

4 THEORETISCHER RAHMEN

4.0 Überblick

Dieses Kapitel geht auf einige morphologietheoretische Grundannahmen ein, die **mo_lex** zugrundeliegen (4.1), und leistet eine grobe Einordnung von **mo_lex** im Vergleich zu anderen morphologischen oder für die morphologische Theorie-Bildung relevanten Formalismen (4.2). Nicht berücksichtigen werde ich hier Schnittstellen-Probleme, die ich in §2/2 und §2/3 unter stärkerer Einbeziehung der albanischen Verbflexion behandle.

4.1 Linearität und Endlichkeit

Für die phonologische Theorie-Bildung scheint es unumgänglich zu sein, auf die lineare Abfolge ihrer Minimal-Einheiten zu referieren, während in der Syntax-Theorie zunehmend versucht wird, von der linearen Abfolge zugunsten rein hierarchischer Strukturen wegzustrahieren und die tatsächliche Abfolge von Teilstrukturen entweder in kanonischer Weise aus den hierarchischen Strukturen abzuleiten (Kayne 1994) oder als Ergebnis anderer Sprachmodule (Phonologie, Morphologie, Synthesis) aufzufassen (Chomsky 1994, Thümmel 1991). Für die Morphologie wird das Linearisierungs-Problem üblicherweise nur in Schnittstellen-Kontexten aufgegriffen, etwa bei der Behandlung von Infixen, die die morphologisch "geforderte" Linearität aufbrechen, um phonologischen Constraints zu genügen (vgl. Prince & Smolensky 1993:Kap. 4), oder bei der Diskussion darüber, wie syntaktische Prozesse die Abfolge von Morphemen in morphologischen Wörtern beeinflussen (Baker 1988, Anderson 1992, Halle & Marantz 1993). Hier vertrete ich die Auffassung, daß Allomorphie am besten durch den Bezug auf lineare Repräsentationen und mit Hilfe endlicher Kontext-Spezifikationen beschreibbar ist, eine der wesentlichen Voraussetzungen für **mo_lex**¹.

4.1.1 Regularität und Linearität

Der folgende Kontrast läßt sich ohne empirische Unterschiede wie in (2) oder (2') beschreiben²: (2') ist dabei die durch **mo_lex** nahegelegte Perspektive.

- (1) *hip-ën* *tho-të*
 einsteigen-3sg sagen-3sg
- (2) 3sg wird bei *hipi* als *-i*, bei *them* als *-ë* realisiert.
- (2') 3sg wird unmittelbar rechts vom Stamm für *hipi* als *-ën*, unmittelbar rechts vom Stamm für *them* als *-të* realisiert.

Die Qual der Wahl zwischen (2) und (2') ergibt sich daraus, daß Morpheme, die eine morphologische Einheit bilden, meistens strikten Reihenfolge-Beschränkungen unterliegen. Z.B. gibt es im Albanischen keine Verbform mit der Morphem-Abfolge 3sg-gehen. Auch (3) liefert kein eindeutiges Argument für oder gegen (2'):

- (3) *hip-e-t*
 gehen-nak-3sg

¹Dies scheint auch für *Lexical Insertion* in Distributed Morphology (vgl. 4.2.1) zu gelten, obwohl in DM auch nichtlineare morphologische Regeln zur Anwendung kommen.

²Zugrunde liegt hier die Annahme, daß Allomorphie für ein bestimmtes Morphem immer in Abhängigkeit von anderen Morphemen besteht.

Zwar läßt sich die alternative Realisierung von **3sg** in (3) leicht in (2') integrieren, etwa durch die Feststellung: 3sg wird unmittelbar rechts von *nak* zu *-t*, aber eine empirisch adäquate Formulierung läßt sich auch nach dem Schema von (2) finden:

- (4) 3sg wird in **nak**-Formen von *hipi* als *-t* in anderen Formen als *-ën* realisiert.

4.1.2 adm-Formen

adm-Formen bestehen (s. §2/1) aus einer Kurzform des Partizips und einer "abgeschwächten" Form des Auxiliars *ka-m*, "haben". Die Reihenfolge ist par-Aux im Gegensatz zu Aux-par bei **prf**-Formen:

- (5) *punua-0-ka-m* *ka-m punua-r*
arbeiten-par-Aux-1sg Aux-1sg arbeiten-par
"Ich habe gearbeitet." "Ich arbeite ja!"

Daß Auxiliar und Partizip in **adm**-Formen eine morphologische Einheit bilden, läßt sich an vielen Punkten zeigen: Die Verkürzung des Partizips (*punua-0* statt *punuar*), die Abschwächung des Stammvokals in **adm-imf**-Formen (*punua-kësh-a* vs. *kish-a punua-r* im **plq**). Bestimmte Nichtstandard-Formen zeigen weitere idiosynkratische Allomorphie. So existieren zu den Standard-**adm**-Formen in (6a) die Alternativen in (6b), während es für **prf**- und **plq**-Formen keine derartigen Varianten gibt (gekennzeichnete Formen sind Nichtstandard):

- (6)

adm 1sg	a.	<i>pun-ua-ka-m</i>	b.	<i>(pun-o-ka-m)</i>
prf 1sg		<i>ka-m pun-ua-r</i>		<i>*ka-m pun-o-r</i>

Für die **3sg-adm-imf** existiert eine Alternativ-Form, deren Endung deutlich von der entsprechenden **plq**-Form abweicht, während die Standard-Form einer Alternativ-Form für das **plq**entspricht:

- (7)

adm inf 3sg	<i>(punua-ke-j)</i>	<i>punua-kësh-0</i>	<i>*punua-kësh-te</i>
plq 3sg	<i>*ke-j punuar</i>	<i>(kish-0 punuar)</i>	<i>kish-te punuar</i>

