müssen damit nur noch die diese Spezialisierung kennzeichnenden zusätzlichen Eigenschaften angeben.

Indem wir die Tabelle (ex 100) nach diesem Kriterium auswerten, kommen wir zu einer *Hierarchie* von Zeichentypen, innerhalb derer sich die Rollen von oben nach unten vererben:

(ex 101) *Die Hierarchie der Zeichen:*



In den letzten Abschnitten haben wir diese Konzepthierarchie bereits genutzt, um — zusammen mit der Beschreibung bestimmter Merkmalsausprägungen — die Gültigkeitsbereiche der morphologischen Regeln zu kennzeichnen.

4.6.3 Die Hierarchie morphologischer Zeichen

Morphologische Zeichen werden als Instanzen der entsprechenden Zeichentypen dargestellt, hängen also gewissermaßen "am unteren Ende" der im letzten Abschnitt beschriebenen Hierarchie. Zur Vereinfachung der Darstellung lexikalischer Einträge können wir natürlich beliebige weitere Untertypen der schon dargestellten Zeichentypen einführen. Beispielsweise können wir mit dem Zeichentyp

(ex 26)
$$\begin{array}{lll} \text{k\_cons\_verb\_root\_sign} & \textbf{:<} & \text{root\_sign} \\ & \textbf{and} & \text{cat : k\_cons\_verb.} \end{array}$$

bei der Beschreibung entsprechender Verbwurzeln Zeichentyp und Verbkategorie zusammenfassen.

Während diese Vereinfachung in unserer Modellierung nicht vorgenommen werden soll, da sie den Beschreibungsaufwand nicht wesentlich vereinfacht, die Lesbarkeit der Lexikoneinträge jedoch gefährdet, lassen sich verschiedene Einträge für gleiche Verben mit diesem Mittel sinnvoll zusammenfassen.

Erinnern wir uns an die Einträge des Verbes 'aru':

(ex 102) *Lexikoneiträge des Verbes 'aru':*

$$a. \quad \left[\begin{array}{lll} <\textbf{\textit{aru}}> & & \textit{root\_sign} \\ \textbf{phon\_base} & : & \text{'a'} \\ \textbf{lexeme} & : & \{\mathsf{aru}\} \\ \textbf{morpheme} & : & \{\mathsf{aru}\} \\ \textbf{morphemes} & : & <\{\mathsf{aru}\}> \\ \textbf{cat} & : & \text{r\_cons\_verb} \end{array}\right]$$

$$b. \quad \left[\begin{array}{lll} <\textbf{\textit{aru, negation}}> & & \textit{composed\_morphological\_sign} \\ \textbf{phon\_base} & : & \text{'na'} \\ \textbf{lexeme} & : & \{\mathsf{aru}\} \\ \textbf{morphemes} & : & <\{\mathsf{aru}\}, \{\mathsf{negation}\}> \\ \textbf{cat} & : & \text{adjective} \end{array}\right]$$

Diese Zeichen lassen sich unter Herausfilterung der gemeinsamen Merkmale sinnvoller in Form der folgenden Hierarchie darstellen:

(ex 103) *Lexikoneiträge des Verbes 'aru' als Konzepthierarchie:*

$$\left[\begin{array}{lll} & & \textit{base\_sign} \\ \textbf{lexeme} & : & \{\mathsf{aru}\} \end{array}\right]$$

$$\left[\begin{array}{lll} <\textbf{\textit{aru}}> & & \textit{root\_sign} \\ \textbf{phon\_base} & : & \text{'a'} \\ \textbf{morpheme} & : & \{\mathsf{aru}\} \\ \textbf{morphemes} & : & <\{\mathsf{aru}\}> \\ \textbf{cat} & : & \text{r\_cons\_verb} \end{array}\right] \qquad \left[\begin{array}{lll} <\textbf{\textit{aru, negation}}> & & \textit{composed\_morphological\_sign} \\ \textbf{phon\_base} & : & \text{'na'} \\ \textbf{morphemes} & : & <\{\mathsf{aru}\}, \{\mathsf{negation}\}> \\ \textbf{cat} & : & \text{adjective} \end{array}\right]$$

Entsprechend können wir die verschiedenen Realisierungen von Suffixen und Flexiven durch diese Darstellung vereinfachen, was für das Kausativsuffix beispielhaft vorgeführt werden soll:

