Akademia Bydgoska im. Kazimierza Wielkiego
Wydział Humanistyczny
Katedra Filologii Germańskiej

Marcin Junczys-Dowmunt

Ein Finite-State-Modell für einfach und mehrfach zusammengesetzte Komposita

Praca magisterska
napisana pod kierunkiem
prof. Marka Cieszkowskiego

Bydgoszcz 2005

Akademia Bydgoska im. Kazimierza Wielkiego
Wydział Humanistyczny
Katedra Filologii Germańskiej

Marcin Junczys-Dowmunt

Model skończenie stanowy wyrazów pojedynczo i wielokrotnie złożonych

Praca magisterska
napisana pod kierunkiem
prof. Marka Cieszkowskiego

Bydgoszcz 2005

Inhaltsverzeichnis

Abbildungsverzeichnis

Einführung

Ein Grundsatz der Computerlinguistik beruht auf der Annahme, dass menschliche Sprachprozesse mithilfe des Computers beschrieben und simuliert werden können. Zwar ordnet man die Computerlinguistik eher der Sprachwissenschaft zu, dennoch bezieht sie als interdisziplinäre Wissenschaft vielfach Methoden und Modelle aus der Informatik, der Mathematik, der kognitiven Psychologie und anderen verwandten Bereichen. Besonders das Konzept der formalen Sprachen und Grammatiken sowie die damit verbundene Automatentheorie, deren Ursprünge in der Mathematik und der theoretischen Informatik anzusetzen sind, haben einen bedeutenden Einfluss sowohl auf die Grundlagenforschung als auf konkrete Anwendungen innerhalb der Computerlinguistik. Sprachliche Vorgänge auf mathematische Modelle zurückzuführen und diese anschließend auf einem Rechnersystem zu realisieren, ist ein erklärtes Ziel. Unlösbar verbunden mit den formalen Sprachen und den endlichen Automaten sind Parser-Anwendungen. Grammatiken dienen im Allgemeinen zur Generierung von formalen Sprachen, Automaten sind in der Lage zu erkennen, ob ein Ausdruck zu einer bestimmten Sprache oder Sprachklasse gehört und Parser schließlich liefern Strukturbeschreibungen, auf deren Grundlage zum Beispiel semantische Interpretationen möglich sind.

Ein für die deutsche Sprache typisches und auch sehr interessantes Phänomen sind die Komposita. Ihnen ist zu verdanken, dass Deutsch als die Sprache mit „den langen Wörtern" bekannt ist. Gerade die Komposita stellen nicht nur bei der maschinellen Übersetzung eine besondere Hürde dar, aber auch für ausländische Deutschlerner ergeben sich Schwierigkeiten. Diese sind darin begründet, dass eine Vielzahl der Komposita, vor allem die Determinativkomposita, okkasionale Bildungen sind und daher auch in keinem Wörterbuch enthalten sein können. Ein deutscher Muttersprachler hat in den meisten Fällen keine Probleme, die Bedeutung eines Kompositums anhand der Bedeutungen seiner Komponenten und des Verhältnisses dieser Komponenten zu einander zu erkennen. Ist dies nicht ohne weiteres möglich, ist eine Betrachtung des unmittelbaren Kontextes üblicherweise ausreichend für die Bedeutungserschließung.

Eine Übersetzungsanwendung wird mit einem neuen Kompositum nun ganz ähnliche Probleme haben: sofern das angetroffene Wort nicht in seinem internen Lexikon enthalten ist, weiß das Programm nichts mit ihm anzufangen. Eine Bestandsaufnahme aller potentiellen Komposita ist unmöglich, da es theoretisch unendlich viele Komposita gibt und jedes dieser Komposita erneut als Glied einer anderen Zusammensetzung auftauchen kann. Diese Eigenschaft der Komposita macht eine Strukturana-

lyse notwendig, die die Zusammensetzungen auf seine unmittelbaren Konstituenten zurückführt, und zwar soweit, dass diese Konstituenten in einem mehr oder weniger kompletten Lexikon aufgenommen werden können. Aufgrund ihrer besonderen Eigenschaften sind Komposita im Sprachsystem zwischen Lexik, Morphologie und Syntax anzusiedeln. Eine zuverlässige Strukturanalyse müsste allen Systemebenen Rechnung tragen und dabei auch semantische wie auch pragmatische Informationen berücksichtigen.

Diesem Anspruch kann man nur schwer gerecht werden, weshalb das Modell auf eine Oberflächenanalyse (Shallow Parse) beschränkt bleibt. Demnach ist das Parsing von Komposita kein triviales Problem, kann aber auf eine Problemstellung reduziert werden, die eingeschränkt genug ist, um für eine Formalisierung mit Finite-State-Modellen geeignet zu sein.

Ziel dieser Arbeit ist es, ein formales, mathematisch begründetes Finite-State-Modell für deutsche einfach und mehrfach zusammengesetzte Komposita vorzustellen. Mithilfe dieses Modells wird gleichzeitig eine auf mehreren Ebenen arbeitende Parsing-Anwendung beschrieben, die durch eine Implementation der beschriebenen Automaten direkt funktionsfähig wird. Die Arbeitsebenen des Parsers betreffen die graphematische Form des Kompositums und mehrere voranalysierte Zwischenstufen, die während des letzten Analysevorgangs disambiguiert werden. Der Parser wird eine Zusammensetzung segmentieren und die gefundenen Segmente mit entsprechenden linguistischen Informationen annotieren. Bei morphologisch mehrdeutigen Komposita werden mehrere Interpretationsmöglichkeiten angegeben, sofern die letzte Analysestufe keine Disambiguierung erreichen konnte, wobei eine semantische Analyse der erhaltenen Strukturen und die damit in Verbindung stehende Untersuchung der Idiomatisierung nicht zum Umfang der Arbeit gehören. Grundlage aller Formalismen bleiben dabei immer die linguistischen Eigenschaften und Besonderheiten der Komposita – vor allem auf morphologischer Ebene.

KAPITEL 1

Vorbemerkungen zur Komposita-Analyse

Komposita sind der linguistische Gegenstand dieser Arbeit. Im Folgenden werden Wesen, Eigenschaften und Struktur der deutschen Komposita in Kürze dargestellt. Vorgehensweisen bei der Segmentierung der Komposita werden besprochen. Anschließend richtet sich das Augenmerk auf die Kompositionsfuge, da das Finite-State-Modell auch das Vorkommen der Fugenelemente und deren Varianten beschreiben wird. Die herausgearbeiteten Regelmäßigkeiten werden später zur Disambiguierung von gefundenen Komposita-Segmentierungen verwendet. Auch werden andere Phänomene an der Kompositionsfuge wie Tilgung und orthografische Besonderheiten wie der Durchkopplungsbindestrich berücksichtigt.

1.1 Das Finite-State-Modell und Wortbildungsmodelle

In diesem Abschnitt wird versucht, eine Brücke zwischen den formalen Mitteln aus Kapitel 2 und dem linguistischen Untersuchungsgegenstand dieser Arbeit, den Komposita, zu schlagen. Es werden Parallelen und Unterschiede des entwickelten Finite-State-Modells zu klassischen Wortbildungsmodellen aufgezeigt. Weiterhin wird dargelegt, warum bewusst auf die Einbeziehung bestimmter Aspekte in das Modell verzichtet wurde und teilweise verzichtet werden musste.

Komposita sind Wortbildungsprodukte. Dem Aufbau von Wortbildungsprodukten liegen bestimmte Wortbildungsmodelle zugrunde. „Ein Wortbildungsmodell ist ein morphologisch-syntaktisch und lexikalisch-semantisch bestimmtes Strukturschema, nach dem Reihen gleich strukturierter Wortbildungsprodukte mit unterschiedlichem lexikalischem Material erzeugt werden können" (Fleischer und Barz 1995, S. 53).

Die Herausarbeitung der Gemeinsamkeiten aller untersuchten Elemente ist das Wesen einer Modellierung, wobei der Abstraktionsgrad eines Modells von der gewählten Beschreibungsgenauigkeit abhängt (vgl. Fleischer und Barz 1995, S. 53).

In der vorliegenden Arbeit wird ein Modell mit mehreren Abstraktionsgraden erstellt. Während bei der morphologischen Analyse einerseits direkt von der lexikalischen Oberfläche ausgegangen wird, also kaum abstrahiert wird, werden bei der Modellierung der Distributionsregeln der Kompositakonstituenten überwiegend abstrakte Kategorien verwendet.

Ein Modell, das auf Transducern beruht, kann sowohl zu Analysezwecken, als auch zur Generierung von Strukturen genutzt werden.[1] Ähnlich wie bei Fleischer und Barz, die feststellen, dass ihre Modellbeschreibung nicht ausreicht, um als vollständiger generativer Mechanismus zu fungieren, der nur die Bildung richtiger Wortbildungsprodukte zulässt, erhebt auch das in dieser Arbeit vorgestellte Modell der Komposita nicht den Anspruch auf Vollständigkeit. Während Fleischer und Barz die Ausdrücke „generativ" und „Mechanismus" noch in Anführungszeichen setzen, ist das in der Arbeit beschriebene Finite-State-Modell im wörtlichen Sinne generativ, da seine Implementierung am Rechner wünschenswert ist, was aus dem Modell letztendlich einen Parser oder Generator macht.

Ein Wortbildungsmodell sollte Angaben über folgende Strukturmerkmale enthalten (vgl. Fleischer und Barz 1995, S. 54):

a) Eine Morphemcharakteristik der Konstituenten, das heißt Informationen, ob es sich zum Beispiel um Grundmorpheme oder Grundmorphemkomplexe handelt. Im Falle von Grundmorphemen sollten auch Informationen über Wortart und semantische Klasse enthalten sein.

b) Die Reihenfolge der Konstituenten

c) Die Wortart und semantische Klasse des finalen Wortbildungsproduktes

d) Formativstrukturelle Spezifika des Wortbildungsproduktes, wie zum Beispiel morphophonologische und graphische Charakteristika

e) Satzsyntaktisches Verhalten des Wortbildungsproduktes

f) Wortbildungsbedeutung

Zu a) Das vorgestellte Finite-State-Modell verfügt über die geforderte Morphemcharakteristik aufgrund der morphologischen Informationen, die im Lexikon einem Lexikoneintrag zugeordnet werden. Der Lexikoneintrag an sich beantwortet die Frage nach der Morphemklasse, da neben den Fugenelementen überwiegend Grundmorpheme und Grundmorphemkomplexe erfasst sind. In der Einführung zu dieser Arbeit

[1] Dies kann durch die Umkehrung von Transducern oder ganzen Transducerkaskaden erreicht werden (siehe 2.2.2). Formal werden umgekehrte Transduktionen angenommen, die ein Analyseergebnis auf die Ursprungsform abbilden können.

wurde festgestellt, dass semantische Aspekte nicht zum Untersuchungsgegenstand gemacht werden. Insofern bleiben die Anforderungen in Bezug auf die semantischen Klassen der Konstituenten unberücksichtigt. Die Wortart hingegen wird nach der Segmentierung durch Tagging (siehe 1.5.1) annotiert.

Zu b) Es sind gerade Reihenfolgen, die von Finite-State-Modellen besonders effizient erfasst werden können. Reguläre Sprachen, die durch endliche Automaten definiert werden können, sind entweder links- oder rechtsrekursiv, was dargestellt als Baumstruktur in einem linearen Ableitungsbaum resultiert. Es sind vor allem die Regeln, die die Reihenfolge der Konstituenten betreffen, die in dieser Arbeit modelliert werden.

Zu c) Die Wortart von Kompositа wird eindeutig durch die Wortart der letzten Konstituente bestimmt. Da die Wortarten der Konstituenten erfasst werden, wird auch die Wortart des finalen Wortbildungsproduktes berücksichtigt. Vermeintliche Sonderfälle der Zusammensetzung, bei denen die Wortart der letzten Konstituente nicht der Wortart der Zusammensetzung entspricht, sind keine Komposita, sondern nach Auffassung des Autors Derivate, Konversionen oder Inkorporationen (siehe 1.4).

Zu d) Das Modell ist konzipiert für eine computerlinguistische Anwendung und beschränkt sich auf die Analyse von schriftlich vorliegenden Daten. Tatsächlich müssen die Daten in einer für einen Computer verständlichen Form vorliegen, weshalb hier von Zeichenketten ausgegangen wird. Morphophonologische Charakteristika bleiben insofern zwangsläufig unberücksichtigt, wobei die Vorschaltung eines Systems, das akustische Daten in Zeichenketten umwandelt, durchaus realisierbar ist.

Zu e) Bis auf die Wortart des Wortbildungsproduktes werden keine weiteren syntaktisch bedeutungsvollen Merkmale berücksichtigt. Das würde ähnlich wie die semantische Analyse die Verbindung mit weiteren Modellen auf verschiedenen Ebenen der Sprache voraussetzen.

Zu f) Die Wortbildungsbedeutung wird als semantische Größe nicht behandelt.

Eichinger (2000, S. 9) beruft sich bei der Betrachtung von komplexen Wörtern auf das systematische Wissen eines Sprechers über syntagmatische und paradigmatische Zusammenhänge, in die das komplexe Wort eingeordnet werden kann.

Bei der syntagmatischen Einbindung unterscheidet Eichinger zwei Ebenen – eine syntagmainterne und eine syntagmaexterne Ebene. Auf der syntagmainternen Ebene wird die Struktur der komplexen Wörter beschrieben. Dabei kann es sich um Regularitäten der Verkettung von Bestandteilen handeln sowie um eine Modellierung der Beziehungen innerhalb der Wörter. Diese Beziehungen können bereits ausreichen, um einige Hypothesen zu der Bedeutung der gesamten Konstruktion anzustellen. Die zweite, syntagmaexterne Ebene umfasst die Einbettung des komplexen Wortes in eine syntagmatisch größere Einheit, wie einen Satz oder allgemeiner einen Text. Die syntaktische Umgebung sowie ein weiterer Kontext geben Hinweise zur Bedeutung.

Ein Beispiel ist die Paraphrasierung komplexer Wörter, in der die syntagmatischen Beziehungen ausformuliert werden.

Durch die Lexikalisierung von Teilkonstituenten ergeben sich mögliche paradigmatische Beziehungen, die ebenfalls auf einer syntagmainternen und einer syntagmaexternen Ebene betrachtet werden können. Im ersten Fall sind im Lexikon des Sprechers bereits ähnliche Bildungen bekannt, die bei der Interpretation des komplexen Wortes herangezogen werden können. Noch abstrakter ist EICHINGERs Verständnis von einer syntagmaexternen Paradigmatik, bei der er von einer Einbettung in Wirklichkeitsbezüge, in das Wissen des Sprechers über die Zusammenhänge zwischen den Dingen ausgeht.

Die syntagmaexternen Aspekte der beschriebenen Syntagmatik und Paradigmatik bleiben in dieser Arbeit zwangsweise ausgespart. Syntagmainterne Aspekte werden jedoch passend zu beiden Sichtweisen erfasst. Beziehungen zwischen Konstituenten werden sowohl mithilfe von allgemeingültigen Regeln (Syntagmatik) als auch im Zusammenhang mit bestimmten konkreten Gliedern (Paradigmatik) modelliert.

Im Fall der Komposita verweisen FLEISCHER UND BARZ auf einige Fixpunkte, deren Modellierung aufgrund der genannten Einschränkungen schwierig ist. Genannt werden dabei die folgenden Aspekte (FLEISCHER UND BARZ 1995, S. 93):

a) Binarität der UK-Struktur

b) Theoretisch unbegrenzte Komplexität der UK

c) Determinatives oder kopulatives Verhältnis der UK

d) Wortartcharakteristik der ersten UK

e) Appellativischer, onymischer oder phraseologischer Charakter der ersten UK

Zu a) Die Binarität der Komposita-Struktur ist ohne die semantische Komponente nicht eindeutig erfassbar. Das vorgestellte Modell verfügt, wie bereits erwähnt, über keine semantischen Daten der Konstituenten und kann dementsprechend auch keine semantischen Regeln beinhalten. Bei einem mehrfach zusammengesetzten Kompositum können mehrere Strukturtypen zugrunde liegen (vgl. DUDENREDAKTION 1998, S. 482) und ohne semantische Kriterien kann in vielen Fällen nur schwerlich zwischen Links- und Rechtsverzweigung oder Mischformen unterschieden werden. Das Kompositum *Armbanduhr* ist ein Beispiel für eine linksverzweigende Struktur, die Konstituentengrenze verläuft zwischen *Armband* und *Uhr*, wobei die erste Konstituente weiterhin untergliedert werden kann. Für einen Rechner ohne semantische Informationen gibt es allerdings keinerlei Anhaltspunkte, warum es z. B. keine *Banduhr* geben sollte. Es wird also keine hierarchische Strukturanalyse unternommen, da diese ohnehin alle möglichen Verzweigungsstrukturen zurückgeben müsste und damit genauso wenig aussagekräftig wäre wie der Mangel an solchen Strukturen. Bei z. B. viergliedrigen Komposita gäbe es fünf mögliche Baumstrukturen, bei fünfgliedrigen bereits 14 und bei sechsgliedrigen 42.

Zu b) Diesem Aspekt wird Rechnung getragen. Zwar bleibt die Modellierung auf die Oberfläche der Komposita beschränkt, dafür ist bei der Analyse keine maximale Zahl an Konstituenten gegeben. Es ist eine der Haupteigenschaften formaler Sprachen im Allgemeinen und endlicher Automaten im Speziellen, dass sie mit endlichen Mitteln unendliche Mengen und Strukturen beschreiben können.

Zu c) Auch hier ist eine Klassifizierung ohne semantische Einheit kaum zu realisieren, da sich Determinativ- und Kopulativkomposita aufgrund ihrer rein morphemischen Struktur nicht unterscheiden. Beide sind Zusammensetzungen von Grundmorphemen oder Grundmorphemkonstruktionen und bei beiden können Fugenelemente auftreten. Auch können die Konstituenten von Kopulativkomposita an der Stelle von Determinante oder Basis eines Determinativkompositums auftauchen und umgekehrt.

Zu d) Die Wortarten der gefundenen Komponenten werden erfasst, dies wurde bereits erwähnt. Bei mehrfachen Zusammensetzungen kann es jedoch schwierig werden zu bestimmen, wann genau die erste Konstituente endet und die zweite beginnt.

Zu e) Dieser Aspekt ist abhängig von den Daten, die im Lexikon gespeichert sind. Es ist an sich kein Problem, eine eigene Kategorie für Eigennamen anzunehmen und diese im Modell zu berücksichtigen. Eine Verbindung mit dem Durchkopplungsbindestrich könnte ebenfalls ganz einfach implementiert werden. Auch Phrasen können so in das Lexikon aufgenommen werden. Es wird in dieser Arbeit jedoch aus praktischen Gründen darauf verzichtet.

Die Anforderungen an ein Wortbildungsmodell, die von Fleischer und Barz (1995) gestellt werden, umfassen Aspekte mehrerer Ebenen des Sprachsystems. Es sollen morphologische, syntaktische, semantische und auch pragmatische Aspekte berücksichtigt werden. Diesen Kriterien wird das vorgestellte Finite-State-Modell nur bedingt gerecht (siehe oben die Ausführungen zu Syntagmatik und Paradigmatik). Die Verbindung mit Modellen, die Regelmäßigkeiten auf verschiedenen Sprachebenen beschreiben, wäre nötig, um alle Bedürfnisse zu erfüllen. Stattdessen wird vielmehr ein Zwischenergebnis geliefert, das als Grundlage zu weiteren Analysen in größeren Kontexten verwendet werden kann.

1.2 Zum Wesen der Komposita

„Unter Zusammensetzungen (Komposita) verstehen wir Wörter, die ohne zusätzliche Ableitungsmittel aus zwei oder mehreren selbstständig vorkommenden Wörtern gebildet sind. Dabei stellt der (isolierbare) erste Bestandteil – von den wenigen Kopulativkomposita und Vergleichskomposita des Typus Himmelskuppel abgesehen – das Bestimmungsglied dar, der zweite das Grundwort (die Basis), das die Wortart der ganzen Zusammensetzung festlegt" (Dudenredaktion 1998, S. 432).

Andere gebräuchliche Bezeichnungen für Bestimmungsglied und Grundwort sind entsprechend Erst- und Zweitglied.

In der generativen Grammatik wird eine andere Terminologie verwendet. So wird das Grundwort als Kopf bezeichnet und das Bestimmungswort einfach als Nicht-Kopf. Als Kopf bezeichnet man im Allgemeinen den einfacheren Teil eines komplexen Gebildes X, der die grammatischen Eigenschaften von X bestimmt (vgl. Sternefeld 2004, S. 5).

Der Wortbildungsprozess, der zu der Entstehung von Komposita führt, heißt Komposition. Eichinger (2000, S. 71) bezeichnet die Komposition als „vielleicht zentralste Art der Wortbildung".

Beide Bestandteile sind in der Regel ebenfalls wortfähig und innerhalb des Kompositums nicht umstellbar (außer bei den wenigen Kopulativkomposita), da außer der Wortart auch die semantische Klasse durch das Grundwort bestimmt wird (vgl. Dudenredaktion 1998, S. 432). Deutsche Komposita werden anders als z. B. in der englischen Sprache traditionell zusammengeschrieben, was eine Konstituentenanalyse zusätzlich erschwert.

„Die beiden UK[2] können entweder in einer Beziehung der Unter- bzw. Überordnung stehen (Subordination), oder sie können gleichgeordnet sein (Koordination). Im ersten Fall handelt es sich um Determinativkomposita, im zweiten Fall um Kopulativkomposita" (Fleischer und Barz 1995, S. 45).

Laut Dudenredaktion (1998, S. 481) spricht man von Kopulativkomposita, wenn beide Glieder des Kompositums der gleichen Bezeichnungsklasse angehören, einander gleichgeordnet sind und wenn die Reihenfolge der Glieder theoretisch vertauschbar ist.

„Die Möglichkeit, Kopulativkomposita zu bilden, die von der Syntax mit ihren vielen Kopulativkonstruktionen her eigentlich sehr nahe liegt, wird im Deutschen wenig genutzt (0,4%; ohne die Namensverbindungen). Die Bildungen sind größtenteils das Ergebnis einer sehr bewussten Sprachprägung Einzelner. In der Gemeinsprache herrscht eindeutig die Zusammensetzung vom Typ des Determinativkompositums vor" (Dudenredaktion 1998, S. 481).

Determinativkomposita werden von der Dudenredaktion (1998, S. 482) auch als Haupt- und Grundtyp der Substantivzusammensetzungen bezeichnet. Im Gegensatz zum Kopulativkompositum führt eine Verstellung der Glieder zu einer Bedeutungsveränderung. „Im Unterschied zu den Kopulativkomposita, die fast immer aus zwei Lexemen bestehen, treten bei Determinativkomposita häufig auch Zusammensetzungen als Grund- oder Bestimmungswörter auf" (Dudenredaktion 1998, S. 482).

Vor allem beim Substantiv ist das Determinativkompositum am häufigsten. Es wird sogar angezweifelt, ob die im Vergleich so seltenen Kopulativkomposita überhaupt in stabiler Form existieren (Eichinger 2000, S. 117).

[2] UK = Unmittelbare Konstituenten: die beiden Konstituenten, aus denen eine Konstruktion unmittelbar gebildet ist und in die sie sich auf der nächstniedrigeren Ebene zerlegen lässt. (vgl. Fleischer und Barz 1995, S. 43)

„Anders als bei syntaktischen Fügungen und ihren Konstituenten werden Beziehungen zwischen den Kompositionsgliedern nicht durch Flexive angezeigt" (Glück 2000, S. 361). Es findet also üblicherweise keine innere Flexion statt, zu einigen wenigen Ausnahmen existieren äquivalente Formen ohne innere Flexion.

1.3 Komposita nach Wortart des Grundwortes

An dieser Stelle werden die Komposita nach der Wortart des Grundwortes bzw. des Zweitgliedes eingeteilt. Fast alle Wortarten bilden Komposita, an dieser Stelle werden jedoch nur die Substantive, Adjektive und Verben besprochen. Bei den übrigen Wortarten sind die Komposita überwiegend lexikalisiert und oftmals nur schwierig von Zusammenrückungen zu unterscheiden.

Eine andere Betrachtungsweise der Komposita, die das Erstglied in den Vordergrund stellt, wird in Kapitel 1.6 angewendet. Den Stammparadigmen werden Kompositionsparadigmen zur Seite gestellt. So wird neben der Flexionsstammform und der Derivationsstammform auch eine Kompositionsstammform angenommen.

1.3.1 Nominale Komposita

„Substantivische Komposita sind unter Verwendung von Einheiten aller Wortarten als Erstglied bildbar; auch Adverbien, Präpositionen, Konjunktionen und sonstige Partikeln können dabei Verwendung finden. Stark ausgeprägt ist die Determinativkomposition aus zwei substantivischen UK, während die kompositionelle Verbindung von UK beim Adjektiv und noch mehr beim Verb weniger entwickelt ist. Beträchtlich stärker als bei den anderen Wortarten ist auch die polymorphemische Komposition (mit vier und mehr Grundmorphemen) vertreten" (Fleischer und Barz 1995, S. 84).

Nach der Klassifizierung aufgrund der Wortart des Zweitgliedes können die Komposita nach der Wortart des Erstgliedes subklassifiziert werden. Hier werden einige besonders häufige Typen kurz vorgestellt.

- Der Typ Substantiv+Substantiv: Laut Dudenredaktion (1998, S. 483) stellen Bildungen des Typs Substantiv+Substantiv vier Fünftel aller Substantivkomposita dar. Beispiele: *Gartentor*, *Kindergeschrei*, *Armbanduhr*

  Hierbei können beide unmittelbaren Konstituenten sowohl Simplizia als auch komplexe Substantive sein.

- Der Typ Adjektiv(Partizip)+Substantiv: Der Anteil dieser Substantivzusammensetzungen liegt bei 6%. Das Bestimmungswort wird im Allgemeinen ohne Fugenelement mit dem Substantiv verbunden. Es kommen auch Zusammenset-

zungen mit Superlativformen als Erstglied vor. Beispiele: *Kleinkind*, *Höchstpreis*, *Gebrauchtwagen*

- Der Typ Verb+Substantiv: Der Anteil dieses Strukturtyps beträgt je nach Textart 5–10%. Es ist unter synchroner Betrachtungsweise nicht immer möglich ein verbales Erstglied von einem substantivischen Verbalabstraktum zu unterscheiden. In solchen Fällen sind beide Interpretationen zutreffend. Beispiele: *Schreibmaschine*, *Mischgetränk*, *Lebewesen*

- Der Typ Adverb/Partikel/Präposition+Substantiv: Bei diesem Strukturtyp werden Zusammensetzungen mit flexionslosen Wörtern als Erstglied zusammengefasst. Besonders Zusammensetzungen mit Präpositionen sind häufig. Beispiele: *Vorstadt*, *Nebenzimmer*, *Unterangebot*, *Sofortprogramm*

Außer den vorgestellten Strukturtypen treten auch weitere Strukturen als Erstglieder auf, die nicht einfach als Wortarten klassifiziert werden können. Unter Verwendung des Durchkopplungsbindestrichs können Eigennamen, Wortgruppen und auch ganze Sätze als Erstglied verwendet werden (vgl. Fleischer und Barz 1995, S. 122).

