Universität Zürich
Institut für Computerlinguistik
Prof. Dr. M. Hess

Morphologieanalyse in der Computerlinguistik

Dreitägige Hausarbeit der Philosophischen Fakultät der Universität Zürich

Luzius Thöny
Breitestr. 4
8400 Winterthur
079 779 40 86
lucius.antonius@gmail.com
13.-16. März 2007

Inhaltsverzeichnis

1 Einleitung **1**
1.1 Gegenstand der Morphologie 1
1.2 Terminologische Abgrenzungen 2
1.2.1 Flexion - Derivation - Komposition 2
1.2.2 Produktivität - Motiviertheit 4

2 Bestehende Ansätze **5**
2.1 FSM-basierte Systeme 5
2.1.1 Grundlegendes zu FSM-Systemen 5
2.1.2 Finite-State Toolkits 7
2.1.3 GERTWOL 8
2.1.4 SMOR 9
2.1.5 WordManager / canoo.net 10
2.1.6 TAGH 10
2.1.7 mOLIFde 11
2.2 Nicht FSM-basierte Systeme 11
2.2.1 Functional Morphology 11
2.2.2 Morphology Induction 12

3 Verbleibende Probleme und mögliche Lösungsstrategien **13**
3.1 Nicht-konkatenative Eigenschaften von Sprachen 13
3.2 Einschränkung der Wortbildungsregeln 14
3.3 Priorisierung komplexer Analysen 14

4 Aktuelle Tendenzen in der Forschung **15**
4.1 Gewichtete Automaten 15
4.2 Finite-State Registered Automata (FSRA) 16
4.3 Integration von Morphologiekomponenten 17

5 Schluss **17**

Bibliographie **17**

Zusammenfassung

Diese Arbeit beschäftigt sich mit den Methoden der Morphologieanalyse in der Computerlinguistik. Es sollen wichtige Ansätze für die Handhabung der morphologischen Analyse und Generierung unter spezieller Berücksichtigung von Derivation und Komposition dargestellt und verglichen werden. Der Schwerpunkt wird dabei – entsprechend der Lage in der Forschung – auf *Finite State Tools* gelegt. Bei der Präsentation einiger konkreter Systeme beschränke ich mich auf die deutschsprachigen, insbesondere GERTWOL, SMOR und TAGH. Anhand der verschiedenen Systeme wird erläutert, welches die grössten Probleme bei der Umsetzung solcher Morphologien sind, und welche Strategien in der Forschung verfolgt werden, um diese Herausforderungen zu meistern.

15. März 2007
gesetzt mit XeLaTeX
10pt Junicode

1 Einleitung

Bestrebungen, Systeme zu entwickeln, die Wortformen morphologisch analysieren können, gibt es schon seit Anfang der Achtzigerjahre. Damals existierte allerdings kein sprachübergreifendes Modell für ein solches System, sodass die frühen Ansätze mehrheitlich mit *copy-paste* von bestimmten Endungen arbeiteten (Karttunen/Beesley 2005). Diese Situation änderte sich, als der Finne Koskenniemi mit seinem Landsmann Karttunen sowie zwei Forschern aus dem Xerox Research Center, Kay und Kaplan, zusammentraf. Diese vier Herren legten die Grundsteine für die Verarbeitung von Morphologie mithilfe von Finite-State-Tools, einer Methode, die seither zum defacto Standard der Disziplin avanciert ist. Richtig ins Rollen kamen die Bestrebungen mit der Publikation von Koskenniemis Monographie von 1983, in der er erstmals den Two-Level Formalismus vorstellte. Als 1986/1987 auch noch ein Compiler entwickelt wurde, der die Regeln des Two-Level Formalismus automatisch in Transduktoren umwandeln konnte, stand der Entwicklung von brauchbaren Morphologieanalysesystemen nichts mehr im Weg (Koskenniemi schrieb die Transduktoren vor der Verfügbarkeit des `twolc`-Compilers noch von Hand, vgl. Karttunen/Beesley 2005:76). In dieser Tradition ist das von Karttunen mit Koskenniemis Vornamen benannte System *KIMMO*, sowie dessen populärer Ableger *PC-KIMMO* (Antworth 1990) entstanden. Im Verlauf der Neunzigerjahre kam dann als weiteres Hilfsmittel `lexc` hinzu, das die Formulierung von Lexikoneinträgen samt Fortsetzungsklassen erleichtern konnte, und die Implementationen der verschiedenen Ersetzungsoperatoren wurden verfeinert (Karttunen/Beesley 2005:78). Mit dem Titel Beesley/Karttunen 2003 liegt ein neueres Standardwerk vor, das die Methoden der *Finite State Morphology* anschaulich und ausführlich, wenn auch mit aussschliesslichem Fokus auf das von den Autoren (mit)entwickelte XFST-Toolkit, darlegt.

In dieser Arbeit sollen die grösseren deutschen Morphologiesysteme, die allesamt auf *Finite-State-Machine*-Technik (FSM-basiert) beruhen, vorgestellt und verglichen werden. Ein Schwerpunkt wird dabei auf die Unterschiede gelegt, durch welche die Ansätze untereinander abweichen. In einem weiteren Teil werden verbleibende Probleme der bestehenden Systeme aufgezeigt und die zur Zeit in der Forschungs besprochenen Lösungsansätze diskutiert. Ein Schlusswort rundet die Arbeit ab.

1.1 Gegenstand der Morphologie

Formal gesehen ist die Aufgabe der Morphologieanalyse eine Abbildung der Menge aller möglichen Wörter, d.h. der Potenzmenge eines Eingabealphabets Σ_I, auf die Menge aller (aus Lemmas und morphosyntaktischen Merkmalen bestehenden) Analysen, d.h. der Potenzmenge der Verknüpfung eines Ausgabealphabets Σ_O mit der Menge aller morphologischen Merkmale Σ_M. Diese Abbildung ist demnach die Teilfunktion $\Sigma_I^* \rightarrow \wp(\Sigma_O^* \cdot \Sigma_M^*)$ (Geyken/Hanneforth 2006:56). In der umgekehrten Richtung kann die Analysefunktion auch als Generator für morphologische Formen dienen.

Was nun durch diese Abbildung etwas verschleiert wird, ist der Umstand, dass sie in der Realität sehr unterschiedliche linguistische Vorgänge abdecken muss. In der Flexionsmorphologie gibt es eine Vielzahl von theoretisch und praktisch dinstinkten Phänomenen, die behandelt werden müssen, und zusätzlich erwartet man von einem elaborierten Morphologiesystem heute auch, dass es mit verschiedenen Arten von Derivation und Komposition umgehen kann. Für die Architektur eines Morphologiesystems ist es von einschneidender Bedeutung, welche theoretischen Auffassungen von Flexion und Wortbildung beim Entwurf mit in das System einfliessen. Es kann sich daher lohnen, vor der Besprechung einzelner Morphologiesysteme die Teilgebiete der Morphologie und insbesondere deren terminologische Abgrenzungen etwas zu beleuchten.

1.2 Terminologische Abgrenzungen

Man hat unter Morphologie traditionellerweise in erster Linie Flexionsmorphologie verstanden (Hacken/-Lüdeling 2002:68 bemängelnd). Derivation und Komposition haben deutlich weniger Prominenz in frühen Systemen, wie z.B. ein Blick in Antworth 1990 verrät. Die Gründe dafür sind wohl zweifältig: (1) haben sich ein beträchtlicher Teil der frühen Bestrebungen im englischsprachigen Raum abgespielt, wo Derivation und insbesondere Komposition im täglichen Sprachgebrauch weniger prominente Phänomene sind als anderswo. Was die Komposition im Englischen angeht, ist allerdings nicht ganz klar, ob sie wirklich weniger wichtig ist als z.B. im Deutschen, denn ob *computer science* als ein Nominalkompositum analysiert werden soll, ist diskutierbar. Tatsache ist jedenfalls, dass ein englischsprachiges computerlinguistisches System ohne weiteres damit auskommt, *computer* und *science* morphologisch als einzelne Nomen zu klassifizieren, und die Bestimmung der Abhängigkeit zwischen diesen Nomen der Syntaxanalyse zu überlassen. So gesehen fährt man also im Englischen auch dann gut, wenn auf morphologischer Ebene keine Komposita erkannt werden, obwohl die meisten Linguisten *computer science* wahrscheinlich übereinstimmend als Kompositum klassifizieren würden. (2) Derivation und Komposition wurden eventuell darum als eine Art "Anhängsel" der Flexionsmorphologie betrachtet, weil sie auf einer rein oberflächlichen Ebene ähnlich funktionieren: Wie bei der Flexion beinhalten diese Prozesse die Kombination von Morphemen (in der üblichen Terminologie Stämme und Affixe), und decken gewisse "Reaktionen" ab, welche durch Verbindung mancher Morpheme ausgelöst werden (Umlaute etc.). Dennoch gibt es wesentliche Unterschiede zwischen diesen Bereichen der Morphologie, die nun in einem kurzen Überblick erläutert werden sollen.

1.2.1 Flexion - Derivation - Komposition

Flexion:

- bildet Wortfomen zu *einem* Lemma (Lexem) → hat keinen Einfluss auf das Lexikon
- die Wortart bleibt erhalten
- die Semantik des Wortes ändert sich nicht, nur dessen morphosyntaktische Eigenschaften
- ist "automatisch"[1] insofern, als jedes Wort einer flektierenden Klasse auch nach allen üblichen Kategorien flektiert werden kann. Ausnahmen dazu sind begrenzt und vorwiegend semantisch bedingt (so hat z.B. das Pluraletantum *Eltern* keinen Singular oder das Modalverb *müssen* keinen Imperativ)

Derivation:

- bildet neue Lemmas (Lexeme) aus vorhandenem Wortmaterial → vergrössert das Lexikon
- das entstandene Wort gehört meistens einer anderen Wortart an
- die Semantik des neuen Lexems ist bis zu einem gewissen Grad vom Ableitungsmuster her vorhersehbar
- die Ableitungssuffixe sind generell unfreie Morpheme (treten nicht als eigenständige Wörter auf)
- unterliegt starken Restriktionen, was die Bildbarkeit zu bestimmten Lexemen betrifft. Diese Restriktionen sind schwierig zu fassen und hängen von einer Vielzahl von Faktoren ab. Dazu zählen: (1) semantische Faktoren: Das Diminutivsuffix *-chen* tritt beispielsweise nur an Nomen für konkrete Gegenstände und Lebewesen an; *un-* kann nicht bereits negierenden Lexemen präfigiert werden (**unillegal*); (2) Blockierungen: **fraulich* wird von *weiblich* blockiert; (3) Herkunft des Lexems: Die Ableitung *-är* > *-arist* (wie in *monetär –Monetarist*) tritt nur an

[1] Unglücklicherweise wird dies in der englischsprachigen Literatur öfters *produktiv* genannt (Sproat 1992:24, Jurafsky/Martin 2000:58). Den Terminus der *Produktivität* sollte man m.E. besser auf Wortbildungsmuster einschränken, um Verwechslungen zu vermeiden.