Diese Daten deuten darauf hin, daß nicht die Präsenz von Morphemen in einer Verbform Allomorphie hervorruft, sondern Morpheme in einer bestimmten linearen Position. Die Stichhaltigkeit dieses Arguments leidet aber darunter, daß sich eine alternative Erklärung anbietet. Vielleicht triggern nur solche Morpheme Allomorphie, die mit dem Morphem ein morphologisches Wort bilden. In der Tat kann man die Auxiliare im **prf** von ihren Partizipien trennen, nicht jedoch im **adm**. Das spricht dafür, daß **adm**-Formen in höherem Grad Wort-Status haben als **prf**-Formen:

- (8) a. *Ka-m shpejt mësuar Shqip.* b. **Mësua-0 shpejt ka-m Shqip.*
Aux-1sg schnell lernen-par Albanisch lernen-par schnell Aux-1sg Albanisch
"Ich habe schnell Albanisch gelernt." "Ich lerne ja schnell Albanisch."

4.1.3 Pronominale Klitika

Gruppen von pronominalen Klitika in **imv**-Formen liefern hier noch eindeutige Evidenz für die Bedeutung der linearen Morphemabfolge. Im Gegensatz zum Kontrast **prf/adm** sind die hier zu beobachtenden Alternativformen auch semantisch äquivalent:

- (9) a. *Bjer-m-a-ni!* b. *M-a bi-ni!*
bringen-d1s-a3s-2pl d1s-a3s bringen-2pl
"Bringt es mir!" "Bringt es mir!"

Wieder bietet sich die Erklärung an, daß Allomorphie nur wortintern getriggert wird. Dagegen sprechen jedoch die alternativen Morphem-Formen im **aor-nak** und im **kon-prs**, die von Morphemen (*të* in (10a), *u* in (10b)) abhängen, die sicher nicht Bestandteil des Hauptverbs sind, wenn es *ma* in (9) nicht ist:

(10)

	aor akt	aor nak		prs ind	prs kon
1sg	<i>mëso -va</i>	<i>u mësova</i>		<i>mësoj-0</i>	<i>të mësoj</i>
2sg	<i>mëso -ve</i>	<i>u mësove</i>		<i>mëson-0</i>	<i>të mëso-sh</i>
3sg	<i>mëso-i</i>	<i>u mësoa-0</i>		<i>mëson-0</i>	<i>të mëso-jë</i>

nak-u (s. §2/2) ist ein Klitikon mit denselben Stellungseigenschaften wie die pronominalen Klitika (z.B. *ma* in (9)). *të* müssen wir sogar noch weiter außen in der Wortstruktur ansetzen, da pronominale Klitika in **kon**-Formen grundsätzlich *zwischen të* und dem Verbstamm stehen:

- (11) *Të më mëso -sh!*
 kon e1s lernen-2sg
 "Daß du mir (ja) lernst!"

Ich folgere, daß auch präverbales *ma* in (9) problemlos den alternativen Stamm *hjer-* auslösen würde, wenn die bloße Anwesenheit von Morphemen für die Allomorphie verantwortlich wäre. Hier erweist sich also die Annahme, daß Allomorphie durch bestimmte lineare Abfolgen von Morphemen bedingt ist, als empirisch überlegen.

4.1.4 Endliche Kontexte

mo_lex enthält eine drastische Restriktion, was mögliche Kontexte für die Allomorphie angeht: Diese müssen endlich (d.h. durch ERAs ausdrückbar) sein. Die damit verbundene These, daß Allomorphie sich nur auf endlich viele Kontexte bezieht, ist empirisch widerlegbar. Z.B. gibt es Derivationsprozesse, die grundsätzlich beliebig iterierbar sind, wie Possibilitativ und Kausativ im Ungarischen:

- (12) *ugr-at-hat-tat-hat...at-0*
 springen-kaus-poss-kaus-poss-...-kaus-1sg
 "er kann springen...lassen können lassen."

Nun ist es denkbar, daß die Realisierung des **1sg**-Morphems vom Stamm (*ugr-*) abhängt, egal, wieviele Affixe zwischen diesem Morphem und dem Stamm stehen. Tatsächlich ist *ugr-ik* ein irreguläres Verb, das die Endung *-ik* anstelle des (regulären) Nullmorphems verlangt. Wenn auch die Formen in (12) *-ik* tragen müßten, würde dies rekursive *Reguläre Ausdrücke* erfordern, etwa: "*ugr (kaus|poss)*\3sg --> ik*". Wie in allen anderen mir bekannten Fällen besteht jedoch keine solche Abhängigkeit: Sobald andere Affixe zwischen *ugr-* und das Flexionsmorphem treten, trägt das Verb die Default-Null-Endung.

Albanische **kon**-Formen mit Klitik-Gruppen stellen auf den ersten Blick ein Problem für die Formulierung durch ERAs dar, da zwischen Endung und **kon**-Partikel beliebige Klitik-Gruppen und sogar der Negations-Partikel *mos* treten können:

- (13) a. *Të bë-sh të fala!*
 kon machen-2sg Grüße
 "Daß du ja Grüße ausrichtest!"
- b. *Të më bë-sh të fala!*
 kon e1s machen-2sg Grüße
 "Daß du mir ja Grüße ausrichtest!"
- c. *Të m'u bë-sh të fala!*
 kon e1s-d3p machen-2sg Grüße
 "Daß du mir ihnen ja Grüße ausrichtest!"

d.
Të mos m'í-a prish-ni qejfin!
 kon nicht e1s-d3p-a3s verderben-2pl Spaß
 "Daß ihr mir ihnen ja nicht den Spaß verderbt!"