1.3.2 Adjektivische Komposita

Ähnlich wie bei Substantiven sind bei adjektivischen Komposita prinzipiell alle Wortarten als Erstglied möglich, doch neben Substantiven, Adjektiven und Verben treten andere Wortarten kaum auf. Kopulativkomposita sind bei Adjektivzusammensetzungen häufiger als bei Substantiven (vgl. Fleischer und Barz 1995, S. 225). Komposita mit partizipialem Zweitglied werden laut Fleischer und Barz (1995, S. 241) ebenfalls zu der Gruppe der Adjektivzusammensetzungen gezählt. Auch hier können verschiedene Strukturtypen beschrieben werden:

- Der Typ Substantiv+Adjektv: Adjektivkomposita mit substantivischem Erstglied sind durchweg Determinativkomposita. Beispiele: *knietief*, *ofenwarm*, *altersschwach*

- Der Typ Adjektiv+Adjektiv: Hier kommen sowohl Determinativkomposita (Beispiele: *schwerkrank*, *dünnflüssig*) als auch Kopulativkomposita (Beispiele: *taubblind*, *süßsauer*) vor.

- Der Typ Verb+Adjektiv: Laut Fleischer und Barz (1995, S. 247) ist die adjektivische Komposition mit verbalem Erstglied im stetigen Ausbau begriffen. Besonders in technischen Texten treten solche Konstruktionen auf. Beispiele: *tragfähig*, *röstfrisch*, *triefnass*

1.3.3 Verbale Komposita

Die Komposition ist bei der Bildung der Verben weniger relevant als bei Substantiven und Adjektiven. Verbzusammensetzungen sind stets zweigliedrig und trennbar, wobei sie im Infinitiv, in der Partizipialform und bei der Endstellung im Nebensatz nicht getrennt erscheinen (vgl. Dudenredaktion 1998, S. 448). Man unterscheidet außerdem die so genannten Pseudokomposita von den echten Verbzusammensetzungen. Optisch handelt es sich um Komposita, bei einer diachronischen Betrachtungsweise ergibt sich jedoch, dass diese Pseudokomposita eigentlich Ableitungen aus Substantiven darstellen, z. B. *wehklagen* aus *Wehklage*. Diese Pseudokomposita sind nicht trennbar (vgl. Dudenredaktion 1998, S. 449).

- Der Typ Substantiv+Verb: Neben den oben genannten Pseudokomposita gibt es auch vereinzelte Verbindungen aus Substantiv und Verb nach Art der Zusammenrückungen (vgl. Dudenredaktion 1998, S. 449). Beispiele: *standhalten*, *teilnehmen*

  Verben, die durch ein substantivisches Erstglied näher bestimmt werden, treten in technischen Texten auf. Solche Konstruktionen sind jedoch selten (vgl. Dudenredaktion 1998, S. 450). Beispiele: *punktschweißen*, *feuerverzinken*

- Der Typ Adjektiv+Verb: Verbindungen mit Adjektiven als Erstglied sind häufiger als verbale Zusammensetzungen mit Substantiven oder Verben. Sie werden traditionell zusammengeschrieben (vgl. Dudenredaktion 1998, S. 450). Beispiele: *freischaufeln*, *heißlaufen*, *hochstapeln*

- Der Typ Verb+Verb: Bei den Verben sind die sonst so häufigen Zusammensetzungen von Wörtern der gleichen Wortart kaum üblich. Allenfalls in der expressionistischen Dichtung und in technischen Texten finden sich Beispiele. Die Einteilung in Kopulativ- und Determinativkomposita ist hier nicht einfach. Im Allgemeinen sind die lyrischen Formen eher kopulativ, die technischen Ausdrücke eher determinativ (vgl. Dudenredaktion 1998, S. 450). Beispiele: *grinsheucheln*, *spritzlöten*

  Außerdem treten noch Zusammensetzungen von Infinitiven auf, die üblicherweise auf die Zweitglieder *bleiben*, *lassen* und *lernen* beschränkt sind (vgl. Fleischer und Barz 1995, S. 296). Diese Formen werden nach der Rechtschreibreform getrennt geschrieben, ehemals war die Zusammenschreibung üblich.

Bei den verbalen Komposita macht die Getrenntschreibung der finiten Formen es nötig, sich allein auf die infiniten zusammen geschriebenen Formen zu beschränken. Finite Formen kommen im Textfluss vor, wo deren Konstituenten mehrere Wörter weit von einander entfernt positioniert sein können. Die Erkennung solcher Elemente ist auch vom Standpunkt der Syntax aus kein triviales Problem.

1.4 Sonderfälle

Wie es oft bei der Untersuchung sprachlicher Phänomene der Fall ist, gibt es auch im Bereich der Wortbildung und speziell bei der Komposition Ausnahmen, die sich nicht den herausgearbeiteten allgemeinen Regeln unterwerfen wollen. Teilweise ist es nicht einfach eindeutige Beschreibungen anzugeben, da verschiedene Quellen nicht selten verschiedener Ansicht sind, wie bestimmte Erscheinungen zu interpretieren sind.

Einige dieser Ausnahmen werden an dieser Stelle in Kürze besprochen. Die Possessivkomposita heben sich strukturell nicht von den „normalen" Determinativkomposta ab, entsprechen aber nicht deren semantischen Eigenschaften. Bei Zusammenrückungen und Zusammenbildung hingegen handelt es sich um Grenzfälle der Komposition mit anderen Wortbildungsarten.

1.4.1 Possessivkomposita

Possessivkomposita sind nach Glück (2000, S. 539) exozentrische Komposita, die zwar wortsyntaktisch dem Determinativkompositum entsprechen, semantisch aber keine Untergruppe der im Grundwort bezeichneten Einheit bilden, so ist z. B. ein *Langbein* kein langes Bein, sondern eine Person mit langen Beinen. Possessivkomposita treten hauptsächlich als Personen-, Tier- und Pflanzenbezeichnungen auf und sind nur noch in geringem Maße motiviert. Man kann bei den Possessivkomposita annehmen, dass ihr Bestand lexikalisiert ist und okkasionale Bildung eher selten, obwohl prinzipiell möglich sind. Mehr zur Handhabung lexikalisierter Zusammensetzungen unter 1.5.1.

1.4.2 Zusammenbildungen

Laut Glück (2000, S. 812) handelt es sich bei den Zusammenbildungen um Grenzfälle zwischen Ableitung und Komposition, bei denen die Wortbildungstruktur nicht eindeutig ist. So kann z. B. *Autofahrer* als Zusammensetzung gesehen werden (*auto + fahrer*) oder als Ableitung (*autofahren + -er*).

Erben (2000, S. 35) bezeichnet Zusammenbildungen als besondere Art der Ableitung, wo eine Wortgruppe als Basis einer suffixalen Ableitung dient. Ein Komposita-Charakter wird ausgeschlossen.

Von Inkorporation dagegen spricht Eichinger (2000, S. 136) und meint damit eine allmähliche Univerbierung von Elementen, die in der syntagmatischen Abfolge eines Satzes nebeneinander stehen. Damit grenzt Eichinger die Zusammenbildungen von den Komposita ab, da das vermeintliche Grundwort einer Zusammenbildung nicht als eigenständiges Lexem auftritt. Vielmehr wird eine Art substantivisches Suffix

angenommen. Bei Adjektiven ist die Argumentation ähnlich, auch wenn in diesem Fall die Interpretation als echtes Kompositum häufiger möglich ist.

Für die Zwecke dieser Arbeit ist die Sichtweise von Eichinger die nützlichste. Sofern diese uneigenständigen Substantiv-Suffixe keinen Lexemcharakter haben, werden sie im Lexikon und damit bei der Segmentierung nicht berücksichtigt (siehe 1.5.1). Somit werden eventuelle Zusammenbildungen auch nicht fälschlich als Komposita erkannt. Ist ein Suffix im oben genannten Sinne dennoch im Lexikon enthalten, handelt es sich um ein homonymes Lexem, womit eine Interpretation als Kompositum wieder legitim ist.

1.4.3 Zusammenrückungen

Als Zusammenrückungen bezeichnet Erben (2000, S. 34) einen Sonderfall der Zusammensetzung, wo eine syntaktische Gruppe – unter Beibehaltung der Wortfolge und eventueller flexivischer Relationsmorpheme – zu einem Wort bzw. einem Kompositionsglied eines neuen Wortes „zusammengerückt" wird, z. B. *Brave-Mädchen-Image* als Kompositum aus Zusammenrückung und Substantiv.

Eichinger (2000, S. 31) würde Formen wie z. B. *Vergißmeinnicht* am liebsten in den Bereich der Konversion verbannen, sieht sie aber auch ähnlich wie die Zusammenbildungen im Umfeld der Inkorporation. Auch Fleischer und Barz (1995, S. 213) argumentieren ähnlich bei Satz- und Wortgruppen-Konversionen.

Das obige Beispiel *Brave-Mädchen-Image* und auch *Graue-Maus-Dasein* sind bei Fleischer und Barz Komposita mit einem Erstglied, das aus einer Wortgruppe oder einem ganzen Satz besteht, die Bezeichnung „Zusammenrückung" wird hier nicht gebraucht.

Im Modell müssen solche Bildungen unberücksichtigt bleiben, da die Erkennung dieser Formen als Kompositionsglied eine satzsyntaktische Analyse nötig machen würde. Schließlich können ganze, sogar komplexe Sätze als Zusammenrückungen vorkommen. Die Fälle von Zusammenrückungen, die als Konversionen erklärt werden können, entfallen ohnehin aus dem Kompositionsschema.

1.5 Komposita-Segmentierung und Segment-Tagging

„Sprachliche Ausdrücke, gleichgültig unter welchem Gesichtspunkt man sie betrachten will, sind nicht als Mengen diskreter Einheiten gegeben, sondern als Kontinua. Für viele sprachpraktische und sprachwissenschaftliche Manipulationen ist es jedoch sinnvoll, die Kontinua in Mengen diskreter Einheiten, nämlich in Folgen von Segmenten aufzulösen. Das Verfahren einer solchen Auflösung heißt Segmentierung" (Glück 2000, S. 616).

Deutsche Komposita können trotz ihrer komplexen Struktur als Paradebeispiele für die erwähnten Kontinua herhalten. Obwohl sie aus mehreren Wörtern bestehen, sind die Grenzen zwischen den Bestandteilen durch die Zusammenschreibung üblicherweise nicht markiert. Ein menschlicher Sprecher führt ebenfalls eine Segmentierung durch, sofern die gesamte Struktur nicht bereits lexikalisiert ist.

Eng mit dem Vorgang der Segmentierung ist das linguistische Tagging verbunden. Tagging im weiteren Sinne ist eine Bezeichnung für eine in der Regel automatische Annotation von Sprachdaten. Unter Tagging im engeren Sinne wird speziell die Annotation von Wortformen in einem laufenden Text mit Wortartenkategorien verstanden. (vgl. Glück 2000, S. 720)

Tagging in Verbindung mit der Segmentierung von Kompostia bedeutet die Annotation der identifizierten Segmente mit grammatischen Daten, die im Lexikon enthalten sind.

Im Folgenden werden vor allem die linguistischen Grundannahmen beschrieben, die mit der Segmentierung und dem anschließenden Tagging verbunden sind. Wie die Vorgänge im Modell implementiert werden, wird in Kapitel 3 besprochen. Weiterhin wird an dieser Stelle die zentrale Rolle des Lexikons für die erfolgreichen Segmentierungs- und Taggingprozesse dargelegt.

1.5.1 Definition des Segmentbegriffes

Will man nun eine Segmentierung eines Kompositums durchführen, muss zuerst festgelegt werden, welchen Charakter die gesuchten Segmente haben sollen. Bei morphologischen Konstituentenanalysen werden Wortbildungsprodukte auf Morpheme zurückgeführt, wobei Grundmorpheme als auch reine Wortbildungsmorpheme (Präfixe, Infixe, Suffixe) bestimmt werden. Für den Zweck dieser Arbeit ist eine derart weitgehende Analyse unnötig, wenn nicht sogar kontraproduktiv, da es schon schwierig genug ist, aufgrund der Kompositionsglieder die Gesamtbedeutung eines Kompositums zu erschließen. Eine tiefere Analyse muss nicht unbedingt eine höhere Durchsichtigkeit zur Folge haben.

Bei Komposita können die beiden Kompositionsglieder, wie bereits erwähnt wurde, entweder aus Simplizia oder wiederum komplexen Wortbildungsprodukten bestehen, die sowohl Ableitungen und Konversionsergebnisse (vgl. Eichinger 2000, S. 116) als auch Zusammensetzungen sein können. Trotz dieser vielfältigen Möglichkeiten ist allen Kompositionsgliedern der Wortcharakter gemeinsam.

Nimmt man nun das Wort als Segment an, ist ein Kompositionsglied, das nicht selbst aus einem Kompositum besteht, nicht weiter analysierbar. Die Segmentierung endet hier. Handelt es sich bei dem Kompositionsglied um ein weiteres Kompositum, können wieder zwei Wörter identifiziert werden. Dieser Prozess wird so lange wiederholt, bis keine weiteren Segmentierungsschritte mehr durchgeführt werden können.

Es stellt sich allerdings die Frage, ob eine solche Analyse nicht immer noch zu weit geht. Bei motivierten und durchsichtigen Konstruktionen ist eine genauere Analyse sicher sinnvoll, mit zunehmender Idiomatisierung nimmt der Nutzen jedoch ab, da die Bedeutung gar nicht oder kaum aus den Bedeutungen der Glieder gefolgert werden kann. „Durch Idiomatisierung entsteht die Notwendigkeit, komplexe Ausdrücke ins Lexikon einer Sprache aufzunehmen" (Glück 2000, S. 285). Ist eine Zusammensetzung erst einmal lexikalisiert, kann man sich die Konstituentenanalyse, die ja letztendlich der Bedeutungserschließung dienen soll, sparen, denn die Bedeutung kann in diesem Fall einfach nachgeschlagen werden (vgl. Eichinger 2000, S. 10).

In dieser Arbeit wird also eine Komposita-Segmentierung vorgeschlagen, die sich an der Opposition des Begriffspaares lexikalisiert/nicht-lexikalisiert orientiert. Ein ähnliches Vorgehen wird auch bei Rackow et al. (1992, S. 2) beschrieben. Die weitere Segmentierung eines bereits identifizierten Segmentes kann abgebrochen werden, wenn das Segment trotz seiner weiterhin komplexen Struktur lexikalisiert ist. Alternativ dazu kann die Segmentierung auch fortgesetzt werden. Da sowohl der komplexe Ausdruck als auch dessen Segmente im Lexikon enthalten sind, werden in diesem Fall einfach beide Interpretationen wiedergegeben. Man kann diese Interpretationen dann als strukturelle Mehrdeutigkeiten betrachten (siehe 1.5.3).

Nun muss noch geklärt werden, wie die Lexikalisierung eines komplexen Ausdrucks festgestellt werden kann. Im Falle eines menschlichen Sprechers ist dies einfach: kennt er die Bedeutung eines komplexen Ausdrucks, ohne dass er dessen Komponenten analysieren muss, ist dieser Ausdruck lexikalisiert. Für das in dieser Arbeit entwickelte Finite-State-Modell kann ähnlich verfahren werden. Ist ein komplexer Ausdruck als Ganzes im Lexikon des Modells gespeichert, gilt dieser Ausdruck als lexikalisiert.

Dies kann man auf alle Wörter erweitern. Ein Wort ist lexikalisiert bzw. ein Lexem, wenn es im Lexikon enthalten ist. Als Referenz-Lexikon dient das im Modell integrierte Lexikon. Umgekehrt wird ein Wort, das nicht im Lexikon enthalten ist, markiert mit der Eigenschaft „nicht-lexikalisiert". In diesem Fall wird eine weitere Segmentierung versucht, wie sie oben beschrieben wurde. Hinzu kommt noch der Fall, dass ein komplexer Ausdruck nicht lexikalisiert ist und auch die Segmentierung keine weiteren Segmente im Lexikon finden kann.[3] In diesem Fall ist die Segmentierung des komplexen Ausdruckes gescheitert. Ein Segment wird nur als solches erkannt, wenn es lexikalisiert ist.

Daraus ergibt sich nebenbei auch eine einfache Definition von Okkasionalismen. Ist ein Wort nicht lexikalisiert, kann aber gemäß des verwendeten Lexikons in Lexeme zerlegt werden, handelt es sich um eine erfolgreich identifizierte okkasionale Bildung.

Im Falle der Komposita ist eine solche Segmentdefinition jedoch noch nicht völlig zufriedenstellend. Sie lässt nämlich die Fugenelemente außer acht. Zwar werden sie dem Erstglied der Komposition zugerechnet, können aber kaum als Bestandteile eines Lexems betrachtet werden, da sie oft genug nicht einmal dem Flexionsparadigma

[3] Bei einem menschlichen Sprecher kann man dies als Wissenslücke interpretieren

der mit ihnen verbundenen Lexeme entsprechen (vgl. Erben 2000, S. 64). Erst recht nicht können sie selbst als eigenständige Lexeme gewertet werden. Im Allgemeinen werden sie nicht einmal als Morpheme betrachtet, da sie außer bei einigen Fällen von vermeintlicher Pluralmarkierung semantisch leer sind (vgl. Fleischer und Barz 1995, S. 137). Einfach ignorieren kann man sie jedoch nicht, da eine computergestützte Segmentierung an die in einer Zeichenkette vorhandenen Symbole gebunden ist. Außerdem können sie, wie später noch dargelegt wird, trotz ihrer semantischen Leere aufgrund ihrer distributiven Eigenschaften von Nutzen sein.

Es ist in diesem Fall am einfachsten, diese Fugenelemente ebenfalls als mögliche Segmente zu betrachten. Auch die Fugenelemente sind lexikalisiert in dem Sinne, dass sie im Lexikon des Modells enthalten sind. Eine Aufnahme der Fugenelemente ins Lexikon ist dagegen aus Implementationsgründen sinnvoll, da der Segmentierungsprozess Zeichenketten analysiert und diese aufgrund des Lexikons identifiziert. Sowohl Lexeme als auch Fugenelemente stellen mögliche Segemente dar und sind zu allererst Zeichenketten. Erst durch den Vergleich mit dem Lexikon, der Zeichen für Zeichen vorgenommen wird, kann eine Unterscheidung von Lexem und Fugenelement stattfinden.

1.5.2 Das Lexikon und linguistisches Tagging

Mit der Segmentierung allein ist es noch nicht getan. Die Segmente müssen linguistisch annotiert werden, damit eine weitere Verarbeitung möglich ist. Während die Segmentierung das Lexikon als Bestandsaufnahme von potentiellen Segmenten auffasst, werden durch das Tagging die im Lexikon einzelnen Einträgen zugeordneten Informationen den identifizierten Segmenten angefügt. Das Lexikon ist also für beide Vorgänge, Segmentierung und Tagging, grundlegend. Dies bedeutet, dass das Modelllexikon mindestens zwei Arten von Daten enthalten muss:

- Repräsentationen der einzelnen Lexeme als Zeichenketten. Diese sind notwendig, um die gesuchten Segmente überhaupt identifizieren zu können. Die Buchstabenfolgen eines unsegmentierten komplexen Wortes müssen mit den Buchstabenfolgen der Lexikoneinträge verglichen werden können. Die Art wie die Gesamtheit der Lexikoneinträge organisiert ist, wird auch als Makrostruktur des Lexikons bezeichnet (vgl. Carstensen et al. 2001, S. 397). Wie genau das möglichst effizient mit Hilfe von Finite-State-Mitteln umgesetzt werden kann, wird später bei der Modellierung des Lexikons ausführlich besprochen (siehe 3.1.1).
- Informationen, die die einzelnen Lexeme linguistisch kategorisieren. Dies sind die Daten, die durch das Tagging den Lexemen ihre grammatische Bedeutung verleihen. So werden in diesem Modell hauptsächlich Informationen bezüglich Wortart und Flexionsklasse verwendet. Es werden auch wortartspezifische Daten annotiert, wie z. B. Genus beim Substantiv. Man spricht hier auch von

der Mikrostruktur eines Lexikons bzw. eines Lexikoneintrags (vgl. Carstensen et al. 2001, S. 398). Eine Menge von grammatischen Informationen, die zu einer Interpretation eines Eintrages gehören, werden auch als Tagsets bezeichnet. Einem Eintrag, der lexikalisch mehrdeutig ist, sind mehrere Tagsets zugeordnet.

Da beide Vorgänge abhängig sind von bestimmten Suchprozessen im Lexikon, bietet sich eine parallele Anwendung von Segmentierung und Tagging an. Wird ein Segment identifiziert, wird es direkt mit den entsprechenden Lexikoninformationen annotiert, und nicht erst nachdem alle Segmente gefunden worden sind. Auf diese Weise werden Mehrfachaufrufe derselben Einträge verhindert. Das Ergebnis einer Segmentierung eines komplexen Wortes mit parallelem Tagging ist eine Menge identifizierter Segmente mit entsprechenden grammatischen Annotationen.

1.5.3 Lexikale und strukturelle Ambiguität

Bei Glück (2000, S. 37) wird Ambiguität (auch Ambivalenz, Mehrdeutigkeit, Vieldeutigkeit) unter zwei Gesichtspunkten definiert. Laut der ersten Auffassung ist Ambiguität eine grundlegende Eigenschaft der meisten Lexeme, Derivations- und Flexionsmorpheme in einer flektierenden Sprache wie dem Deutschen. Die Mehrdeutigkeit der Lexeme wird in der Wortbildung teils disambiguiert, teils vermehrt. Diese lexikalische Ambiguität wird in der europäischen strukturellen Semantik als Homonymie, Polysemie und Multisemie untersucht.

Die zweite Auffassung bezeichnet Ambiguität auf der Ebene der Syntax als Eigenschaft eines komplexen Zeichens, aufgrund unterschiedlicher Strukturbeschreibungen mehr als eine Interpretation zu besitzen. In diesem Sinne unterscheiden sich ambige Ausdrücke von polysemen, welche als mehrdeutig aufgrund von nicht-strukturellen Faktoren verstanden werden. Die in dieser Arbeit als strukturelle Ambiguität bezeichnete Art der Ambiguität hat ihren Grund dementsprechend in der Mehrdeutigkeit der grammatischen Strukturzusammenhänge und Formkennzeichnungen. Da komplexe Zeichen nicht nur auf syntaktischer Ebene auftauchen und Komposita eindeutig komplex sind, kann diese Auffassung auch auf die Strukturbeschreibungen von Komposita angewendet werden.

Während des Segmentierungsvorganges eines Kompositiums können prinzipiell beide Arten von Ambiguität auftreten. Lexikalische Ambiguität ist gegeben, wenn einem identifiziertem Segment mehrere Einträge im Lexikon entsprechen. Da in dem verwendeten Lexikon keine semantischen Informationen kodiert sind, handelt es sich hier immer um Arten der Homonymie und nicht der Polysemie. Die Einträge unterscheiden sich aufgrund einzelner oder auch mehrerer grammatischer Eigenschaften wie z. B. Genus (*die Kiefer – der Kiefer*) oder verschiedenen Pluralformen (*Bank* : *Bänke – Banken*), wobei in beiden Fällen unterschiedliche Flexionsklassen annotiert sind. Lexikalische Mehrdeutigkeit tritt also beim Tagging der identifizierten Segmente auf.

Strukturelle Ambiguität kann unabhängig vom Tagging allein durch die reine Segmentierung gegeben sein. Können zum Beispiel mehrere alternative Zerlegungen eines Kompositums aufgrund verschiedener Morphem- oder Segmentgrenzen ausgemacht werden, müssen beide Segmentierungen akzeptiert werden, sofern nicht andere Kriterien zur Disambiguierung angewendet werden. Ein Beispiel für eine nicht-eindeutige Segmentierung ist das Kompositum *Druckerwartung*. Es kann segmentiert werden in zwei vorerst nicht annotierte Strukturen:

(1) *drucker* + *wartung*
(2) *druck* + *erwartung*

Vermutlich ist Lesart (1) die wahrscheinlichere von beiden, dennoch kann Lesart (2) nicht einfach ignoriert werden, da sie prinzipiell möglich ist. Hinzu kommt, dass der Segmentierungsmechanismus über keine Informationen über die Wahrscheinlichkeit bestimmter Lesarten verfügt.

Noch uneindeutiger wird das Ergebnis des Analyseprozesses, wenn sowohl lexikalische als auch strukturelle Ambiguität vorliegt. Werden die beiden Segmentierungsvorschläge von *Druckerwartung* entsprechend annotiert, ergeben sich weitere Lesarten. So kann *druck* sowohl als Substantiv (z. B. im Sinne von psychischem Druck) als auch als Verbstamm (der Vorgang des Druckens) interpretiert werden. Es ergeben sich also drei Lesarten:

(1) *drucker* (N) + *wartung* (N)
(2) *druck* (N) + *erwartung* (N)
(3) *druck* (V) + *erwartung* (N)

Jede dieser Lesarten könnte man wiederum als lexikalische Mehrdeutigkeit des gesamten Zeichens *Druckerwartung* betrachten, die nur durch die Einbettung in einen Kontext disambiguiert werden könnte. Zum Beispiel wird in einer technischen Geräteanleitung wohl nur die Lesart (1) akzeptiert. Tatsächlich sind dies Lesarten, die alle als grammatische korrekt akzeptiert werden müssen.

An dieser Stelle wurden jedoch noch keine Disambguierungsmechanismen vorgestellt, die falsche Lesarten, die ebenfalls auftauchen können und werden, eliminieren. Ein naiver Algorithmus, der Segmentierung und Tagging nur aufgrund der Lexikoninformationen durchführt, wird mehr (falsche) Lesarten finden.

(1) *drucker* (N) + *wartung* (N)
(2) *druck* (N) + *erwartung* (N)
(3) *druck* (V) + *erwartung* (N)
(4)* *druck* (N) + *er* (I) + *wartung* (N)
(5)* *druck* (V) + *er* (I) + *wartung* (N)

Da Fugenelemente (hier mit (I) für Interfix bezeichnet) ebenfalls im Lexikon enthalten sind, werden die Lesarten (4) und (5) erkannt. Erst distributive Regeln von Morphemen und Fugenelementen können solche falschen Segmentierungen ausschließen.

So würde Lesart (4) ausgeschlossen werden, sobald berücksichtigt wird, dass sich wenn überhaupt nur Substantive, die im Plural die Endung *-er* aufweisen, mit dem Fugenelement *-er-* verbinden. Lesart (5) entfällt, wenn man die Regel anwendet, dass Verbstämme üblicherweise ohne Fugenelement stehen oder mit *-e-* in einigen Fällen. Mehr zur Verwendung von Regularitäten an der Kompositionsfuge zur Disambiguierung von Segmentationsergebnissen unter 1.6.4.

Auch als ein Fall von struktureller Ambiguität kann der bereits erwähnte Sachverhalt betrachtet werden, wenn Komposita als Ganzes im Lexikon enthalten sind, die eigentlich noch weiter segmentiert werden könnten und parallel auch weiter segmentiert werden.

(1) *reise* (N) + *schreibmaschine* (N)
(2) *reise* (V) + *schreibmaschine* (N)
(3) *reise* (N) + *schreib* (V) + *maschine* (N)
(4) *reise* (V) + *schreib* (V) + *maschine* (N)

Solange noch keine Disambiguierung stattgefunden hat, kann die Anzahl aller (inklusive der grammatisch falschen) Lesarten mithilfe der folgenden Formel berechnet werden:

$$\text{Anzahl der Lesarten} = \sum_{i=1}^{|S_g|} \Big(\prod_{j=1}^{|S_i|} |s_{i,j}| \Big)$$

Hier steht $s_{i,j}$ (wo $j = 1, 2, \ldots$) für ein einzelnes Segment und $|s_{i,j}|$ für die Anzahl der lexikalischen Ambiguitäten dieses Segments. S_i (wo $i = 1, 2, \ldots$) ist die Menge aller Segmente $s_{i,j}$, die zu einer Segmentierungsart[4] gehören. $|S_i|$ ist die Anzahl der Segmente einer Segmentierungsart S_i . Sei $k = |S_i|$, dann ist $S_i = \{s_{i,1}, s_{i,2}, \ldots, s_{i,k}\}$. Die Indeces der Segmente entsprechen der Reihenfolge der Segmente. S_g ist die Menge aller verschiedenen Segmentierungsarten S_i und $|S_g|$ ist die Anzahl dieser Segmentierungsarten. Sei $n = |S_g|$ dann ist $S_g = \{S_1, S_2, \ldots, S_n\}$.