(ex 104) *Lexikoneiträge des Kausativ-Suffixes als Konzepthierarchie:*

suffix_sign	
morpheme	: {causative}
morphemes	: <{causative}>
cat	: vowel_verb
subcat_base	: a_base

<causative, vowel_verb> *suffix_sign*	
subcat_cat	: vowel_verb
phon	: ‘sase’

<causative, cons_verb> *suffix_sign*	
subcat_cat	: cons_verb
phon	: ‘se’

4.6.4 Lexikalische Regeln

Zu guter letzt können wir einige der bisher explizit angeführten Merkmalswerte mittels lexikalischer Regeln erschließen. Sehen wir uns dazu die Zeichen für das Verb ‘kaku’ und das Präsensflexiv an:

(ex 105) a. *Der Lexikalische Eintrag des Verbes ‘kaku’:*

<kaku> *root_sign*	
phon_base	: ‘ka’
lexeme	: {kaku}
morpheme	: {kaku}
morphemes	: <{kaku}>
cat	: k_cons_verb

b. *Der Lexikalische Eintrag des Präsensflexives für vokalische und konsonantische Verben:*

flexive_sign	
morpheme	: {flexive}
morphemes	: <{flexive}>
cat	: verb
subcat_base	: k_base

<flexive, vowel_verb> *flexive_sign*	
subcat_cat	: vowel_verb
phon	: ‘ru’

<flexive, cons_verb> *flexive_sign*	
subcat_cat	: cons_verb
phon	: ‘u’

Bei Zeichen vom Typ ‘root_sign’ ergibt sich offensichtlich der Füller der Rolle ‘morphemes’ aus dem Wert des Attributes ‘lexeme’. Bei Zeichen für Morpheme — also solchen des Typs ‘morpheme_sign’ — kann weiterhin aus der Rolle 'morpheme' durch Listenbildung der Wert der Rolle 'morphemes' berechnet werden. Die zur Wirkung kommenden Regeln lassen sich folgendermaßen beschreiben:

(ex 106) *a.* *Aleitung des Attributes* 'morpheme' *aus der Rolle* 'lexeme' *bei Wurzel-Zeichen:*

$$\left[\ \textit{root\_sign}\ \right] \Longrightarrow \left[\begin{array}{ll}\textbf{morpheme} & : \mathrm{L} \\ \textbf{lexeme} & : \mathrm{L}\end{array}\right]$$

b. *Ableitung der Morphemliste* 'morphemes' *aus dem Attribut* 'morpheme' *bei Morphem-Zeichen:*

$$\left[\ \textit{morpheme\_sign}\ \right] \Longrightarrow \left[\begin{array}{ll}\textbf{morphemes} & : \langle \mathrm{M} \rangle \\ \textbf{morpheme} & : \mathrm{M}\end{array}\right]$$

Dank dieser Regeln müssen im Lexikon nur noch solche Eigenschaften angegeben werden, die sich nicht erschließen lassen:

(ex 107) *a.* *Der Lexikalische Eintrag des Verbes 'kaku':*

$$\left[\begin{array}{lr}\langle \textit{\textbf{kaku}} \rangle & \textit{root\_sign} \\ \textbf{phon\_base} & : \text{'ka'} \\ \textbf{lexeme} & : \{\mathsf{kaku}\} \\ \textbf{cat} & : \text{k\_cons\_verb}\end{array}\right]$$

b. *Der Lexikalische Eintrag des Präsensflexives für vokalische und konsonantische Verben:*

$$\left[\begin{array}{lr} & \textit{flexive\_sign} \\ \textbf{morpheme} & : \{\mathsf{flexive}\} \\ \textbf{cat} & : \text{verb} \\ \textbf{subcat\_base} & : \text{k\_base}\end{array}\right]$$

$$\left[\begin{array}{lr}\langle \textit{\textbf{flexive, vowel\_verb}} \rangle & \textit{flexive\_sign} \\ \textbf{subcat\_cat} & : \text{vowel\_verb} \\ \textbf{phon} & : \text{'ru'}\end{array}\right] \qquad \left[\begin{array}{lr}\langle \textit{\textbf{flexive, cons\_verb}} \rangle & \textit{flexive\_sign} \\ \textbf{subcat\_cat} & : \text{cons\_verb} \\ \textbf{phon} & : \text{'u'}\end{array}\right]$$