"klassische" Stämme an; (4) bestehende Morphemstruktur (vorhandene Suffixe) der Ableitungsbasis: *gelb* > *gelblich*, aber nicht **eisernlich* < *eisern*[2]; (5) syntaktische Eigenheiten der Basis: *-bar* tritt nur an transitive Verben an.

Komposition:

- bildet neue Lemmas (Lexeme) aus vorhandenem Wortmaterial → vergrössert das Lexikon
- das entstandene Wort übernimmt Wortart und morphosyntaktische Eigenschaften des Kopfes (Hinterglied)
- die Semantik des neuen Wortes ist bis zu einem gewissen Grad aus den Einzelbedeutungen der komponierten Lexeme verständlich
- die Komponenten eines Kompositums sind eigenständige Wörter, die auch als Simplexe auftreten können
- ist ebenfalls restringiert, aber nicht so stark wie die Derivation; dies vorwiegend nach semantischen Kriterien

Die Aufstellung legt den Schluss nahe, dass Derivation und Komposition gar nicht so eng mit der Flexion verwandt sind, wie es scheint. Ob die Zusammenfassung von Flexion, Derivation und Komposition unter dem Label *Morphologie* terminologisch wirklich eine gute Wahl war, kann man also in Frage stellen. In diesem Zusammenhang mag es auch erwähnenswert sein, dass in der historischen Sprachwissenschaft unter Morphologie meistens nur Flexionsmorphologie verstanden wird, und diese der Wortbildung als Überbegriff für Derivation und Kompositon entgegengesetzt wird.

Ein Argument, das *für* eine gewisse Nähe zwischen diesen Teilgebieten spricht, ist der Umstand, dass es sowohl zwischen Flexion und Derivation sowie zwischen Derivation und Komposition Grenzfälle gibt, die weder ganz in den einen noch ganz in den anderen Bereich fallen, wie ich sogleich erläutern möchte.

Grenzfälle zwischen Flexion und Derivation Das Paradebeispiel für ein Phänomen, das zwischen Flexion und Derivation steht, ist die *Steigerung der Adjektive*. Traditionellerweise wird die Steigerung als ein Flexionsphänomen behandelt. Allerdings gibt es ein triftiges Argument dafür, dass es sich um einen Wortbildungsprozess handelt: Gesteigerte Adjektive werden zusätzlich flektiert (*das niedlichere Kücken - die niedlicheren Kücken*), was einen Lexemstatus andeutet. Für Steigerung als Flexionsprozess können folgende Argumente sprechen: (1) Wortart bleibt erhalten (kein zwingendes Argument); (2) Flexion ist "automatisch", d.h. jedes Adjektiv kann gesteigert werden. Ausnahmen sind (wie oben bei der Flexion erwähnt) selten und semantisch bedingt (*der *pensioniertere Bauer*).

Ein zweiter Grenzfall sind die *Partizipien des Präsens und des Präteritums*. Auch Standardgrammatiken sind bei der Klassifikation nicht immer schlüssig, wie z.B. der Duden 2005:345 mit der Feststellung "Partizipien verhalten sich teilweise wie Adjektive" zeigt. Ähnlich sehen dies Gallmann/Sitta 2001:35f. und 69f., die schreiben: "Oft besteht zwischen echten Adjektiven und adjektivischen Partizipien keine scharfe Grenze". Nun spricht einiges dafür, dass sich Partizipien nicht nur wie Adjektive verhalten, sondern tatsächlich Adjektive *sind*:

- beide Arten von Partizipien flektieren wie Adjektive (*ein glucksendes Huhn - viele glucksende Hühner*; *ein geröstetes Hähnchen - viele geröstete Hähnchen*)
- die Präsenspartizipien können adverbial verwendet werden (*glucksend Körner picken*)

Für die Zugehörigkeit der Präteritumspartizipien zum Verbalparadigma spricht jedoch, dass sie heute überwiegend in periphrastischen Verbalkonstruktionen verwendet werden (*habe das Hähnchen geröstet*), wodurch sie von den Sprechern tendenziell eher als Verbalformen wahrgenommen werden.

[2] In *eisern* steckt bereits ein adjektivisches Ableitungssuffix – es ist eine Stoffableitung zu *Eisen*.

Grenzfälle zwischen Derivation und Komposition Viele komplexe Verben stellen eine Mischform zwischen Derivation und Komposition dar. Bildungen wie *auffüllen* oder *hineinlegen* sind als Komposita verständlich, da die "Präfixe" noch deutlich als Präpositionen, Richtungsadverbien etc. identifizierbar sind – sie sind also aus freien Morphemen aufgebaut. Bei vielen Verben sind zudem die Präfixe abtrennbar, d.h. deren Status als freie Morpheme können unmittelbar an Texten aufgezeigt werden. Bei anderen komplexen Verben hingegen, wo das Präfix stärker mit dem Verbstamm verschmolzen, also das Verbalkompositum bereits als ganzes lexikalisiert ist, kann dieses nicht mehr abgetrennt werden, und man kann synchron auch nicht mehr von einem Kompositum sprechen (z.B. *erholen*, *entdecken*). Zwischen diesen Vorgängen eine Grenze zu ziehen, ist ausgesprochen schwierig.

Die Verwicklungen zwischen Derivation und Komposition können verständlich gemacht werden, wenn man sich in Erinnerung ruft, dass viele Derivationsmuster historisch gesehen aus bestimmten Kompositatypen entstanden sind; so z.B. die Adjektivableitung mit *-lich*, die auf Verbindungen mit dem althochdeutschen Substantiv *līh* 'Körper' zurückgehen (Erben 2000:148).

1.2.2 Produktivität - Motiviertheit

Bei der Spezifikation eines Morphologiesystems muss man sich also darüber klar werden, wie viele dieser Phänomene man tatsächlich abdecken will. Hier gibt es unterschiedliche Strategien. Zum einen kann man versuchen, eine möglichst weitgehende Analyse der Wortformen zu ermöglichen. Dies bedeutet, eine grosse Anzahl an Derivations- und Kompositionstypen zu implementieren, mit deren Restriktionen (und Möglichkeiten, diese Restriktionen umzusetzen), man sich einzeln zu befassen hat. Am anderen Ende der Skala steht die Strategie, nur soviel umzusetzen, wie unbedingt nötig ist. Das Schlüsselwort in diesem Zusammenhang ist die *Produktivität* von Derivations- und Kompositionsmustern. Viele Systeme versuchen lediglich, die produktiven Prozesse abzudecken, die in einer Sprache wirklich lebendig sind, nicht aber diese, deren Ableitungen bereits lexikalisiert sind. Dies geschieht in Übereinstimmung mit der Absicht, mit Wortbildungsmechanismen in erster Linie die Erkennung von "ungesehenen" Wörtern, d.h. solchen, die noch nicht im Lexikon des Morphologiesystems vorhanden sind, zu ermöglichen, nicht aber, bereits im Lexikon erhaltene Formen weiter zu "dekomponieren". Beispielsweise kann man sich durchaus fragen, ob es Sinn macht, das Substantiv *Ankunft* in *An-kunft* zu zerlegen oder sogar auf das Verb *kommen* zu beziehen. Obwohl dies historisch sicher richtig ist, gehört *Ankunft* wohl als eigenes Lemma ins Lexikon, und braucht nicht weiter analysiert zu werden. Für die Erkennung neuer Wörter kann der Bezug von *Ankunft* auf *(an)-kommen* keinen Dienst leisten.

Die meisten computerlinguistischen Morphologiesysteme zielen dabei in die zweite Richtung und implementieren nur produktive Wortbildungsmuster. Nun unterscheiden sich aber in der Fachliteratur die Ansichten darüber, was denn unter Produktivität genau zu verstehen ist (vgl. Hacken/Lüdeling 2002:66: "Productivity is a difficult and controversial concept"). Ein Grund dafür mag sein, dass in der Wortbildungsliteratur m.E. oft nicht genau zwischen *Produktivität* und *Motiviertheit* (*Durchsichtigkeit*) von Bildungen unterschieden wird. Der Unterschied liegt darin, dass *motivierte* Bildungen für die Sprecher zwar noch durchsichtig, also als Ableitungen erkennbar sind, das zugrundeliegende Muster jedoch nicht mehr für Neuprägungen verwendet wird. *Produktive* Muster hingegen bilden bis dato unbekannte Lexeme, d.h. ad-hoc Bildungen und wirkliche Neuprägungen. In Titeln wie Fleischer/Barz 1995, Erben 2000 oder Booij 2005 werden produktive und motivierte Bildungen oft undifferenziert behandelt. Man vgl. hierzu z.B. Fleischer/-Barz 1995:54f, wo an "nicht mehr [...] produktiven Bildungen", die "durch gespeicherte Wortbildungsprodukte im Lexikon vertreten sind", lediglich implizite Derivate (Bsp. *Wurf, Flug*) und einige verdunkelte Präfixverben (*ob-siegen*) und Adjektivableitungen (*tör-*

icht) genannt werden. Gleichzeitig enthält deren Liste mit Ableitungssuffixen (ab S. 146) viele Suffixe, bei denen es zumindest fraglich ist, ob sie wirklich noch zu Neuprägungen fähig sind und somit in einem Morphologieanalysesystem implementiert werden sollen. S. 170 nennen Fleischer/Barz z.B. die Bildung auf *-sel* (*Stöpsel*, *Mitbringsel*) explizit "produktiv", obwohl man sich kaum vorstellen kann, dass heute noch eine Neuprägung nach diesem Muster vorgenommen werden könnte.