Die Anzahl der Lesarten von *Druckerwartung* berechnet sich demnach beispielsweise wie folgt:

$$S_g = \left\{ \begin{array}{lllll} S_1 = & \{s_{1,1} = drucker & , & s_{1,2} = wartung & \}, \\ S_2 = & \{s_{2,1} = druck & , & s_{2,2} = erwartung & \}, \\ S_3 = & \{s_{3,1} = druck & , & s_{3,2} = er \quad , \quad s_{3,3} = wartung & \} \end{array} \right\}$$

$$\begin{array}{lll} |s_{1,1}| = 1 & |s_{1,2}| = 1 & \\ |s_{2,1}| = 2 & |s_{2,2}| = 1 & \\ |s_{3,1}| = 2 & |s_{3,2}| = 1 & |s_{3,3}| = 1 \end{array}$$

[4] Als Segmentierungsart wird die reine Segmentierung ohne Tagging bezeichnet. Zwei Segmentierungsarten sind gleich, wenn alle Segmentgrenzen übereinstimmen. Bei der Segmentierung von *Druckerwartung* gehören Lesarten (2) und (3) zu einer Segmentierungsart sowie Lesarten (4) und (5) zu einer anderen.

$$
\begin{aligned}
\sum_{i=1}^{|S_g|}\Big(\prod_{j=1}^{|S_i|}|s_{i,j}|\Big) &= \sum_{i=1}^{3}\Big(\prod_{j=1}^{|S_i|}|s_{i,j}|\Big) \\
&= \prod_{j=1}^{|S_1|}|s_{1,j}| + \prod_{j=1}^{|S_2|}|s_{2,j}| + \prod_{j=1}^{|S_3|}|s_{3,j}| \\
&= \prod_{j=1}^{2}|s_{1,j}| + \prod_{j=1}^{2}|s_{2,j}| + \prod_{j=1}^{3}|s_{3,j}| \\
&= |s_{1,1}|\cdot|s_{1,2}| + |s_{2,1}|\cdot|s_{2,2}| + |s_{3,1}|\cdot|s_{3,2}|\cdot|s_{3,3}| \\
&= 1\cdot 1 + 2\cdot 1 + 2\cdot 1\cdot 1 \\
&= 1 + 2 + 2 \\
&= 5
\end{aligned}
$$

Die Segmente $s_{2,1}$ und $s_{3,1}$ haben jeweils zwei lexikale Deutungsmöglichkeiten. Alle übrigen Segmente haben jeweils eine lexikale Deutungsmöglichkeit. Die Anzahl der Lesarten einer Segmentierungsart ist gleich dem Produkt der lexikalen Deutungsmöglichkeiten aller Segmente dieser Segmentierungsart. Für S_1 gibt also genau eine Lesart ($1\cdot 1$). Für S_2 gibt es demnach genau zwei Lesarten ($2\cdot 1$) und für S_3 ebenfalls zwei Lesarten ($2\cdot 1\cdot 1$). Anschließend wird die Anzahl der Lesarten aller Segmentierungsarten summiert. Insgesamt gibt es also fünf Lesarten. Wie gesagt, handelt es sich hier um naive Lesarten, die nicht linguistisch bereinigt sind. Die letztendliche Anzahl der grammatisch korrekten Lesarten lässt sich nicht anhand der Kombinationen von lexikaler und strukturaler Ambiguität berechnen. Es ist prinzipiell möglich, dass alle, einige oder keine Lesarten korrekt sind. Sind die Informationen im Lexikon unvollständig, bedeutet dies, dass einige unter Umständen korrekte Lesarten erst gar nicht gefunden werden.

1.6 Die Kompositionsfuge

Die Morphemgrenze zwischen den Konstituenten einer Wortbildungskonstruktion wird als Fuge bezeichnet (vgl. Fleischer und Barz 1995, S. 136). Unterschieden wird dabei zwischen Derivations- und Kompositionsfuge, wobei an dieser Stelle ausschließlich die Kompositionsfuge behandelt wird. An der Kompositionsfuge kann es zu verschiedenen Erscheinungen kommen. So können bestimmte Elemente entweder getilgt oder andere Elemente hinzugefügt werden. Die hinzugefügten Elemente werden in der entsprechenden Literatur als Fugenelemente oder Interfixe bezeichnet.

Langer (1998, S. 3) verwendet eine andere Terminologie als die gemeinhin übliche, da seiner Meinung nach die Bezeichnung Fugenelement oder Interfix darauf hindeutet, dass diese Elemente zwischen den Kompostionsgliedern auftreten. Da sie aber

zum Erstglied gezählt werden (siehe unten), bilden sie zusammen mit dem Erstglied die Kompositionsform des Erstglieds. Von der Kompostionsform ausgehend bezeichnet Langer die Fugenelemente als Kompositionssuffixe des Erstglieds. Diese Bezeichnung ist wohl nicht weniger problematisch als die übliche, da eine Kategorisierung als Suffix wiederum auf einen Morphemcharakter hindeutet, der bei den Fugenelementen zumindest diskutiert werden kann.

Dafür, dass die Fugenelemente dem Erstglied zugerechnet werden müssen, zählt Fuhrhop (1998, S. 187) drei Argumente auf: Erstens wird das Fugenelement vom Erstglied bestimmt. Zweitens steht die Form des Fugenelements im engen Zusammenhang mit dem Flexionssystem des Erstgliedes. Drittens verbleibt das Fugenelement bei Koordination beim Erstglied, wie z. B. bei *Frühlings- und Herbsttage*. Analog zu Langer spricht Fuhrhop von Kompostionsstammformen des Erstgliedes. Als Kompositionsstammform wird die Form des Erstgliedes bezeichnet, die letztendlich in die Komposition eingeht. Tritt ein Fugenelement auf, wird es zur Kompositionsstammform gerechnet. Dieser Begriff wird im Folgenden weiterverwendet.

Die Fugenelemente sind überwiegend auf Substantiv- oder Verbstämme als Erstglieder beschränkt. Obwohl die Fugenelemente wie die Flexionsmorpheme der entsprechenden Erstglieder aussehen, können sie nicht als Erscheinungen der Flexion gewertet werden, da mitunter auch Fugenelemente interfigiert werden, die nicht zum Flexionsparadigma des entsprechenden Ersgliedes gehören, z. B. *Liebesdienst* mit dem Fugenelement *-s-*. In einigen Fällen kann ein Fugenelement (oder Interfix) noch den Plural eines Erstgliedes markieren, wenn die Form mit der Pluralform des entsprechenden freien Lexems übereinstimmt, aber auch das ist nicht in allen Fällen sicher. Nicht selten wird die Fuge aufgrund von Kompositionsmustern übernommen, z. B. *Strahlenkranz* gegenüber *Tortenstück* – viele Strahlen, aber nur eine Torte (vgl. Dudenredaktion 1998, S. 495).

Bei manchen Erstgliedern werden durch die Fugengestaltung Homonymendifferenzierung, Wortartenunterschiede und auch semantische Differenzen gekennzeichnet (vgl. Fleischer und Barz 1995, S. 137), z. B. *Landesverteidigung* und *Landebahn*, wo im ersten Kompositum *Land* im Sinne von *Staat* verwendet wird, während im zweiten Fall der Verbstamm von *landen* vorliegt. Diese Eigenschaften wurden bereits im Zusammenhang mit der Segmentierung der Komposita erwähnt und sind der Hauptgrund für die Beschäftigung mit den Fugenelementen in dieser Arbeit.

Fleischer und Barz erklären die Schwierigkeiten mit der Herausarbeitung von Regularitäten in der Distribution von Fugenelementen mit dem „Widerstreit zweier Regularitätskonzeptionen" – der „Orientierung an grammatischen Regeln" einerseits und der „Orientierung an lexikalischen Mustern" andererseits. So kann sich die an ein bestimmtes Erstglied gebundene Fugengestaltung lexikalisierter Komposita gegen grammatische Regularitäten durchsetzen.

Ob ein Fugenelement gesetzt wird oder nicht, hängt laut Dudenredaktion (1998, S. 495) von der Beschaffenheit des Erstgliedes ab:

a) insbesondere von der Wortart des Bestimmungswortes,

b) von seiner morphologischen Grundausstattung (Flexionsklasse),

c) von seiner Lautstruktur (Umfang, Silbenzahl, Auslaut),

d) von seiner Wortbildungsstruktur (davon, ob es sich um ein Simplex, eine Ableitung oder eine Zusammensetzung handelt)

e) zum Teil auch davon, ob das Kompositum nur eine oder mehrere der im Bestimmungswort bezeichneten Sachen oder Personen voraussetzt,

f) von regionalen Bedingungen.

Eine systematische Beschäftigung mit dem Auftreten der Fugenelemente liefert Fuhrhop (1998, S. 187–220), in der die obigen klassischen Kriterien genauer untersucht werden und nach Funktionen der Fugenelemente gefragt wird.

Fuhrhop konzentriert sich anschließend auf die produktiven Fugenelemente und sucht nach den Grundlagen der Intuition, mit der Sprecher Fugenelemente setzen. Die unproduktiven Fugenelemente werden ausdrücklich ausgespart.

Sowohl Dudenredaktion als auch Fleischer und Barz nennen als Fugenelemente *-(e)s-*, *-e-*, *-(e)n-*, *-er-* und *-ens-*. Bei Fuhrhop werden die Fugenelemente *-s* und *-es*, sowie *-en* und *-n* deutlich unterschieden. Der fehlende Bindestrich auf der rechten Seite des Fugenelementes ist durch die Zugehörigkeit zur Kompositionsstammform begründet.

1.6.1 Bildung der Kompositionsstammformen mit Fugenelementen

Im Folgenden werden die von Fuhrhop (1998) gefundenen Regularitäten zusammengestellt und aufgrund der Einbindungsmöglichkeiten in das beschriebene Modell kommentiert.

Fuhrhops Hauptkriterium zur Erstellung einer Systematik der Fugenelemente ist deren Produktivität, ein Aspekt, der in deskriptiven Untersuchungen übergangen wird. Zunächst werden die Fugenelemente nach ihrer Form eingeteilt, die Erstglieder werden aufgrund dieser Einteilung auf die Entstehung von Basismengen oder Reihen untersucht. Die Kriterien, die den Reihen zugrunde liegen, werden gesammelt und an anderen Fällen geprüft.

Kompositionsstammformenbildung mit *-s*

Das Fugen-*s* nimmt eine Sonderstellung unter den Fugenelementen ein, da es als einziges produktiv neben der paradigmischen Form unparadigmisch auftritt. Mit

dem Merkmalspaar paradigmisch/unparadigmisch ist hier das Auftreten zu den Fugenelementen homophoner und homographer Flexionssuffixe in den entsprechenden flektierten Wortformen des Erstglieds gemeint. So tritt *-s* auch in Kompositionsstammformen auf, zu deren Flexionsparadigma es aber nicht gehört und somit als unparadigmisch bezeichnet wird. Entspricht das Fugen-*s* dem Genitiv-*s* des Erstgliedes, ist es paradigmisch. Eine andere interessante Eigenschaft beruht auf der Tatsache, dass *-s* niemals in Kompostionsstammformen vorkommt, deren Pluralform mit *-s* gebildet wird, auch dann nicht, wenn die Genitiv-Singular-Form formgleich ist (vgl. FUHRHOP 1998, S. 197).

Bei den Kompositionsstammformen mit unparadigmischen Fugen-*s* nennt FUHRHOP vier Klassen femininer Substantive. Die Erstglieder sind:

a) suffigierte feminine Substantive (z. B. *Tapferkeitsmedallie, Versicherungsvertreter*);

b) alte Ableitungen mit *-t* aus Partikelverben (z. B. *Abfahrtszeit*, *Aufsichtsverpflichtung*);

c) (synchron) Simplizia (z. B. *Anstaltsleiter*, *Arbeitserlaubnis*);

d) formal ein Kompositum mit nur einem Beleg: *Hochzeitsfeier*.

In a) bilden alle diese Substantive bis auf Formen mit *-ei*, *-erei* und *-in* ihre Kompositionsstammformen regelmäßig mit Fugen-*s*. In Fällen einer expliziten Pluralbedeutung ist aber auch die entsprechende Pluralform möglich. Bei b) – d) sind auffällige Gemeinsamkeiten zu erkennen. So sind alle Belege mehrsilbig und enden auf *-t*. Es ergibt sich zusammenfassend, dass Mehrsilbigkeit und damit verbundene morphologische Komplexität notwendige Bedingungen für eine unparadigmische Kompsitionsstammform mit *-s* sind.

Die paradigmischen Formen des Fugen-*s* sind schwieriger zu erfassen. Da das paradigmische *-s* niemals einer Pluralform entspricht, kann es nur der Genitiv-Singular-Form formgleich sein (vgl. FUHRHOP 1998, S. 198).

Die Gruppe der deverbalen Substantive weist keine eindeutigen Kompositionsstammformen auf. Bei den deverbalen Substantiven auf *-en* sind es die einfachen oder präfigierten deverbalen neutralen Substantive, die ihre Kompositionsstammform auf *-s* bilden (Beispiele: *Essensmarke*, *Wissensdurst*, *Einkommensgrenze*). Entsprechende Bildungen mit maskulinem Genus weisen keine Fugenelemente auf, z. B. *Hustensaft*. Neutrale Erstglieder vom Typ *Erdbebenvorsorge* sind selbst Komposita, die entsprechenden Verben Rückbildungen aus diesen Substantiven und weisen ebenfalls keine Fugenelemente auf. Die genannten Formen unterscheiden sich von den substantivierten Infinitiven, deren Kompositionsstammform der der zugrundeliegenden Verben entspricht, durch ihre Pluralfähigkeit. Daraus zieht FUHRHOP den Schluss, dass der Gebrauch des Fugen-*s* in dieser Gruppe mit zunehmender Lexikalisierung eintritt.

Implizite Ableitungen aus Verben weisen teilweise Fugen-*s* auf, teilweise erscheinen sie ohne Fugenelement. Die impliziten Ableitungen komplexer präfigierter Verben treten mit Fugenelement auf (z. B. *Anfangsgehalt*, *Bestandsaufnahme*, *Vorschlagsrecht*). Fuhrhop (1998, S. 201) nennt dabei einige Ausnahmen, vor allem die auf einen koronalen Frikativ auslautenden Ableitungen wie z. B. *Aufschluss*. Außerdem werden von Fuhrhop *Ausritt* und *Ausklang* als Ausnahmen bezeichnet, die fugenlose Kompositionsstammformen bilden. Dagegen stehen aber vom Autor dieser Arbeit gefundene Formen wie *Ausklangsfigur* und *Ausklangsfeier*[5]. Als Komposition mit *Ausklang* als Kompositionsstammform ohne Fugen-*s* wurde nur *Ausklangambiente* gelistet. Fuhrhop liefert selbst keine entsprechenden Beispiele. Implizite Ableitungen einfacher Verben wie z. B. *Brauchtum* und *Schlagball* treten ohne Fugen-*s* auf. Ausnahmen sind z. B. *Handelsmann*. Fuhrhop hebt hier die Mehrsilbigkeit der Ausnahmen hervor. Die Ableitungen ohne Fugenelement sind einsilbig. Damit wird die Verbindung zum unparadigmischen Fugen-*s* hergestellt, das niemals nach einsilbigen Erstglied folgt.

Das unparadigmische *-s* tritt regelmäßig nach bestimmten femininen Ableitungssuffixen auf (siehe oben). Bei maskulinen Substantiven ist das paradigmische *-s* nur nach dem Suffix *-ling* regelmäßig, z. B. *Lehrlingsgehalt* oder *Lieblingsgetränk*. Nach *-er* tritt es nur nach Ortsnamen auf, die man als lexikalisiert betrachten kann, z. B. *Eberswalde*. Ist das Zweitglied ein relationales Substantiv, tritt in bestimmten Fällen ein *-s* hinter *-er*, z. B. *Bauersfrau*, *Richterssohn*. Bei diesen Erstgliedern ist die Kompositionsstammform mit *-s* nicht die einzige, es tritt immer auch die Kompositionsstammform ohne Fugenelement auf.

Bei den neutralen Substantiven tritt *-s* regelmäßig nach den Suffixen *-sal* und *-tum* auf (Fleischer und Barz 1995, S. 139).

Einsilbige maskuline oder neutrale Substantive, die mit Fugen-*s* auftreten, gibt es nur wenige, z. B. *Amtsgericht* oder *Wolfsgrube*. Es könne keine besonderen Eigenschaften ausgemacht werden, die diese Substantive zu Basismengen zusammenfassen. Für eine Auflistung siehe Fuhrhop (1998, S. 202).

Die genannten Kriterien für das unparadigmische *-s* können mit Finite-State-Mitteln relativ einfach dem Modell hinzugefügt werden. Sowohl die Feststellung von entsprechenden Suffixen als auch die Erkennung der Komplexität der Erstglieder wird realisiert (siehe dazu 3.2). Die Wortart sowie Genus im Falle der Substantive sind im Lexikon kodiert. Da die suffigierten Feminia überwiegend Abstrakta sind, kann an den enthaltenen Flexionsinformationen erkannt werden, ob es sich um Singulariatantum handelt, bei denen eine pluralische Verwendung ausgeschlossen werden kann.

Schwieriger gestaltet sich die Einbindung der beschriebenen Regularitäten für das paradigmische *-s*. Neben der Suffixgestaltung und der Komplexität der Erstglieder spielt auch deren Wortbildungscharakteristik eine Rolle. Es müsste festgestellt werden, ob

[5]Quelle: Wortschatz Deutsch – `http://wortschatz.uni-leipzig.de`

bestimmte Substantive deverbalen Ursprungs sind, was über Zusatzinformationen im Lexikon erreicht werden könnte. Im Falle der einsilbigen Maskulina, die *-s* verlangen, ist eine Einbindung entsprechender Informationen ins Lexikon empfehlenswert und auch praktikabel, da deren Zahl nur gering ist.

Kompositionsstammformbildung mit *-es*

-es gehört zu den paradigmischen Fugen und kann damit nur bei maskulinen und neutralen Stämmen auftreten, deren Genitiv-Singular-Formen es entspricht. Fast bei allen diesen Erstgliedern kommen alternative Kompositionsstammformen vor, fugenlose oder auch mit anderen Fugenelementen. Zum Fugen-*s* verhält es sich dagegen komplementär. Notwendige Bedingung für das Auftauchen eines Fugen-*es* ist die direkte Nachbarschaft zu einer Akzent tragenden Silbe, was im Allgemeinen ein einsilbiges Erstglied bedeutet. Beispiele sind *Armeslänge*, *Haaresbreite* oder *Standesamt*. Diese Formen sind überwiegend lexikalisiert.

Für das beschriebene Modell ist eine Implementierung wieder schwierig. Da die Komposita mit *-es* überwiegend lexikalisiert sind, können als solche auch vollständig ins Lexikon aufgenommen werden oder sie werden dem Regelsystem als „lexikalisierte" Regeln hinzugefügt. Die angegebenen Regeln helfen eher *-es* in bestimmten Fällen auszuschließen, was ebenfalls nützlich sein kann.

Kompositionsstammformbildung mit *Schwa*

Die Substantive, die *Schwa* im Plural aufweisen, können auch ihre Kompositionsstammform mit *Schwa* bilden. Tritt in der Pluralform der Umlaut auf, lautet auch die Kompositionsstammform um. Beispiele sind *Hundeleine*, *Gänsefeder*. Bei *Maus* kann *Schwa* auch unparadigmisch mit fehlendem Umlaut vorkommen, z. B. *Mausefalle*. Diese unparadigmischen Bildungen sind lexikalisiert. Prinzipiell sind hier mehrere Kompositionsstammformen möglich. *Schwa* wird besonders bei pluralischer Bedeutung verwendet, wie z. B. *Ärztekammer* gegenüber *Arztpraxis*.

Bei den Verben ist *Schwa* das einzige Fugenelement, z. B. *Ankleidekabine*, *Sorgepflicht*, ansonsten treten sie in ihrer Stammform auf. Laut Fuhrhop (1998, S. 206) ist die Verbfuge nur tendenziell erfassbar und phonologisch begründet. In vielen Fällen tritt *Schwa* nach stimmhaften Obstruenten auf.

Auch hier ist eine genaue Einbindung ist das Modell schwierig. Die Fuge mit *Schwa* wird vor allem als potentielles Fugenelement berücksichtigt werden, dass auftreten kann, aber nicht auftreten muss, umgekehrt aber auch nicht ausgeschlossen werden kann. Zumindest kann so eine Einschränkung auf eine Substantivklasse und auf die Verbstämme formuliert werden.

Kompositionsstammformbildung mit *-en/-n*

Alle schwachen Maskulina bilden ihre Kompostionsstammform mit *-en*. Für Substantive wie z. B. *Bärenkäfig*, *Studentenausweis* ist dies eine hinreichende Bedingung. Endet die Grundform eines schwachen Substantivs auf *Schwa* wird das Fugen-*n* gesetzt, z. B. *Botengang*. Hier liegt Allomorphie vor.

Substantive, die ihren Plural mit *-en* bilden, können auch entsprechende Kompositionsstammformen besitzen, wobei diese im Allgemeinen auch Pluralbedeutung hat. Diese Substantive haben aber stets wenigstens zwei Kompositionsstammformen, z. B. *Schriftprobe* und *Schriftenverzeichnis*.

Daneben existieren einige Substantive, die verschiedenen Flexionsklassen angehören, aber dennoch ihre Kompositionsstammform ausschließlich mit *-en* bilden, z. B. *Gefahrenbereich*, *Nachrichtenmagazin*, *Instrumentenbau*. Für eine vollständigere Liste siehe Fuhrhop (1998, S. 207).

Auch die nicht-schwachen auf *Schwa* auslautenden Substantive bilden ihre Kompositionsstammform gewöhnlich mit *-n*, z. B. *Blumenwiese*, *Bienenstock*. Daneben gibt es aber auch viele Ausnahmen, in denen es zu *Schwa*-Beibehaltung z. B. *Erdkundelehrer*, *Schwa*-Tilgung z. B. *Eckfenster* oder *Schwa*-Ersetzung z. B. *Geschichtsbuch* kommen kann.

Die *Schwa*-Beibehaltung tritt vor allem bei Substantiven auf, die ihren Plural anders bilden als auf *-n* z. B. *Leuteschinder* oder mit deren Pluralform eine Bedeutungsveränderung (z. B. *Erdkundelehrer* aber *Urkundenfälschung*) verbunden ist.

Die Beispiele mit *Schwa*-Tilgung bilden ihren Plural mit *-n*. Hier sind alternative Kompositionsstammformen mit *-n* möglich, jedoch nie allein mit *Schwa*. Die beiden möglichen Kompositiosstammformen stehen nebeneinander, z. B. *Sprachschule* und *Sprachenschule*. Wobei die Tendenz in Richtung *-n* geht, was mitunter am Einfluss der schwachen Substantive liegt.

Schwa-Ersetzung ist schwierig zu erfassen. Fuhrhop zeigt nur einige wenige Einzelbeispiele auf, die sich keinen einheitlichen Kriterien unterwerfen.

Soweit es sich um schwache Substantive handelt, ist die Einbindung in das Modell unproblematisch, da die Fugengestaltung eindeutig ist und nur von der Flexionsklasse abhängt. Ob *-en* oder *-n* in der Fuge auftaucht kann durch eine Überprüfung auf *Schwa* im Auslaut eindeutig geklärt werden.

Ein Großteil der übrigen Substantive auf *Schwa* kann ebenfalls durch eine Analyse des Auslauts eingeordnet werden. Ist *Schwa* erst einmal festgestellt, muss das Flexionsparadigma berücksichtigt werden. Fälle, in denen mehrere Kompositionsstammformen möglich sind, müssen als solche berücksichtigt werden. Beispiele, die sich keinen besonderen Kriterien unterordnen wollen, müssen einzeln eingebunden werden.

Kompositionsstammformbildung mit *-er*

Auch *-er* taucht nur als Fugenelement auf, wenn das Substantiv im Plural *-er* aufweist, z. B. *Kinderarzt*, *Lichterglanz* oder *Wörterbuch*. Paare wie *Wortkunde – Wörterbuch* deuten darauf hin, dass die Kompositionsstammform auf *-er* bei Pluralbedeutung verwendet werden, obwohl bei einigen Formen der Mehrzahlbezug nicht hergestellt werden kann.

Ähnlich wie bei den Bildungen auf *Schwa* können hier nur Tendenzen im Modell festgehalten werden. So ist *-er* prinzipiell möglich, wird aber weder gefordert noch ausgeschlossen. Andere Kompositionsstammformen sind prinzipiell möglich und nach anderen Kriterien verteilt.

Kompositionsstammformen ohne Fugenelement

Fuhrhop konzentriert sich auf die Verteilung der Fugenelemente und verzichtet damit auf die Beschreibung von Kompositionsstammformen, die ohne diese Elemente gebildet werden. Die älteren Quellen wie Fleischer und Barz (1995, S. 139) und Dudenredaktion (1998, S. 503) nennen unter anderem die folgenden Regelmäßigkeiten, wobei hier nur die Regularitäten erfasst werden, die sich in das Finite-State-Modell einfügen lassen und von Kriterien wie morphologischen Eigenschaften, Silbenzahl und Suffixgestaltung abhängen.

Die sogenannte Nullfuge steht regelmäßig nach den Suffixen *-bold*, *-chen*, *-ei*, *-er*, *-ich(t)*, *-ig*, *-lein*, *-nis* und *-rich*. Ohne Fuge stehen auch einsilbige Feminina. Die Nullfuge steht auch nach allen Substantiven, die *-s* als Pluralendung annehmen und nach den Singulariatantum ohne charakterischen Wortausgang.

Ansonsten lassen sich lediglich Tendenzen feststellen, die schwierig zu implementieren sind, da prinzipiell auch andere Kompositionsstammformen zugelassen werden müssen.

1.6.2 Systematisierung der Regularitäten

Im vorangegangenen Unterkapitel wurden die Fugenelemente nach ihrer Form beschrieben. Für eine Implementierung ist eine Einteilung nach den Eigenschaften der Erstglieder vorteilhafter, denen anschließend Gruppen von Fugenelementen oder die Nullfuge zugeordnet werden. So können aus ähnlichen Eigenschaften Regeln resultieren, die für verschiedene Fugenelementen der Regel-Struktur nach ähnlich sind. Eigenschaften der Erstglieder, die in dem beschriebenen Modell berücksichtigt werden, sind:

	Wortart	Genus	Sg.	Pl.	Struktur	Suffix/Auslaut	Fuge
1	Substantiv			P1*			-, -e
2	Substantiv	neut/mask	S2*	P3*			-en
	Substantiv	neut/mask	S2*	P3*		-e*	-n
3	Substantiv	fem	S3	P3*		-e*	-n
4	Substantiv	neut/mask		P4*			-, -er, -s
5	Substantiv			P5*			-
6	Substantiv	neut/mask	S1*		einfach*		-, -es
7	Substantiv	fem*	S3		einfach*		-
8	Substantiv	fem*	S3	P3	komplex*	-t*	-s, -en
	Substantiv	fem*	S3	–P*	komplex*	-t*	-s
9	Substantiv	neut/mask*			komplex*	-bold*, -chen*, -er*, -ich(t)*, -ig*, -lein*, -nis*, -rich*	-
10	Substantiv	neut/mask*			komplex*	-ling*, -sal*, -tum*	-s
11	Substantiv	fem*	S3	P3	komplex*	-heit*, -ion*, -ität*, -keit*, -schaft*, -ung*	-s, -en
	Substantiv	fem*	S3	–P*	komplex*	-heit*, -ion*, -ität*, -keit*, -schaft*, -ung*	-s
12	Substantiv	fem*	S3	P3	komplex*	-ei*	-, -en
	Substantiv	fem*	S3	–P*	komplex*	-ei*	-
13	Substantiv			–P*		ohne Suffix*	-
	Verb						-, -e
	Adjektiv						-

Abbildung 1.1: Bildung der erfassten Kompositionsstammformen

a) morphologisch, wie Wortart, grammatisches Geschlecht, Zugehörigkeit zu Flexionsklassen und Informationen, ob es sich um Singulariatantum oder Pluraliatantum handelt;

b) suffix- und auslautabhängig, da eine Vielzahl von Suffixen und Auslauten genau festlegt, ob und welches Fugenelement nach ihnen gesetzt werden kann;

c) komplexitätsabhängig, womit die Silbenzahl gemeint ist, da einsilbige Erstglieder oft ohne Fugenelement stehen, während mehrsilbige verstärkte Fugenverwendung zeigen;

d) lexikalisiert, da einige Erstglieder Ausnahmen zu den genannten Regularitäten darstellen, aber durch Lexikalisierung eindeutig sind und nur eine Kompositionsstammform aufweisen.