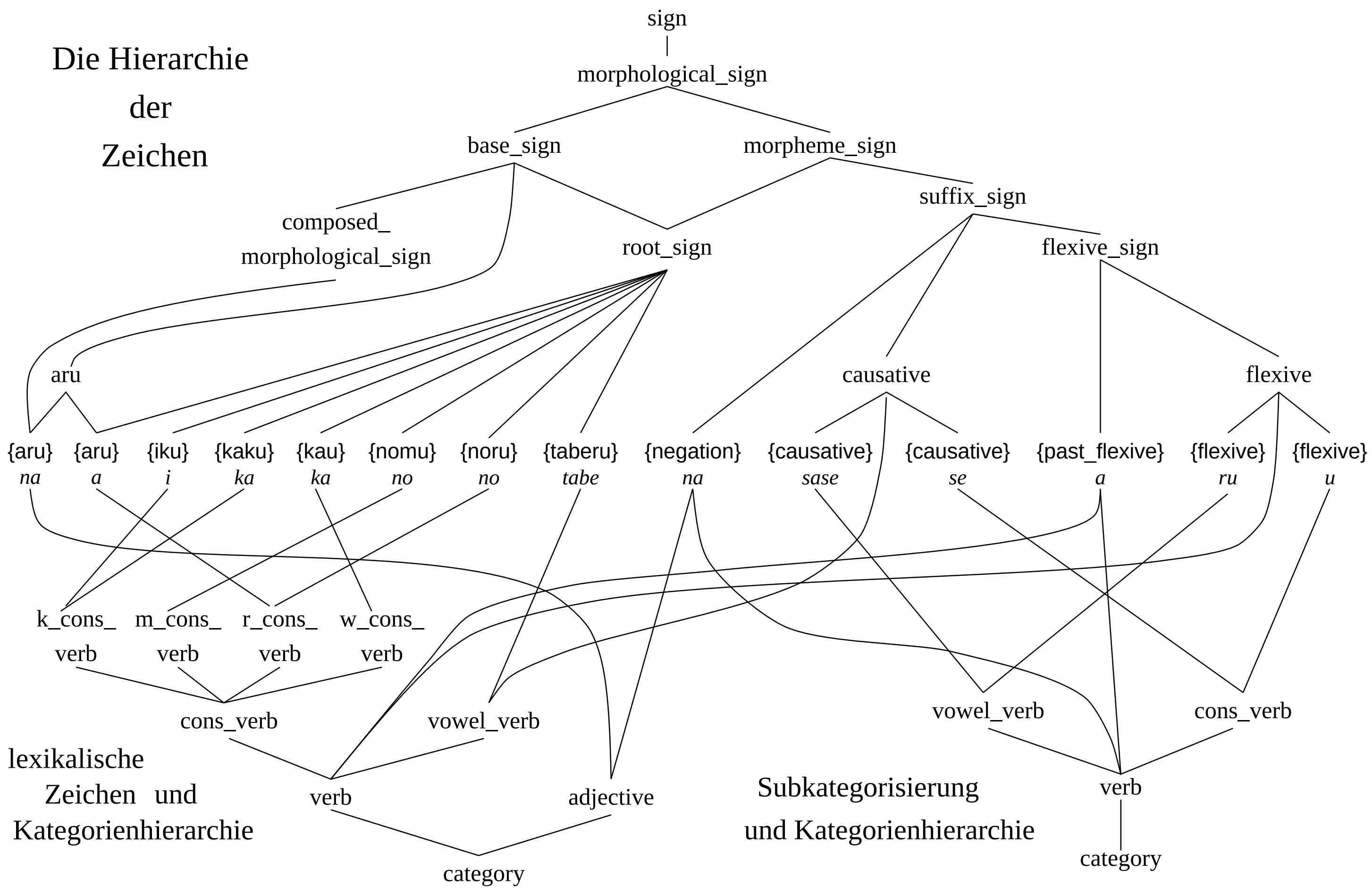
4.6.5 Zusammenfassung

Die Organisation der Zeichen in Hierarchien, die Formulierung auf diesen basierender lexikalischer Regeln und die Ausnutzung der Vererbungsmechanismen ermöglichen uns, bei der Angabe lexikalischer Einträge Redundanzen zu vermeiden. Aus allen drei Aspekten ergibt sich eine Struktur, in die sich lexikalische Einträge so einordnen lassen, daß nur noch die ihnen wesentlichen, nicht aus allgemeinen Prinzipien erschließbaren Charakteristika beschrieben werden müssen.