Natürlich kann man die Abstufung zwischen produktiven und (un)motivierten Bildungen noch weiter verfeinern, was auch gemacht wird (vgl. Glück 2000 unter *Motiviertheit* und *Produktivität*), doch muss (darf) der Computerlinguist/die Computerlinguistin vielleicht eine etwas pragmatischere Haltung einnehmen. Für das Design eines Morphologiesystems ist auf jeden Fall an einer (möglicherweise etwas willkürlichen) Stelle eine Abgrenzung zwischen produktiven und motivierten Bildungen zu schaffen. Eine vernünftige Strategie scheint mir, nur erstere als Regeln zu implementieren, und bei letzteren eine möglichst umfassende Aufnahme ins Lexikon anzustreben.

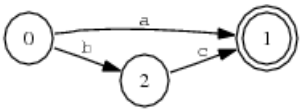
2 Bestehende Ansätze

FSM-basierte Morphologieanalysesysteme sind gegenüber solchen, die auf andere Techniken setzen, relativ klar in der Überzahl. Dies rechtfertigt ihre prominente Platzierung in diesem Kapitel und eine knappe Einführung in die Grundlagen des Ansatzes.

2.1 FSM-basierte Systeme

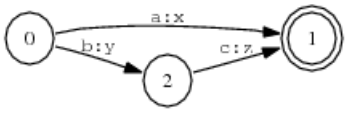
2.1.1 Grundlegendes zu FSM-Systemen

Endliche Automaten bestehen aus Knoten, von denen einer als Startknoten und mindestens einer als Endknoten markiert ist, sowie aus beschrifteten, gerichteten Kanten zwischen diesen Knoten. Die Knoten bezeichnen die Zustände des Automaten und die Kanten stellen die Übergänge zwischen den Zuständen dar. Die Menge der Kantenbeschriftungen liefert das Alphabet *Sigma* des Automaten. Jede Möglichkeit, vom Startzustand in einen Endzustand zu gelangen, bezeichnet einen Pfad durch den Automaten. Notiert man die Kantenbeschriftungen entlang eines Pfades, erhält man ein Wort, das vom Automaten akzeptiert wird. Die Menge aller Wörter, die man auf diese Weise – durch beschreiten aller Pfade durch den Automaten – erhalten kann, bilden zusammen die Sprache, welche der Automat akzeptiert. Ihre Eigenschaften sind die einer **regulären Sprache**.



Ein endlicher Automat.

Transduktoren unterscheiden sich von Automaten dadurch, dass sie als Kantenbeschriftungen nicht nur einfache Alphabetszeichen besitzen, sondern Paare von Zeichen. Dadurch bekommt der Transduktor zwei "Seiten", d.h. eine obere, wozu die Zeichen auf der linken Seite der Kantenbeschriftungen zählen, sowie eine untere, wozu die Zeichen auf der rechten Seite der Kantenbeschriftungen zählen. Dies führt dazu, dass Pfade nicht mehr Wörter beschreiben, sondern Relationen zwischen Wörtern. Die Menge der oberen Wörter bildet so eine separate Sprache im Vergleich zur Sprache der unteren Wörter. Transduktoren beschreiben folglich nicht eine Sprache, so wie Automaten, sondern eine Relation zwischen zwei Sprachen. Man kann sich einen Transduktor als einen Übersetzer denken, der eine Sprache in eine andere umzusetzen vermag, und zwar sowohl in die eine wie auch in die andere Richtung.



Ein Transduktor.

Reguläre Ausdrücke sind eine Notationsform für reguläre Automaten. Automaten können maschinell

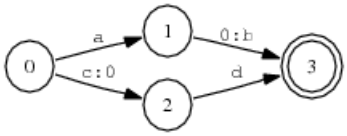
aus solchen regulären Ausdrücken kompiliert werden. Automaten wiederum akzeptieren Sprachen. Sprachen können ihrerseits mit regulären Ausdrücken beschrieben werden. Die drei Konzepte, reguläre Ausdrücke, Automaten und Sprachen, sind somit ineinander überführbar und darum in einem gewissen Sinn äquivalent.

Als Beispiel kann der erste Automat aus 2.1.1 dienen. Er wird durch den regulären Ausdruck `[a|b c]` beschrieben und akzeptiert die Sprache {`"a", "bc"`}.

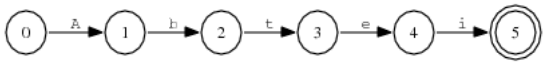
Zu den Merkmalen endlicher Automaten zählt, dass sie entweder **deterministisch** oder **nicht-deterministisch** sind, je nachdem, ob sie Knoten enthalten, von denen mehrere Kanten mit der gleichen Beschriftung wegführen. Falls dies der Fall ist, spricht man von nicht-deterministischen Automaten; ansonsten von deterministischen. Etwas einfacher formuliert bedeutet dies, dass ein Automat an einer bestimmten Stelle nicht "eindeutig" sein kann. Ein nicht-deterministischer Automat kann algorithmisch in einen determinierten umgewandelt werden. Im XFST-Toolkit (mehr dazu später) gibt es für diesen Vorgang sogar einen eigenen Befehl: `determinize net` (vgl. Beesley/Karttunen 2003:74f u. 195).

Automaten gelten als **zirkulär**, sobald sie eine unendliche Sprache akzeptieren. Ein simpler, zirkulärer Automat kann mit dem regulären Ausdruck `[a*]` definiert werden:

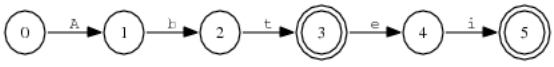
Eine weitere wichtige Eigenschaft von Automaten ist das Vorhandensein von **Epsilon-Übergängen**. Unter einem Epsilon-Übergang versteht man eine Kante, die keine Zeichen aus dem Alphabet trägt, also einen leeren Übergang symbolisiert. Bei Transduktoren sind Kantenbeschriftungen der Art ϵ`:a` bzw. `a:`ϵ möglich, die dazu verwendet werden, zusätzliche Zeichen einzufügen oder zu löschen. Eine wichtige Tatsache ist dabei, dass die Wortpaare einer Relation sich aufgrund von Epsilon-Übergängen in der Zeichenlänge unterscheiden können. Dies hat Konsequenzen für die Operationen, die für die Relationen anwendbar sind: Nur Relationen, bei denen alle Wörter auf der Unter- und der Oberseite gleich lang sind, sind unter Schnitt, Subtraktion und Komplementbildung abgeschlossen (Beesley/Karttunen 2003:55). Ein Automat mit Epsilon-Übergängen, der die Relation {`<a, ab>, <cd, d>`} beschreibt, sieht so aus (0 steht in der Grafik für ϵ):



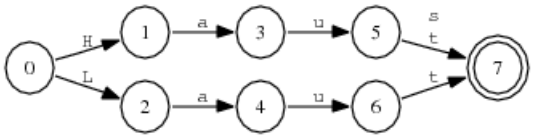
Da endliche Automaten algorithmisch minimiert werden können, zählt es zu ihren Eigenschaften, dass sie auf erstaunlich kleinem Raum eine **grosse Datendichte** erreichen. Dadurch, dass Teilstrukturen zu verschiedenen Pfaden gehören und somit "geteilt" werden können, wachsen die Automaten beim Vergrössern ihrer Abdeckung nicht unbedingt im erwarteten Mass an. Als Beispiel betrachte man den Automaten, der die Sprache {"Abtei"} erkennt:



Erweitert man diesen so, dass er zusätzlich das Wort "Abt" erkennt, so stellt sich heraus, dass weder ein zusätzlicher Zustand noch eine zusätzliche Kante eingefügt werden muss. Es genügt, lediglich einen Zustand als zusätzlichen Endpunkt zu markieren:



Zu welch unvorhergesehenen Effekten die automatische Minimierung führen kann, mag an folgendem Beispiel erläutert werden. Der Automat für die Sprache {"Haus", "Haut", "Laut"} sieht folgendermassen aus:



Fügt man diesem zusätzlich die Fähigkeit hinzu, das Nomen "Laus" zu erkennen, führt dies zu folgendem Automaten:

Dieser Automat erkennt die Sprache {"Haus", "Haut", "Laut", "Laus"}. Man beachte, dass der zweite Automat tatsächlich kleiner ist als der erste, obwohl er mehr Wörter erkennt. Während der erste Automat noch 7 Zustände und 8 Kanten besass, kommt der zweite mit 4 Zuständen und 6 Kanten aus. Der Grund liegt darin, dass die Buchstabenfolgen *au* im ersten Fall nicht zusammengelegt werden können, da nach dem *u* nur ein *s* kommen darf, wenn am Anfang ein *H* gestanden hat. Dies ist eine Fernabhängigkeit, die mit endlichen Automaten grundsätzlich nicht modelliert werden kann – weshalb die Teilstruktur *au* in diesem Fall nicht fusioniert werden kann. Im zweiten Beispiel ist in allen Fällen ein Ausgang auf *s* oder *t* möglich, egal ob das Wort mit *H* oder *L* begonnen hat. Deshalb kann die Unterscheidung der zwei Fälle aufgegeben und die Teilstruktur *au* zusammengelegt werden.

Da Sprachen und Relationen Mengen sind, können auch die gebräuchlichen **Mengenoperationen** wie Vereinigung, Schnitt, Komplementbildung etc. auf sie angewendet werden. Einschränkungen dazu sind im obigen Abschnitt zu den Epsilon-Übergängen bereits erwähnt worden. Zusätzliche Operationen, die mit Transduktoren möglich sind (und in der Praxis eine grosse Rolle spielen) sind die Komposition, bei der die untere Seite eines Transduktors mit der oberen Seite eines zweiten verknüpft wird, sowie die Ersetzungsoperatoren, die bedingte Ersetzungen mit Restriktionen (wahlweise) auf der Ober- oder Unterseite des Transduktoren zulassen (Beesley/Karttunen 2003:28 und 132ff.).