Dennoch sind die Möglichkeiten für Morphem-Kombinationen endlich und lassen sich durch den *ERA* in (14a) ausdrücken (Ich verzichte auf Darstellung von Klammern.). Das **mo_lex**-Constraint, das die Ausnahmeform der **2sg** im Konjunktiv regelt, ist dementsprechend (bis auf Klammern) (14b):

- (14) a. (neg2)?(e1[sp])?(d[1-3][sp])?(a[1-3][sp])?
 b. (kon)(neg2)?(e1[sp])?(d[1-3][sp])?(a[1-3][sp])?{Vstamm} \2sg

4.2 mo_lex im Vergleich

Die Theorien, die ich hier bespreche, sind, was Anspruch und Grad der Formalisierung angeht, sehr heterogen und sollen - wenigstens in Ansätzen - einiges vom Spektrum aktueller morphologischer Theoriebildung abdecken. In **4.2.1** behandle ich Distributed Morphology (DM, Halle & Marantz 1993), ein Modell von Morphologie, das in der Gesamt-Konzeption viele Gemeinsamkeiten mit (und in Einzelheiten sicher ebensoviele Unterschiede zu) **mo_lex** aufweist. **4.2.2** geht auf DATR ein (Evans & Gazdar 1990, 1996). In **4.2.3** vergleiche ich formale Eigenschaften von Two-Level-Grammatiken mit denen von **mo_lex** und gehe auf Vorteile von **mo_lex** gegenüber der Morphotaktik von Koskeniemi (1983) ein. In **4.2.4** untersuche ich kurz die Möglichkeit, **mo_lex** im Sinn von Optimality Theory (Prince & Smolensky 1993) als System mit verletzbaren Constraints aufzufassen.

4.2.1 Distributed Morphology (DM)

4.2.1.1 DM

mo_lex, insbesondere seine Verwendung in dieser Arbeit, hat einige wesentliche Gemeinsamkeiten mit DM (Halle und Marantz 1993, im Weiteren: [H&M]):

- Morphologie wird als postsyntaktisch gesehen (vgl. §2/2 und Trommer in Vorb.).
- Phonologisches Material wird den Bestandteilen von vollständigen morphotaktischen Strukturen zugeordnet, die ausschließlich abstrakte morphosyntaktische Information enthalten.
- Die Morphologiekomponente besteht in beiden Ansätzen aus den drei Modulen morphotaktische Constraints (in DM: Regeln), willkürliche Allomorphie, (DM: *Vocabulary Insertion*) und Morphophonologie (DM: *Readjustment Rules*, s. auch §2/3).

Demgegenüber bestehen einige grundsätzliche Unterschiede:

- DM ist eine erklärtermaßen derivationale Theorie. **mo_lex** ist kompatibel mit einem vollständig constraintbasierten Grammatik-Modell (vgl. §2/2 und §2/3).
- Der Input für die Allomorphie-Komponente in **mo_lex** ist bereits linearisiert. Diese Linearisierung geschieht in DM im Rahmen von *Vocabulary Insertion*.

Im Folgenden gehe ich nur auf *Vocabulary Insertion* in DM ein, da **mo_lex** als Repräsentationssprache für den Bereich der willkürlichen Allomorphie konzipiert ist, und die Behandlung von Morphotaktik und Morphophonologie in dieser Arbeit zu spekulativ ist, um eine geeignete Vergleichsbasis zu liefern.

4.2.1.2 Vocabulary Insertion in DM

Morpheme sind in DM vor der (postsyntaktischen) lexikalischen Insertion (*Vocabulary Insertion*) Bündel morphosyntaktischer Features ohne phonologische Information, etwa [+ga +nom +sig +mas]. Vokabular-Einträge sind geordnete Paare von unerspezifizierten Merkmalsbündeln und phonologischen Matrizen:³

- (15)
- 1) ([+ga +nom +sig +mas], *i*)
 - 2) ([+nomen]\[+ga], *e*)
 - 3) ([+ga], *të*)
 - 4) ([+nomen +nom], *djali*)
 - 5) ([+nomen +akk], *djalin*)
 - 6) ([+komp], *tij*)
 - 7) ([+adj], *madh*)⁴

Diese Einträge bilden eine Spezifitäts-Hierarchie. Z.B. ist ([+ga +nom +sig +mas], *i*) spezifischer als ([+ga], *të*). Auch ([+nomen]\[+ga], *e*)⁵ - "[+nomen]" ist eine Kontextbeschränkung ähnlich wie in **mo_lex** - kommt in dieser Hierarchie vor [+ga]. Präzedenz nach Featurespezifität geht dabei vor Spezifität nach Kontext, d. h. die Anordnung aller drei erwähneter Feature-Bündel ist wie in (15). Jedes abstrakte Morphem wird bei der lexikalischen Insertion mit dem *Vocabulary Item* assoziiert, dessen Feature- (und Kontextrestriktionen) es am genauesten matcht.⁶

- (16)
- | | | |
|--|----|---------------------------------------|
| [+nomen +nom] [+ga +nom +sig +mas] [+adj] | => | <i>djali i madh</i> |
| [+nomen +akk] [+ga +akk +sig +mas] [+adj] | => | <i>djalin e madh</i> |
| [+nomen +akk] [+kom] [+ga +akk +sig +mas] [+adj] | => | <i>djalin më të madh</i> ⁷ |

Z.B. matcht [+ga +nom +sig +mas] sowohl [+ga +nom +sig +mas] als auch [+ga]. Relevant ist aber nur der erste Match, da er der passendste ist. [+ga +akk +sig +mas] hingegen ist mit dem ersten Eintrag inkompatibel, da sich +akk und +nom gegenseitig ausschließen, und matcht [+nom]. [H&M] merken an, daß es für bestimmte Paare von Feature-Bündeln nicht möglich ist, das spezifischere zu finden (etwa [+def] und [+nom]), und mutmaßen mit Noyer (1992), daß eine Hierarchie von Features (z.B. +def > +nom) die nötige Ordnung schafft. Andernfalls, und dies entspricht der Praxis bei ihren konkreten Beispielen, "the correct output can be obtained by imposing an extrinsic order of precedence between the two Vocabulary items in question" ([H&M]:120).