Die Kriterien a)–c) durchdringen sich oftmals gegenseitig und müssen kombiniert werden. Andererseits impliziert ein Kriterium nicht selten ein anderes, wie z. B. klar ist, dass alle Erstglieder mit *ung*-Suffix immer Feminina sind. In solchen Fällen kann die Information bezüglich des grammatischen Geschlechts als redundant angesehen werden und muss in einer entsprechenden Regel nicht berücksichtigt werden. Nicht ignoriert werden darf dagegen die Tatsache, dass die Feminina auf *-ung* neben dem unparadigmischen *-s* ihre Kompositionsstammform im Fall von pluralischer Bedeutung auch mit *-en* bilden können, sofern es sich nicht um Singulariatantum handelt.

In Abbildung 1.1 sind die Kriterien a)–c) zusammengestellt. Als Vorlage zur Einteilung in Flexionsklassen dienen die Deklinationsschemata für Substantive aus Dudenredaktion (1998, S. 226 u. 229). Mit * sind die Kriterien gekennzeichnet, die notwendig sind, um einem Erstglied ein oder mehrere Fugenelemente oder die Nullfuge zu zuordnen. Die Angabe der Wortart ist immer ein notwendiges Kriterium und wird hier nicht noch einmal gekennzeichnet. Oft ist eine Kombination von mehreren Kriterien nötig.

Die in d) gemeinten Ausnahmen sind z. B.: Das Substantiv *Liebe*, das ausschließlich in der unparadigmischen Kompositionsstammform *Liebes* auftaucht; Beispiele für lexikalisierte paradigmische Kompositionsstammformen sind einsilbige Maskulina und Neutra *Amt*, *Glück*, *Krieg*, *Reich* und andere, die alle regelmäßig *-s* in der Fuge aufweisen. Ebenfalls zu der Gruppe der lexikalisierten Kompositionsstammformen werden hier die nicht-schwachen Substantive gezählt, die ihre Kompositionsstammform regelmäßig auf *-en* bilden und im entsprechenden Abschnitt besprochen wurden.

1.6.3 Einige Worte zum Bindestrich

Der Gebrauch des Bindestriches wird bei Fleischer und Barz (1995, S. 142) kurz angerissen. Die Fälle, in denen er obligatorisch verwendet werden muss, werden in der vorliegenden Arbeit nicht erfasst, da Zusammensetzungen mit Wortgruppen und Kurzwörtern nicht einbezogen werden, auch Zusammensetzungen mit Eigennamen bleiben außen vor. Einzig die Rolle des Bindestriches bei der Rezeptionserleichterung muss berücksichtigt werden, weil sonst eine Fülle von Komposita, die in authentischen Texten vorkommen können, nicht als solche erkannt werden würde. Letztendlich ist der Bindestrich für die computerbasierte Analyse auch nur einen Zeichen unter vielen. Er muss also in seiner Funktion als Erläuterungsbindestrich berücksichtigt werden, bei der er nicht obligatorisch gesetzt werden muss, aber durchaus gesetzt werden kann.

Es handelt sich bei dem Bindestrich um eine Erscheinung an der Kompositionsgrenze, die tatsächlich zwischen den Gliedern steht und damit das Erstglied vom Zweitglied

trennt. Befindet sich ein Fugenelement an derselben Kompositionsgrenze, bleibt es beim Erstglied.

Eine Einbeziehung des Erläuterungsbindestriches in den Segmentierungsprozess hat direkte Folgen für das Segmentierungsergebnis. Bei einer naiven Analyse des Kompositums *drucker-wartung* entfallen sofort alle Segmentierungsarten, die *erwartung* als Zweitglied enthielten. Der Erläuterungsbindestrich erfüllt also auch hier seine Aufgabe.

Um bei der Segmentierung miteinbezogen werden zu können, wird der Bindestrich einfach als Lexikoneintrag verstanden und stellt damit eine Art abstraktes Segment für sich dar. Die auf der Ebene der Implementation angenommene Interpretation aller Einträge als Zeichenketten lässt dies zu. Im Segmentierungsergebnis ist der Bindestrich nicht mehr vorhanden, da die Segmentgrenzen dort anders dargestellt werden und der Bindestrich nicht mehr benötigt wird.

1.6.4 Die Fuge im Verhältnis zur Segmentierung

Fragen bezüglich der strukturellen Mehrdeutigkeit der Kompositasegmentierung wurden bereits in 1.5.3 besprochen. Die Bedeutung der Fuge in diesem Prozess wird jedoch nur am Rande behandelt. Obwohl die überwiegende Mehrheit der einfach zusammengesetzten Komposita keine Fugenelemente aufweisen, können Regeln, die die Distribution der Interfixe beschreiben, auch in diesen Fällen zur Disambiguierung beitragen. So kann verhindert werden, dass Fugenelemente erkannt werden, die gar nicht vorhanden sind. Mögliche Uneindeutigkeiten ergeben sich vor allem durch die Formgleichheit häufiger Suffixe und Präfixe mit den Fugenelementen. Alle drei Erscheinungen treffen bei den Komposita an der Kompositionsfuge aufeinander. So kommt z. B. *er* in allen drei Fällen vor, als Präfix, Suffix und als Fugenelement. Hinzu kommt, dass es als Präfix und Suffix oftmals zu deverbalen Substantiven gehört. Als Suffix kennzeichnet es nämlich die deverbale Ableitung, als Präfix ist es Überbleibsel des zugrunde liegenden präfigierten Verbs. Die Verben wiederum bilden ihre Kompositionsstammformen als reine Stämme, die oft bei den deverbalen Substantiven als Grundmorpheme ausgemacht werden können. Dies kann zu falschen Segmentierungsarten führen, die ohne die Distributionsregeln für Fugenelemente unerkannt bleiben würden. Daneben existieren auch viele Mehrdeutigkeiten, die schlichtweg zufällig sind und aus den vielfältigen Kombinationsmöglichkeiten der Kompositionsglieder oder deren Homonymie resultieren.

Die Regularitäten an der Fuge ergeben einen morphologischen Kontext der Kompositionsglieder unter einander, der zur Disambiguierung eingesetzt werden kann, bevor auf Kriterien aus höheren Sprachebenen Bezug genommen werden muss. Selbst der Bindestrich kann als Phänomen an der Fuge dazu beitragen. Im Grunde ist es dieser morphologische Kontext auf den alle Elemente des Finite-State-Modells hinarbeiten. Es werden möglichst viele Eigenschaften der Segmente aufgrund der beschriebenen

Kriterien festgestellt und anschließend in eine Beziehung zu einander gesetzt. Obwohl nur eine kleine Zahl der Komposita tatsächlich Fugenelemente aufweist, ist die Wirkung dieser Regeln bei der Segmentierung nicht nur auf diese gefugten Komposita beschränkt.

KAPITEL 2

Verwendete Finite-State-Mittel

2.1 Endliche Automaten

Endliche Automaten (EA) und Finite-State-Transducer (FST) sind die beiden Hauptkonzepte, die in dieser Arbeit verwendet werden. Beide Maschinenklassen verarbeiten Zeichenketten, die im Weiteren als Wörter bezeichnet werden. Wörter werden mithilfe eines Alphabets gebildet, das einfach eine Menge von Symbolen oder Zeichen ist. Ein Alphabet kann endlich (wie das deutsche Alphabet) oder auch unendlich (wie die Menge der reellen Zahlen) sein. Ein Wort ist eine endliche Sequenz von Symbolen. Die Menge aller Wörter über einem Alphabet Σ (Sigma) heißt Stern von Sigma und wird mit Σ^* (sprich: Sigma Stern) bezeichnet. Eine formale Sprache ist dann einfach eine Teilmenge von Σ^*.

2.1.1 Definitionen

Der endliche Automat ist ein mathematisches Modell eines Systems mit diskreten Ein- und Ausgaben. Das System befindet sich in einem aus einer endlichen Anzahl von internen Zuständen. Der Zustand eines Systems umfasst die Informationen, die sich aus den bisherigen Eingaben ergeben haben und die benötigt werden, um die Reaktion das Systems auf noch folgende Eingaben zu bestimmen.

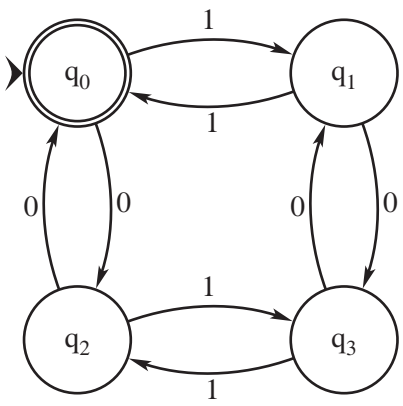
Einige grundlegende theoretische Eigenschaften endlicher Automaten (EA) machen diese zu flexiblen, effizienten und mächtigen Werkzeugen. Ein EA besteht aus einer endlichen Menge von Zuständen und einer Menge von Übergängen, die mit einem Eingabesymbol aus einem Alphabet Σ gekennzeichnet sind und einen Zustand in einen anderen überführen (vgl. Hopcroft und Ullman 2000, S. 16).

In dieser Arbeit werden endliche Automaten als Akzeptoren verstanden, die überprüfen, ob ein Wort im obigen Sinne zu einer durch den Automaten definierten Sprache

$Q = \{q_0, q_1, q_2, q_3\}$
$\Sigma = \{0, 1\}$
$F = \{q_0\}$

Funktion δ

Zustände	Eingabesymbole	
	0	1
q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2



Quelle: Hopcroft und Ullman (2000, S. 17)

Abbildung 2.1: Das Transitionsdiagramm für einen endlichen Automaten

gehört. Ist dies der Fall, akzeptiert der Automat das Wort, sonst wird das Wort nicht akzeptiert.

Formal ist ein EA ein 5-Tupel $M = (Q, \Sigma, \delta, q_0, F)$ bestehend aus

- einer endlichen Menge Q von Zuständen,
- einem endlichen Alphabet Σ,
- einer Übergangsfunktion δ,
- einem Anfangszustand (auch Startzustand genannt) q_0,
- einer Menge F von Endzuständen (auch Akzeptanzzustände genannt).

Wird ein EA als gerichteter Graph (Transitionsdiagramm) dargestellt, interpretiert man die Zustände aus der Menge Q als Knoten des Graphs. Die Struktur des Graphs, also welche Knoten auf welche Weise mit anderen Knoten verbunden sind, wird durch die Übergangsfunktion δ festgelegt, die oft mithilfe einer Zustandstafel dargestellt wird. Gibt es bei Eingabe des Symbols a einen Übergang von Zustand q in den Zustand p ($\delta(q, a) = p$), dann existiert im Transitionsdiagramm ein mit a markierter Pfeil vom Zustand q zum Zustand p. Wenn zwei Zustände durch gleichgerichtete Übergänge verbunden sind, kann man diese Übergänge auch zu einem einzigen Übergang zusammenfassen, der dann mit allen verwendeten Symbolen beschriftet ist. Der Anfangszustand wird üblicherweise mit einem einzelnen Pfeil kenntlich gemacht. Endzustände werden mit einem doppelten Rand dargestellt. Ein EA akzeptiert ein Wort, wenn die Übergangsfolge, die der Symbolfolge des Wortes entspricht, den Anfangszustand in einen der Endzustände überführt.

Abbildung 2.1 zeigt einen endlichen Automaten, der jede Zeichenkette akzeptiert, die aus einer gerade Anzahl von Einsen und einer geraden Anzahl von Nullen besteht.

Andere Zeichenketten werden nicht akzeptiert. Dieser EA besteht aus vier Zuständen, das Alphabet enthält zwei Symbole, 0 und 1, q_0 ist hier gleichzeitig Start- sowie Endzustand. Die Übergangsfunktion δ ist hier als Zustandstafel gegeben, so dass in jedem Schnittpunkt von Zustand (Zeile) und Eingabesymbol (Spalte) der neue Zustand zu finden ist, z. B. für $\delta(q_0, 0) = q_2$.

Sei „110101" eine Beispielzeichenkette mit einer jeweils geraden Anzahl von Nullen und Einsen. Der EA beginnt die Berechnung im Zustand q_0 und liest das erste Symbol des Wortes ein, eine „1". In der Tabelle der Funktion δ reicht es, den Zustand q_0 in der linken Spalte zu finden; in derselben Zeile ist der Ergebniszustand unter dem Eingabesymbol „1" der Zustand q_1. Auf dem Transitionsdiagramm sieht man, dass ein mit einer „1" bezeichneter Pfeil vom Zustand q_0 auf den Zustand q_1 gerichtet ist. Der Automat befindet sich nun im Zustand q_1 und liest das nächste Zeichen ein, wieder eine „1". Vom Zustand q_1 aus verursacht die Eingabe einer „1" einen Übergang zurück zu q_0, nach denselben Regeln wie zuvor. Die Eingabe einer „0" führt zu q_2, das nächste Symbol „1" zu q_3, die nächste „0" zu q_1 und schließlich das letzte Symbol, eine „1", zu q_0. Nach der Eingabe des letzten Symbols befindet sich der Automat in einem Endzustand. Die Zeichenkette „110101" wird akzeptiert. Würde sich der Automat nach dem Einlesen des letzten Symbols nicht in einem Endzustand befinden, würde die Eingabe verworfen. In diesem Beispiel wird deutlich, dass ein Endzustand erreicht werden kann, noch bevor eine Eingangszeichenkette komplett eingelesen wurde. Der Automat beendet in einem solchen Fall jedoch nicht seine Arbeit. Dies bedeutet lediglich, dass er die bisher eingelesene Zeichenkette akzeptiert. Man spricht in diesem Fall von Präfixen. In diesem Beispiel ist die Folge „11" solch ein Präfix.

Die Übergangfunktion δ gibt an, welcher Folgezustand beim Lesen eines einzelnen Zeichens erreicht wird. Sie lässt sich auf eine Übergangfunktion $\delta' : Q \times \Sigma^* \to Q$ erweitern, die festlegt, welcher Zustand beim Lesen eines ganzen Wortes erreicht wird.

Sei $q \in Q$ und $a \in \Sigma, w \in \Sigma^*$, dann ist

- $\delta'(q, \varepsilon) = q$, wobei ε hier für das leere Wort steht.
- $\delta'(q, aw) = \begin{cases} \delta'(\delta(q, a), w) & \text{falls } \delta(q, a) \neq \perp \\ \perp & \text{sonst} \end{cases}$

Der Einfachheit halber wird die Funktion δ' im Weiteren mit der Funktion δ gleichgesetzt, was auch in der entsprechenden Fachliteratur (vgl. Asteroth und Baier 2002, S. 224) üblich ist.

Nachdem die Übergangsfunktion δ von einzelnen Symbolen zu ganzen Wörtern erweitert wurde, kann mit ihrer Hilfe die Sprache definiert werden, die durch einen EA akzeptiert wird.

Eine durch einen Automaten M akzeptierte Sprache $L(M)$ ist die Menge aller Wörter, die den Anfangszustand q_0 in einen beliebigen Endzustand überführen.

$$L(M) = \left\{ w \in \Sigma^* : \delta(q_0, w) \in F \right\}$$

Im Fall des Beispielautomaten aus Abbildung 2.1 ist $L(M)$ die unendliche Menge der Zeichenketten mit einer geraden Anzahl von Nullen und einer geraden Anzahl von Einsen.

Ein durch eine solche Übergangsfunktion definierter Automat wird als deterministischer endlicher Automat (DEA) bezeichnet in Opposition zu den nicht-deterministischen Automaten (NEA), die in Kapitel 1.2.2. gesondert besprochen werden. Bei einem DEA handelt es sich um eine Unterklasse der nicht-deterministischen Automaten, die durch eine einfachere Implementierung und eindeutige Ergebnisse gekennzeichnet ist. Determinismus heißt in diesem Fall, dass für jedes Wort, für jede Sequenz von Symbolen, genau ein bestimmter Pfad, eine bestimmte Folge von Übergängen, im Transitionsdiagramm existiert. Für den Entwurf von effizienten Algorithmen ist dies eine besonders interessante Eigenschaft, da sie eine optimale (lineare) Laufzeit garantiert. Bei einem Analysevorgang einer Symbolfolge, die aus n Elementen besteht, werden so genau n Vergleiche angestellt.

Formal besteht der Unterschied zwischen Determinismus und Nicht-Determinismus bei EA in der Definition der Übergangsfunktion δ, die bei einem DEA, wie zuvor bereits beschrieben, nur einen einzigen Zielzustand zurückgibt. Bei einem NEA wird generell eine Menge von Zuständen zurückgegeben, die einen, mehrere oder auch gar keine Zielzustände enthalten kann. Das Transitionsdiagramm eines DEA weißt eine Besonderheit auf, die durch die Eindeutigkeit der Übergangsfunktion gegeben ist. Jeder Knoten des gerichteten Graphs eines DEA ist durch höchsten genau einen Übergang mit einem bestimmten Eingangssymbol mit einem anderen Zustand verbunden. Es ist nicht möglich, dass in einem DEA zwei oder mehr Übergänge mit der gleichen Etikette von einem Knoten ausgehen (von verschiedenen Knoten aus, aber sehr wohl). So ist gewährleistet, dass der Pfad im Graph bei einer bestimmten Zeichenfolge ebenfalls vorbestimmt ist. Bei einem NEA muss indes ein Suchprozess stattfinden, da es oft mehrere mögliche Pfade gibt.

2.1.2 Nicht-Deterministische Automaten und Determinierung

Wie bereits beschrieben, unterscheidet sich ein nicht-deterministischer endlicher Automat (NEA) von der deterministischen Variante formal gesehen einzig durch die Definition der Übergangsfunktion δ. Während in einem Transitionsdiagramm eines DEA, von einem Zustand aus nur ein Pfeil mit einem bestimmten Symbol ausgehen kann, ist bei einem NEA durchaus üblich, dass von einem Zustand aus mehrere Übergänge für dasselbe Symbol zu verschiedenen anderen Zuständen existieren. In

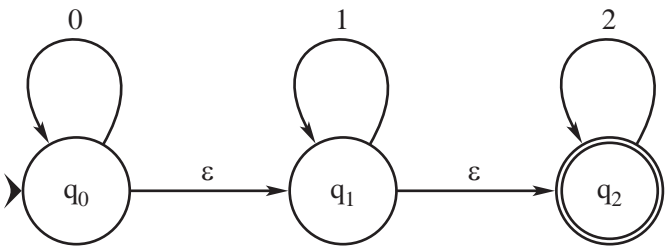
diesem Fall ist nicht mehr eindeutig, welcher Pfad bei Eingabe eines Wortes eingeschlagen werden muss. Was auf den ersten Blick nach einer unnötigen Komplikation klingt, besonders da für jeden NEA ein äquivalenter DEA angegeben werden kann (vgl. Hopcroft und Ullman 2000, S. 22), hat jedoch auch Vorteile. So ist es für einen menschlichen Designer oftmals sehr viel einfacher einen NEA zu entwerfen, als dies bei einem DEA der Fall wäre. Wurde ein NEA erst einmal konzipiert, kann er später dank automatischer Prozesse zu einem DEA umberechnet werden. Der neue DEA kann allerdings im schlimmsten Fall aus bis zu 2^n Zuständen bestehen, wenn der NEA aus n Zuständen aufgebaut ist. Im Falle eines NEA mit insgesamt 10 Zuständen wären dies bis zu 1024 Zustände eines äquivalenten DEA. Dieser Vorgang wird als Determinierung (meist mithilfe der Potenzmengenkonstruktion) bezeichnet.

Wie schon angedeutet, ist auch eine NEA formal durch ein 5-Tupel $M = (Q, \Sigma, \delta, q_0, F)$ beschrieben, in dem alle Elemente den Elementen eines DEA entsprechen. Lediglich die Übergangsfunktion δ ist deutlich weiter gefasst.

Bei einem nicht-deterministischen endlichen Automaten ist die Übergangfunktion die Relation $\delta: Q \times \Sigma \rightarrow 2^Q$. Obwohl es sich hier nicht mehr um eine Funktion im mathematischen Sinne handelt, da einem Element unter Umständen mehrere andere Elemente zugeordnet werden, wird die Bezeichnung Übergangsfunktion im Weiteren beibehalten, um Verwirrungen zu vermeiden. Jedem Paar (bestehend aus einem Zustand und einem Eingabesymbol) wird hier eine Menge von Zuständen zugeordnet. Im vorigen Unterkapitel wurde erwähnt, dass jeder DEA gleichzeitig ein NEA ist; umgekehrt ist dies nur erfüllt, wenn die Menge von zurückgegeben Zuständen immer aus höchstens einem Element besteht.

Eine weitere Klasse nicht-deterministischer endlicher Automaten sind Automaten mit so genannten ε-Transitionen. Hierbei handelt es sich um Übergänge, die stattfinden, selbst wenn kein Eingabesymbol eingelesen wurde. Man spricht von spontanen Zustandsänderungen, die man umgangssprachlich auch als Sprünge bezeichnen kann. Für solche Automaten wird die Übergangsfunktion nochmals modifiziert zu $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$. Hier wird jedem Paar aus Zustand und Eingabesymbol eine Menge von Zuständen zugeordnet, die alle Zustände enthält, die durch das angegebene Eingabesymbol sowie durch das Befolgen aller ε-Transitionen (auch über mehrere ε-Transitionen hinweg) erreicht werden können.

Abbildung 2.2 zeigt einen NEA mit ε-Transitionen, der als Sprache alle Wörter akzeptiert, die aus einer beliebigen Anzahl von Nullen, auf die eine beliebige Anzahl von Einsen folgt, auf die wiederum eine beliebige Anzahl von Zweien folgt, bestehen. Andere Zeichenketten werden verworfen. Obwohl in der Zustandstafel für ein Paar aus dem Zustand q_0 und dem Eingabesymbol „0" nur die Menge $\{q_0\}$ angeben ist, ist das Ergebnis einer erweiterten Übergangsfunktion $\{q_0, q_1, q_2\}$. Zwar wird q_0 als Ergebnis wiedergegeben, aber von q_0 aus sind über ε-Transitionen auch die Zustände q_1 und q_2 erreichbar, da für ε-Transitionen keine Eingabesymbole nötig sind.



Zustände	Eingabesymbole			
	0	1	2	ε
q_0	$\{q_0\}$	$\emptyset$	$\emptyset$	$\{q_1\}$
q_1	$\emptyset$	$\{q_1\}$	$\emptyset$	$\{q_2\}$
q_2	$\emptyset$	$\emptyset$	$\{q_2\}$	$\emptyset$

Quelle: Hopcroft und Ullman (2000, S. 25)

Abbildung 2.2: Endlicher Automat M_1 (NEA) mit ε-Transitionen

Abbildung 2.3 zeigt einen NEA ohne ε-Transitionen, der dem NEA aus Abbildung 2.2 entspricht und dieselbe Sprache akzeptiert ($L(M_2) = L(M_1)$). Für jeden ε-erweiterten NEA existiert ein äquivalenter NEA ohne ε-Übergänge (vgl. Asteroth und Baier 2002, S. 243).

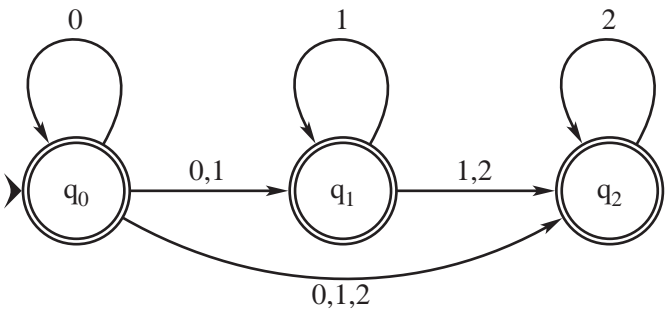
Auf genaue Beschreibungen, wie die Übergangsfunktionen für einen NEA und einen NEA mit ε-Transitionen erweitert werden müssen, um ganze Wörter zu akzeptieren, wird an dieser Stelle verzichtet (vgl. Asteroth und Baier 2002, S. 228 und S. 242). Die Sprache, die durch einen NEA oder einen ε-erweiterten NEA akzeptiert wird, ist etwas anders definiert als bei der deterministischen Variante. Eine durch einen NEA M akzeptierte Sprache L ist die Menge aller Wörter, die den Anfangszustand q_0 in eine Menge von Zuständen überführt, in der mindestens ein Endzustand enthalten ist, oder anders formuliert, der Durchschnitt der Menge der Ergebniszustände mit der Menge der Endzustände des Automaten ist nicht leer:

Eine durch einen NEA M akzeptierte Sprache $L(M)$ ist die Menge aller Wörter, die den Anfangszustand q_0 in eine Menge von Endzuständen überführt.

$$L(M) = \left\{ w \in \Sigma^* \colon \delta(q_0, w) \cap F \neq \emptyset \right\}$$

Bei der hier verwendeten Übergangsfunktion δ handelt es sich um die erweiterte Variante, die ganze Wörter als Argument annimmt.

Der Automat M_3 von Abbildung 2.3 kann, nachdem die ε-Transitionen entfernt wurden, determinisiert werden. Das heißt, ein NEA ohne ε-Übergänge kann in einen äquivalenten DEA umberechnet werden, in einen DEA, der dieselbe Sprache akzeptiert wie der Ausgangsautomat und sonst keine anderen Wörter, $L(M_3) = L(M_2)$.



Zustände	Eingabesymbole		
	0	1	2
q_0	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
q_1	$\emptyset$	$\{q_1, q_2\}$	$\{q_2\}$
q_2	$\emptyset$	$\emptyset$	$\{q_2\}$

Quelle: Hopcroft und Ullman (2000, S. 28)

Abbildung 2.3: Endlicher Automat M_2 (NEA) ohne ε-Transitionen

Abbildung 2.4 zeigt einen solchen DEA, der durch die Potenzmengenkonstruktion (vgl. Asteroth und Baier 2002, S. 231) aus M_2 entstanden ist.