2.1.2 Finite-State Toolkits

Es gibt eine ganze Bandbreite an FSM-Toolkits, die das Erstellen und Bearbeiten von endlichen Automaten ermöglichen. Die Zahl der kommerziellen FSM-Toolkits übersteigt mittlerweile zehn, und freie/fast freie Lösungen gibt es sogar noch mehr[3]. Neben XFST und SFST ist mir jedoch keine Library bekannt, die direkt im Hinblick auf morphologische Anwendung konstruiert worden wäre (obwohl man natürlich auch mit diesen zwei einiges mehr anfangen kann). Ein bekanntes FSM-Toolkit, das mit *gewichteten* (engl. *weighted*) Automaten umgehen kann und für Verarbeitung von gesprochener Sprache eingesetzt wird, ist die *AT&T FSM Library*[4].

XFST - Xerox Finite State Tools Die XFST[5] Tools wurden am *Palo Alto Research Center* (PARC) der Firma Xerox entwickelt und stellen ein mächtiges, weit gereiftes Werkzeug dar. Ausführlich beschrieben werden die Tools in Beesley/Karttunen 2003. Zum Inventar zählen eine interaktive bzw. skriptbare `xfst`-Shell, wo on-the-fly Automaten definiert, kompiliert, verarbeitet, getestet und ausgegeben werden können. Für den Lexikographen bietet das Tool `lexc` die nötige Funktionalität. Wer mit klassischen Zwei-Ebenen-Regeln arbeiten will, kann dafür `twolc`, den Two-Level-Compiler, einsetzen. Mit den XFST Tools wurden schon verschiedenste grosse Morphologiesysteme realisiert. Die Programme sind zu nicht-kommerziellen Zwecken mit gewissen Einschränkungen (mit dem Buch zusammen) erhältlich und werden für kommerzielle Anwender lizensiert. Implementiert wurde XFST in der Sprache C.

[3] Eine mehr oder weniger aktuelle Liste ist unter `<http://www.ling.helsinki.fi/events/FSMNLP2005/cfp.shtml>` einsehbar.

[4] `<http://www.research.att.com/~fsmtools/fsm/>`

[5] `<http://www.fsmbook.com/>`

SFST - Stuttgart Finite State Transducer Tools Bei SFST[6] handelt es sich um ein Toolkit, das an der Universität Stuttgart entwickelt worden ist. Im Gegensatz zu XFST ist es freie Software und kann ohne Restriktionen verwendet werden. Leider bietet SFST aber nicht den vollen Funktionsumfang von XFST und ist auch in der Handhabung etwas weniger elaboriert. Immerhin bietet SFST in der neusten Version auch Ersetzungsoperatoren an, wie sie von XFST her schon bekannt sind. Ein ganz knapper Abriss über die Programmiersprache von SFST ist in Schmid 2006 nachzulesen. SFST ist in C++ programmiert.

2.1.3 GERTWOL

GERTWOL hat als erste deutschsprachige Morphologie einen gewissen Bekanntheitsgrad erreicht. Das System wurde in den Neunzigerjahren von der finnischen Firma Lingsoft, Inc. entwickelt. Es handelt sich um eine traditionelle Implementation der Two-Level Morphologie, so wie sie von der gleichen Firma auch für Englisch (ENGTWOL), Finnisch (FINTWOL) und Schwedisch (SWETWOL) realisiert wurde. Sie verwendeten dazu die damals verfügbaren Two-Level Tools von Xerox. Einiges zum Aufbau des Systems ist in den Artikeln Haapalainen/Majorin 1994, Haapalainen/Majorin 1995 und Koskenniemi/Haapalainen 1996 nachzulesen. Auf der Homepage von Lingsoft, Inc.[7] findet man eine Online-Demo von GERTWOL - die Produkte der Firma werden zu Forschungszwecken zugänglich gemacht, und ansonsten kommerziell vertrieben.

GERTWOL besteht aus einem Lexikon und einer Menge von morphophonemischen Regeln. Das Lexikon enthält alle Morpheme, sowie die Informationen darüber, auf welche Weise diese Morpheme miteinander verknüpft werden können. Mit den Regeln werden nicht-konkatenative Phänome wie der Umlaut behandelt. Leitgedanke der Designer war es von Anfang an, besonders unregelmässige und strikt unproduktive Alternationen nicht mit Regeln zu behandeln, sondern direkt ins Lexikon aufzunehmen (Koskenniemi/-Haapalainen 1996:121).

Die Entwickler von GERTWOL haben sich entschieden, nur eine kleine Anzahl an Flexionsklassen ins System aufzunehmen. Für schwache Verben gibt es 12 Klassen, für starke ebenfalls 12, für Substantive 10 und für Adjektive sind es deren 17. Das sind auffallend wenige Klassen, besonders bei den Substantiven. GERTWOL kennt jedoch zu vielen Klassen noch Unterklassen, was gerade bei den Substantiven ins Gewicht fällt. Die Unterscheidung in Hauptklassen erfolgt nämlich bei den Substantiven aufgrund der Pluralbildung, und wenn man die weiteren Merkmale Umlaut, Genitivendung, optionales *e* im Dativ Singular und Konsonantengemination hinzunimmt, ergeben sich beinahe 300 Klassen (Haapalainen/Majorin 1994).

Diese Klasseneinteilung hat natürlich Auswirkungen darauf, wie die Lexikoneinträge auszusehen haben. Die genannten Merkmale müssen – da sie wie gesagt nicht in der Klassenkennzeichnung enthalten sind – bei jedem Lexikoneintrag dazugeschrieben werden. Ein Beispielseintrag für die Substantive kann z.B. `Phosphat S1(=s/es)/nt;` lauten, wobei `S1` die Flexionsklasse, `s/es` die Genitivbildung, `=` die Restriktion des Dativ-*e*, und *nt* das Genus anzeigt. Umlaut würde mit einem `+` und Gemination mit einem `*` im Lexikon markiert.

Den Ablaut behandelt GERTWOL nicht wie den Umlaut als morphophonemische Regel, sondern codiert die Stammformen direkt ins Lexikon. Ein Beispieleintrag: `laufen U3: U3 lauf läuf lauf lief lief lauf`.

Die Gestaltung der Lexikoneinträge ist eine wichtige Eigenschaft eines Morphologiesystems, weil sie einen entscheidenden Einfluss darauf hat, auf welche Art und Weise das Lexikon später aufgefüllt werden kann und welche bestehenden Ressourcen dazu eingesetzt werden können. Dies ist früh genug zu bedenken,

[6] <http://www.ims.uni-stuttgart.de/projekte/gramotron/SOFTWARE/SFST.html>

[7] <http://www2.lingsoft.fi/cgi-bin/gertwol>

da man sich bereits beim Entwurf auf ein Lexikonformat festlegen muss, das anschliessend kaum mehr geändert werden kann.

Wie viel Derivation GERTWOL genau macht, bleibt manchmal unklar. Die von GERTWOL gelieferte Analyse `"her|aus|finden"` z.B. gibt keine Auskunft darüber, ob die Präfixe *her-* und *aus-* als produktive Ableitungen erfasst worden sind, oder ob das Lemma schon mit diesen Präfixen im Lexikon stand. Nach Haapalainen/Majorin 1994 ist eine beträchtliche Anzahl von Präfigierungsmustern und sogar teilweise eine Doppelpräfigierung von Verben möglich. Bei den Substantiven sind beispielsweise die Ableitungen mit *-ung* (Abstrakta) und *-chen* (Diminutiva) implementiert. Dazu kommen einige Ableitungsmuster, die nur bei bestimmten Suffixen greifen: Etwa die Movierung von Nomina Agentis mit *-in* (*Lehr-er-in*) oder die Adjektivisierung von Personenbezeichnungen auf *-ist* mit *-isch* (*athe-ist-isch*). Auch einige Konversionen (Ableitungen ohne morphologische Kennzeichnung) werden gemacht: *(das) Laufen*, *(das) Schönste* etc.

Für die Bildung von Komposita nimmt GERTWOL an, dass die Fugenelemente flexionsklassenspezifisch sind. Diese Annahme liegt schon darum nahe, weil die Fugenelemente grösstenteils Flexionselemente (Gen. Sg. oder Pl.-Endungen) sind, wie z.B. durch die Gegenüberstellung von *das Licht des Tages* und *das Tageslicht* deutlich wird (Erben 2000:69). Ausnahmen, die es dazu gibt, sind in der Regel durch analogische Übertragung eines solchen Flexionselements auf ein Substantiv einer anderen Klasse zu erklären (*Meinungs-umfrage*, obwohl der Gen. Sg. nicht **Meinungs* lautet). Die Kompositionstypen, von denen GERTWOL solche mit Nomen und solche mit Adjektiven im Hinterglied kennt, werden, soweit dies aus den Forschungsberichten hervorgeht, nicht eingeschränkt. Nach diesen Mustern lässt GERTWOL also alle denkbaren Komposita zu.

Über die Abdeckung von GERTWOL liegen mir nur Zahlen von 1994 vor. Damals umfasste das Lexikon inklusive Eigennamen und Abkürzungen ca. 80'000 Einträge. Die Verarbeitungsgeschwindigkeit liegt bei ca. 2000 Wortanalysen pro Sekunde (Koskenniemi/-Haapalainen 1996:139).

2.1.4 SMOR

SMOR ist ein deutsches Morphologiesystem, das von Schmid et al. 2004 vorgestellt wurde. Es setzt für die Implementierung auf die SFST-Tools und deckt Flexion, Derivation und Komposition ab. Leider liegen ausser der einen, kurzen Publikation von 2004 keine weiteren Informationen zum System vor, und es ist nicht bekannt, ob es derzeit noch weiter entwickelt wird. Eine ältere Version von SMOR mit einigen beispielhaften Lexikoneinträgen ist mit den SFST-Tools erhältlich; der Rest ist, so weit ich weiss, nicht zugänglich.