4.2.1.3 Ersetzungs-Hierarchien in DM

Stellen wir zuerst fest, daß wir mehrere Hierarchien vor uns haben: willkürliche Anordnung der Morphemeinträge (bzw. die Feature-Hierarchie), Featurespezifität, Kontextspezifität und die Präferenz von Feature- vs. Kontextspezifität. Dies steht im Gegensatz zu **mo_lex**, wo eine einzige Hierarchie (die

³Einträge enthalten bei [H&M] außerdem Angaben über die Linearisierung von Morphemen, etwa ob ein Suffix oder Präfix vorliegt. Der dahinterstehende Mechanismus wird von [H&M] nur andeutungsweise beschrieben. In jedem Fall hängt er eng mit ihrem Modell von Morphotaktik zusammen, und ich werde darauf hier nicht weiter eingehen.

⁴Die Notation der Einträge weicht von der von [H&M] in nicht relevanten Punkten ab.

⁵Beispiele sind aus Trommer (in Vorb.) und beschreiben verschiedene Realisierungen des sogenannten *vorangestellten* (oder *Gelenk-*)*Artikels*.

⁶Es bleibt bei [H&M] unklar, ob mit Match Unifikation oder Subsumption gemeint ist, d.h. matcht [+mas +sig] [+mas +def] oder nicht? Ich nehme an, daß Subsumption gemeint ist, daß hier also kein Match vorliegt.

⁷*djali i madh*, "der grosse Junge"; *djalin e madh* "den grossen Jungen"; *djalin më të madh*, "den größten Jungen".

Reihenfolge der Constraints) die Zuordnung von Morphemen und phonologischem Material organisiert⁸. Der **mo_lex**-Mechanismus ist konzeptuell einfacher, während die Verfahrensweise in DM mehr inhaltlich motiviert erscheint, indem sie sich erklärtermaßen an *Paninis Prinzip* (speziellere Information überschreibt allgemeinere) orientiert. Nun gibt es für *Paninis Prinzip* in der Literatur zwei Interpretationen, als Axiom (wie in DM) oder als Theorem (wie in OT, Prince und Smolensky 1993:81).

Nehmen wir, um dies zu verdeutlichen, an, daß wir in DM wie in **mo_lex** ausschließlich die Reihenfolge der Constraints (bzw. Morphem-Einträge) berücksichtigen, und postulieren die Reihenfolge (1) [+ga], (2) [+ga +nom +sig +mas]. Da jedes Morphem, das (2) matcht, notwendigerweise auch (1) matcht, und (1) höher in der Hierarchie steht, wird keine Feature-Kombination je durch (2) realisiert, d.h. die Reihenfolge (2), (1) ist die einzige, die elementaren theoretischen Ökonomieprinzipien genügt. Ähnliches gilt für die Abfolge (1) [+ga], (2)[+nomen][+ga]. Wiederum hätte diese Reihenfolge denselben Effekt wie eine Theorie ohne (2) und wäre aus diesem Grund auszuschließen.⁹

Die Reduktion von *Paninis Theorem* auf Ökonomie-Kriterien ist - wieder aus Gründen der konzeptuellen Einfachheit - der Annahme vorzuziehen, daß *Paninis Prinzip* axiomatischen Status hat. Im Kontext einer Interpretation von **mo_lex**-Constraints in OT (vgl. 4.2.4) erlaubt sie eine integrierte Behandlung von Default-Phänomenen in Phonologie, Syntax und Morphologie.

Diese Argumentation gibt natürlich keinen Aufschluß darüber, ob 1) oder 2) in (15) spezifischer ist, noch ermöglicht sie die explizite Formulierung hierarchischer Beziehungen zwischen einzelnen Features (s. 4.2.1.1). Weitere empirische Untersuchungen müssen zeigen, ob solche Stipulationen haltbar sind, sie bilden in jedem Fall nicht den Kern von *Paninis Prinzip*. **mo_lex** gestattet allerdings, diese Restriktionen implizit in der Constraint-Anordnung einzuhalten und so Hierarchien in der Art von DM zu implementieren.

4.2.1.4 Features und Strings

Ein weiterer Unterschied zwischen **mo_lex** und *Vocabulary Insertion* besteht darin, daß **mo_lex** keine Features beinhaltet. Wie Trommer (in Vorb.) zeigt, ist eine solche Feature-Notation als Mittel der Unterspezifikation auch in **mo_lex** sinnvoll. Formal gesehen ist es problemlos, **mo_lex** mit atomaren Features anzureichern, indem man **mo_lex**-Symbole als Bündel elementarer Features darstellt. Wie die einschlägige Literatur zu *Finite-State-Transducern* zeigt ([K&K]:361, Johnson 1972:Kap. 3), wird durch eine solche Erweiterung die Mächtigkeit von *Regulären Relationen* nicht erweitert, solange diese Features nur endlich viele atomare Werte haben.