Abschließend wollen wir anhand der Graphik (fig 13) die Einordnung einiger schon behandelter Zeichen in dieses Schema exemplarisch darstellen. Aus

Gründen der Übersichtlichkeit verzichten wir bei der Angabe von Zeichentypen auf die Attribute und kennzeichnen Instanzen einzig durch Morphem und Form.

Zusätzlich zu der Einordnung in die Hierarchie der Zeichentypen werden die lexikalischen Einträge entsprechend der Merkmale 'cat' und 'subcat_cat' — soweit diese für sie spezifiziert sind — in die Kategorienhierarche eingeordnet.



Die Hierarchie der Zeichen

(fig 13)

5 Ergebnisse und Ausblick

Eine kurze Zusammenfassung der Ergebnisse soll den Abschluß dieser Arbeit bilden. In aller Kürze charakterisiert, läßt sich der vollzogene Prozess als Annäherung eines genau bestimmten, aber trotzdem recht unformal erscheinenden "Inhaltes" und einer inhaltlich neutralen, aber durch die Notwendigkeit exakter Formulierungen gekennzeichneten "Form" beschreiben: Der "Inhalt" ist das morphologische Verhalten japanischer Verben, die "Form" der mathematisch strukturierte, auf Vererbungshierarchien und Defaultfolgerung basierende Formalismus. Das Ziel der Arbeit war somit das Gelingen dieser Zusammenführung, also eine funktionierende Modellierung.

Versetzt man sich nach dem Erreichen dieses Zieles zurück an die zwei Ausgangspunkte der Näherung, erscheint das Resultat, den verschiedenen Perspektiven entsprechend, in doppelter Gestalt: der linguistischen und der formalen.

Aus linguistischer Sicht erhalten wir eine formale Beschreibung des morphologischen Verhaltens japanischer Verben. Die in dieser Beschreibung behandelten Verbformen wurden so ausgewählt, daß sie für die Probleme, die in einer morphologischen Beschreibung auftreten, exemplarisch sind. Das hier formulierte linguistische Modell kann damit leicht zu einer umfassenden Darstellung der japanischen Verbmorphologie ausgeweitet werden. Die Art der Darstellung hat zwei Vorteile für die maschinelle Sprachverarbeitung: Sie ist der intuitiven Darstellung grammatischer Regeln, wie man sie in didaktisch ausgerichteten Grammatiken findet, nahe und daher dem Menschen leicht verständlich; sie ist weiterhin formal, d.h. für die Anwendung auf dem Computer geeignet.

Aus der formalen Perspektive ergeben sich genau genommen zwei Resultate: Das eine betrifft die Entwicklung eines Formalismus, der auf Vererbungshierarchien und Defaultfolgerung beruht und speziell für die Beschreibung linguistischer Phänomene adäquat ist. Das andere besteht in der konkreten Umsetzung der morphologischen Modellierung in das existierende System FLEX, das — unter bestimmten Einschränkungen — als "Prototyp" des entwickelten Formalismus gesehen werden kann: Als lauffähiges Computerprogramm errechnet es zu einer Liste von Morphemen die jeweils korrekte, orthographische Form.

Mit der Beschreibung der japanischen Verbmorphologie durch eine funktionierende FLEX-Terminologie ist der Nachweis erbracht, daß sich morphologische Sachverhalte mit den Mitteln der hierarchischen Vererbung und der Defaultfolgerung darstellen lassen. Zugleich zeigt die abstrakte Notation, die der konkreten FLEX-Modellierung zugrundeliegt, und sich nur geringfügig von dieser unterscheidet, daß eine solche Darstellung nicht nur möglich ist, sondern — dank ihrer Deklarativität — auch Anforderungen an Einfachheit und Klarheit genügen kann.

Resultat der abtrakten Notation ist der Verweis auf einen linguistischen Formalismus, der sich mit den Anforderungen der Sprachbeschreibung überdeckt. Wir haben gesagt, daß sich FLEX-Terminologie und abstrakte morphologische Notation nur geringfügig unterscheiden. In bezug auf den Nachweis der *Möglichkeit* einer Beschreibung ist dieser Unterschied tatsächlich gering; für das resultierende System hingegen wirkt sich schon diese geringe Abweichung durch den Verlust der Anwendungsuniversalität aus: Wir haben FLEX dort, wo die bereitgestellten Ausdrucksmittel nicht ausreichten, durch compute-Prädikate erweitert. Diese Prädikate aber haben eine prozedurale Semantik, die nicht durch das unterliegende FLEX-Kalkül berücksichtigt wird. Während das Kalkül Konsequenzen der Terminologie unabhängig von einer vorgegebenen Berechnungsreihenfolge ermittelt, ergibt sich durch das prozedurale Verhalten der compute-Prädikate eine solche Reihenfolge. Aus der "Axiomatik" der Verbmorphologie, die sich zur Vervollständigung von Zeichen, zur Analyse oder zur Generierung anwenden läßt, wurde durch diese Erweiterungen ein "Berechnungsverfahren", das nur eine dieser Anwendungen erlaubt.