Wie man leicht erkennt, hat sich die Zahl der Zustände vergrößert, wenn auch nicht bei weitem so dramatisch wie für den schlimmsten Fall angenommen. Aber auch bei einer exponentiellen Vergrößerung der Zustandszahl lassen sich noch Modifikationen vornehmen, die die Anzahl der Zustände wieder verkleinern. Man spricht in diesem Fall von einer Minimierung. Oft genug werden bei der Potenzmengenkonstruktion unnötig viele Zustände und Übergänge konstruiert, die mithilfe der Minimierung wieder entfernt werden. Das Ergebnis einer Minimierung ist ein so genannter Mi-

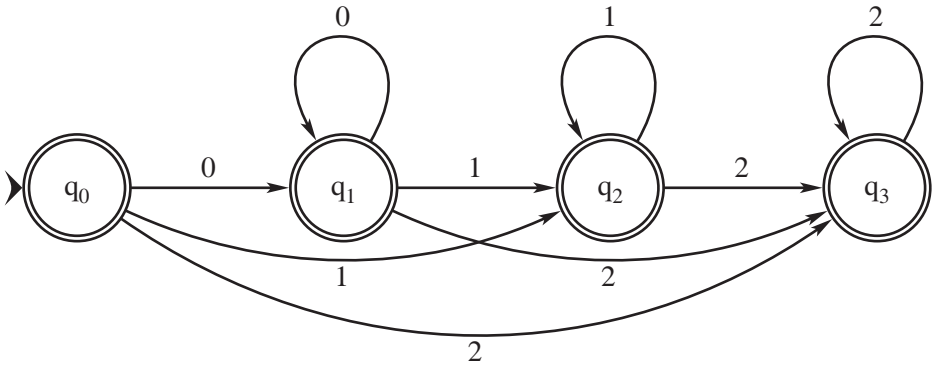


Abbildung 2.4: Determinierungsergebnis M_3 (DEA)

nimalautomat. Das heißt, der DEA M_3 wurde durch einen äquivalenten DEA M_4 ersetzt, dessen Zustandsraum minimal unter allen äquivalenten DEA ist (Abbildung 2.5). Auch hier gilt $L(M_4) = L(M_3)$. Es kann demnach kein deterministischer Automat angegeben werden, der dieselbe Sprache beschreibt und über eine noch kleinere Struktur verfügt. Das Ergebnis der Minimierung ist optimal.

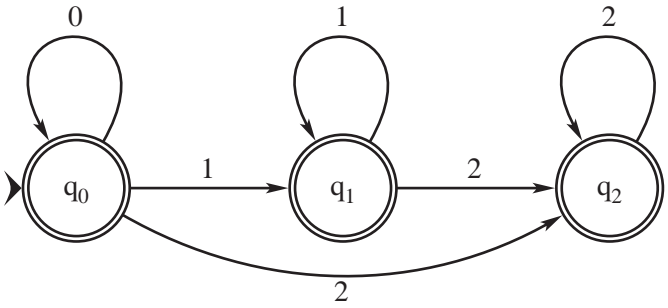


Abbildung 2.5: Minimierungsergebnis M_4 (DEA)

Die Konstruktion eines minimalen DEA aus einem ε-erweiterten NEA ist ein automatischer Prozess und kann Computern überlassen werden. Die Methoden dazu sind mathematisch erforscht. In unserem Beispiel wurde letztendlich aus einem ε-erweiterten NEA M_1 durch die Entfernung der ε-Übergänge, anschließende Determinierung und Minimierung ein minimaler DEA M_4 berechnet. Für die formalen Sprachen, die von beiden Automaten definiert werden, gilt $L(M_4) = L(M_1)$. Je nach Anwendungsgebiet ist es sinnvoll oder weniger sinnvoll diese Vorgänge auszuführen. Auch ε-erweiterte NEA können am Rechner implementiert werden, sie verbrauchen im Allgemeinen weniger Speicherplatz, als ihre deterministischen Äquivalente, funktionieren aber langsamer, da statt deterministischer Wege, eine Suche stattfinden muss. Bei sehr großen Aufblähungserscheinungen durch die Determinierung kann eine Suche allerdings vorgezogen werden.

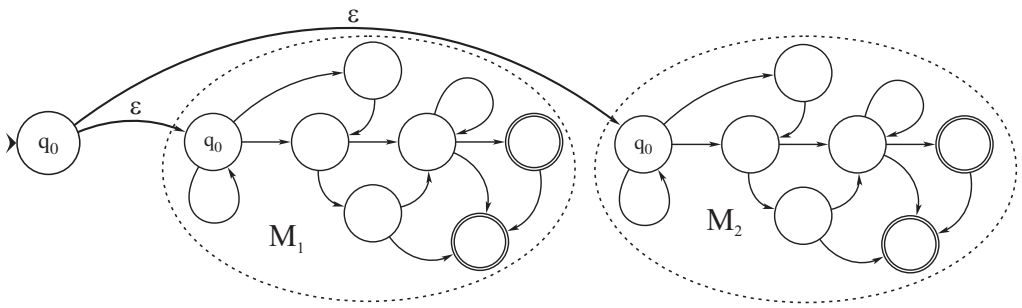
2.1.3 Abschlusseigenschaften endlicher Automaten

Neben den Möglichkeiten der Determinierung und Minimierung bieten endliche Automaten bestimmte Operationen, die sie zu vorzüglichen Werkzeugen in der Computerlinguistik machen. Es ist im Allgemeinen sehr schwierig, ein Modell zu entwickeln, dass wenn schon nicht alle, dann wenigstens die meisten Eventualitäten eines bestimmten linguistischen Phänomens beschreibt. Am besten nähert man sich schrittweise von einem allgemeinen, ungenauen Modell einem Modell an, das immer mehr Ausnahmen und Einzelfälle erfasst. Während die schrittweise Erweiterung von z. B. Phrasenstrukturgrammatiken mit erheblichen Schwierigkeiten verbunden sein kann, die zu nicht selten zu Veränderungen im gesamten Regelwerk führt, existieren für endliche Automaten (und Transducer, siehe nächstes Kapitel) bestimmte Methoden, die den Aufbau komplexer Systeme aus Teilelementen extrem vereinfachen.

Diese Eigenschaften endlicher Automaten heißen Abschlusseigenschaften, d. h. endliche Automaten sind unter bestimmten mathematischen Operationen abgeschlossen. Wird eine dieser Operationen über einem oder mehreren Automaten ausgeführt, ist das Ergebnis wieder ein endlicher Automat. Die Klasse formaler Sprachen, die von endlichen Automaten definiert wird, entspricht genau der Klasse der regulären Sprachen (vgl. Hopcroft und Ullman 2000, S. 30). Deren gut erforschte Abschlusseigenschaften werden direkt auf endliche Automaten übertragen

Vereinigung

Die Menge der regulären Sprachen ist abgeschlossen unter Vereinigung: Wenn M_1 und M_2 zwei endliche Automaten sind, so ist es möglich einen Automaten $M_1 \cup M_2$ zu berechnen, für den gilt $L(M_1 \cup M_2) = L(M_1) \cup L(M_2)$ (vgl. Roche und Schabes 1997, S. 6). Der Ergebnisautomat wird alle Wörter akzeptieren, die von einem der beiden Ursprungsautomaten akzeptiert werden. Auf diese Weise kann man zwei oder mehr Modelle einfach mit einander verbinden und zu einem Modell kombinieren. Tatsächlich ist diese Operation sehr einfach mit ε-Übergängen zu erreichen. Man fügt einen neuen Anfangszustand hinzu und verbindet diesen Zustand durch ε-Übergänge mit allen bisherigen Startzuständen beider Automaten, wie auf Abbildung 2.6 zu sehen ist. Auf diese Weise entsteht natürlich ein ε-erweiterter NEA – selbst wenn beide Teilautomaten DEA sind. Durch die im vorigen Kapitel beschriebenen Methoden, lässt sich der vereinigte Automat wieder in einen (minimalen) DEA umwandeln, sofern dies für notwendig erachtet wird.



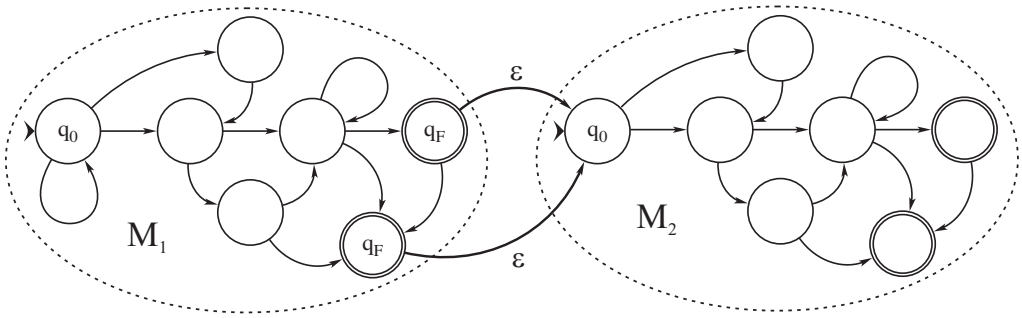
Quelle: Jurafsky und Martin (2000, S. 51)

Abbildung 2.6: Die Vereinigung zweier endlicher Automaten

Konkatenation

Die Menge der regulären Sprachen ist abgeschlossen unter Konkatenation: Wenn M_1 und M_2 zwei endliche Automaten sind, so ist es möglich einen Automaten $M1 \cdot M2$

zu berechnen, für den gilt $L(M_1 \cdot M_2) = L(M_1) \cdot L(M_2)$ (vgl. Roche und Schabes 1997, S. 6). Der Ergebnisautomat wird alle Wörter akzeptieren, die aus Zusammensetzungen der Wörter der Ursprungsautomaten bestehen – auf ein Wort des ersten Automaten muss ein Wort des zweiten Automaten folgen. Hierbei ist die Reihenfolge wichtig, da es sich bei der Konkatenation um keine kommutative Operation handelt. Wenn beide Ursprungsautomaten verschiedene Sprachen definieren, gilt $L(M_1 \cdot M_2) \neq L(M_2 \cdot M_1)$.



Quelle: Jurafsky und Martin (2000, S. 50)

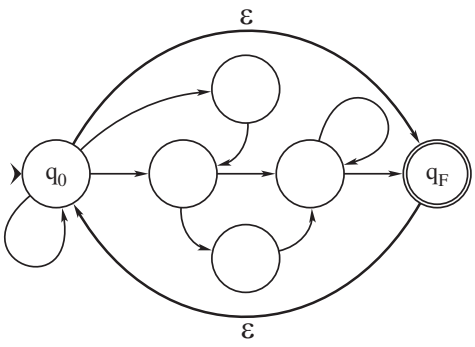
Abbildung 2.7: Die Konkatenation zweier endlicher Automaten

Ähnlich wie bei der Vereinigung kann die Konkatenation relativ einfach durch die Verwendung von ε-Übergängen hergestellt werden. Es genügt, alle Endzustände des ersten Automaten durch ε-Übergänge mit dem Startzustand des zweiten Automaten zu verbinden. Auch hier entsteht wieder ein ε-erweiterter NEA.

Kleene-Abschluss

Die Menge der regulären Sprachen ist abgeschlossen unter Kleene-Abschluss: Wenn M ein endlicher Automat ist, so ist es möglich einen Automaten M^* zu berechnen, für den $L(M^*) = L(M)^*$ gilt (vgl. Roche und Schabes 1997, S. 6). Der Ergebnisautomat akzeptiert alle Wörter, die aus einer beliebigen Anzahl von Wörtern des Ursprungsautomaten bestehen, darunter auch das leere Wort.

Ähnlich wie zuvor ist auch diese Konstruktion recht einfach herzustellen. Es werden alle Endzustände des Ursprungsautomaten über ε-Übergänge mit seinem Startzustand verbunden (dies implementiert die Wiederholung). Außerdem wird auch der Startzustand durch ε-Übergänge mit den Endzuständen verbunden (dies implementiert die Möglichkeit das leere Wort zu akzeptieren). Der letzte Schritt wird weggelassen, wenn statt dem Kleene-Stern die Operation Kleene-Plus realisiert werden soll ($L(M^+) = L(M)^+$). In diesem Fall werden keine leeren Worte akzeptiert.



Quelle: Jurafsky und Martin (2000, S. 51)

Abbildung 2.8: Der Kleene-Abschluss eines endlichen Automaten

Durchschnitt

Die Menge der regulären Sprachen ist abgeschlossen unter Durchschnitt: Wenn $M_1 = (Q_1, \Sigma, \delta_1, q_{0,1}, F_1)$ und $M_2 = (Q_2, \Sigma, \delta_2, q_{0,2}, F_2)$ zwei endliche Automaten sind, so ist es möglich einen Automaten $M_1 \cap M_2$ zu berechnen (auch Produktautomat genannt), für den $L(M_1 \cap M_2) = L(M_1) \cap L(M_2)$ gilt (vgl. Roche und Schabes 1997, S. 6). Die Konstruktion eines solchen Automaten ist deutlich komplizierter als in den vorherigen Fällen. Der Produktautomat unterliegt der Vorstellung, dass M_1 und M_2 parallel geschaltet werden. Ist w das Eingabewort, dann startet die synchrone Bearbeitung des Wortes w durch M_1 und M_2. Sobald einer der Automaten frühzeitig verwirft, dann auch der Produktautomat. Nur wenn beide Automaten akzeptieren, akzeptiert auch der Produktautomat. Das ist dadurch zu erklären, dass ein Wort, wenn es zu der Schnittmenge der beiden durch die Automaten definierten Sprachen gehört, es auch zu den beiden Sprachen der einzelnen Automaten gehören muss. Der Produktautomat kann folgendermaßen konstruiert werden (vgl. Asteroth und Baier 2002, S. 237): $M_1 \cap M_2 = (Q_1 \times Q_2, \Sigma, \delta, (q_{0,1}, q_{0,2}), F_1 \times F_2))$ wobei die neue Übergangsfunktion δ wie folgt definiert wird: $\delta((q_1, q_2), a) = \Big\{(p_1, p_2): \quad p_1 \in \delta_1(q_1, a) \wedge p_2 \in \delta_2(q_2, a)\Big\}$. Die Zustände des Ergebnisautomaten bestehen dann aus Paaren von Zuständen der Ursprungsautomaten.

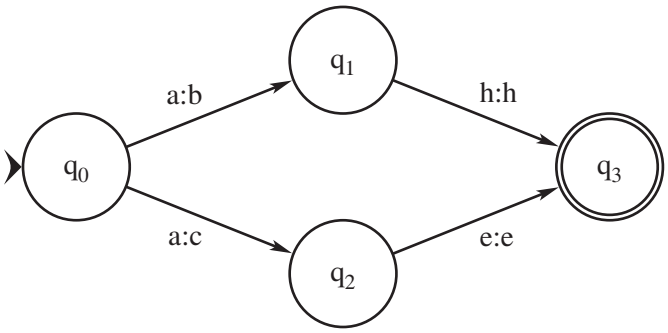
Außer den genannten Operationen sind endliche Automaten auch unter Komplementbildung, Umkehrung und Differenz abgeschlossen. Diese Abschlusseigenschaften werden in der vorliegenden Arbeit jedoch nicht verwendet und daher nicht weiter besprochen. Transducer, die im nächsten Kapitel besprochen werden, weisen weitere Abschlusseigenschaften auf, die mit dem Relationscharakter von Transducern zusammenhängen.

2.2 Finite-State-Transducer

Bei Finite-State-Transducern (FST) handelt es sich um eine Erweiterung des Konzeptes der endlichen Automaten. Man bezeichnet Transducer auch als Automaten mit Ausgabe oder als Mealy-Maschinen. In dieser Arbeit wird aus praktischen Gründen die englische Bezeichnung Transducer verwendet, wie dies auch überwiegend in der entsprechenden deutschsprachigen Fachliteratur der Fall ist.

2.2.1 Definitionen

Wie bereits besprochen, liest ein endlicher Automat ein Symbol von einem Eingabeband und geht abhängig von seinem Ausgangszustand und dem eingelesenem Symbol in einen anderen (im Fall von Schleifen auch in denselben) Zustand über. Wenn mit einem solchen Zustandübergang eine Ausgabeoperation verbunden ist, spricht man von einem Transducer. Abbildung 2.9 zeigt einen einfachen Transducer. Wie man sieht, sind die Übergänge mit Paaren von Symbolen gekennzeichnet, von denen das jeweils linke ein Eingabesymbol ist und das jeweils rechte als Ausgabesymbol bezeichnet wird.



Quelle: Roche und Schabes (1995, S. 219)

Abbildung 2.9: Ein einfacher FST

Formal lässt sich ein Transducer je nach Interpretation auf verschiedene Weisen darstellen (vgl. Roche und Schabes 1997, S. 6). In der vorliegenden Arbeit werden drei Darstellungsweisen verwendet, die später je nach Notwendigkeit angenommen werden. Während gewöhnliche endliche Automaten nur über einem Alphabet arbeiten, werden für Transducer zwei Alphabete angenommen, ein Eingabealphabet, in dem alle einlesbaren Symbole enthalten sind, und einem Ausgabealphabet, aus dem alle auszugebenden Symbole stammen. Je nach Interpretation wird mit diesen Alphabeten verschieden verfahren.

Interpretation 1: Ein Transducer ist ein 5-Tupel $T = (Q, \Sigma, \delta, q_0, F)$ bestehend aus den folgenden Elementen

- einer endlichen Menge Q von Zuständen,
- einem endlichen Alphabet $\Sigma \subseteq \Sigma_1 \times \Sigma_2$ bestehend aus komplexen Symbolen,
- einer Übergangsfunktion $\delta : Q \times \Sigma \to 2^Q$,
- einem Anfangszustand q_0,
- einer Menge von $F \subseteq Q$ Endzuständen.

Die komplexen Symbole aus Σ sind Paare von Eingabe- und Ausgabesymbolen aus den jeweiligen Alphabeten. Σ ist eine Teilmenge der Menge $\Sigma_1 \times \Sigma_2 = \{(i, o) : i \in \Sigma_1 \wedge o \in \Sigma_2\}$ aller möglichen Paare. Einem Eingabesymbol wird somit ein Ausgabesymbol zugeordnet, das beim Einlesen des Eingabesymbols ausgegeben wird. Ein Symbolpaar (i, o) wird auch als $i : o$ notiert. Besonders in Tranistionsdiagrammen ist dies üblich, wie auch in Abbildung 2.9. Besteht ein Paar aus zwei identischen Symbolen, wird der Einfachheit halber oft auch nur ein einzelnes Symbol verwendet, im Falle eines Transducers entspricht i dann *i:i*. Die Übergangsfunktion δ arbeitet ähnlich wie im Falle der bereits beschriebenen NEA, als Argument wird hier außer dem Ausgangszustand ein komplexes Symbol angenommen. Befindet sich der Automat also in einem Zustand q, erreicht er über das komplexe Symbol $a = i : o$ eine Menge von Zuständen ($\delta(q, a) = \delta(q, i : o) = \{p_1, p_2, \ldots\}$).

Diese Interpretation eines FST wird verwendet, um verschiedene Operationen ausführen zu können. So werden zum Beispiel die Vereinigung, der Schnitt und die Komposition von Transducern, sofern diese Operationen erlaubt sind, auf dem zugrunde liegenden endlichen Automaten eines FST ausgeführt. Auch eine gewisse Determinierung und Minimierung sind so möglich. Nach dieser Auffassung ist der FST von Abbildung 2.9 übrigens deterministisch, da von keinem Zustand zwei oder mehr Übergänge mit dem gleichen Symbolpaar ausgehen

Interpretation 2: Ein Transducer ist ein 7-Tupel $T = (Q, \Sigma_1, \Sigma_2, \delta, \sigma, q_0, F)$ bestehend aus den folgenden Elementen

- einer endlichen Menge Q von Zuständen,
- einem endlichen Eingabealphabet Σ_1,
- einem endlichen Ausgabealphabet Σ_2,
- einer Übergangsfunktion $\delta : Q \times \Sigma_1 \to 2^Q$,
- einer Ausgabefunktion $\sigma : Q \times \Sigma_1 \times Q \to 2^{\Sigma_2^*}$,
- einem Anfangszustand q_0,

- einer Menge von $F \subseteq Q$ Endzuständen.

Besonderes Augenmerk gilt hier besonders der Ausgabefunktion σ (auch Emissionsfunktion genannt). Diese Funktion ist dafür zuständig, jedem Tripel bestehend aus Ausgangszustand, Eingabesymbol und Zielzustand eine Menge von Ausgabesymbolen zu zuordnen. Es muss hier noch der Zielzustand angegeben werden, da sonst nicht immer klar ist, welches Ausgabesymbol gesucht ist. Hier wird eindeutig zwischen Eingabe- und Ausgabealphabet unterschieden. Nach dieser Auffassung ist der Transducer aus Abbildung 2.9 nicht deterministisch, da es zum Beispiel vom Startzustand aus zwei Übergänge gibt, die beide mit dem gleichen Eingabesymbol bezeichnet sind. Bei Transducern ist Nicht-Determinismus kein triviales Problem. Um die korrekte Menge an Ausgabewörtern zu erhalten, müssen erst Suchvorgänge im FST ausgeführt werden. Man erhält auf diese Weise alle Pfade, die für ein bestimmtes Eingabewort zu einem der Endzustände führen. Jedem dieser Pfade kann ein anderes Ausgabewort zugeordnet sein.

An dieser Stelle sei die Menge $P(x)$ definiert als die Menge aller Pfade, die bei Eingabe des Wortes x vom Startzustand q_0 zu einem der Endzustände führen. Ein solcher Pfad wird auch als erfolgreicher Pfad bezeichnet. Die Ausgabefunktion σ kann anschließend über die Elemente dieser Menge erweitert werden und so das Wort ausgeben, das einem Pfad zugeordnet wird.

Diese Interpretation beschreibt die Arbeitsweise eines FST am besten, da sowohl die Übergangsfunktion als auch die Ausgabefunktion nur vom internen Zustand des FST und den eingelesen Eingabesymbolen abhängig sind. In der Realität sind dies genau die Daten, die während des Arbeitsprozesses eines FST zur Verfügung stehen.

Interpretation 3: Ein Transducer T wird definiert durch eine Transduktion $|T|$ der Menge der Wörter über dem Eingabealphabet Σ_1 auf die Potenzmenge der Menge der Wörter über dem Ausgabealphabet Σ_2 ($|T| : \Sigma_1^* \rightarrow 2^{\Sigma_2^*}$)

Eine Transduktion ist eine Relation, d. h. es werden Elemente einer Menge mit Elementen einer anderen Menge in Relation gesetzt. Im Gegensatz zu Funktionen, die einem Element immer genau ein Element zuordnen, können durch Relationen (Transduktionen) Elementen auch Mengen von Elementen zugeordnet werden. In Anlehnung an die in der vorigen Interpretation eingeführte Ausgabefunktion und ihre Erweiterung auf erfolgreiche Pfade über einem Eingabewort ist die Menge der Ausgabeworte, die durch einen Transducer T einem Eingabeworte $w \in \Sigma_1^*$ zugeordnet wird, wie folgt definiert, wobei $P(w)$ die Menge aller erfolgreichen Pfade für das Eingabewort w ist:

$$|T|(w) = \bigcup_{\pi \in P(w)} \sigma(\pi)$$

Da das Ergebnis einer Transduktion $|T|$ eine Menge von Wörtern ist, ist es sinnvoll die Transduktion auf Mengen zu erweitern, was im nächsten Kapitel für die Abschlusseigenschaft der Komposition benötigt wird. Ist $W \subseteq \Sigma_1^*$ eine Menge von Eingabewörtern, dann reicht es jede Menge von Ausgabewörtern, die jedem Eingabewort zugeordnet ist, zu vereinigen:

$$|T|(W) = \bigcup_{w \in W} |T|(w)$$

Diese Interpretation ist deutlich abstrakter als die beiden ersten und dient meist der Notation verschiedener mathematischer Eigenschaften von FST. Auf diese Weise lassen sich z. B. Kompositionen, Umkehrungen und andere Konstruktionen einfacher darstellen, was später auch beim Aufbau des in dieser Arbeit beschriebenen Modells verwendet wird. Der FST aus Abbildung 2.9 wird kann nach dieser Interpretation aufgrund der Mengen seiner Eingabe- und Ausgabewörter definiert werden.

$$W = \{ah, ae\}$$

$$|T|(W) = |T|(\{ah, ae\}) = |T|(ah) \cup |T|(ae) = \{bh\} \cup \{ce\} = \{bh, ce\}$$

Die interne Struktur (z. B. das Transitionsdiagramm) eines FST, der auf eine solche Weise definiert wird, ist im Grunde unerheblich. Jeder FST, der die Menge $\{ah, ae\}$ gemäß des obigen Beispiels auf die Menge $\{bh, ce\}$ abbildet, ist äquivalent zu T. Die Transduktion $|T|$ definiert in diesem Fall den Transducer T.

2.2.2 Abschlusseigenschaften von Transducern

Ähnlich wie endliche Automaten haben Finite-State-Transducer verschiedene Abschlusseigenschaften, die von Modellen genutzt werden können.

Vereinigung, Konkatenation und Kleene-Abschluss

Wird ein FST gemäß Definition 1 als zugrunde liegender endlicher Automat verstanden, lassen sich die Abschlusseigenschaften Vereinigung, Konkatenation und Kleene-Abschluss direkt auf die Klasse der FST übertragen. Die Operationen werden wie in den Beispielen aus Kapitel 1.2.3. ausgeführt. Als ε-Transitionen dienen im Fall von FST Paare der Form ε:ε. Anschließend können auch diese FST wie Automaten ohne Ausgabe determiniert und minimalisiert werden.

Durchschnitt

Die Klasse der FST ist im Allgemeinen nicht abgeschlossen unter der Operation des Durchschnitts (vgl. Roche und Schabes 1997, S. 18). Eine besondere Klasse der FST, die ε-freien Transducer, sind jedoch sehr wohl abgeschlossen unter Durchschnitt. Dies sind FST, die weder im Eingabe- noch im Ausgabealphabet ε enthalten. Dies kann zum Beispiel erreicht werden, in dem man ε durch ein anderes Symbol ersetzt, das nach der Anwendung des FST wieder aus dem Ausgabewort entfernt wird. Die Klasse der ε-freien Transducer wird aufgrund dieser Eigenschaft intensiv in verschiedenen computerlinguistischen Anwendungen genutzt. Der Durchschnitt zweier FST wird ähnlich konstruiert wie der Durchschnitt zweier Automaten, auch hier wird die Interpretation des zugrunde liegenden Automaten verwendet.

Komposition

Zwei Transducer T_1 und T_2 sind abgeschlossen unter der Operation der Komposition. Werden diese FST als Transduktionen $|T_1| : 2^{\Sigma_1^*} \rightarrow 2^{\Sigma_2^*}$ und $|T_2| : 2^{\Sigma_2^*} \rightarrow 2^{\Sigma_3^*}$ aufgefasst, kann man eine Transduktion $|T_1 \circ T_2| : 2^{\Sigma_1^*} \rightarrow 2^{\Sigma_3^*}$ definieren. Ist $W \in \Sigma_1^*$ wie schon zuvor eine Menge der Eingabewörter von T_1 dann ist $|T_1 \circ T_2|(W) = |T_2|\Big(|T_1|(W)\Big)$ eine Menge von Ausgabewörtern von T_2. Hierbei wird die Menge $|T_1|(W)$ der Ausgabewörter als Menge von Eingabewörtern von $|T_2|$ verwendet. Die beiden Transduktionen werden also in Serie geschaltet. Die Ausgabe eines Transducer wird zur Eingabe eines anderen. Man spricht in diesem Fall auch von einer Transducerkaskade. Es können auch mehr als zwei Transducer zu einer Kaskade zusammengeschaltet werden. Der Nutzen einer solchen Kaskade ist folgender: Man kann nun einfache Transducer konstruieren, die nur Zwischenergebnisse liefern. Auf diese Zwischenergebnisse können weitere Transducer verwendet werden, bis das gewünschte Endergebnis vorliegt.

Eingabe		**f**	**o**	**x**	**+N**	**+P**		
Zwischenergebnis		**f**	**o**	**x**	^	**s**	#	
Ausgabe		**f**	**o**	**x**	**e**	**s**		

Quelle: Jurafsky und Martin (2000, S. 77)

Abbildung 2.10: Komposition zweier FST

Abbildung 2.10 zeigt die Arbeitsweise einer solchen Komposition für zwei Transducer. Der erste Transducer erstellt allgemeine Pluralformen englischer Nomina. Die Pluralform von *fox* wird im Grunde regelmäßig gebildet durch das Anfügen eines Plural-s. Dies wird im Zwischenergebnis erreicht, wobei ^ die Morphemgrenze markiert und # das Wortende. Allerdings wird im Englischen nach Zischlauten wie *x*, *ch*, *sh* ein *e* vor das Plural-s geschoben, was phonetische Gründe hat. Der zweite Transducer ist dafür zuständig, aus dem abstrakten Zwischenergebnis eine richtige englische Pluralform des Wortes *fox* zu bilden, also das *e* einzuschieben und die Marker zu entfernen.