4.2.1.5 Fazit: mo_lex und Vocabulary Insertion in DM

Die Tabelle in (17) soll noch einmal die wichtigsten Gemeinsamkeiten und Unterschiede zwischen **mo_lex** und *Vocabulary Insertion* in DM gegenüberstellen.

⁸Andere Desambiguierungsmechanismen in **mo_lex** betreffen nur constraintinterne Konflikte.

⁹Viele (**f**)lex-Implementationen geben für (1) die Fehlerwarnung "Pattern can never be matched" aus:

(1) A printf("A");
 A/B printf("A/B");

(17)

	mo_lex	Vocabulary Insertion
Input	linearisierte morphotaktische Strukturen	nichtlinearisierte morphotaktische Strukturen
Funktion	"Abstrakte" Morpheme werden in konkrete umgesetzt.	"Abstrakte" Morpheme werden in konkrete umgesetzt und linearisiert.
Grundmechanismus	Das passendste Morphem "gewinnt".	Das passendste Morphem "gewinnt".
Hierarchien	Reihenfolge der Constraints	<ul style="list-style-type: none"> • Reihenfolge der Morphemeinträge • Spezifität der Feature - Bündel • Kontext vs. kein Kontext
Atomare Einheiten	Strings/ <i>Reguläre Ausdrücke</i>	Feature-Bündel

4.2.2 DATR

4.2.2.1 Zu DATR

Die grundlegenden Mechanismen von DATR (Evans & Gazdar 1990, 1996) sind Knoten, Pfade, Defaults und Vererbung. Z.B. läßt sich ein Teil der albanischen Endungsallomorphie wie folgt darstellen:¹⁰

- (18) END/1sg = 0 /* hap-0, öffnen-1sg, "ich öffne" */
 END/1sg/aor = a /* hap-a, öffnen-aor:1sg, "ich öffnete" */
 END/1sg/aor/shoh = shë /* pa-shë, sehen-aor:1sg, "ich sah" */¹¹

END ist dabei ein Knoten. END/1sg und END/1sg/aor sind Pfade. Knoten sind Informationseinheiten, deren Strukturierung durch Pfadangaben gegeben ist. Intuitiv gesprochen, besagt die erste Zeile, daß eine Endung in der **1sg** leer ist, die zweite, daß eine Endung in der **1sg** im **aor** -a ist, die dritte, daß die Endung in der **1sg** im **aor** für das Verb *shoh* -shë ist. Diese Aussagen stehen offensichtlich in Widerspruch zueinander, ein Widerspruch, den DATR durch Bevorzugung längerer Pfade auflöst. Kürzere Pfade liefern Default-Werte, die durch Angaben in längeren Pfaden überschrieben werden. (19) zeigt die DATR-Ergebnisse für einige Anfragen zu dieser Theorie:

(19)	Eingabe: END/1sg/prs/hap END/1sg/aor/hap END/1sg/aor/shoh/	Ausgabe: 0 a shë
------	--	----------------------------------

Vererbung ist eine Relation zwischen Knoten. Z.B. besagt der erste DATR-Satz in (20), daß der Knoten ZHYEJ alle Pfadangaben vom Knoten LYEJ erbt:

¹⁰In Anhang D befindet sich ein größeres DATR-Fragment, das repräsentative albanische Verbformen generiert.

¹¹Zu Zwecken der Exposition nehme ich hier, anders als in §2/5, kein separates **aor**-Morphem an.

- (20) ZHYEJ: \diamond =LYEJ. */* zhe-va, beschmutzen-aor:1sg, "ich beschmutzte" */*
/ zhy-het, beschmutzen-nak:3sg, "man beschmutzt" */*
- LYEJ/STAMMVOKAL/PAR = ye */* lye-r, ölen-par, "geölt" */*
 LYEJ/STAMMVOKAL/AOR = e */* le-va, ölen-aor:1sg, "ich ölte" */*
 LYEJ/STAMMVOKAL/NAK = y */* ly-het, ölen-nak:3sg, "man ölt" */*

(21)	Eingabe:	Ausgabe:
	LYEJ/STAMMVOKAL/AOR	e
	ZHYEJ/STAMMVOKAL/AOR	e
	LYEJ/STAMMVOKAL/PAR	ye
	ZHYEJ/STAMMVOKAL/PAR	ye
	LYEJ/STAMMVOKAL/NAK	y
	ZHYEJ/STAMMVOKAL/NAK	y

Wie wir in §2/5 genauer sehen werden, weicht die Stammbildung von *zhyej* aber *minimal* von LYEJ ab: **par**-Formen haben den Stamm-Vokal -y- statt -ye- wie bei *lyej*. Dies läßt sich in die DATR-Theorie integrieren, indem wir die Angaben für ZHYEJ in (20) *minimal* erweitern:

- (20) ZHYEJ: \diamond =LYEJ.
- ZHYEJ/STAMMVOKAL/PAR = y */* zhy-rë, beschmutzen-par, "beschmutzt" */*

(21)	Eingabe:	Ausgabe:
	LYEJ/STAMMVOKAL/AOR	e
	ZHYEJ/STAMMVOKAL/AOR	e
	LYEJ/STAMMVOKAL/PAR	ye
	ZHYEJ/STAMMVOKAL/PAR	y

Hier sehen wir auch den zweiten DATR-Default-Mechanismus am Werk. Von anderen Knoten geerbte Information wird von expliziten Pfadangaben am "Zielknoten" überschrieben.