Das Problem entsteht also aus der teilweisen Divergenz der linguistischen Anforderungen an die Ausdrucksmittel eines Formalismus und die Beschreibungsmittel, die von FLEX zur Verfügung gestellt werden und betrifft somit die Frage der *Beschreibungsadäquatheit*.

Ein Nebeneinanderstellen der Beschreibungsmittel, die in der abstrakten morphologischen Darstellung auftreten, und damit wirklich benötigt werden, und der Mittel, die durch FLEX vorgegeben werden, verweist damit auf einen Formalismus, der der linguistischen Materie wirklich entgegenkommt. die "Nähe" eines solchen zum existierenden System FLEX hingegen zeigt, daß sich ein solcher auch realisieren läßt.

Die eben geschilderten Ergebnisse korrespondieren in direkter Weise mit den Themen weiterführender Untersuchungen: Aus Sicht der Linguistik ergibt sich das Interesse an einer Ausweitung der schon vollzogenen Beschreibung: Anhand von charakteristischen "Problemfällen" haben wir Struktur und Prinzipien einer Modellierung der japanischen Verbmorphologie aufgezeigt. Es liegt damit nahe, diese "Problemfälle" auf der begründeten formalen Basis aufbauend zu einer umfassenderen Beschreibung der japanischen Verbformen zu erweitern. Verbvalenz und Semantik schließen sich dabei an die Modellierung der Verbformen an. Schließlich ließe sich das entstehende Modell in eine Gesamtbeschreibung der japanischen Sprache integrieren.

Die zwei folgenden Themen betreffen wiederum die formale Seite unserer

Modellierung:

Wir haben auf das Hervortreten der Umrisse eines adäquaten linguistischen Formalismus als Ergebnis des Vergleichs zwischen konkreter Modellierung und abstrakter Notation schon hingewiesen. Damit ergibt sich die genaue Ausarbeitung einer Spezifikation und die Implementierung eines entsprechenden Systems als Thema sich anschließender Untersuchungen. Ein solches System wäre geeignet, die rein deklarative Grammatik, die in dieser Arbeit parallel zur FLEX-Beschreibung entwickelt wurde, zu implementieren, und böte somit die formale Grundlage zu der eben angesprochenden umfassenden Beschreibung der japanischen Grammatik. Die beschriebenen Thematiken ergänzen sich damit — wenn sie auch ganz verschiedene Aspekte behandeln — zu einem gemeinsamen Ganzen.

Aber wenn das so entstehende System dank der Deklarativität auch den Anforderungen nach Eleganz und Anwendungsunabhängigkeit entgegenkommt, müßte es gerade aus diesen Gründen Anforderungen an "reale" Systeme widersprechen: Es wäre sicherlich nicht sonderlich effizient.

Damit aber sind wir bei dem dritten Thema für eine fortführende Untersuchung: Kriterien der Beschreibungseleganz, wie das der Deklarativität einer Darstellung, resultieren letztendlich in der mathematischen Begründung des Formalismus und damit auch der Beschreibung. Systeme, die diesen Kriterien genügen, stehen damit jedoch nur scheinbar im Widerspruch zu Anforderungen "realer" Anwendungen: Erst durch die mathematische Grundlegung wird die fehlerfreie, axiomatische Spezifikation komplexer Sachverhalte möglich. Damit aber ist die Grundlage für anwendungsbezogene Optimierungen gelegt: Auf die mathematische Definition aufsetzend lassen sich Systeme angeben, die unter Verlust bestimmter allgemeiner Eigenschaften wie der Vollständigkeit des Kalküls auf bestimmte Funktionen hin optimiert sind. Von der allgemeinen theoretischen Spezifikation ausgehend kommen wir damit zu anwendungsbezogenen und effizienten Programmen.

Das dritte Thema wäre damit die Untersuchung der Frage, wie sich aus der deklarativen Beschreibung effiziente, auf bestimmte Probleme wie Generierung oder Analyse spezialisierte Systeme ableiten lassen, die den Anforderungen des Einsatzes von computerlinguistischen Systemen genügen.