SMOR beansprucht für sich, im Gegensatz zu GERTWOL und WordManager, die hauptsächlich auf ein umfassendes Lexikon setzen würden, das einzige deutsche Morphologiesystem zu sein, das gleichzeitig produktive Wortbildung und Flexionsmorphologie abdecke (Schmid et al. 2004:1). Diese Behauptung mag in bezug auf WordManager zutreffen, doch scheint sie mir gegenüber GERTWOL nicht ganz nachvollziehbar, da GERTWOL ebenfalls beide Komponenten mitbringt.

Um die Flexion zu behandeln, setzt SMOR wie die klassischen Two-Level-Systeme auf einen konkatenativen Ansatz mit Fortsetzungsklassen. Restriktionen bei der Anfügung von Affixen werden mittels Merkmalen und Filtern gemacht, d.h. die Affixe werden bereits im Lexikon mit Merkmalen versehen, deren Übereinstimmung später mit Regeln, die als Filter dienen, erreicht werden kann. Für morphophonemische Phänomene werden Two-Level-Regeln eingesetzt.

Stärker noch als bei GERTWOL werden die Lemmas im Lexikon von SMOR mit Meta-Informationen angereichert. Diese beziehen sich zum Teil auf den morphologischen Status des Eintrags (Stamm, Suffix, Präfix), auf die Wortart, die Herkunft des Wortes (heimisch, fremd, klassisch), auf die Art des Stammes (Basis, Ableitung, Kompositum), auf die Komple-

xität (Simplex, deriviert mit Suffix, deriviert mit Präfix) und zuletzt auch einem Kürzel, das die Flexionsklasse angibt. Hierin geht SMOR also grundsätzlich andere Wege als GERTWOL, indem mit jedem Eintrag im Lexikon eine beträchtliche Menge an Zusatzinformation mitgeliefert wird. Dies hat zwei unmittelbare Konsequenzen: (1) Der Aufwand für die Erstellung des Lexikons ist beträchlich höher. Bestehende lexikalische Ressourcen müssen höchstwahrscheinlich zuerst mit diesen Informationen angereichert werden, bevor sie ins Lexikon von SMOR aufgenommen werden können, und falls dies von Hand gemacht werden muss, kommt dies einem enormen Aufwand gleich. (2) Dem System stehen für die Flexion, aber insbesondere für die Wortbildung, z.B. durch die Angabe der Herkunft der Stämme, wichtige Informationen zur Verfügung, die in GERTWOL ganz fehlen. SMOR kann also Restriktionen, die bei der Wortbildung greifen, aufgrund der Meta-Informationen aus dem Lexikon fassen. Gewisse Ableitungen können beispielsweise nur für "klassische" Stämme erlaubt werden. GERTWOL hat keine Möglichkeit, eine solche Bildung zu restringieren, und man muss sich damit abfinden, dass in diesen Fällen stark übergeneriert wird.

Über Zustandekommen, Gliederung und Anzahl der Flexionsklassen liegen keine Informationen vor. Ferner geht aus der Publikation Schmid et al. 2004 auch nicht hervor, wie gross das Lexikon von SMOR ist, oder wie es aufgebaut wurde. Man darf aber annehmen, dass ein grosses Lexikon vorhanden ist, da SMOR in einem Evaluationstest für 66.8% eines Korpus von 80 Millionen Token eine Analyse geliefert hat.

2.1.5 WordManager / canoo.net

WordManager, ein in Hacken/Lüdeling 2002 beschriebenes System, ist in erster Linie eine lexikalische Ressource, obwohl es von den Autoren als "System für morphologische Wörterbücher" (*system for morphological dictionaries*) angepriesen wird. Entstanden ist WordManager unter der Leitung von Domenig an den Universitäten Basel, Amsterdam und Lugano. WordManager kann eigentlich nur halb als FSM-basierter Ansatz gelten, da es zu einem wesentlichen Teil aus einer lexikalischen Datenbank besteht. Da Teile des Systems dennoch mithilfe von Finite-State-Tools realisiert worden sind (Hacken/Lüdeling 2002:78), habe ich mich für die Einreihung an dieser Stelle entschieden.

Aufgrund des Aufbaus von WordManager ist das System für Wörter, die nicht im Lexikon sind, auf einen separaten *Guesser* (Originalname: *Unknown Word Analyser*) angewiesen. Es handelt sich also bei WordManager nicht um ein komplettes Morphologiesystem, zumal auch – so weit ich sehe – keine Generierung von Wortformen möglich ist. Damit erübrigen sich auch die Fragen danach, auf welche Weise einzelne Phänomene wie Umlaut oder Ablaut bzw. gewisse Derivationsmuster behandelt werden, denn die Antwort ist, dass alles im Lexikon codiert ist.

WordManager umfasste im Jahr 2002 ingesamt 200'000 deutsche Lexeme, womit es GERTWOL und möglicherweise auch SMOR deutlich übertrifft. WordManager kann auf der Webseite von canoo.net[8] ausprobiert werden.

2.1.6 TAGH

Obwohl sich TAGH schon seit 5 Jahren in Entwicklung befindet, ist bisher nur eine Publikation (Geyken/Hanneforth 2006) erschienen. Das System basiert auf der Potsdamer FST Bibliothek und deckt Flexion, Derivation und Komposition ab. Soweit die Architektur des Systems von Geyken/Hanneforth dargelegt wird, ist sie am ehesten mit derjenigen von SMOR vergleichbar. TAGH setzt ebenfalls auf stark annotierte Lexikoneinträge und behandelt Stammalternationen von der Art des Ablauts genauso wie SMOR im Lexikon. TAGH bringt aber im Gegensatz zu den anderen Systemen einige Neuerungen, die deut-

[8] <http://www.canoo.net>.

lich weiter gehende Einschränkungen von Derivations- und Kompositionsmustern ermöglichen, als das bis dato der Fall war. Dies sind im Wesentlichen die folgenden zwei Punkte:

- mit TAGH ist im deutschsprachigen Raum erstmals eine Morphologie mithilfe von gewichteten Automaten realisiert worden. Damit kann insbesondere eine Priorisierung von verschiedenen Analysen erreicht werden. Wie dies genau funktioniert, wird in Abschnitt 4.1 thematisiert.
- TAGH ist die erste mir bekannte Morphologie, die eine Semantikkomponente beinhaltet. Mithilfe von Konzepteen, die aus LexikoNet gewonnenen sind, können auf diese Weise Derivations- und Kompositionsprozesse stark eingeschränkt werden. Die von mir in der Einleitung beschriebene semantische Restriktion, dass die Verkleinerung mit *-chen* nur an konkrete Gegenstände und Lebewesen antritt, kann in TAGH also tatsächlich modelliert werden.

Leider ist über TAGH sonst nicht viel bekannt. Die Autoren Geyken und Hanneforth verraten lediglich, dass TAGH bei grossen Millionen-Korpora Erkennungsraten von 98-99% erreicht. Eine Evaluation dieser Analysen haben sie allerdings bisher nicht gemacht. Über die Grösse des Lexikons und weitere Parameter von TAGH ist nichts bekannt.

2.1.7 mOLIFde

Das Morphologiesystem mOLIFde ist an der Universität Zürich in Entwicklung. Aufgrund seines inkompletten Zustandes (beschränktes Lexikon, unvollständige Derivation und Komposition, keine grossflächigen Tests) beschränke ich mich hier auf die Nennung einiger Eckdaten und auffälliger Merkmale, durch die es sich von den anderen Systemen abhebt:

- in mOLIFde werden keine Stammalternationen oder Metainformationen im Lexikon gespeichert. Die Lemmas werden lediglich mit einem einzigen Flexionscode im Lexikon abgelegt, der für jedes Wort eindeutig festlegt, wie es flektiert. Dies führt zwar zu einer Vervielfachung der Flexionsklassen, doch bringt dies durch Vereinfachung der Systemarchitektur wesentliche Vorteile.
- Derivation wird nicht zur Laufzeit gemacht. Dies geschieht aus der Überlegung heraus, dass bei der Derivation immer neue Lexeme entstehen, die als solche wiederum im Lexikon abgelegt werden müssen. Bei mOLIFde werden die Derivierungsprozesse von den Entwicklern bei der Kompilation ausgelöst, und die Resultate werden direkt zurück ins Lexikon geschleust.
- mOLIFde ist meines Wissens das einzige System, das per Abgleich mit dem "Korpus Internet" (via Suchmaschinen) unwahrscheinliche Ableitungen ausfiltert.
- mOLIFde versucht sich nach Möglichkeit an bestehende Standards (EAGLES für die Codierung von morphosyntaktischen Merkmalen; OLIF als Standard für den lexikographischen Austausch) zu halten.

2.2 Nicht FSM-basierte Systeme

In grundsätzlich verschiedene Richtungen zielen die zwei *Functional Morphology* (FM) und *Morphology Induction* genannten Vorgehen, die ganz ohne endliche Automaten oder Transduktoren auskommen. So weit ich informiert bin, sind beides bisher eher Konzeptstudien und können noch keine fertigen, vollwertigen Analysesysteme vorweisen.

2.2.1 Functional Morphology

Forsberg/Ranta 2006 bietet eine Beschreibung eines *Functional Morphology*[9] (FM) genannten Konzepts, das in der funktionalen Programmiersprache Haskell realisiert worden ist. FM unternimmt, wie die meisten Morphologiesysteme, eine Trennung zwi-

[9] <http://www.cs.chalmers.se/~markus/FM/>

schen den sprach-unabhängigen Komponenten des Systems, sowie den sprachabhängigen Komponenten, wobei besonderer Wert darauf gelegt worden ist, dass letztere auch für nicht-programmierende Linguisten leicht zu erlernen sein sollen. Im Funktionsumfang beschränkt sich FM auf die Flexionsmorphologie – Wortbildung wird nicht berücksichtigt. Für die Flexion haben die Entwickler ein von ihnen als *Wort und Paradigma* (engl. *word-and-paradigm*) bezeichnetes Konzept gewählt, gemäss welchem jedes Wort im Lexikon mit einem Zeiger auf ein bestimmtes Paradigma ausgestattet ist. Ein Vorteil von FM, der von den Autoren als Kriterium hervorgehoben wird, warum der FM eine längere Lebensdauer vorausgesagt werden könne als anderen Systemen, ist ein Feature, das den Export der Morphologie als XFST-Sourcecode bzw. in den Formaten weiterer Toolkits, oder auch in Tabellenform etc. ermöglicht. Lexika bestehen für mehrere Sprachen, wobei das grösste (Schwedisch) 20'000 Einträge umfasst.