Statt einer Kaskade von Transducern kann auch ein einziger Transducer berechnet werden, der direkt dieselbe Transduktion beschreibt, wie die Kaskade. Der neue Transducer hat damit die folgende Eigenschaft: $T_3 = T_1 \circ T_2$. Auf diese Weise entfallen alle Zwischenergebnisse einer Kaskade.

2.2.3 Transducer mit Endausgabefunktion

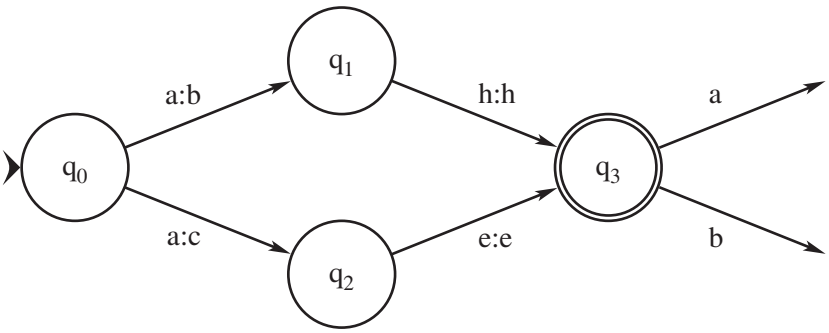
Für bestimmte Anwendungen ist es günstig das Konzept des Transducers noch um eine Endausgabefunktion zu erweitern, da sich dank dieser Funktion leicht Mehrdeutigkeiten modellieren lassen, wie sie oft im Falle von linguistischen Phänomenen auftreten. Erreicht ein Transducer einen Endzustand, wird eine Zusatzausgabe an die bisherige Ausgabe angehängt (konkateniert). Gibt die Endausgabefunktion eine Menge von Ausgaben wieder, wird die bisherige Ausgabe vor jedes Element dieser Menge gestellt, was letztendlich auch in einer Menge von Ausgabewörtern der gesamten Transduktion resultiert. Ein Transducer ist dann ein 8-Tupel $T = (Q, \Sigma_1, \Sigma_2, \delta, \sigma, \rho, q_0, F)$, wo alle bekannten Elemente wie zuvor definiert werden. Lediglich die Funktion $\rho : F \rightarrow 2^{\Sigma_2^*}$ ist neu.

Die Interpretation eines Transducers als Transduktion muss in diesem Fall ebenfalls erweitert werden. Wenn π ein erfolgreicher Pfad ist, dann ist $f(\pi)$ der Endzustand, der am Ende dieses Pfades steht. Damit sieht die Transduktion folgendermaßen aus:

$$|T|(w) = \bigcup_{\pi \in P(w)} \sigma(\pi) \cdot \rho(f(\pi))$$

Ansonsten ändert sich nichts. Ist ein solcher Transducer deterministisch in dem Sinne, dass der Eingabe deterministisch gefolgt werden kann, spricht man auch von einem *p*-subsequentiellen Transducer, wobei p die größtmögliche (endliche) Anzahl der Elemente der Menge $\rho(q)$ ist, wo $q \in F$ darstellt (vgl. Mohri und Allauzen 2002, S. 3). Gibt es zum Beispiel höchstens zwei Elemente, die die Endausgabefunktion zurückgibt, heißt eine solcher Automat 2-Subsequentieller Transducer.

Abbildung 2.11 zeigt einen 2-subsequentiellen Transducer. Die Endausgabefunktion ist dargestellt in Form von Pfeilen, die vom Endzustand q_3 ausgehen, aber in keinen



Quelle: vgl. Mohri (1997, S. 4)

Abbildung 2.11: Ein einfacher FST mit Endausgabefunktion

anderen Zustand übergehen. Sei $w = bb$, dann ist $|T|(w) = |T|(bb) = \{ab\} \cdot \{a, b\} = \{aba, abb\}$.

Transducer mit Endausgabefunktion werden zum Beispiel für Lexika eingesetzt. Ist ein Eingabewort ein Lexikoneintrag, können über die Endausgabefunktion die Daten ausgegeben werden, die mit einem Lexikoneintrag verbunden sind. Wenn es sich um Einträge handelt, die mehrere Bedeutungen haben, kann jede einzelne Bedeutung an die Standardausgabe angehängt werden. Ein solcher Transducer wird in dieser Arbeit als Lexikon verwendet.

2.3 Reguläre Sprachen und menschliches Parsing

Wie das menschliche Sprachmodell aussieht, ist unbekannt. Aussagen darüber kann man höchstens als Vermutungen bezeichnen. Dennoch wird schon seit Jahrzehnten versucht, die Anwendung von formalen Sprachen auch mit psychologischen Aspekten der menschlichen Sprachverarbeitung in Einklang zu bringen. Generell herrscht die Ansicht, dass natürliche Sprachen (gemeint sind vor allem Syntaxmodelle) keine regulären Sprachen sind und somit nicht von endlichen Automaten dargestellt werden können. Ein Hauptargument ist hier die Rekursion, die bei mittig verschachtelten Strukturen auftaucht, wie zum Beispiel Sätzen mit verschachtelten Relativsätzen. Nun haben menschliche Sprecher ab einem gewissen Verschachtelungsgrad ebenfalls Probleme, solche Sätze zu verarbeiten. Allerdings lässt sich nicht sinnvoll begründen, dass eine Verschachtelung des Grades N grammatisch richtig ist, eine Verschachtelung des Grades $N+1$ aber nicht mehr, weshalb angenommen wird, dass die menschliche Sprache kontextfrei ist und damit eine unendliche Zahl von Verschachtelungen als grammatisch richtig akzeptiert werden muss (vgl. Jurafsky und Martin 2000, S. 493).

Die Schwierigkeiten des menschlichen Sprechers werden meist mit Gedächtnisproblemen erklärt. Der Mensch verfügt nicht über genug Speicher, um solche Strukturen

uneingeschränkt verarbeiten zu können. Diese Hypothese schient wiederum auf den regulären Charakter von Sprache hinzudeuten, da kontextfreie Grammatiken, die mit einem eingeschränkten Speicher arbeiten müssen, sehr wohl mit einem endlichen Automaten modelliert werden können. Die Idee, dass die menschlichen Probleme bei mittig verschachtelten Sätzen mithilfe eines menschlichen Parsers mit einer endlichen Menge von Zuständen erklärt werden könnten, wurde von einigen Forschern (vgl. Church 1980, S. 12) aufgenommen und auch auf andere Phänomene übertragen.

Ein weiteres interessantes Argument scheint die Linearität menschlicher Sprachverarbeitung zu sein; Menschen verstehen auch längere Sätze anscheinend ohne größere merkbare zeitliche Verzögerungen. Dies lässt vermuten, dass weder kontextfreie noch andere mächtigere Grammatikformalismen wahrscheinliche Modelle für den menschlichen Parser darstellen, da das Parsen eines Satzes mit n Konstituenten mithilfe kontextfreier Grammatiken bis zu n^3 Schritte erfordern kann. Bei komplizierten Grammatikmodellen kann dies noch deutlich höher ausfallen. Eine mögliche Lösung für dieses Problem stellt die Annäherung durch Finite-State-Modelle an komplexere Formalismen dar.

van Noord (2000, S. 1) gibt zusammenfassend drei Beobachtungen an, die darauf hinweisen, dass menschliche Sprachverarbeitung auf Modellen mit endlichen Zuständen beruht:

- Menschen verfügen nur über einen endlichen (kleinen, eingeschränkten) Speicher für Sprachverarbeitung;
- Menschen haben Schwierigkeiten mit bestimmten grammatischen Konstruktionen, die mit Finite-State-Modellen nicht beschrieben werden können;
- Menschen verarbeiten natürliche Sprache sehr effizient (in linearer Zeit).

Auch Kornai (1985) stellt in seinem Artikel drei weitere Argumente für die Regularität von natürlichen Sprachen vor. Das obige Problem der theoretisch unendlichen Verschachtelung wird von ihm verworfen und zählt nicht als Beispiel für die Unendlichkeit natürlicher Sprachen, da die Grammatikalität dieser Sätze fraglich ist. Als einziges Beispiel echter Unendlichkeit gibt er koordinierende Phrasen an, die uneingeschränkt rekursiv sein können und auch von menschlichen Sprechern problemlos akzeptiert werden. Diese Art von sprachlicher Unendlichkeit kann ohne weiteres mit regulären Sprachen modelliert werden.

Kornais zweites Argument kommt aus der Neurologie. Zitiert werden Quellen, nach denen einzelne Neuronen durch endliche Automaten modelliert werden können. Dreidimensionale Matrizen endlicher Automaten können wiederum durch einen einzigen endlichen Automaten ersetzt werden. Daraus folgert der Autor eine obere Grenze der möglichen Komplexität des Gehirnbereiches, der für die Sprache zuständig ist, woraus wiederum geschlossen werden kann, dass Sprache regulär sein muss, da das Gehirn ein endlicher Automat ist.

Das dritte Argument beruht auf der Annahme, dass es in jeder Sprache nur eine endliche und auch ziemlich beschränkte Zahl von Wortarten und Morphemklassen gibt. Ein Automat beschreibt alle möglichen Kombinationen von Wortarten oder Morphemklassen in einem Satz. Soll dieser Automat einen Satz akzeptieren, gibt es nur eine bestimmte Zahl von Wahlmöglichkeiten für die jeweils nächste Wortart, die keinesfalls unendlich ist. Von einem Zustand aus, gibt also nur so viele Folgezustände wie Wahlmöglichkeiten für die nächste Wortart, wobei zusätzliche Einschränkungen bezüglich der Grammatikalität gelten. In einem minimierten Automat werden Phrasen, die gegeneinander substitutioniert werden können, immer zu demselben Zustand führen. Somit können sowohl koordinierende als auch suboordinierende Sätze akzeptiert werden, da ein Satz, der die Stelle eines Satzteils einnimmt, durch einen einfacheren Satzteil ersetzt werden kann, der wiederum aus einem Wort einer bestimmten Wortart bestehen kann.

2.4 Vor- und Nachteile von Finite-State-Modellen

Auch wenn bisher nicht einmal klar gesagt werden kann, ob die natürlichen Sprachen im menschlichen Geist überhaupt mithilfe von formalen Sprachen welchen Typs auch immer definiert werden können, ist die Theorie der formalen Sprachen ein viel genutztes Werkzeug in der Computerlinguistik. Je nach Klasse und Komplexität sind bestimmte Sprachen entweder ausdruckstärker und zugleich schwieriger zu verarbeiten oder weniger ausdrucksstark, aber dafür effizienter was die Verarbeitung betrifft. Es muss also ein Kompromiss gefunden werden zwischen Expressivität eines Modells und der Effizienz seiner Implementierung am Rechner. Selbst die heutigen Rechnergenerationen sind schnell an der Grenze ihrer Möglichkeiten, wenn die benötigten Algorithmen exponentielle Bearbeitungszeiten aufweisen. Andererseits ist es wünschenswert, dass ein Modell die Anforderungen einer zugrunde liegenden linguistischen Theorie erfüllt.

Dieser Widerspruch zwischen Mächtigkeit einerseits und Effizienz andererseits wird auch am Beispiel von Finite-State-Modellen deutlich. Aus linguistischer Sicht werden Finite-State-Modelle im Allgemeinen als nicht expressiv genug eingeschätzt. Ihre Anwendung und die damit verbundene Forschung produziert kein neues Wissen im Hinblick auf linguistische Theorien (im vorigen Kapitel wurde beschrieben, das einige Forscher durchaus anderer Meinung sind). Neuere Formalismen wie z. B. HPSG (Head-Driven Phrase Structure Grammar) oder LFG (Lexical Functional Grammar) können sowohl komplett am Rechner implementiert werden, als auch als Beschreibungsmodelle von linguistischen Phänomenen verwendet werden, die weit über bloße Syntaxtheorien hinausgehen. Dabei ist eine Implementierung nicht erforderlich. Finite-State-Modelle komplexer Phänomene dagegen sind unleserlich für den Menschen, da sie nur schwer ganzheitlich erfasst werden können. Ein Graph mit vielen Tausenden Knoten und Übergängen wirkt eher Angst einflößend als modelltheoretisch überzeugend. Dies ist auch der Grund warum im nächsten Kapitel viele Au-

tomaten nicht mehr grafisch, sondern anhand ihrer Eigenschaften beschrieben werden. Selbst einfachere Automaten müssen aus Platz- und Übersichtlichkeitsgründen vereinfacht werden, indem z. B. mehrere Übergänge zu einem Übergang zusammen gefasst werden, dessen neues Symbol eine Verkettung der einzelnen Symbole darstellt.

Die Nachteile auf theoretischer Ebene werden jedoch durch viele praktische Eigenschaften wieder wettgemacht. Finite-State-Modelle sind vom mathematischen Standpunkt her deutlich besser erforscht als andere Sprachklassen. Sie sind effizienter was Zeitaufwand und Speicherverbrauch betrifft. Durch ihre Abschlusseigenschaften können Module zu einem einzigen Modell kombiniert werden, das anschließend durch Determinierung und Minimierung zu einer optimalen Repräsentation berechnet werden kann. Hinzu kommen neuerdings Techniken, die es erlauben Formalismen von höherer Komplexität mit Finite-State-Mitteln zu simulieren. Man versucht sich diesen höheren Formalismen mit möglichst einfachen Mitteln zu nähern und dabei diese Einfachheit beizubehalten. Allerdings sind dies wie gesagt nur Annäherungen, die in bestimmten Situationen versagen können und es wahrscheinlich auch tun. Diese Möglichkeiten werden jedoch zum Wohle der Effizienz in Kauf genommen.

KAPITEL 3

Das Finite-State-Modell

In diesem Kapitel werden die in Kapitel 2 beschriebenen Formalismen verwendet, um ein Finite-State-Modell für ein- und mehrfach zusammengesetzte Komposita zu entwerfen. Die strukturellen Eigenschaften der Komposita sowie die Distribution der Fugenelemente wurden in Kapitel 1 beschrieben.

Im Anschluss wird gezeigt, wie Finite-State-Mittel zur Modellierung eines Lexikons mit Kompositasegmenten verwendet werden. Dieses Lexikon wird zu einem Mechanismus erweitert, der in der Lage ist, Komposita zu segmentieren. Die zunächst naive Segmentierung muss disambiguiert werden. Dazu werden die Erstglieder in Bezug auf die Anzahl der Silben und das Vorkommen bestimmter Suffixe hin untersucht. Auch hier werden Automaten und Transducer verwendet. Die Struktur der Komposita und die Bildung der Kompositionsstammformen wird mithilfe von Transducer-Regeln dargestellt. Diese Regeln sind letztendlich für die Disambiguierung verantwortlich. Schließlich werden alle Modellteile mit einander in Verbindung gesetzt und die gesamte Funktionalität wird beschrieben.

3.1 Das Lexikon

Im Folgenden wird die Struktur des Lexikons des Finite-State-Modells besprochen. Die im Lexikon enthaltenen Daten werden dargestellt. Weiterhin wird gezeigt, wie durch einige formale Modifikationen aus einem deterministischen Lexikon-Transducer, der lediglich in der Lage ist, einzelnen Kompositasegmenten ihre Tagsets zu zuweisen, ein Transducer konstruuiert wird, der eine komplette Kompositasegmentierung durchführen kann. Es wird außerdem diskutiert, ob ein solcher Transducer determiniert werden kann. Hierzu wird ein Beweis angeführt, dass dies nicht der Fall ist.

3.1.1 Lexikoneinträge und Tagsets

In Unterkapitel 1.5.2 wurde die Bedeutung des Lexikons für das Tagging besprochen. An dieser Stelle wird die Mikrostruktur des Lexikons bzw. der Aufbau der Tagsets im Einzelnen beschrieben. Aufgrund der im Lexikon enthaltenen linguistischen Daten werden in Unterkapitel 3.3 die Distributionsregeln zur Analyse möglicher Kompositionsstammformen entworfen. Diese Daten müssen den in 1.6.2 beschriebenen morphologischen Kriterien entsprechen.

Die Wortart ist das Hauptkriterium für die Wahl der Fugenelemente. Somit werden alle Segmente im Lexikon bezüglich ihrer Wortart erfasst. Nur die Substantive und Verben nehmen in ihren Kompositionsstammformen Interfixe an. Die übrigen Wortarten bilden Kompositionsstammformen, die ihren Grundformen entsprechen. Adjektive sind hier eine Ausnahme, da unter Umständen auch deren gesteigerte Form in die Komposition eingehen kann.

Außer im Fall der Substantive ist die Wortart das einzige benötigte Kriterium zur Bestimmung der Interfixdistribution. Bei Verben ist das Erscheinen der Fuge sehr unregelmäßig und es wird prinzipiell angenommen, dass beide Kompositionsstammformen, mit und ohne *-e*, möglich sind. Bei den Substantiven wurden viele weitere Kriterien ausgemacht, von denen zumindest ein Teil bequem im Lexikon kodiert werden kann.

Die morphologische Grundausstattung der Substantive wird im Lexikon erfasst. Angegeben wird das grammatische Geschlecht, die Flexionsklasse im Singular und die Flexionsklasse im Plural. In Verbindung mit den Flexionsklassen wird ebenfalls festgehalten, ob es sich um Singulariatantum oder Pluraliatantum handelt, was Einfluss auf die Wahl von Interfixen mit pluralischer Restbedeutung hat.

Ein einzelnes Tagset besteht aus mehreren Tags, von denen jedes eine grammatische Eigenschaft beschreibt. Innerhalb eines Tagsets sind die Tags in einer bestimmten Reihenfolge enthalten, um gewährleisten zu können, das bekannt ist, welcher grammatischen Eigenschaft der Wert des Tags entspricht. Auf Abbildung 3.1 sieht man zwei Tabellen. In der ersten sind die Bedeutungen aller verwendet Tags zusammengestellt. Die zweite Tabelle beschreibt die mögliche Positionen der Tags innerhalb der einzelnen Tagsets. Die Bezeichnungen der Flexionsklassen stammen aus Dudenredaktion (1998, S. 226 u. 229). Im Lexikon liegt ein Tagset als Zeichenkette vor, die eine Sequenz von Tags darstellt. Aus diesem Grund ist jedem Tag ein „+" vorangestellt, um die einzelnen Tags von einander unterscheiden zu können. Das Sonderzeichen „#" kennzeichnet immer das Ende eines Tagsets.

Dem Lexikoneintrag *drucker* entspricht z. B. das Tagset +N+MS+S1+P2#, das ein maskulines Substantiv mit */e/s*-Singular und ∅-Plural beschreibt. Einträge, die lexikalisch mehrdeutig sind, verfügen über mehrere Tagsets, so z. B. der Eintrag *druck* mit +N+MS+S1–P# und +V#. Im folgenden Unterkapitel wird gezeigt, wie die Tagsets

Tag	Bedeutung	Tag	Bedeutung
+N	Substantiv	+V	Verb
+A	Adjektiv	+F	Flexionsloses Wort
+I	Interfix		
+MS	Maskulinum	+FM	Femininum
+NT	Neutrum		
+S1	*/e/s*-Singular	+S2	*/e/n*-Singular
+S3	∅-Singular	–S	Pluraliatantum
+P1	*/e/*-Plural	+P2	∅-Plural
+P3	*/e/n*-Plural	+P4	*er*-Plural
+P5	*s*-Plural	–P	Singulariatantum
#	Ende des Tagsets		

1. Stelle	2. Stelle	3. Stelle	4. Stelle	5. Stelle
+N	+MS, +NT, +FM	+S1, +S2, +S3, –S	+P1, +P2, +P3, +P4, +P5, –P	#
+V	#			
+A	#			
+P	#			
+I	#			

Abbildung 3.1: Bedeutung und mögliche Reihenfolge der Tags

in einem Lexikon-Transducer einzelnen Segmenten zugeordnet werden. Anschließend werden die Tagsets auch bei der Segmentierung ganzer Komposita verwendet.

3.1.2 Ein Transducer als Basis-Lexikon

In Kapitel 1.5 wurde die Bedeutung des Lexikons für die Segmentdefinition, sowie für die Vorgänge der Segmentierung und des Taggings besprochen. Im Folgenden wird gezeigt, wie beide Prozesse mithilfe eines einzelnen Transducers realisiert werden können. Der Transducer wird ein Kompositum als Eingabe annehmen und eine segmentierte Form zurückgeben. Falls die in Kapitel 1.5.3 beschriebenen Mehrdeutigkeiten vorliegen, ergibt sich eine Menge von Lesarten, deren Elementezahl sich gemäß der Formel aus 1.5.3 berechnen lässt.

Deterministische und azyklische Transducer sind besonders geeignet für die Beschreibung von computerbasierten Lexika. Um ein Wort mit n Buchstaben in einem auf einem deterministischen Transducer basierenden Lexikon ausfindig zu machen, reichen genau n Schritte. Man muss lediglich dem Pfad im Transducer folgen, des-

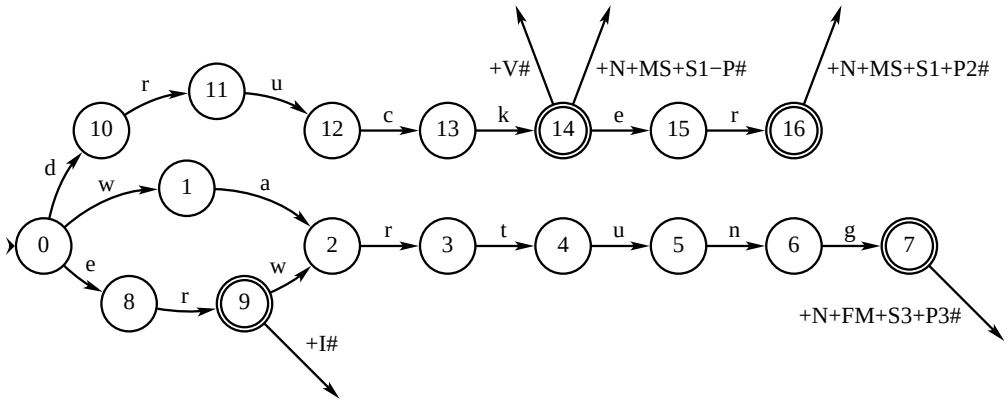


Abbildung 3.2: Ein vereinfachter Lexikon-FST T_{dic}

sen Eingangssymbole dem gesuchten Wort entsprechen. In einem deterministischen Transducer gibt es nur einen solchen Pfad. Die Tagsets, die einem Lexikoneintrag zugeordnet sind, werden meist am Ende eines solchen Pfades ausgegeben, wofür die Endausgabefunktion ρ zuständig ist, wie sie in Kapitel 2.2.3 eingeführt wurde. Damit handelt es sich bei einem solchen Transducer um einen p-subsequentiellen Transducer. Der Wert von p wird dabei durch die größte Anzahl der möglichen lexikalen Mehrdeutigkeiten eines Eintrags bestimmt.

Abbildung 3.2 zeigt einen 2-subsequentiellen Transducer T_{dic}, der alle möglichen Segmente des Kompositums *Druckerwartung* als Lexikoneinträge enthält. Dabei handelt es sich um ein Minmalbeispiel eines Lexikons. Die tatsächliche Version verfügt über mehr als 40.000 Einträge, entspricht aber in Sachen Funktionalität dem Beispiel T_{dic}. Folgt man auf Abbildung 3.2 einer Sequenz von Übergängen vom Startzustand bis zu einem Endzustand, entspricht die Verkettung der Symbole dieser Übergänge einem Lexem. Dies ist auch dann der Fall, wenn der Endzustand nicht der letzte erreichbare Zustand ist, sondern noch weiteren Übergängen gefolgt werden kann. In diesem Fall ist ein Lexikoneintrag einfach nur ein Präfix[6] eines anderen Eintrags.

Der Beispieltransducer T_{dic} gibt neben den Tagsets, die einem Lexikoneintrag entsprechen, auch noch einmal den Eintrag selbst aus, an dessen Ende die annotierten Informationen bei der Ausgabe anhängt werden. Da dieser Transducer keine Modifikationen an den Segmenten selbst vornimmt, kann die Ausgabefunktion σ gegenüber den Definitionen in Interpretation 2 (S. 45) stark vereinfacht werden. In diesem Fall wird $\sigma(q, a, q') = a$ angenommen. Für den Lexikoneintrag *druck* sieht die Ausgabe von T_{dic} dann folgendermaßen aus:

$$|T_{\text{dic}}|(\text{druck}) = \{\text{druck}_{+\text{N+MS+S1–P\#}}, \text{druck}_{+\text{V\#}}\}$$

[6] Gemeint ist hier nicht das Präfix als wortbildendes Morphem, sondern als Zeichenkette, die zwei Einträgen als Anfangssequenz gemeinsam ist.

Für diesen Eintrag gibt die Endausgabefunktion ρ zwei mögliche Tagsets wieder, was in Form von lexikaler Mehrdeutigkeit in zwei möglichen Interpretationen von *druck* resultiert. In dem Beispiel sind die Tags klein gedruckt, was nur der besseren Lesbarkeit dienen soll; tatsächlich handelt es sich bei jedem Element der Ergebnismenge um einfache Zeichenketten, in denen alle Zeichen gleichberechtigt sind.

Auf der Ebene des Lexikons spielt sich die Unterscheidung von Lexemen und Interfixen ab, die in der Transducerstruktur alle in Form von Symbolsequenzen verzeichnet sind, sich aber aufgrund der zugeteilten Tagsets unterscheiden.

An dieser Stelle sei erwähnt, das die Konstruktion eines solchen p-subsequentiellen Transducers auf einem Algorithmus von Daciuk et al. (1998) zur Konstruktion von minimalen azyklischen Automaten aufgrund von alphabetisch sortierten Wortlisten basiert. Da der Transducer auf allen Übergängen identische Ein- und Ausgabesymbole besitzt, kann er formal als einfacher Automat interpretiert werden. In dieser Arbeit wird jedoch eine Endausgabefunktion verwendet, die bei Daciuk et al. nicht vorgesehen ist. Der Algorithmus wurde entsprechend angepasst, indem die Äquivalenzbedingungen von Endzuständen modifiziert wurden. So reicht die Bedingung, dass zwei Endzustände äquivalent sind, wenn ihre rechtsseitigen Sprachen identisch sind, nicht mehr aus. Hinzu kommt die Bedingung, dass die Werte der Endausgabefunktion ρ ebenfalls identisch sein müssen. Zwei Endzustände $q, q' \in F$ sind nicht äquivalent, wenn $\rho(q) \neq \rho(q')$.

Ein so definiertes Lexikon, das alle möglichen Kompositasegmente einschließlich der Interfixe enthält, dient als Grundlage für den eigentlichen Segmentierungsvorgang. Das Lexikon selbst kann in dieser Form nicht alle möglichen Komposita enthalten, da es im Grunde nur die kompakte Form einer Wortliste mit Annotationen darstellt. Ein azyklischer endlicher Automat ist nicht in der Lage unendliche Strukturen darzustellen. Der Beweis dafür ist einfach und folgt aus der Definition der endlichen Automaten. Die Menge der Zustände sowie die Menge der Übergänge ist endlich, woher die Bezeichnung für diese Formalismen herrührt. Die längste mögliche Struktur eines endlichen Automaten mit einen endlichen Menge an Zuständen ist ein linearer Automat. Beginnt man vom Anfangszustand aus, den Übergängen dieses Automaten zu folgen, erreicht man irgendwann zwangsläufig einen letzten Zustand, von dem aus es keine weiteren Übergänge mehr gibt. Da Zyklen ausgeschlossen wurden, folgt daraus, dass keine potentiell unendlichen Strukturen dargestellt werden können und somit auch nicht die deutschen Komposita.