Literatur

Carpenter, B. 1992 . *The Logic of Typed Feature Structures with Applications to Unification-based Grammars, Logic Programming and Constraint Resolution*, volume 32 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, New York.

Chomsky, N. 1981 . *Lectures on Government and Binding*. Studies in Generative Grammar 9. Foris, Dordrecht.

Ebbinghaus, H.-D., Flum, J., und Thomas, W. 1992 . *Einführung in die mathematische Logik*. BI-Wiss.-Verl., Mannheim, Leipzig, Wien, Zürich.

Gazdar, G., Klein, E., Pullum, G. K., und Sag, I. A. 1985 . *Generalized Phrase Structure Grammar*. Basil Blackwell, Oxford.

Grewendorf, G., Hamm, F., und Sternefeld, W. 1996 . *Sprachliches Wissen: eine Einführung in moderne Theorien der grammatischen Beschreibung*. Suhrkamp, Frankfurt am Main.

Gunji, T. 1987 . *Japanese Phrase Structure Grammar*. Reidel, Dordrecht.

Hoppe, T., Kindermann, C., Quantz, J. J., Schmiedel, A., und Fischer, M. 1993 . BACK V5 Tutorial and Manual. KIT Report 100, Department of Computer Science, Technische Universität Berlin, Berlin, Germany.

Iida, M., Manning, C., O'Neill, P., und Sag, I. 1994 . The lexical integrity of japanese causatives. Presented at the LSA 1994 Annual Meeting, Boston. Unpublished ms., Stanford University.

Ishii, K., Kono, M., Miyazawa, Y., Nagano, N., Watanabe, H., und Watanabe, N. 1989 . *Hiroo Japanese Center: The Complete Japanese Verb Guide*. Charles E. Tuttle, Tokyo.

Kato, Y. 1985 . *Negative Sentences in Japanese*. PhD thesis, Sophia University, Tōkyō.

Lewin, B. 1990 . *Abriss der japanischen Grammatik*. Harrassowitz, Wiesbaden. 3., verbesserte Auflage.

Luck, K. v. und Owsnicki-Klewe, B. 1990 . KL-ONE: Eine Einführung. IWBS Report 106, IBM Germany Scientific Center, IWBS, Stuttgart, Germany.

Lyons, J. 1968 . *Introduction to Theoretical Linguistics*. Cambridge University Press, London, New York.

Makino, S. und Tsutsui, M. 1986 . *A Dictionary of Basic Japanese Grammar*. The Japan Times, Tōkyō.

McClain, Y. M. 1981 . *Handbook of Modern Japanese Grammar*. Hokuseidō Press, Tōkyō.

Pollard, C. und Sag, I. A. 1987 . *Information-based Syntax and Semantics, Vol. 1.* Number 13 in Lecture Notes. CSLI Publications, Stanford University. Distributed by University of Chicago Press.

Pollard, C. und Sag, I. A. 1994 . *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago.

Quantz, J. und Royer, V. 1992 . A preference semantics for defaults in terminological logics. pages 294–305.

Quantz, J. J. und Ryan, M. 1993 . Preferential default description logics. KIT Report 110, Department of Computer Science, Technische Universität Berlin, Berlin, Germany.

Rickmeyer, J. 1983 . *Morphosyntax der japanischen Gegenwartssprache*, volume 2 of *Deutsch und Japanisch im Kontrast*. Groos, Heidelberg.

Rickmeyer, J. 1995 . *Japanische Morphosyntax*. Groos, Heidelberg. (Früher u.d.T.: Morphosyntax der japanischen Gegenwartssprache, Rickmeyer 1983).

Shieber, S. M. 1986 . *An Introduction to Unification-Based Approaches to Grammar*. CSLI Lecture Notes 4. Center for the Study of Language and Information, Stanford University, Stanford, Cal.

Suska, S. 1994 . A proof theory for preferential default description logics. KIT Report 117, Department of Computer Science, Technische Universität Berlin, Berlin, Germany.

Udo, M. 1982 . Syntax and morphology of the japanese verb: A phrase structural approach. unpublished M.A. Thesis, University College London.

von Stechow, A. und Sternefeld, W. 1988 . *Bausteine syntaktischen Wissens, Ein Lehrbuch der generativen Grammatik*. Westdeutscher Verlag, Opladen.