2.2.2 Morphology Induction

Die Idee des induktiven Lernens von Morphologie ist es, in zwei Schritten die Morphologie einer Sprache automatisch aus einer Textmenge herauszulösen. Neuere Literatur zu diesem Vorgang ist Hammarström 2006 (Beschreibung eines Induktionsalgorithmus) und Creutz et al. 2006 (Vorstellung eines Induktionstools).

Der erste Schritt in einem Induktionsalgorithmus ist die Segmentierung des vorhandenen Wortmaterials in Stämme und Affixe. Für die Zerlegung der Wörter werden statistische Verfahren angewendet. Anschliessend wird mit den gewonnenen Suffixen eine nach "Wichtigkeit" (engl. *salience*) geordnete Liste erstellt. Im zweiten Schritt werden aus den Endungen "Paradigmen" erstellt, worunter lediglich Mengen von Endungen verstanden werden (Hammarström 2006:289). Dies kann so ablaufen, dass die Menge aller Endungen, die an einem bestimmten Stamm beobachtet wurden, zu einem solchen Paradigma zusammengestellt wird. Mit verschiedenen Heuristiken können so "wahrscheinliche" Paradigmen ermittelt werden.

Das Vorgehen, welches ohne Überwachung (engl. *unsupervised*) abläuft, ist aber nicht unproblematisch. Hammarström nennt das Induzieren von Paradigmen aus den Mengen von Endungen sogar "exceedingly difficult", da es zu jedem Set von Endungen eine enorme Anzahl von möglichen Paradigmen gibt, und die Paradigmen einer Sprache sich ausserdem oft überschneiden. Trotzdem sind die Resultate, wenn man z.B. die knappe Präsentation bei Creutz et al. 2006 vergleicht, eigentlich erstaunlich gut.

Aus linguistischer Perspektive muss man sagen, dass ein induktives Verfahren einige grundsätzliche Probleme hat. Zum einen eignet es sich nur für rein konkatenative Verfahren. Sobald sprachliche Prozesse wie Ablaut oder Umlaut ins Spiel kommen, welche das Aussehen der beteiligten Morpheme verändern, muss das System kapitulieren. Es ist in diesem Zusammenhang vielleicht auch erwähnenswert, dass gemäss Creutz et al. 2006 bisher vorwiegend mit Finnisch und Türkisch gearbeitet wurde – zwei agglutinierenden Sprachen, die sich durch eine besonders "konkatenative" Morphologie auszeichnen. Ein zweites Problem ist das Unvermögen, zwischen Flexion und Wortbildung zu unterscheiden. Für ein induktives System sind alles gleichwertige Morpheme, und es spielt für die Segmentierung keine Rolle, welche linguistischen Prozesse den komplexen Formen zugrunde liegen. Drittens leistet das System keine eigentliche Analyse der Wortformen, sondern nur Segmentierungen, da ihm über morphosyntaktische Eigenschaften der einzelnen Morpheme keine Informationen vorliegen.

Obwohl das Verfahren also einen grossen Vorteil bietet, nämlich dass weder die Erstellung eines Lexikons noch die Formulierung von morphophonemischen Regeln oder Wortbildungsmustern geleistet werden muss, glaube ich aufgrund der genannten Punkte nicht, dass es die FSM-basierten Ansätze ernsthaft konkurrieren kann. Vielleicht ist der direkte Vergleich mit jenen Systemen aber auch nicht ganz gerechtfer-

tigt, da die induktiven Verfahren vorwiegend in der Verarbeitung von gesprochener Sprache eingesetzt zu werden scheinen, was natürlich im Gegensatz zu den klassischen, text-basierten Verfahren eine etwas andere Ausrichtung bedingt.

3 Verbleibende Probleme und mögliche Lösungsstrategien

Bei der Besprechung der einzelnen Systeme wurde bereits einiges zu deren Stärken und Schwächen angeführt. Gewisse Probleme ziehen sich aber durch alle Ansätze hindurch. In diesem Kapitel sollen diese gemeinsamen Probleme genannt und Strategien aufgezeigt werden, wie man diese Herausforderungen angehen kann.

3.1 Nicht-konkatenative Eigenschaften von Sprachen

Ob komplexe Wörter durch eine einfache Verkettung (*Konkatenation*) von Morphemen aufgebaut werden können, ist eine inhärente Eigenschaft einer natürlichen Sprache. Da dies von Sprache zu Sprache sehr verschieden sein kann, entstehen unterschiedliche Bedürfnisse an ein Morphologiesystem, je nachdem, mit welcher Sprache man arbeitet. Zu den sogenannt nicht-konkatenativen Phänomen zählen im Deutschen z.B. der Umlaut und der Ablaut. In anderen Sprachen kommen z.B. Vokalharmonieeffekte (Finnisch), Reduplikationen (Tagalog, Malay), oder, als besondere Herausforderung, gewisse Verschachtelungseffekte vor, die aus der "Auffüllung" von Konsonantenstrukturen mit bestimmten Vokalmustern bestehen (Arabisch; engl. *interdigitation* oder *templatic morphology*; vgl. Beesley/Karttunen 2003:376ff.). Solche Phänomene stellen ein grundsätzliches Problem für die vorherrschenden FSM-basierten Systeme dar, weil sie alle mit nicht-lokalen Abhängigkeiten beschrieben werden müssen. Auf die eine oder andere Weise ist jedoch meistens eine Behandlung möglich, wobei der Aufwand je nach Lösungsansatz beträchlich schwankt. Die deutschsprachigen Systeme bieten alle eine Behandlung der im Deutschen vorhandenen nicht-konkatenativen Prozesse, nach den folgenden Strategien:

- Abwälzung ins Lexikon. Besonders unregelmässige Wörter (z.B. bei Suppletion: *sein - bin - ist - war*) können als Vollformen ins Lexikon aufgenommen werden, wodurch man sich eine aufwändige Behandlung im Regelsystem ersparen kann. Diese Strategie wird auch oft beim Ablaut angewendet, z.B. in GERTWOL und in SMOR durch die Angabe von Stammformen direkt im Lexikon. In mOLIFde gibt es keine Stammformen im Lexikon, da für jedes Ablautsmuster ein eigenes Paradigma existiert, und im Lexikon deshalb nur der Code für das zutreffende Paradigma gesetzt werden muss.
- Unterspezifikation im Lexikon. Dies eignet sich zur Behandlung von Umlauts- oder Harmoniephänomenen. Ein Substantiv mit Umlaut im Plural kann beispielsweise im Lexikon als *HAus-* abgelegt werden, wobei das grosse *-A-* ein unterspezifiziertes Phonem darstellt. Dieses kann durch eine dedizierte Regel später als *-a-* oder *-ä-* realisiert werden, und zwar unter Berücksichtigung des vorliegenden Numerus. Unterspezifikation im Lexikon wird meines Wissens bei keinem der deutschsprachigen Analysesysteme angewendet.
- Merkmalsprüfung mit Unifikation. Um dem Problem von nicht-lokalen Abhängigkeiten beizukommen, sind Mechanismen erfunden worden, um die endlichen Automaten und Transduktoren mit zusätzlichen Fähigkeiten auszurüsten. Der bekannteste Ansatz dafür sind die von den XFST-Entwicklern Beesley und Karttunen empfohlenen *Flag Diacritics* (vgl. Beesley/Karttunen 2003:339ff.). Im XFST Framework können damit Automaten mit "unsichtbaren" Übergängen angereichert werden, die das Setzen und Prüfen von Merkmalen er-

lauben. Beispielsweise können so Zirkumfixe, z.B. die deverbale Wortbildung mit *Ge-* und *-e* (*Ge-schrei-e*) im Deutschen behandelt werden, indem das Präfix *ge-* mit einem Merkmal "Zirkumfix" versehen wird, dessen Vorhandensein bei der Suffigierung von *-e* geprüft werden kann. Ein Überblick über weitere unifikationsbasierte Mechanismen bietet Amtrup 2003:224ff.

3.2 Einschränkung der Wortbildungsregeln

Will man dem Verhalten einer natürlichen Sprache zumindest annäherungsweise gerecht werden, muss man Wege finden, die Derivation und Komposition nach den in der Einleitung dargelegten Punkten einzuschränken. In diesem Punkt bestehen bei den existierenden deutschsprachigen Systemen grosse Unterschiede. GERTWOL, das fast unbeschränkt ableiten und komponieren lässt, steht am einen Ende der Skala, und TAGH, das mit relativ starken Restriktionen ausgerüstet ist, befindet sich am anderen Ende. SMOR liegt irgendwo dazwischen. Mögliche Massnahmen:

- Anreicherung der Lexikoneinträge mit Metainformationen, die für die Wortbildung ausschlaggebend sind. Dies geschieht bei SMOR und TAGH, bei denen beispielsweise die etymologische Herkunft des Stammes ("heimisch", "klassisch") die erlaubten Derivationen einschränkt. Bei GERTWOL und mOLIFde steht keine derartige Information zur Verfügung.
- Einschränkung der Bildungen gemäss morphologischer Struktur der Basis. Dies ist mit kontextabhängigen Regeln relativ gut zu meistern. Die deutschsprachigen Systeme bringen meines Wissens alle einen Mechanismus mit, der z.B. die vom Adjektivsuffix *-bar* "abhängige" Bildung auf *(-bar)-keit* erfassen kann. Dieser Typ von Derivation scheint übrigens besonders regelmässig, beinahe schon "automatisch" bildbar zu sein.
- Einschränkung nach syntaktischen Eigenheiten der Basis. Mir ist kein System bekannt, dass eine Restriktion wie die Zulassung der *-bar* Ableitung nur bei transitiven Verben ermöglichen würde.
- Semantische Komponente. Nur bei TAGH sind Einschränkungen aufgrund von semantischen Kriterien möglich. Realisiert wird dies durch eine "flache" semantische Klassifikation des Lexikons, wodurch, um das erwähnte Beispiel noch einmal aufzunehmen, die Diminutivbildung mit *-chen* auf konkrete Gegenstände und Lebewesen limitiert werden kann. Um zu beurteilen, wie gut die damit erzielten Resultate sind, muss man allerdings eine weitere Publikation der TAGH-Autoren abwarten, da sie im einzigen vorliegenden Artikel keine Evaluation präsentieren.