Damit das Lexikon Komposita enthalten kann, muss die Lexikonstruktur modifiziert werden. Dies wird im folgenden Unterkapitel beschrieben.

3.1.3 Lexikonstruktur und Segmentierung

Das Basis-Lexikon T_{dic} enthält Kompositasegmente. Ein Lexikon, das alle möglichen Zeichenketten akzeptiert, die aus Sequenzen dieser Segmente bestehen, wird neben

vielen ungrammatischen Sequenzen auch alle grammatisch richtigen Komposita akzeptieren. Ziel dieses Unterkapitels ist es, formal einen Transducer zu beschreiben, der innerhalb dieser Sequenzen die enthaltenen Segmente identifiziert und entsprechend annotiert. Bei möglichen strukturellen und lexikalen Mehrdeutigkeiten sollen diese Vervielfachungen der Interpretationsmöglichkeiten berücksichtigt werden. An dieser Stelle ist die grammatische Richtigkeit der Segmentierungen noch nicht relevant.

In Unterkapitel 2.1.3 wurden die Abschlusseigenschaften von Automaten besprochen, darunter auch der Kleene-Abschluss. Mithilfe des Kleene-Abschlusses können Automaten konstruiert werden, die beliebige Folgen von Wörtern eines Ursprungsautomaten als Eingabewörter akzeptieren. Der Kleene-Abschluss des Basislexikons T_{dic} stellt einen Transducer $(T_{\text{dic}})^+$ her, der alle möglichen Folgen von enthaltenen Einträgen als Eingabewort annimmt. Da das Kleene-Plus verwendet wird, bleibt hier das leere Wort ausgespart.

Der Kleene-Abschluss eines Automaten wird erzeugt, indem jeder Endzustand über einen ε-Übergang mit dem Anfangszustand verbunden wird. Formal kann dies durch eine Modifikation der Übergänge erreicht werden, einfacher ist jedoch eine Modifikation der über einem einzelnem Eingabesymbol definierten Übergangsfunktion dahin, dass der Startzustand q_0 zu jeder Zustandsmenge hinzugefügt wird, in der auch ein Endzustand erhalten ist, ansonsten wird die Zustandsmenge nicht erweitert. Formal sieht dies folgendermaßen aus:

$$\hat{\delta}(q,a) = \begin{cases} \delta(q,a) \cup \{q_0\} & \text{falls } \exists p \in \delta(q,a) \wedge p \in F \\ \delta(q,a) & \text{sonst} \end{cases}$$

Die neue Funktion $\hat{\delta}$ wird dann einfach statt der üblichen über einzelnen Eingangssymbolen definierten Übergangsfunktion δ verwendet. Abbildung 3.3 zeigt das graphische Ergebnis einer solchen Modifikation, die hier in Form von zusätzlichen ε-Übergängen dargestellt wird.

Damit sind die Modfikationen jedoch noch nicht abgeschlossen. Der neue Transducer $(T_{\text{dic}})^+$ akzeptiert jetzt alle möglichen Folgen von Segmenten, markiert aber noch nicht die Grenzen zwischen ihnen. Auch werden nur die Tagsets des letzten gefunden Segments an das komplexe Eingabewort angehängt. Das Kompositum *Druckerwartung* als Eingabewort liefert nun das folgende Ergebnis:

$$|(T_{\text{dic}})^+|(\text{druckerwartung}) = \{\text{druckerwartung}_{+\text{N}+\text{FM}+\text{S3}+\text{P3}\#}\}$$

Will man dem Eingabewort *Druckerwartung* über die Pfade in $(T_{\text{dic}})^+$ folgen, zeigt sich, dass in Wirklichkeit drei mögliche Pfade existieren, die auf verschiedene Segmentierungsarten hinweisen, aber dieselbe Ausgabe liefern. Da zu jedem Segment Tagsets im Lexikon enthalten sind und letztendlich die Tagsets auch mit den Seg-

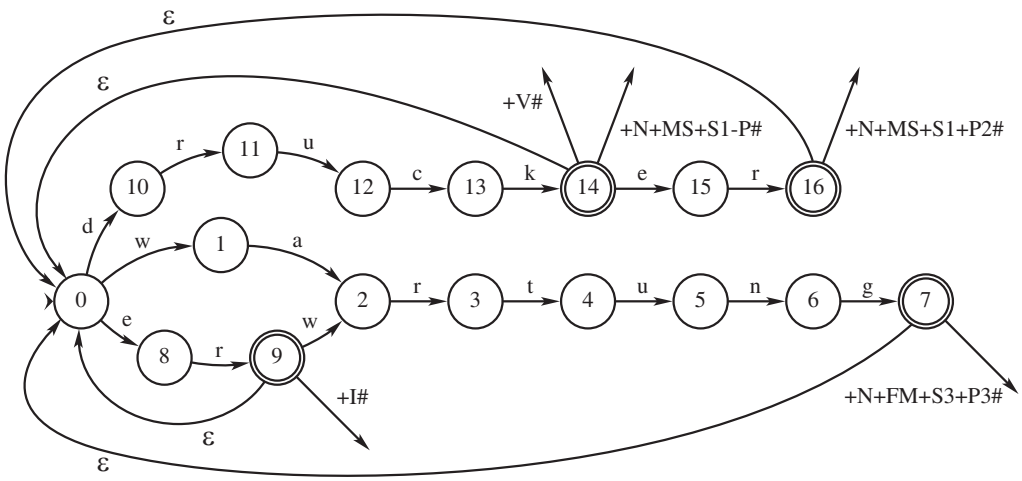


Abbildung 3.3: Der Lexikon-FST nach dem Kleeneabschlus $(T_{\text{dic}})^+$

menten im Analyseergebnis in Verbindung gebracht werden sollen, bieten sich diese Informationen an, um die Segmente von einander zu trennen.

Die Endausgabefunktion ρ ist zuständig für das Hinzufügen der Tagsets am Ende der Ausgabe. Um die Segmentgrenzen zu markieren, müsste sie aber sozusagen im „Vorübergehen" Tagsets beim Erreichen eines Endzustandes an die bisherige Ausgabe anhängen, und zwar auch dann, wenn die Ausgabe noch nicht abgeschlossen ist. Zu diesem Zweck muss jeder Zustand, sobald er erreicht wird, darauf geprüft werden, ob ihm linguistische Informationen zugeordnet sind. Ist dies der Fall, müssen diese Informationen angehängt werden. Die Funktion ρ verliert bei einer solchen Modifikation ihren Endausgabecharakter und wird von nun an gemäß ihrer Hauptaufgabe als Tagging-Funktion bezeichnet. Um die ständige Überprüfung auf Segmentenden zu gewährleisten, muss die Tagging-Funktion bei jedem Zustandswechsel aufgerufen werden. Zu diesem Zweck wird die vereinfachte Ausgabefunktion σ verändert:

$$\hat{\sigma}(q, a, q') = \sigma(q, a, q') \cdot \hat{\rho}(q') = a \cdot \hat{\rho}(q')$$

Somit wird bei jedem Übergang für den Zielzustand eine erweiterete Tagging-Funktion $\hat{\rho}$ aufgerufen, die gleich beschrieben wird. Bisher ist ρ nur für Endzustände, also die Menge F, definiert und wird zu $\hat{\rho}$ erweitert, die für die Menge aller Zustände Q definiert wird:

$$\hat{\rho}(q) = \begin{cases} \rho(q) & \text{falls } q \in F \\ \varepsilon & \text{sonst} \end{cases}$$

So ist $\hat{\rho}$ für alle Zustände definiert, liefert aber nur eine Ausgabe, wenn einem Zustand Tagsets zugeordnet sind, was nur auf die Endzustände zutrifft. Ist ein Zustand kein

Endzustand, wird das leere Wort zurückgegeben. Damit sind die Funktionen $\hat{\delta}$, $\hat{\sigma}$ und $\hat{\rho}$ durch Modifizierungen der ursprünglichen Funktionen gegeben und beschreiben in Verbindung mit dem Transducer T_{dic} die Funktionsweise eines Transducer T_{seg}, der für die Eingabe von *druckerwartung* folgende Ausgabe liefert, die den zu Anfang des Unterkapitels gestellten Anforderungen entspricht:

$$
\begin{aligned}
&|T_{\text{seg}}|(\text{druckerwartung}) = \{\\
&\quad \text{drucker}_{+\text{N}+\text{MS}+\text{S1}+\text{P1}\#}\text{wartung}_{+\text{N}+\text{FM}+\text{S3}+\text{P3}\#},\\
&\quad \text{druck}_{+\text{N}+\text{MS}+\text{S1}-\text{P}\#}\text{erwartung}_{+\text{N}+\text{FM}+\text{S3}+\text{P3}\#},\\
&\quad \text{druck}_{+\text{V}\#}\text{erwartung}_{+\text{N}+\text{FM}+\text{S3}+\text{P3}\#},\\
&\quad \text{druck}_{+\text{N}+\text{MS}+\text{S1}-\text{P}\#}\text{er}_{+\text{I}\#}\text{wartung}_{+\text{N}+\text{FM}+\text{S3}+\text{P3}\#},\\
&\quad \text{druck}_{+\text{V}\#}\text{er}_{+\text{I}\#}\text{wartung}_{+\text{N}+\text{FM}+\text{S3}+\text{P3}\#}\\
&\}
\end{aligned}
$$

Die Segmente werden durch das jeweils erste „+" deutlich von den nachfolgenden linguistischen Annotationen getrennt. Die Annotationen selbst werden mit einem einzelnem „#" abgeschlossen, das außerdem die Segmentgrenzen eindeutig markiert. Die obige Kompositasegmentierung entspricht den in Unterkapitel 1.5.3 beschriebenen Mehrdeutigkeiten.

Im Lexikon-FST T_{dic} kann der Eingabe deterministisch gefolgt werden. In der modifizierten Version T_{seg} geht der Determinismus durch den Kleene-Abschluss verloren. Während endliche Automaten immer determiniert werden können, ist dies im Falle von Transducern in Bezug auf die Eingabe nicht unbedingt möglich. Die Kriterien, wann ein Transducer determiniert werden kann, sind noch immer Gegenstand der aktuellen Forschung und werden vorwiegend mathematisch oder algorithmisch bestimmt (vgl. z. B. Mohri und Allauzen 2002). Allerdings können aufgrund des Transduktionsergebnisses, der Eigenschaften von sequentiellen Transducern und der Struktur der deutschen Komposita gewisse Rückschlüsse auf die Determinierungsmöglichkeit von T_{seg} gezogen werden.

Bei der Transduktion, die durch einen deterministischen Transducer definiert wird, handelt es sich um eine Funktion, das heißt, dass einem Eingabewort genau ein Ausgabewort zugeordnet wird. Mehrdeutigkeit kann erfasst werden, indem die Ergebnisse einer Endausgabefunktion an das bisherige Ausgabewort angehängt werden und für jedes Element dieser Endausgabe ein neues Ausgabewort in der Ergebnismenge der Transduktion hinzugefügt wird. Die Menge der möglichen Elemente, die eine Endausgabefunktion zurückgeben kann, ist endlich.

Betrachtet man das obige Transduktionsergebnis, zeigt sich, dass von fünf Lesarten nur drei verschiedene Segmentgrenzen aufweisen, die übrigen Lesarten ergeben sich durch die lexikale Mehrdeutigkeit der Segmente. Ignoriert man diese lexikale Mehrdeutigkeit, existieren drei Segmentierungsarten, die direkt auf die Struktur des Kompositums *Druckerwartung* zurückgehen. Daraus folgt, dass ein deterministischer

Transducer ohne Endausgabefunktion nicht in der Lage ist, diese Ergebnisse zu liefern, da dessen Ergebnismenge nur aus einem Element bestehen darf.

Nimmt man nun einen deterministischen Transducer mit Endausgabefunktion an, muss diese Funktion genau drei verschiedene Ausgaben an die deterministische Ausgabe anfügen, um diese Zahl von Elementen zu erreichen. Nun sind die Komposita aber in Bezug auf die Zahl ihrer Segmente unbeschränkt, da jedes Kompositum wiederum als Glied eines anderen Kompositums auftauchen kann. Mit wachsender Komplexität ergeben sich zusätzliche Segmentierungsarten, die ebenfalls nur durch die Menge der Elemente der Endausgabefunktion bestimmt werden können. Daraus ergibt sich, dass die Anzahl der Elemente, die die Endausgabefunktion zurückgeben muss, um alle Segmentierungsarten zu kennzeichnen, proportional zur Komplexität des Eingabewortes ist und außerdem ein Vielfaches der Anzahl der Segmente betragen kann. Aus der potentiellen Unendlichkeit der Komposita folgt, dass keine maximale Anzahl von Elementen angegeben werden kann, die von einer Endausgabefunktion zurückgegeben werden müsste, um eine deterministische Version von T_{seg} zu berechnen. Dies widerspricht der Definition der Endausgabefunktion, deren Ergebnismenge endlich sein muss.

T_{seg} dagegen ist als nicht-deterministischer Transducer in der Lage Komposita von unbeschränkter Komplexität zu segmentieren, ohne dass dessen Struktur abhängig wäre von der Länge der Eingabeworte. Damit folgt aus dem Nicht-Determinismus der Komposita bezüglich ihrer Segmentierung, direkt die Unmöglichkeit einen deterministischen Transducer zu Segmentierungszwecken anzugeben.

3.2 Erstgliedanalyse

Neben den im Lexikon kodierten morphologischen Eigenschaften, haben auch andere Kriterien Einfluss auf die Bildung der Kompositionsstammformen der Erstglieder. Vor allem bei den Distributionsregeln des unparadigmischen *-s* spielen auch Silbenzahl, Auslaut und Suffixgestaltung des Erstglieds eine Rolle. In den folgenden Unterkapiteln werden einige Mechanismen zur Erstgliedanalyse vorgestellt, die zwischen der naiven Segmentierung und der Überprüfung der grammatischen Richtigkeit der gefundenen Segmentierungsarten stattfindet. Die Erstgliedanalyse fügt den Segmenten weitere Tags zu, die anschließend bei der Gestaltung der Distributionsregeln miteinbezogen werden.

3.2.1 FST zur Silbenanalyse

Bezüglich der Silbenzahl der Erstglieder ist eine Unterscheidung zwischen einsilbig und mehrsilbig ausreichend. Es sind z. B. mehrsilbige Feminina mit *-t* im Auslaut, die ihre Kompositionsstammform regelmäßig mit unparadigmischem *-s* bilden. Um Mehrsilbigkeit festzustellen, muss die Struktur der Silbe berücksichtigt werden. Man

unterscheidet hier zwischen Silbenschale (bestehend aus Silbenkopf und Silbenkoda) und dem Silbenkern (vgl. GLÜCK 2000, S. 632). Der Silbenkern ist obligatorisch, während Kopf und Koda nicht bei jeder Silbe vorhanden sind. Im Deutschen ist der Silbenkern vokalisch, die Silbenschale besteht aus Konsonanten. Um die Anzahl der Silben in einem Wort zu ermitteln, reicht es aus, die vorhandenen Silbenkerne zu zählen. Das Vorkommen von Silbenköpfen und -koda ist unerheblich, genauso kann die genaue Position der Silbengrenzen beiseite gelassen werden. Dazu muss nun die Struktur der Silbenkerne bekannt sein. In deutschen Wörtern müssten die Grapheme und Graphemgruppen von Abbildung 3.4 die meisten vokalischen Silbenkerne abdecken:

Einzelvokale:	*a, e, i, o, u, ä, ö, ü*
Doppelvokale:	*aa, ee, ie, oo*
Diphthonge:	*ai, au, äu, ei, eu*

Abbildung 3.4: Graphematische Repräsentationen möglicher Silbenkerne

Problematisch in Bezug auf die Einbindung in einen Transducer sind die Doppelvokale und die Diphthonge. Gerade im Fall von Zusammensetzungen können ähnliche Strukturen auftauchen, bei denen die Silbengrenze zwischen den beiden Vokalen verläuft. Es wird angenommen, dass dieses Problem durch die vorangegangene Segmentierung gelöst wurde und dadurch in einzelnen Segmenten enthaltene Folgen von Vokalen einem Silbenkern entsprechen. Jedes Vorkommen von z. B. *au* wird als ein Silbenkern gewertet und nicht als zwei einzelne Silbenkerne *a* und *u*. So ist z. B. *Maut* einsilbig und nicht mehrsilbig *Ma-ut**. Bei der Modellierung eines entsprechenden Transducers ergeben sich jedoch Schwierigkeiten. Das Beispiel *Maut* würde ohne die Einbeziehung von phonotaktischen Regeln doppeldeutig, einmal als einsilbig und einmal als mehrsilbig, interpretiert werden. Um dies zu verhindern, reicht es festzuhalten, dass nach einem einzelnem Silbenkern *a* kein Silbenkern *u* auftauchen kann. Stattdessen soll eine derartige Folge als einzelner Silbenkern interpretiert werden. Dies kann erreicht werden, indem man anhand der Auflistung der Doppelvokale und Diphthonge die Grapheme isoliert, die zusammen mit einem Einzelvokal eben diese Doppelvokale und Diphthonge bilden. Mit einem vorangestellten *a* bilden *a, i, u* die graphematische Representation der Doppelvokale oder Diphthonge *aa, ai, au*. Es sind also die Vokale *a, i, u*, die nicht direkt auf ein *a* folgen können und gleichzeitig einen zweiten Silbenkern bezeichnen. Eine derartige Regel kann als Transducer beschrieben werden. Hierzu werden erst einige Mengen eingeführt, die die Lesbarkeit der Regel erleichtern sollen. Die Menge $V = \{a, e, i, o, u, ä, ö, ü\}$ ist die Menge der graphematischen Repräsentationen der Einzelvokale. Die Menge K ist die Menge der Grapheme, die für Konsonanten verwendet werden. Eine Menge $V \setminus \{a, i, u\}$ ist die Menge V ohne die Vokale *a, i, u*. Mit einem „?" wird die Menge aller Symbole bezeichnet, die zum Alphabet des Transducers gehört. Einzelsymbole, die keine durch

„:" getrennten Paare von Ein- und Ausgabesymbolen sind, sind Abbildungen auf sich selbst, z. B. entspricht *a* dem Paar *a:a*.

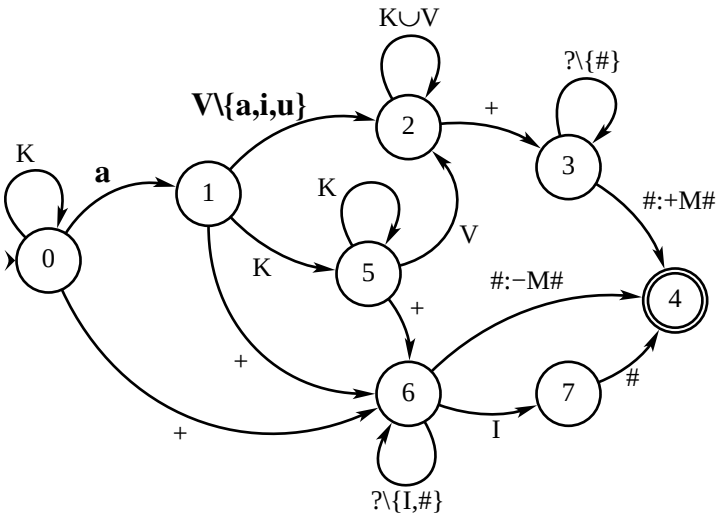


Abbildung 3.5: Regel A_1 für Erstsilben mit Silbenkern *a*

Abbildung 3.5 zeigt einen Transducer A_1, mit dessen Hilfe festgestellt werden kann, ob ein Segment einsilbig oder mehrsilbig ist. Berücksichtigt werden durch diesen Transducer nur Segmente, deren erste Silbe einen durch *a* repäsentierten Silbenkern besitzt. Der Transducer A_2 auf Abbildung 3.6 besitzt die gleiche Funktion für Silbenkerne mit *e*. Andere Silbenkerne werden durch diese Regeln nicht akzeptiert. Die in der Abbildung 3.5 fett gedruckten Symbole illustrieren, wie verhindert wird, dass Diphthonge und Doppelvokale, die *a* als Bestandteil enthalten, als Anzeichen von Mehrsilbigkeit gewertet werden. Auf *a* können hier, wie zuvor gefordert, nur Vokale folgen, die mit *a* keine Diphthonge oder Doppelvokale bilden. Folgt ein anderer Vokal auf *a* ist dies ein Anzeichen für Mehrsilbigkeit. Durch diese Regel wird nur die genaue Form der ersten Silbe beschrieben. Die Gestaltung des Silbenkerns der zweiten Silbe ist unnötig, da jeder auftauchende Vokal auf einen neuen Silbenkern hindeutet, dabei ist unerheblich, ob er Bestandteil eines Diphthongs oder Doppelvokals ist. Aus der Existenz eines zweiten Silbenkerns folgt direkt die Mehrsilbigkeit des Segments. Die Existenz weiterer Silben wird nicht überprüft, sondern nur noch berücksichtigt, um die Regel bis zum Ende des Segments durchlaufen zu lassen. Am Ende wird bei Mehrsilbigkeit ein zusätzliches Tag +M an das Tagset des Segments angehängt. Entsprechend wird –M bei Einsilbigkeit angefügt.

Die Regel ist so gestaltet, dass Interfixe ignoriert und deren Tagsets nicht modifiziert werden. Sie sind ohnehin einsilbig oder im Falle von *-s* nicht einmal silbisch. Die Regel-Transducer A_3 bis A_8 für die übrigen Einzelvokale unterscheiden sich von einander nur durch die fett gedruckten Symbole, so wie A_1 und A_2 auch nur in dieser Hinsicht verändert sind.

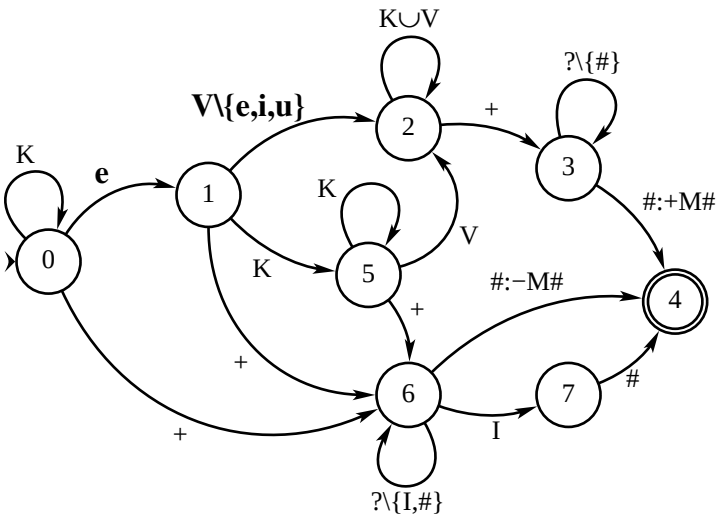


Abbildung 3.6: Regel A_2 für Erstsilben mit Silbenkern *e*

Die Regeln A_9 bis A_{12} beschreiben die Doppelvokale. Die Struktur dieser Regel-Transducer ist einfacher als bei den Einzelvokalen, da nun angenommen wird, dass jeder Vokal, der auf einen Doppelvokal folgt, einen neuen Silbenkern kennzeichnet. Abbildung 3.7 zeigt die Regel A_9 für Segmente, deren erste Silbe den Silbenkern *aa* aufweist. Auch hier genügt es, um die Regeln für die übrigen Doppelvokale zu erhalten, die fett gedruckten Symbole zu ersetzen. Die restliche Struktur des Transducers bleibt erhalten.

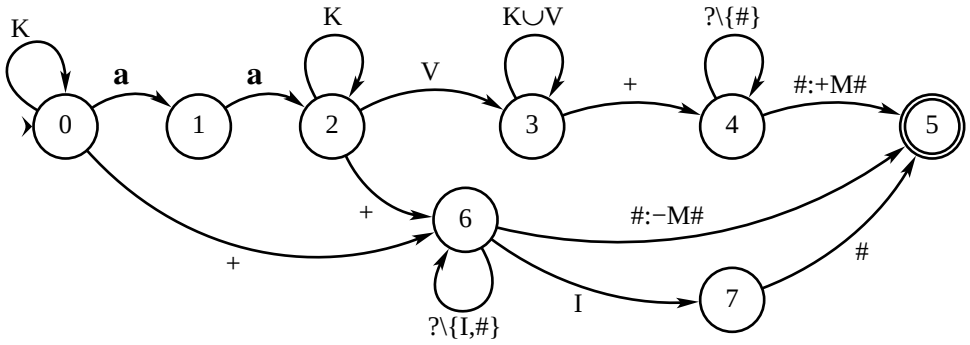


Abbildung 3.7: Regel A_9 für Erstsilben mit Silbenkern *aa*

Vergleicht man beispielsweise A_9 mit A_1 in Hinblick auf die Gestaltung der Vokalfolgen, sieht man, wie sich diese gegenseitig ausschließen. Somit wird eine Doppelinterpretation von *aa* verhindert.

Auf Abbildung 3.8 sieht man die Regel A_{16}, die Silbenkerne mit *eu* modelliert. Auch hier entspricht die Struktur der Regel der der Doppelvokale. Um die Regeln A_{13} bis

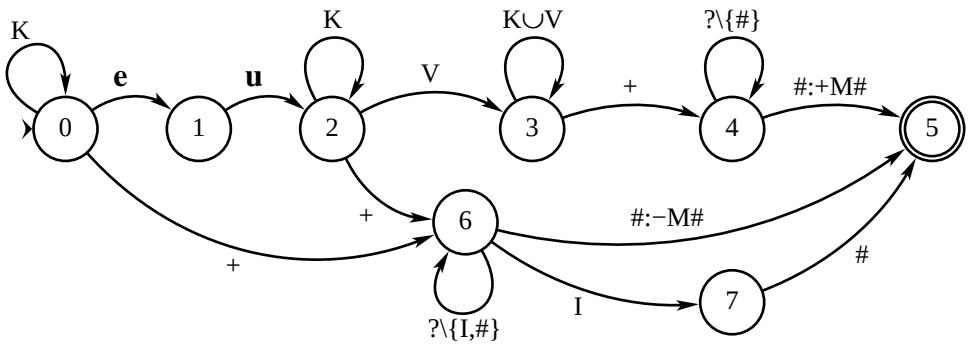


Abbildung 3.8: Regel A_{16} für Erstsilben mit Silbenkern *eu*

A_{16} herzustellen, müssen lediglich die fett gedruckten Symbole mit den entsprechenden Diphthongen ersetzt werden.