4.2.2.2 Defaults und Constrainthierarchien

Auch **mo_lex**-Theorien lassen sich mit dem Begriff "Default" charakterisieren. Z.B.wäre (22) eine entsprechende Formulierung der Endungsallomorphie in (18):

- (22)
- | | | | |
|-----------|---------------------|-----|-----|
| C1 | S(shoh) V K aor\1sg | --> | shë |
| C2 | aor\1sg) | --> | a |
| C3 | (1sg) | --> | ; |

Anstatt durch die Länge der Pfade wird die Defaulthierarchie hier durch die lineare Reihenfolge der Constraints etabliert. Solange es um die Beschreibung atomarer Eigenschaften geht (wie hier den Spell-Out des **1sg**-Morphems), scheint der einzige Unterschied zwischen **mo_lex** und DATR darin zu liegen, daß **mo_lex** sowohl Bedingungen für Statements (linke und rechte Kontexte) als auch die Defaulthierarchie strikt linear kodiert, während in DATR die lineare Struktur von Pfaden rein durch inhaltliche Erwägungen bestimmt wird und die lineare Abfolge von Sätzen grundsätzlich irrelevant ist.

Doch offensichtlich hat **mo_lex** keinen Mechanismus, der der Vererbung von Knoten zu Knoten entspricht. Z.B. läßt sich in **mo_lex** kein Knoten für LYEJ angeben, von dem ZHYEJ alle Information erben kann bis auf die, die es idiosynkratisch selbst spezifiziert. Wir müssen ZHYEJ bei jedem Constraint für LYEJ hinzufügen:

(23)

C1	V(ZHYEJ LYEJ)/K PAR	-->	y	C3	V(ZHYEJ LYEJ)/K AOR	-->	e
C2	V(ZHYEJ LYEJ)/K PAR	-->	ye	C4	V(ZHYEJ LYEJ)/K NAK	-->	y

Mit Hilfe *regulärer Definitionen* läßt sich aber auch dieser Aufwand vermeiden, wenn wir ZHLYEJ als *reguläre Definition* für (LYEJ|ZHYEJ) verwenden. Gleichzeitig werden dadurch Respezifikationen der Theorie leichter, da die Neueinbeziehung von Lexemen nur die Änderung einer einzigen *regulären Definition* erfordert.

(23')

C1	VZHYEJ/K PAR	-->	y	C3	V{ZHLYEJ}/K AOR SIG	-->	e
C2	V{ZHLYEJ}/K PAR	-->	ye	C4	V{ZHLYEJ}/K NAK	-->	y

Eine ähnliche Formulierung wie in (23') ist im übrigen auch in DATR selbst möglich, indem man LYEJ und ZHYEJ als Attribute eines gemeinsamen Knoten interpretiert:

(24) KNOTEN: ZHYEJ/ PAR = y
KNOTEN: LYEJ/ PAR = ye
KNOTEN: \$ZHLYEJ/ AOR/SIG = e
KNOTEN: \$ZHLYEJ/ PRS NAK = y

#VARS \$ZHLYEJ LYEJ ZHYEJ

\$ZHLYEJ ist hier eine Variable über die Attribute ZHYEJ und LYEJ, die in der letzten Zeile von (26) definiert wird. Diese Version der DATR-Theorie orientiert sich an Langers (1993) Vorschlag, DATR auf *Pure DATR* zu reduzieren, eine DATR-Version ohne spezifische Knoten und globale Vererbung, aber mit derselben Ausdruckskraft wie Standard-DATR.

4.2.2.3 DATR vs. mo_lex

DATR ist im Gegensatz zu **mo_lex** eine generelle Repräsentationssprache für lexikalisches Wissen, und ein Vergleich zwischen beiden nicht ohne weiteres möglich. Sicherlich aber bietet **mo_lex** für den Bereich der willkürlichen Allomorphie eine formal restriktivere Alternative, die es wie in DATR erlaubt, effizient mit Defaults zu arbeiten.

4.2.3 Two Level Morphology

4.2.3.1 Ein kurzer Überblick über Two-Level-Morphologie

Two-Level-Morphologie geht auf Koskenniemi (1981) zurück und gehört inzwischen zu den einflußreichsten Theorien in der computerlinguistischen Morphologie. Gut lesbare Einführungstexte sind u.a. Karttunen (1993) und Sproat (1992). Pulman et al. (1992) sowie Antworth (1994) beschreiben ausführlich größere implementierte Systeme. Two-Level-Grammatiken bestehen aus einer Menge von Constraints über zwei Ebenen, die lexikalische ("Eingabe-") und die "Oberflächen-" (Ausgabe-)Ebene, wobei jedem Buchstaben auf der Eingabeseite genau ein Buchstabe auf der Ausgabeseite entspricht. Z.B. legt das folgende Constraint fest, daß jedem *a* auf der Eingabeseite auf der Ausgabeseite ein *b* entspricht, wenn *a* zwischen *c* und *e* steht, und *c* in der Ausgabe zu *d* und *e* zu *f* wird. Bei allen durch Doppelpunkt getrennten Buchstabenpaaren, sogenannten *feasible pairs*, gehört der linke Buchstabe zur Eingabe und der rechte ist der entsprechende Ausgabe-Buchstabe.