3.3 Priorisierung komplexer Analysen

Bei der Analyse von komplexen Wörtern kommt es schnell zu einer Vervielfachung der möglichen Analysen, von denen die meisten recht unwahrscheinlich sind. GERTWOL und SMOR sehen keine Möglichkeit vor, unwahrscheinliche Analysen auszufiltern – sie liefern stets alle gefundenen Analysen. Die bisherigen Bemühungen in diesem Bereich zielen alle in die selbe Richtung: Es soll eine Möglichkeit geben, die Pfade zu *gewichten*, sodass eine Rangliste von möglichst wahrscheinlichen Analysen erstellt werden kann. Zwei Strategien wurden bisher verfolgt:

- Externer Filter. Die von Volk 1999 entworfene Lösung ist ein externer Filter, durch den die Analysen des Morphologiesystems geschleust werden. Der Filter kann durch Anzahl und Typus der Morphemgrenzen eine Priorisierung vornehmen. Kernidee ist es, Wörter mit wenigen und möglichst "schwachen" Morphemgrenzen (Flexionsgrenze vor Derivationsgrenze vor Kompositionsgrenze) zu bevorzugen. Die von Volk erzielten Resultate sind recht gut, doch ist

das Vorgehen insofern unbefriedigend, als dass es ausserhalb des FSM-Frameworks liegt.

- Gewichtete Automaten. Die etwas radikalere Idee ist es, das FSM-Kalkül direkt so zu erweitern, dass einzelne Kanten des Automates gewichtet werden können. Wie das für die Behandlung von deutschen Komposita funktionieren kann, beschreibt Schiller 2006. Für die Funktionsweise von gewichteten Automaten verweise ich auf das folgende Kapitel.

4 Aktuelle Tendenzen in der Forschung

In der Forschung werden zur Zeit für die Behandlung der im vorhergehenden Kapitel beschriebenen verbleibenden Probleme vor allem Erweiterungen des bewährten FSM-Kalküls diskutiert. Wenn man den Tagungsband Yli-Jyrä 2006 als Massstab nimmt, ist unübersehbar, dass der Trend hin zu einer Anreicherung des Funktionsumfangs von FSM-Werkzeugen geht – nur herrscht noch keine Einigkeit darüber, auf welche Weise dies geschehen soll. Die häufigste Methode zur Erweiterung von endlichen Automaten scheint zur Zeit die Einführung von gewichteten Kanten zu sein. Dies möchte ich im folgenden etwas genauer darlegen. Ferner möchte ich eine Technik vorstellen, die in Automaten Register, also eine Art Arbeitsspeicher, einführt. Zum Schluss möchte ich einen dritten Bereich erwähnen, in welchem die Forschung zur Zeit aktiv ist: Die Herausforderung der Integration von FSM-basierten Morphologien in grössere NLP-Systeme.

4.1 Gewichtete Automaten

Findet man für ein bestimmtes Wort mehrere Pfade durch einen Automaten, so gelten im herkömmlichen Ansatz alle Pfade als richtig. Dies ist häufig auch erwünscht und etwa bei der Bestimmung von Flexionsformen ein beabsichtigter Effekt, da für eine bestimmte Form oft mehrere Analysen richtig sind. In anderen Fällen aber, z.B. wenn man bei komponierten und/oder derivierten Wörtern eine Vielzahl von Analysen bekommt, von denen die meisten unwahrscheinlich sind, möchte man die Möglichkeit haben, eine Priorisierung der einzelnen Pfade vornehmen zu können.

Grundlegende Überlegungen dazu, wie man solche Gewichtungen direkt in die Automatentheorie aufnehmen kann, haben Mohri et al. 2000 (AT&T Labs) geleistet. Sie haben mit ihren Erkentnissen auch gleich eine Implementation angefertigt: Das FSM-Toolkit von AT&T unterstützt bereits die Verarbeitung solcher gewichteten Automaten. Auch von den Forschern bei Xerox liegt mittlerweile ein Tool vor, das gewichtete Automaten unterstützt (Kempe et al. 2003).

Konkret bedeutet die Erweiterung, dass alle Kanten einer WFSM (*weighted finite state machine*) mit einem Gewicht versehen werden. Das Gewicht ist nicht mit der Wahrscheinlichkeit für die Kante identisch, sondern ist im Gegensatz zur Wahrscheinlichkeit als eine Art "Aufwand" zu betrachten, die für die Begehung der Kante benötigt wird. Das Gewicht berechnet sich z.B. aus dem umgekehrten Logarithmus der Wahrscheinlichkeit (Nasr/Volanschi 2006:169). Für die Umsetzung der Gewichte verwendet man das Konzept der Halbringe (engl. *semiring*), eine Struktur aus der abstrakten Algebra. Grundlegende (nichtcomputerlinguistische) Literatur zu Halbringen und ihren Anwendungen sind die Titel Golan 1999 und Mohri 2002. Ein Halbring ist eine Struktur der Form $(\mathbb{K}, \bigoplus, \bigotimes, \overline{0}, \overline{1})$. $\mathbb{K}$ ist eine nicht-leere Menge von Gewichten. Die zwei Zeichen $\bigoplus$ und $\bigotimes$ stehen für zwei Operationen, die man üblicherweise Addition und Multiplikation nennt, obwohl sie auch andere Operationen bezeichnen können. $\overline{0}$ und $\overline{1}$ stehen für die zwei "neutralen Elemente" dieser zwei Operationen, d.h. für Elemente, die bei der Anwendung der entsprechenden Operation mit $x \in \mathbb{K}$ wiederum x ergeben. Das neutrale Element für die Addition ist folglich 0 und für die Multiplikation 1, wodurch auch die Schreibweisen $\overline{0}$ und $\overline{1}$ verständlich werden. Jeder Kante des Auto-

maten wird nun ein Wert aus $\mathbb{K}$ zugewiesen, sodass ein gewichteter Automat entsteht. Das Gewicht eines Wortes wird nun so berechnet, dass die Gewichte aller Kanten entlang des Pfades mit der Operation $\bigoplus$ "addiert" werden. Falls mehrere Pfade für das Wort in Frage kommen, werden die einzelnen Gewichte mit der Operation $\bigotimes$ "multipliziert".

Ein anschauliches Beispiel für einen Halbring ist der sogenannte *tropische Halbring* $(\mathbb{R}^+, \min, +, \infty, 0)$. Der *tropische Halbring* wird bei gewichteten Automaten häufig verwendet, da er die (intuitiv richtige) Folge hat, dass die Gewichte entlang einem Pfad addiert werden, während bei mehreren richtigen Pfaden derjenige mit dem kleinsten Gewicht geliefert wird.

Für eine WFSM muss die Notation einer FSM um ein Initialgewicht sowie eine Funktion, die Endzustände auf Gewichte abbildet, erweitert werden. Zudem muss die Definition der Kanten so erweitert werden, dass pro Kante ein Gewicht festgelegt werden kann. Je nach Notation (vgl. z.B. Geyken/Hanneforth 2006) wird so aus dem ursprünglichen 5-Tupel eines endlichen Automaten ein 8-Tupel $(\Sigma, \Delta, Q, q_0, F, E, \lambda, \rho)$ über dem Halbring W, wobei Σ das Eingabealphabet, Δ das Ausgabealphabet, Q die Menge der Zustände, $q_0 \in Q$ den Startzustand, $F \subseteq Q$ die Menge der Endzustände, E die Menge der Kanten (mit Gewichten), $\lambda \in W$ das Anfangsgewicht und ρ die Abbildung der Endzustände in die Menge W bezeichnet.

Gewichtete endliche Automaten und Transduktoren werden bereits erfolgreich für verschiedene computerlinguistische Anwendungen eingesetzt. Dazu zählen die Verarbeitung von gesprochener Sprache (Mohri 1997), eine deutsche Morphologie (Geyken/Hanneforth 2006), ein französischer Tagger und Chunker (Nasr/Volanschi 2006), Analyse von deutschen Komposita (Schiller 2006) und wahrscheinlich noch einiges mehr.

Zur Illustration möchte ich das Vorgehen von Schiller herausgreifen, da sie sich gerade mit komplexen deutschen Wortformen auseinandersetzt. Schiller zeigt, wie man Kompositionsstämme direkt im Lexikon mit Gewichten versehen kann, welche die "Kompositionsfreudigkeit" einzelner Stämme angeben. Für die Verrechnung der Gewichte verwendet sie nicht den *tropischen Semiring* sondern den *realen Semiring* $(\mathbb{R}^+, +, *, 0, 1)$, was den etwas irritierenden Effekt hat, dass die Pfade mit den grössten Gewichten favorisiert werden. Im Gegensatz dazu werden die Gewichte anderswo eher als "Kosten" aufgefasst und die Priorisierung in ansteigender Ordnung gemacht (vgl. z.B. die grössere Gewichtung von starken Morphemgrenzen bei TAGH, Geyken/Hanneforth 2006:63). Schiller präsentiert auch eine Methode, wie die Gewichtungen mit statistischen Verfahren aus einem Korpus gewonnen werden können. Sie kann in ihrem Experiment zeigen, dass von den vielen möglichen Analysen von *Verbraucherzahlen* (*Verbrauch-erz-ahlen*, *Verbraucher-zahlen* etc.) tatsächlich die wahrscheinlichste Lesung *Verbaucher-zahlen* bevorzugt wird.

4.2 Finite-State Registered Automata (FSRA)

Das Konzept von *Finite-State Registered Automata* wurde in mehreren Publikationen entwickelt, wovon Cohen-Sygal/Wintner 2006 die jüngste darstellt. Das Automatenkonzept wird dabei mit einem Register erweitert, sodass an einem beliebigen Punkt im Automaten "Variablenwerte" abgespeichert oder ausgelesen werden können. Der Unterschied zum klassischen *Pushdown-Automaten* (Hopcroft et al. 2002:229ff.) liegt also darin, dass diese Art "Arbeitsspeicher" des Automaten nicht ein einfacher *last-in-first-out*-Stack ist, sondern über Adressen (Variablennamen) einen Zugriff auf beliebige Stellen des Speichers erlaubt. Cohen-Sygal und Wintner schlagen für das Bedienen des Registers auch gleich eine Syntax vor, die auf der XFST-Syntax aufbaut. In einem anschaulichen Beispiel erläutern sie, wie diese Technik für eine besonders elegante Behandlung des Partizip-Präteritum-Zirkumfixes *ge- -t* (*ge-mach-t*) verwendet werden kann. Ob der Ansatz von anderen Forschern übernommen

wird, bleibt abzuwarten – die präsentierten Zahlen zur Effizienz von FSRA machen jedenfalls einen vielversprechenden Eindruck.

4.3 Integration von Morphologiekomponenten

Mit einem ganz anderen Problem beschäftigt sich Amtrup 2003. Seine Überlegungen setzen an dem Punkt ein, wo die Morphologiesysteme brauchbar sind und in grössere Computerlinguistiksysteme eingebaut werden sollen. Amtrup möchte inbesondere die Integration in ein System, das mit Merkmal-Wert-Strukturen arbeitet, wie sie in vielen Grammatikformalismen verwendet werden, vereinfachen. Er schlägt vor, das Morphologieanalysesystem zu befähigen, anstatt simpler morphosyntaktischer Merkmale wie `<Sg>` für Singular oder `<3P>` für die dritte Person direkt Merkmal-Wert-Strukturen ausgeben zu können. Dazu ist bereits auf FSM-Ebene eine Unifikation von Merkmalen nötig. Wie erwähnt, gibt es nun bereits verschiedene Methoden wie z.B. *Flag Diacritics*, um Unifikation von Merkmalen innerhalb von Automaten zu realisieren. Auf diese Weise wird es also möglich, vom Morphologieanalysesystem direkt Merkmals-Wert-Strukturen ausgeben zu lassen. Dies kann z.B. in MÜ-Systemen attraktiv sein (Amtrup 2003:226).

5 Schluss

Der Vergleich der grossen deutschsprachigen Systeme hat kleine und grosse Unterschiede zwischen den verschiedenen Ansätzen aufgezeigt. Gewisse grundsätzliche Probleme finden sich in allen bestehenden Morphologiesystemen wieder. Diese wurden von mir dargestellt und zusammen mit möglichen Lösungsansätzen besprochen. Zum Schluss habe ich einige in der Forschung aktuell diskutierte Bereiche aufgegriffen.

Als ein Resultat dieser Arbeit darf die Einsicht gelten, dass die Wortbildung in der computerlinguistischen Morphologieanalyse lange stiefmütterlich behandelt wurde. Erst mit TAGH liegt ein System vor, das eine relativ ausgeklügelte Methode zur Einschränkung von Wortbildungsprozessen mitbringt. In diesem Bereich liegt in meinen Augen bei allen existierenden Systemen noch Verbesserungspotenzial.

Zu den dargestellten Erweiterungen des FSM-Kalküls (WFSM, FSRA) ist abschliessend zu bemerken, dass sie den bekannten theoretischen Rahmen durch Einführung von Gewichten oder Registern sprengen und somit nicht automatisch auf dessen bekannte Vorzüge aufbauen können. Zu ihrer Verarbeitung sind ganz neue Algorithmen nötig. WFSM und FSRA sind noch weniger gut verstanden und erforscht als die klassischen endlichen Automaten. Daher erklärt sich auch, warum in der Forschung so verschiedene Ansätze gewählt wurden, und es bleibt im Moment noch offen, welche Strategien sich in der Zukunft durchsetzen werden.

Bibliographie

Amtrup, J. W. *Morphology in Machine Translation Systems*. In: *Machine Translation*. Band 18, 2003. S. 213–235.

Antworth, E. L. *PC-Kimmo : A Two-level Processor for Morphological Analysis*. Dallas 1990.

Beesley, K. / Karttunen, L. *Finite State Morphology*. Stanford 2003.

Booij, G. *The Grammar of Words. An Introduction to Linguistic Morphology*. Oxford 2005.

Cohen-Sygal, Y. / Wintner, Sh. *Finite-State Registered Automata and Their Uses in Natural Languages*. In: Yli-Jyrä 2006. 43–54.

Creutz, M. / Lagus, K. / Virpioja, S. *Unsupervised Morphology Induction Using Morfessor*. In: Yli-Jyrä 2006. 300–301.

Duden. *Die Grammatik*. Band 4. Mannheim 2005. 7. Auflage.

Erben, J. *Einführung in die deutsche Wortbildungslehre*. 2000. 4. Auflage.

Fleischer, W. / Barz, I. *Wortbildung der deutschen Gegenwartssprache*. Tübingen 1995.

Forsberg, M. / Ranta, A. *Tool Demonstration: Functional Morphology*. In: Yli-Jyrä 2006. 304–305.

Gallmann, Peter / Sitta, Horst. *Deutsche Grammatik*. Zürich 2001. 3. Auflage.

Geyken, A. / Hanneforth, Th. *TAGH: A Complete Morphology for German Based on Weighted Finite State Automata*. In: Yli-Jyrä 2006. 55–65.

Glück, H. (Hrsg.) *Metzler Lexikon Sprache*. Stuttgart, Weimar 2000. 2. Auflage.

Golan, J. S. *Semirings and their Applications*. Dordrecht, Boston, London 1999.

Haapalainen, M. / Majorin, A. *GERTWOL: Ein System zur automatischen Wortformerkennung deutscher Wörter*. Onlinepublikation. 1994. <http://www.ifi.unizh.ch/CL/volk/LexMorphVorl/Lexikon04.Gertwol.html>.

Haapalainen, M. / Majorin, A. *GERTWOL und Morphologische Disambiguierung für das Deutsche*. Onlinepublikation. 1995. <http://www2.lingsoft.fi/doc/gercg/NODALIDA-poster.html>.

Hacken, P. ten / Lüdeling, A. *Word formation in computational linguistics*. In: *Proceedings of Traitement Automatique de Langue Naturelle (TALN) 2002, Nancy*. Band 2, 2002. S. 61–87. <http://www.loria.fr/projets/JEP-TALN/actes/TALN/tutoriels/Tutoriel03.pdf>.

Hammarström, H. *A New Algorithm for Unsupervised Induction of Concatenative Morphology*. In: Yli-Jyrä 2006. 288–289.

Hopcroft, J. E. / Motwani, R. / Ullman, J. D. *Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie*. München 2002. 2. Auflage. [engl. Original 2001].

Ibarra, O. H. / Dang, Zhe (Hrsg.). *Implementation and Application of Automata. 8th International Conference CIAA 2003*. Berlin, Heidelberg 2003.

Jurafsky, D. / Martin, J. *Speech and Language Processing*. Upper Saddle River 2000.

Karttunen, L. / Beesley, K. *Twenty-Five Years of Finite-State Morphology*. In: Arppe, A. et al. (Hrsg.) *Inquiries into Words, Constraints, and Contexts. Festschrift for Kimmo Koskenniemi on his 60th Birthday*. 71–83. Stanford 2005. <http://csli-publications.stanford.edu/koskenniemi-festschrift/8-karttunen-beesley.pdf>.

Kempe, A. / Baeijs, Ch. / Gaál, T. / Guigne, F. / Nicart, F. *WFSC - A New Weighted Finite State Compiler*. In: Ibarra/Dang 2003. 108–119.

Koskenniemi, K. *Two-level morphology: A general computational model for word-form recognition and production*. Helsinki 1983. [nicht gesehen].

Koskenniemi, K. / Haapalainen, M. *GERTWOL – Lingsoft Oy*. In: Hauser, Roland (Hrsg.) *Linguistische Verifikation : Dokumentation zur Ersten Morpholympics 1994*. 121–140. Tübingen 1996. <http://www.cl.unizh.ch/siclemat/lehre/ss06/mul/script/papers/KoskeniemmiHaapalainen1996.pdf>.

Mohri, M. *Finite-State Transducers in Language and Speech Processing*. In: *Computational Linguistics*. Band 23, 1997(2). S. 269–311. <http://acl.ldc.upenn.edu/J/J97/J97-2003.pdf>.

Mohri, M. *Semiring Frameworks and Algorithms for Shortest-Distance Problems*. In: *Journal of Automata, Language, and Combinatorics*. Band 7, 2002(3). S. 321–350. <http://www.soe.ucsc.edu/classes/cmps290c/Spring04/paps/semi_mohri.pdf>.

Mohri, M. / Pereira, F. / Riley, M. *The Design Principles of a Weighted Finite-State Transducer Library*. In: *Theoretical Computer Science*. Band 231, 2000. S. 17–32. <http://www.cis.upenn.edu/~pereira/papers/tcs.pdf>.

Nasr, A. / Volanschi, A. *Integrating a POS Tagger and a Chunker Implemented as Weighted Finite State Machines*. In: Yli-Jyrä 2006. 167–178.

Ritchie, G. D. / Russell, G. J. / Black, A. W. / Pulman, S. G. *Computational Morphology. Practical Mechanisms for the English Lexicon*. Cambridge, London 1992.

Schiller, A. *German Compound Analysis with* wfsc. In: Yli-Jyrä 2006. 239–246.

Schmid, H. *A Programming Language for Finite State Transducers*. In: Yli-Jyrä 2006. 308–309.

Schmid, H. / Fitschen, A. / Heid, U. *SMOR: A German Computational Morphology Covering Derivation, Composition, and Inflection*. Onlinepublikation. 2004. <http://www.ims.uni-stuttgart.de/www/projekte/gramotron/PAPERS/LREC04/smor.ps.gz>.

Sproat, R. *Morphology and Computation*. Cambridge MA 1992.

Volk, M. *Choosing the right lemma when analysing German nouns*. In: *Multilinguale Corpora: Codierung, Strukturierung, Analyse*. 304–310. Frankfurt 1999.

Yli-Jyrä, A. et al. (Hrsg.). *Finite-State Methods and Natural Language Processing. 5th International Workshop, FSMNLP 2005*. Berlin, Heidelberg 2006.