(25) a:b <= c:d__e:f

Das String-Paar (*cae, dbf*) genügt dem Constraint, da *a* im entsprechenden Kontext zu *b* wird, ebenfalls (*cae, dag*) und (*cbe, dbf*). Im ersten Fall werden Kontextbedingungen nicht erfüllt (*a* steht zwar links von *e*, aber *e* korrespondiert mit *g* statt *f*), im zweiten steht *b* statt *a* im relevanten Kontext. In beiden Fällen ist das Constraint irrelevant. Hingegen verletzt (*cae, dgf*) das Constraint, da *a* im angegebenen Kontext *g* statt *b* entspricht. Anstelle der Buchstabenpaare in (25) können auch (zweistellige) *Reguläre Ausdrücke* stehen. Neben dem Constraint-Format in (25) (*Surface Coercion Rules*) gibt es sogenannte *Surface Restriction* (26a) und *Surface Prohibition Rules* (26b):

(26) a. $a:b \Rightarrow c:d_e:f$ b. $a:b / \leq c:d_e:f$

(26a) fordert, daß jedes Buchstaben-Paar *a:b* im Kontext *c:d_e:f* stehen muß. (26b) stellt fest, daß *a:b* in diesem Kontext unmöglich ist. Nachdem Koskeniemi (1983) in seiner Dissertation Constraints per Hand in *FSTs* übersetzt hatte, haben spätere Arbeiten gezeigt, daß solche Constraints generell *Reguläre Relationen* denotieren ([K&K], Ritchie 1992).

String-Paare in Two-Level-Morphologie sind wohlgeformt, wenn sie allen Constraints genügen, bzw. von allen *FSTs*, die den einzelnen Constraints entsprechen akzeptiert werden. Dies bedeutet formal die Schnittmenge der Sprachen, die von den Constraints erkannt werden.¹²

4.2.3.2 mo_lex als Two-Level-System

Einzelne **mo_lex**-Constraints können wir in einem Two-Level-System als *Surface Coercion Rules* (*SCR*, s. o.) auffassen. Die Kontextbeschränkungen von *SCRs* berücksichtigen, wie oben ausgeführt, sowohl den Kontext der Ausgabe wie den der Eingabe, während **mo_lex**-Constraints ausschließlich den Eingabe-Kontext "sehen". Dies bedeutet aber lediglich, daß die Notation für *SCRs* genereller ist als die von **mo_lex**.¹³ Für jedes **mo_lex**-Constraint gibt es eine äquivalente *SCR*, aber nicht unbedingt umgekehrt. Z.B. entspricht dem **mo_lex**-Constraint in (27a) die *SCR* in (27b):

(27) a. $c \backslash a / d \rightarrow b$ b. $a:b \leq c:@_e:@$

@ steht hier für ein beliebiges Symbol des Ausgabealphabets. Sowohl (a) als auch (b) sind erfüllt, gdw. jedes *a*, das in der Eingabe zwischen *c* und *d* steht, in der Ausgabe zu *b* wird.

Die Auflösung von Regelkonflikten in **mo_lex** kann man mit Hilfe des Negations-Operators implementieren, indem man jedes Constraint *C*, dem ein "allgemeineres" Constraint *C'* vorausgeht, auf die Kontexte beschränkt, für die *C'* nicht gültig ist: Z.B. werden die konfligierenden Constraints in (28a) durch die widerspruchsfreien Constraints in (28b) ersetzt, die sich dann problemlos im Two-Level-Formalismus umsetzen lassen.¹⁴

(28) a. $\begin{array}{l} C1 \quad B \backslash A / C \quad \rightarrow \quad X \\ C2 \quad A \quad \rightarrow \quad Y \end{array}$ b. $\begin{array}{l} C1' \quad B \backslash A / C \quad \rightarrow \quad X \\ C2' \quad A / \sim C \quad \rightarrow \quad Y \\ C2'' \quad \sim B \backslash A \quad \rightarrow \quad Y \end{array}$

4.2.3.3 Koskeniemis (1983) Morphotaktik

Koskeniemi (1983) benützt als Modell für die Morphotaktik einen einfachen *Finite-State-Automaten* (*FSA*), dessen Morpheme mit Features annotiert sind. Die mangelnde linguistische Adäquatheit dieses Modells ist oft moniert worden (z.B. Sproat 1992:127) und hat u.a. dazu geführt, daß neuere Two-Level-Implementierungen diese Morphotaktik durch eine Unifikationsgrammatik im Stil von PATRII (Shieber 1983) ersetzt haben. Hier gebe ich nur ein kurzes Beispiel dafür, daß **mo_lex** bestimmte Mängel von

¹² Aus der *Regularität* der einzelnen Constraints ergibt sich allerdings nicht automatisch, daß Two-Level-Grammatiken als ganzes *Reguläre Relationen* beschreiben. Ein Beweis findet sich bei [K&K] (S. 367).

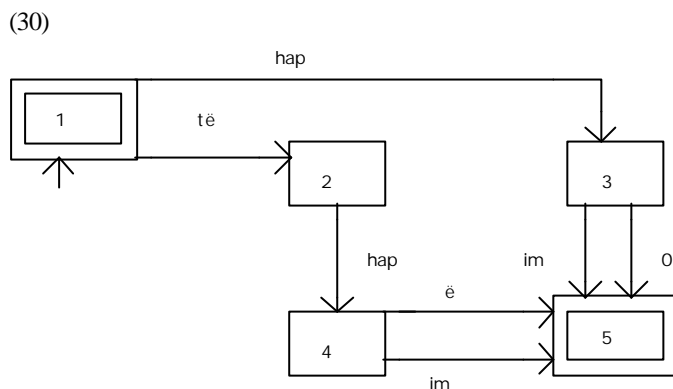
¹³ Die Konstanten stehen hier für Strings. [K&K] erlauben in ihrer Formalisierung von Two-Level-Regeln an denselben Stellen, an denen Paare von Graphemen erlaubt sind, *Reguläre Relationen*.

¹⁴ Eine ausführlichere Beschreibung dieses Verfahrens findet sich in Trommer (in Vorb.).

Koskenniemi Modell vermeidet. Mit FSAs lassen sich Abhängigkeiten zwischen nichtadjazenten Morphemen nicht ausdrücken. Z.B. hängt die Form des **3sg**-Morphems im Albanischen u.a. von der Anwesenheit des **kon**-Partikels *të* ab. Zwischen dem Kongruenz-Morphem und *të* steht jedoch der Verbstamm (vgl. 4.1.4):

- (29) *hap-0* *të hap-ë*
 öffnen-3sg kon öffnen-3sg
 "er öffnet (**ind**)" "er öffnet (**kon**)"

Der FSA in (30) beschreibt zwar die korrekten Formen, aber um den Preis von zwei Übergängen für den Stamm von *hap*:



Auch Endungen wie *-im 1pl*, die für **kon**- und **ind**-Formen identisch sind, müssen in einem einfachen FSA doppelt aufgeführt werden, während in der folgenden **mo_lex**-Theorie, die dieselben Fakten beschreibt, jedem Allomorph genau ein Constraint entspricht:

- (31)
- | | | | |
|-----------|---------------|-----|-----|
| C1 | 1pl | --> | im |
| C2 | S(hap) | --> | hap |
| C3 | kon S(hap)3sg | --> | ë |
| C4 | 3sg | --> | ; |

4.2.4 Optimalitäts-Theorie (OT)

Wie oben (4.2.1) bereits angedeutet, besteht eine enge Affinität zwischen Defaults und einer bestimmten Form von Constraint-Interaktion in Optimalitäts-Theorie (Prince & Smolensky 1993, im weiteren: [P&S]). Grammatiken in OT sind Hierarchien von verletzbaren Constraints, die Input auf den Output abbilden, der diese Constraints "am wenigsten schwer" verletzt. Das Gewicht von Constraintverletzungen ist durch die Dominanz-Relation definiert, die eine lineare Ordnung über den Constraints darstellt. Die Verletzung von Constraints wiegt um so schwerer, je höher diese in der Dominanz-Hierarchie stehen. Bei Mehrfachverletzungen bestimmter Constraints entscheidet die Anzahl der Verletzungen, wenn für mehrere Output-Kandidaten kein höheres Constraint verletzt wird. Die Implementation dieses Grundkonzepts erfolgt im allgemeinen durch Verrechnung von Constraintverletzungen ("Marken"). Zur Veranschaulichung verwende ich einmal mehr ein Beispielfragment aus der albanischen Verbflexion:

(32)

C1	S(hap)	-->	hap	C4	S(jam)	-->	je
C2	S(jam)/3sg	-->	ësh	C5	S(jam)\3sg	-->	të
C3	S(jam)/1sg	-->	ja	C6	[13]sg	-->	;

Wir suchen die optimale Übersetzung für den String "S(jam) 3sg" und vergleichen, inwieweit verschiedene Kandidaten die Constraints in (32) verletzen: Eine Verletzung des Constraints $C = LK \setminus X/RK \rightarrow Y$ liegt vor, wenn in einem Eingabestring E ein String, der X matcht, zwischen einem String, der LK matcht und zwischen einem String, der RK matcht, vorkommt und nicht nach Y übersetzt wird.¹⁵

(33)

Kandidaten	C1	C2	C3	C4	C5	C6
<i>ësh-të</i>				*		*
<i>ësh-0</i>				*	*!	
<i>je-të</i>		*!				*
<i>hap-të</i>		*!		*		*
<i>je-0</i>		*!			*	
<i>hap-0</i>		*!		*	*	
<i>ja-m</i>		*!		*	*	*

Constraintverletzungen sind durch die Marke "*" in der jeweiligen Spalte gekennzeichnet. *ësh-0* und *ësh-të* sind optimaler als alle anderen Kandidaten, weil diese mindestens ein höhergewichtetes Constraint verletzen (**C2**). **C4** ist für die weitere Bewertung irrelevant, da es sowohl von *ësh-të* als auch von *ësh-0* verletzt wird, aber *ësh-0* verletzt **C5** und *ësh-të* nur das niedrigere **C6**, so daß *ësh-të* als optimaler Kandidat übrigbleibt. OT-Constraints sind so konzipiert, daß die Erfüllung eines Constraints u.U. zur Nichterfüllung anderer Constraints führt. So wird jede Erfüllung von **C5** bei der Übersetzung von "3sg" dazu führen, daß **C6** verletzt wird.

Bei der Interpretation von **mo_lex** im Sinn von OT ergeben sich allerdings auch einige Probleme. Ein grundlegendes OT-Constraint (Faithfulness bzw. PARSE, s. [P&S]:Kap. 4), das fordert, daß Ausgabe-Strukturen Eingabe-Strukturen soweit wie möglich entsprechen, ist in **mo_lex** offensichtlich nicht gültig. Eingabe-Strukturen sind abstrakte Morpheme, Ausgabe-Strukturen Phoneme, d.h. die Ausgabe darf (bis auf die lineare Abfolge der Morphem) nicht identisch zur oder auch nur vom selben Format wie die Eingabe sein. Eng damit zusammen hängt die Frage, wie *Negativer Abschluß* im Rahmen verletzbarer Constraints zu realisieren wäre. Wenn wir **mo_lex**-Constraints zulassen, deren Kerne Strings von Symbolen zulassen, müssen wir außerdem eine plausible Deutung der Links-Rechts-Strategie bzw. der Präferenz für längere Strings geben.¹⁶

¹⁵Ein String S_a matcht einen String S_b , gdw. $S_a = S_b$. S_a matcht einen *ERA* in disjunktiver Normalform $(S_1 | S_2 | \dots | S_n)$, gdw. S_a S_1 matcht oder $(S_2 | \dots | S_n)$ matcht.

¹⁶Für eine kritische Wertung von morphologischen Constraints, wie sie hier vorgeschlagen werden s. Golston (in Vorb.).