Nachdem die Regeln für alle möglichen Silbenkerne aus Abbildung 3.4 implementiert wurden, müssen diese zusammengefügt werden. Jede Regel für sich akzeptiert nur Segmente, die genau den durch diese Regel beschriebenen Silbenkern enthalten. Um einen Transducer zu berechnen, der alle Silbenkerne berücksichtigt, genügt es, die Abschlusseigenschaft der Vereinigung (siehe Unterkapitel 2.1.3) auf die Regeln A_1 bis A_{16} anzuwenden. Damit entsteht der neue Transducer:

$$\bigcup_{i=1}^{16} A_i = A_1 \cup A_2 \cup \ldots \cup A_{16}$$

Dieser neue Transducer ist nun in der Lage, jedes Segment als ein- oder mehrsilbig zu kennzeichnen, wobei Interfixe ignoriert werden. Die Regeln sind aber nur für einzelne Segmente definiert, somit kann dieser Transducer ebenfalls nur einzelne Segmente analysieren und nicht Zusammensetzungen aus diesen. Um dies zu ermöglichen, wird wieder einmal der Kleene-Abschluss angewendet. Man erhält damit folgendes Ergebnis:

$$T_{\text{sil}} = (\bigcup_{i=1}^{16} A_i)^+$$

Der Transducer A kann nun in jeder Sequenz von Segmenten die einzelnen Segmente analysieren und jedem Segment die Eigenschaften ein- oder mehrsilbig zuordnen. Wird dieser Transducer auf ein Segmentierungsergebnis von T_{seg} angewendet, ergibt sich folgendes Bild:

$$|T_{\text{sil}}|(\text{druck}_{+\text{N}+\text{MS}+\text{S1}-\text{P}\#}\text{er}_{+\text{I}\#}\text{wartung}_{+\text{N}+\text{FM}+\text{S3}+\text{P3}\#}) = \{\text{druck}_{+\text{N}+\text{MS}+\text{S1}-\text{P}-\mathbf{M}\#}\text{er}_{+\text{I}\#}\text{wartung}_{+\text{N}+\text{FM}+\text{S3}+\text{P3}+\mathbf{M}\#}\}$$

Das Segment *druck* ist einsilbig und wird mit dem Tagset –M gekennzeichnet. *er* ist ein Interfix und wird in Bezug auf die Silbenanalyse ignoriert. Das Segment *wartung* wird als mehrsilbig erkannt und das Tag +M wird angefügt. Eigentlich müsste das letzte Segment nicht analysiert werden, allerdings ist es mühselig, dies zu verhindern. Die Analyse des letzten Segments bringt zwar keinen Nutzen, schadet aber auch nicht.

3.2.2 FSTs zur Suffix- und Auslautbestimmung

Außer der Silbenzahl spielt auch der Auslaut der Segmente eine Rolle bei der Gestaltung der Kompositionsstammform. So sind es *-ung*, *-ling*, *-bold* und andere in Abbildung 1.1 (S. 28) hervorgehobenen Suffixe, die entweder bestimmte Fugenelemente fordern oder ablehnen. Insgesamt sind es 19 verschiedene Suffixe. Daneben werden auch zwei Arten von Auslauten bestimmt, der nicht-silbische Auslaut *-t* und das auslautende Schwa (*-e*). Die Prinzipien, die den Regel-Transducern zur Suffix- und Auslauterkennung zugrunde liegen, sind ähnlich wie bei der Silbenanalyse im vorigen Unterkapitel. Auch hier wird für jedes Suffix ein Transducer entworfen, der in der Lage ist, ein Segment zu untersuchen und bezüglich des Vorhandenseins des gesuchten Suffixes zu kennzeichnen. Dies geschieht wie zuvor über das Anfügen eines entsprechenden Tags an das Tagset des Segments. Abbildung 3.9 zeigt die zusätzlichen Tags, die nach der Untersuchung der Segmente auf spezifische Suffixe oder Auslaute angehängt werden. Sie stehen an sechster Stelle im Tagset der Substantive. Das Symbol „#" verschiebt sich an siebte Stelle.

Tag	Bedeutung	Tag	Bedeutung
+SS	Suffix, das *-s* fordert	+AT	Auslaut mit *-t*
+SN	Suffix mit Nullfuge	+AE	Auslaut mit *-e*
–SA	ohne Suffix oder Auslaut		

Abbildung 3.9: Zusätzliche Tags nach der Suffix- und Auslautanalyse

Eine Transducer-Regel, die feststellt, ob das Suffix *-ung* am Ende eines Segments vorhanden ist, ist auf Abbildung 3.10 zu sehen. Die Regeln unterscheiden sich von einander nur durch die fett gedruckten Symbole und die Zahl der damit verbundenen Zustände, die von der Länge des beschriebenen Suffixes abhängt. Außerdem sind bei verschiedenen Gruppen von Suffixen, verschiedene Tags im Transducer enthalten, der entsprechende Übergang ist ebenfalls fett gedruckt. In diesem fett gedruckten Teil des Transducers stellt die Regel sicher, dass der Transducer das Suffix *-ung* deterministisch erkennt. So wird das Merkmal +SS nur dann dem Tagset angefügt, wenn die Zeichenfolge *ung* genau am Ende des Segments auftaucht und nicht etwa früher wie z. B. im Wort *ungemach*. Auch hier werden die Interfixe als Segmente ignoriert.

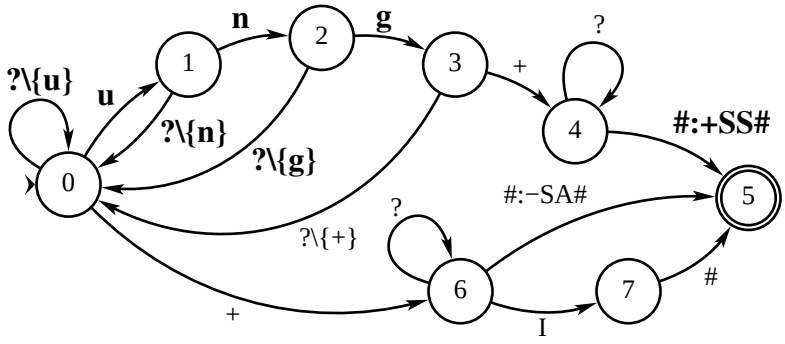


Abbildung 3.10: Regel B_{18} für die Suffixerkennung von *-ung*

Wie gesagt, gibt es 19 verschiedene Suffixe. Die männlichen und neutralen Suffixe, die die Nullfuge fordern, werden in den Regeln $B1$ bis $B9$ erfasst und mit dem Tag +SN markiert. $B10$ bis $B12$ beschreiben die männlichen und neutralen Suffixe, die regelmäßig mit *-s* stehen, markiert werden diese mit dem Tag +SS. Ebenfalls mit +SS werden die weiblichen Suffixe gekennzeichnet, die mit dem unparadigmischen *-s* stehen und den Regeln $B13$ bis $B18$ entsprechen. $B19$ markiert Segmente mit dem weiblichen Suffix *-ei* durch +SN, da auch hier die Fugung im Normalfall nahtlos erfolgt.

Abbildung 3.11 zeigt einen Transducer B_{20}, der feststellt, ob ein Segment auf *-t* auslautet und der, wenn das zutrifft, das Tagset mit +AT erweitert. Die Regel B_{21} für das auslautende *Schwa* ist in seiner Struktur identisch. Als Tag wird +AE angehängt.

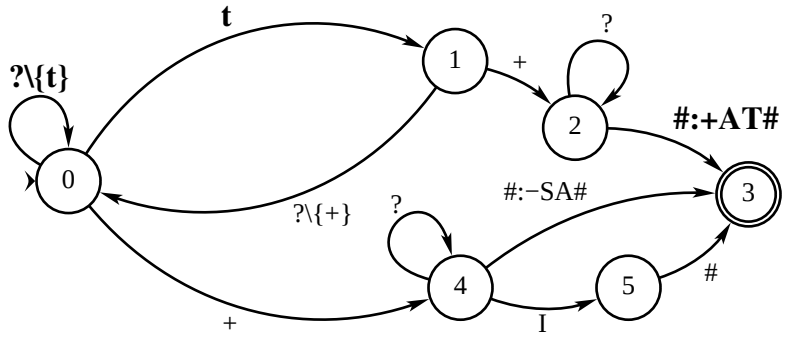


Abbildung 3.11: Regel B_{20} für die Auslauterkennung von *-t*

Allen Regeln ist gemein, dass das Tagset eines Segments, das kein charakteristisches Suffix und keinen gesuchten Auslaut enthält, mit dem Tag –SA erweitert wird.

Die Suffix-Regeln B_1 bis B_{21} werden ähnlich wie die Regeln zur Mehrsilbigkeit zu einem neuen Transducer vereinigt:

$$\bigcup_{i=1}^{21} B_i = B_1 \cup B_2 \cup \ldots \cup B_{21}$$

$$T_{\text{suf}} = (\bigcup_{i=1}^{21} B_i)^+$$

Anschließend wird auch dieser Transducer dem Kleene-Abschluss unterzogen, um Sequenzen von Segmenten verarbeiten zu können. Der neue Transducer T_{suf} kann nun alle in Abbildung 1.1 erfassten Suffixe und Auslaute erkennen und markieren.

3.3 Distributionsregeln

Nachdem die Segmente identifiziert wurden und deren Kategorisierung mithilfe der entsprechenden Tagsets abgeschlossen worden ist, verfügen die Distributionsregeln über alle Informationen, die benötigt werden, um die Disambiguierung der Segmentierungsarten vorzunehmen.

Formal sind für die Modellierung der Distributionsregeln gewöhnliche Automaten ausreichend und Transducer müssen nicht genutzt werden. Dies führt zu einer etwas größeren Flexibilität bei der Anwendung der Abschlusseigenschaften. Um letztendlich doch eine Ausgabe zu erhalten, wird angenommen, dass die Distributionsregeln bei einem positiven Ergebnis die Eingabe identisch auf die Ausgabe abbilden. Entspricht eine Eingabe nicht der modellierten Regel und der Regelautomat verwirft, wird das leere Wort zurückgegeben. Erweitert man die Regelautomaten dann auf Mengen von Eingaben, ist das Ergebnis einer Regel eine Untermenge der Eingabemenge, aus der die falschen Elemente entfernt wurden. Sind keine falschen Elemente vorhanden, entspricht die Eingabemenge der Ausgabemenge. Sind dagegen alle Elemente unakzeptabel, wird die leere Menge zurückgegeben. Insofern kann man auch weiterhin von Transducern sprechen und dabei alle Eigenschaften der einfachen endlichen Automaten nutzen.

3.3.1 Naive Kompositastruktur

In Kapitel 1 wurden an verschiedenen Stellen die Struktureigenschaften der deutschen Komposita beschrieben. Einerseits bestehen syntagmatische Relationen zwischen den Kompositionsgliedern und andererseits paradigmatische innerhalb dieser Glieder. Man kann also von zwei Ebenen des morphologischen Kontextes sprechen, einem lokalen und einem globalen. Lokal und paradigmatisch sind die Bildungsregeln der Kompositionsstammformen der Erstglieder. Global und syntagmatisch sind die Verhältnisse der Kompositionsstammformen der Erstglieds zu den Zweitgliedern.

Beiden Kontexten muss Rechnung getragen werden und beide haben einen großen Einfluss auf die Disambiguierung der Ergebnismenge einer naiven Kompositasegmentierung. Die Abschlusseigenschaften der endlichen Automaten erlauben es, beide Kontexte zu modellieren, wobei die Modellierung der globalen Distributionsregeln komplett über diese Eigenschaften abgewickelt wird. Die Modellierung der lokalen Abhängigkeiten ist deutlich mühseliger. An dieser Stelle wird der Entwurf einer allgemeinen Kompositastruktur vorgestellt, der später zur Grundlage der genaueren Regeln verwendet werden wird.

Diese naive Regel, die im Moment auf nominale Glieder beschränkt bleibt, modelliert bereits vollständig die syntagmatischen Beziehungen der Glieder unter einander und könnte bereits zu einer einfachen Disambiguierung verwendet werden. Die lokalen Kontexte sind jedoch nicht ausformuliert und sehr allgemein gehalten, lassen aber deren zukünftige Struktur erkennen.

Es werden zuerst diese allgemeinen lokalen Kontexte beschrieben, die sich auf die Bildung der Kompositionsstammformen der nominalen Erstglieder beschränken. Die Kompositionsstammformen der Substantive bestehen entweder aus deren Grundform oder der Grundform und einem Fugenelement. Abbildung 3.12 zeigt einen endlichen Automaten KS_1, der alle Segmente akzeptiert, deren Wortart durch das Tag +N markiert ist. Dabei wird die graphematische Form des Segments ignoriert, ebenso wie die übrigen hier nicht benötigten Tags.

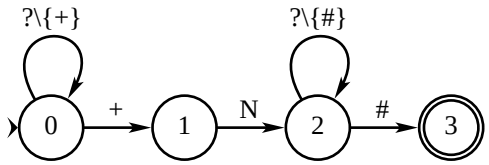


Abbildung 3.12: Nominale Kompositionsstammform ohne Fuge (KS_1)

Die Identifikation läuft beispielsweise für das Segment „drucker$_{+N+MS+S1+P2+M-SA\#}$" folgendermaßen ab: Jedes Zeichen von „drucker" verursacht keinen Zustandswechsel, da die Schleife für jedes Zeichen außer „+" in den Startzustand übergeht. Mit dem ersten „+" wird der Beginn des Tagsets gekennzeichnet und es findet ein Zustandswechsel statt, ebenso wie beim Zeichen „N". Damit wird das erste Tag bei nominalen Segmenten „+N", das die Wortart beschreibt, erfasst. Die übrigen Tags verursachen wieder keinen Zustandswechsel, solange kein „#" eingelesen wird, das das Ende des Segments kennzeichnet. Damit werden durch diesen Automat alle Segmente mit der Wortart Substantiv akzeptiert. Da im Lexikon bei den Substantiven nur die Grundformen enthalten sind, akzeptiert dieser Automat alle nominalen Grundformen, bzw. alle Kompositionsstammformen, die der Grundform entsprechen.

Zu der Struktur dieses Automaten und auch der Automaten, die noch folgen werden, sei gesagt, dass die distributionell bedeutsamen[7] Eigenschaften auf den Übergängen

[7] Damit sind die in der Abbildung 1.1 (S. 28) mit (*) gekennzeichneten Kriterien gemeint.

verzeichnet sind, die Zustandswechsel verursachen. Die Eigenschaften, die ignoriert werden können, werden durch die Schleifen abgedeckt.

In Bezug auf den lokalen Kontext ist das Fehlen der Fuge in diesem Fall die Relation, die durch diesen Automat KS_1 modelliert wird. Obwohl es nicht explizit wird, wird damit ein Verhältnis zwischen den Segmenten modelliert. Die Transducer aus den vorigen Unterkapiteln beschränkten sich bisher auf die Analyse einzelner Segmente, wobei deren Reihenfolge unerheblich war.

Die allgemeine Struktur der Kompositionsstammformen mit Fugenelementen zeigt Abbildung 3.13. Der Automat KS_2 beschreibt alle Kompositionsstammformen die mithilfe von Fugenelementen gebildet werden. Restriktionen bezüglich der Verbindung von bestimmten Substantiven mit bestimmten Fugenelementen werden hier noch nicht berücksichtigt. Von diesem Automat muss ebenfalls nur die Grundform des Substantivs akzeptiert werden, so wie im Beispiel zuvor. Hinzu kommt, dass auf diese Grundform ein Fugenelement folgen muss. Die graphematische Form des Fugenelements wird wieder ignoriert, einzig die Tags „+I" und „#" sind bedeutsam.

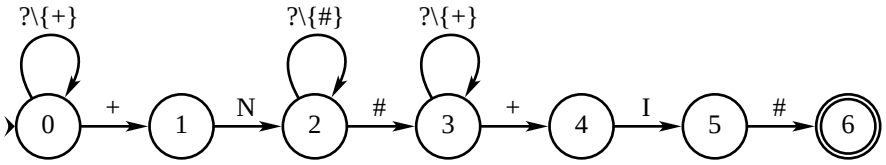


Abbildung 3.13: Nominnale Kompositionsstammform mit Fuge (KS_2)

KS_1 akzeptiert nur die Kompositionsstammformen ohne Fugenelement, KS_2 nur die Kompositionsstammformen mit Fugenelement. Um alle möglichen Kompositionsstammformen akzeptieren zu können, müssen KS_1 und KS_2 mit einander in Beziehung gesetzt werden. Dies wird durch die Vereinigung der beiden Automaten erreicht:

$$KS = KS_1 \cup KS_2$$

Der neue Automat KS akzeptiert nun alle möglichen Kompositionsstammformen, aber auch viele ungrammatische, da er sehr allgemein gehalten ist. Ein Automat, der aus der Vereinigung zweier Automaten entsteht, akzeptiert alle Zeichenketten, die von dem ersten oder dem zweiten akzeptiert werden. Damit modelliert der Automat KS alle Kompostionsstammformen, die nur aus der nominalen Grundform oder der nominalen Grundform mit Fugenelement bestehen. Im naiven Modell sind so alle lokalen Beziehungen erfasst.

Der globale Kontext zwischen den beiden Kompositionsgliedern wird hier beschrieben. Anschließend wird dieses naive Modell auf mehrfach zusammengesetzte Komposita erweitert. Ein Kompositum besteht aus Erstglied und Zweitglied, das im Weiteren als Grundwort bezeichnet wird. Generell kann auf jedes Erstglied jedes beliebiges

Grundwort folgen. Im Fall der nominalen Komposita bedeutet dies, das jedes beliebige Substantiv als Grundwort auftauchen kann. Damit ist ein Automat der identisch ist mit der Struktur von KS_1 ausreichend, um das Grundwort zu akzeptieren. Dieser Automat wird, um Verwirrungen zu vermeiden, NG genannt.

Ein zweigliedriges Kompositum ist lediglich die Verbindung von Erstglied mit dem Grundwort. Formal gesagt, handelt es sich um die Konkatenation aus Erstglied und Grundwort. Das Erstglied wiederum geht in seiner Kompositionsstammform in die Komposition ein. Damit beschreibt die Konkatenation aus Kompositionsstammform und Grundwort die Struktur aller zweigliedriger Komposita. Analog dazu, genügt es die Abschlusseigenschaft der Konkatenation auf die Automaten KS und NG anzuwenden, um einen Automat

$$KS \cdot NG$$

zu erhalten, der genau diese Kompositastruktur beschreibt. Auf eine beliebige nominale Kompositionsstammform muss eine beliebige nominale Grundform folgen, damit dieser Automat die Eingabe akzeptiert. Damit können bereits erste Disambiguierungen durchgeführt werden. Obwohl durch die Beliebigkeit der Glieder keine starken Restriktionen formuliert wurden, können z. B. Segmentierungsarten ausgeschlossen werden, bei denen fälschlicherweise Fugenelemente vor dem Erstglied oder nach dem Grundwort erkannt wurden. An diesen Stellen können keine Fugenelemente auftreten und dies wird durch diese Regel für zweigliedrige Komposita beschrieben. Auch würden z. B. Segmentierungsarten verworfen, die mehr als ein Fugenelement an einer Fuge erkannt haben.

Das Modell soll jedoch nicht auf die zweigliedrigen Komposita beschränkt bleiben. Es muss der potentiellen Unendlichkeit der Komposita Rechnung tragen. Betrachtet man die Oberfläche der mehrfach zusammengesetzten Komposita, sieht man im Grunde eine Sequenz von Grundformen mit oder ohne Fugenelement, also eine Sequenz von Kompositionsstammformen, die immer durch eine Grundform abgeschlossen wird, auf die kein Fugenelement folgen darf. Es muss mindestens eine Kompositionsstammform vorhanden sein auf die eine einzelne Grundform folgt. Eine obere Grenze für die Zahl der Kompositionsstammformen ist nicht gegeben.

Die potentiell unendliche Sequenz von Kompositionsstammformen, die mindestens aus einer Kompositionsstammform bestehen muss, kann ganz einfach mit dem Kleene-Abschluss realisiert werden. Der Automat $(KS)^+$ akzeptiert damit jede Folge von möglichen Kompositionsstammformen. Diese Folge muss nur noch durch das Grundwort abgeschlossen werden, was ähnlich wie zuvor, durch Konkatenation gelöst wird. Damit wird der Automat

$$T_{\text{naiv}} = (KS)^+ \cdot NG = (KS_1 \cup KS_2)^+ \cdot NG$$

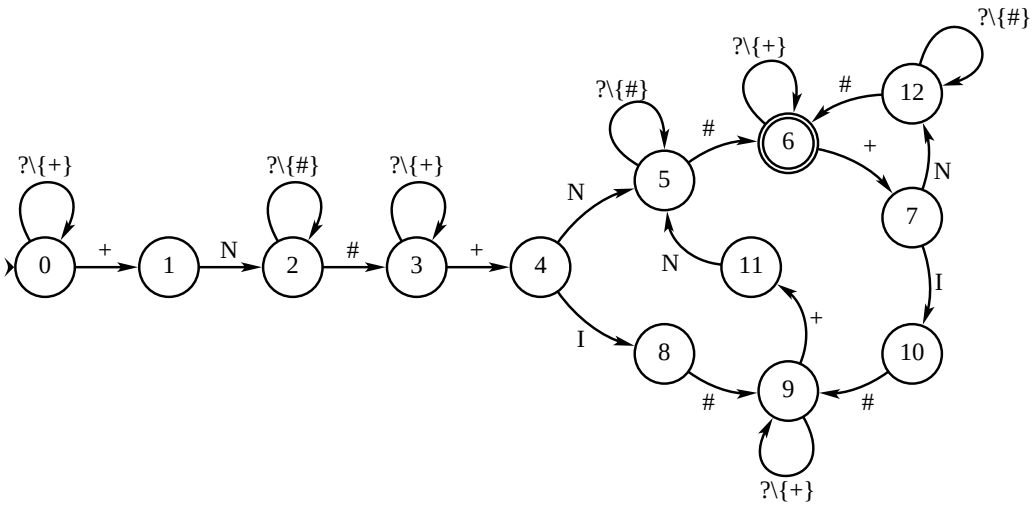


Abbildung 3.14: Allgemeine Kompositastruktur $(KS)^+ \cdot NG$

erstellt, der in seiner minimierten Form auf Abbildung 3.14 dargestellt ist und jetzt die naive Struktur aller Komposita mit mindestens zwei nominalen Gliedern beschreibt. Das Modell ist für das menschliche Auge nicht wirklich lesbar, obwohl es sich hier bislang nur um eine allgemeine Version ohne die zahlreichen Ausnahmen handelt. Aus diesem Grund wird im weiteren Verlauf der Arbeit auf die Darstellung der globalen Distributionsregeln in Diagrammform verzichtet. Statt dessen werden die Regeln im lokalen Kontext modelliert. Auf diese einfachen Komponenten werden dann die Abschlusseigenschaften angewendet, die alles zu größeren Automaten zusammenfügen. Diese Form ist deutlich leichter zu interpretieren. Die Abschlusseigenschaften sind mathematische Operationen, die für die automatische Berechnung konzipiert sind, daher sind deren Ergebnisse auch nicht für eine menschliche Interpretation gedacht. Durch die Information welche Abschlusseigenschaft wie angewendet wird, wird das Modell ausreichend beschrieben, zumal die Abschlusseigenschaften so gewählt werden, um den Anforderungen des Modells zu genügen.

Es werden nur noch Automaten gezeigt, die einzelne Kompositionsstammformen oder Grundwörter beschreiben. Mithilfe der Abschlusseigenschaften werden sie dann zu einander in Beziehung gesetzt.

Dieses vereinfachte Modell akzeptiert auch viele falsche Lesarten. Bei dem Beispiel *Druckerwartung* würde dieses Modell noch keine der möglichen fünf Segmentierungsarten verwerfen, man kann sagen, dass es zu tolerant ist. Um diese Toleranz einzuengen, werden nach und nach zusätzliche Kompositionsstammformen hinzugefügt. Wie das genau realisiert wird, wird im nächsten Unterkapitel besprochen.

3.3.2 Tag-basierte Regeln

Um sich schrittweise einem genauerem Modell nähern zu können, müssen die einzelnen Kompositionsstammformen modelliert werden. Als Grundlage dazu dient die Zusammenstellung auf Abbildung 1.1. Nach den bedeutsamen, mit (*) gekennzeichneten Kriterien werden die Tags ausgewählt, die es erlauben, einem Segment ein oder mehrere Fugenelemente zu zuordnen. Jeder Regelautomat wird so entworfen, dass er nur eine Art von Kompositionsstammform spezifisch beschreibt, und alle anderen Fälle in der allgemeinen Form berücksichtigt. Dabei sind diese spezifische Kompositionsstammform und die allgemeinen Formen disjunkt.

Wird z. B. eine Regel modelliert, die den schwachen nicht auf *Schwa* auslautenden Substantiven das Fugenelement *-en* zuordnet, dann akzeptiert sie nur diese Kompositionsstammformen der schwachen Substantive und keine andere. Sollte ein schwaches nicht auf *Schwa* auslautendes Substantiv eingelesen werden, dass eine andere Fugengestaltung aufweist, verwirft die Regel. Wendet man diese Regel auf ein nichtschwaches Substantiv an, akzeptiert die Regel jede mögliche Kompositionsstammform, da sie in diesem Fall „tolerant" ist. Nur im Fall dieser Gruppe der schwachen Substantive verfügt sie über genaue Informationen. Modelliert wird so der allgemeine Fall aus dem vorigen Kapitel, für den die Regel genau eine Ausnahme spezifiziert. Hat man mehrere solche Regeln, von denen jede eine andere Ausnahme vom allgemeinen Modell beschreibt, und alle anderen Fälle ohne diese Ausnahme akzeptiert, können sie per Durchschnitt zusammengefügt werden. Damit entsteht ein Modell, das immer mehr Ausnahmen enthält und so immer genauer wird bzw. intoleranter falschen Formen gegenüber.

Auf Abbildung 1.1 sind für die nominalen Erstglieder 13 verschiedene Arten von Distributionsregularitäten erfasst. Die Bezeichnungen der Regelautomaten werden dieser Nummerierung entsprechend gewählt. Auf Abbildung 3.15 ist eine modifizierte Version dieser Zusammenstellung zu sehen, die nur die bedeutsamen Kriterien enthält und diese in Form von Tags darstellt. Die Regel, die die Kompositionsstammformen aller schwachen Substantive beschreibt, wird z. B. mit NK_2 bezeichnet. Die Tags, die nötig sind, um die schwachen Substantive zu identifizieren, sind +N, das die Wortart festlegt sowie +S2 und +P3, die die schwache Deklination kennzeichnen. Unterschieden wird dabei noch zwischen schwachen Substantiven ohne spezifischen Auslaut (–SA) und solchen, die auf *Schwa* auslauten (+AE). Soll eine Regel in Form eines Automaten implementiert werden, muss sie alle aufgezählten Tags berücksichtigen. Damit ergibt sich, dass jede Regel für sich auch als Mengendefinition angesehen kann, da sie nur eine bestimmte Klasse von Substantiven beschreibt, auf die die genannten Merkmale zutreffen.

Abbildung 3.16 zeigt die Regel NK_2, die nach den beschriebenen Gesichtspunkten konzipiert wurde. Der Automat ist hier etwas vereinfacht dargestellt, ohne dass dessen Funktionalität dadurch beeinträchtigt wird. Die mit Symbolen von mehr als einem Zeichen Länge versehenen Übergänge bestehen tatsächlich aus einer Folge von

NK	Wortart	Genus	Sg.	Pl.	Struktur	Suffix/Auslaut	Fuge
1	+N			+P1			-, -e
2	+N		+S2	+P3		–SA	-en
	+N		+S2	+P3		+AE	-n
3	+N		+S3	+P3		+AE	-n
4	+N			+P4			-, -er, -s
5	+N			+P5			-
6	+N		+S1		–M		-, -es
7	+N	+FM			–M		-
8	+N	+FM			+M	+AT	-s, -en
	+N	+FM		–P	+M	+AT	-s
9	+N	+NT/+MS			+M	+SN	-
10	+N	+NT/+MS			+M	+SS	-s
11	+N	+FM			+M	+SS	-s, -en
	+N	+FM		–P	+M	+SS	-s
12	+N	+FM			+M	+SN	-, -en
	+N	+FM		–P	+M	+SN	-
13	+N			–P		–SA	-
VK	+V						-, -e
AK	+A						-

Abbildung 3.15: Kompositionsstammformen nach bedeutsamen Tags

Übergängen mit jeweils nur einem Zeichen, so wie das z. B. in Abbildung 3.14 der Fall ist.

Bei der Konstruktion der Regeln muss immer die Form des Segments sowie die Reihenfolge der Tags im Tagset berücksichtigt werden. Zur Identifikation einer bestimmten Gruppe von Substantiven genügt es, zu überprüfen, ob alle Merkmale, die diese

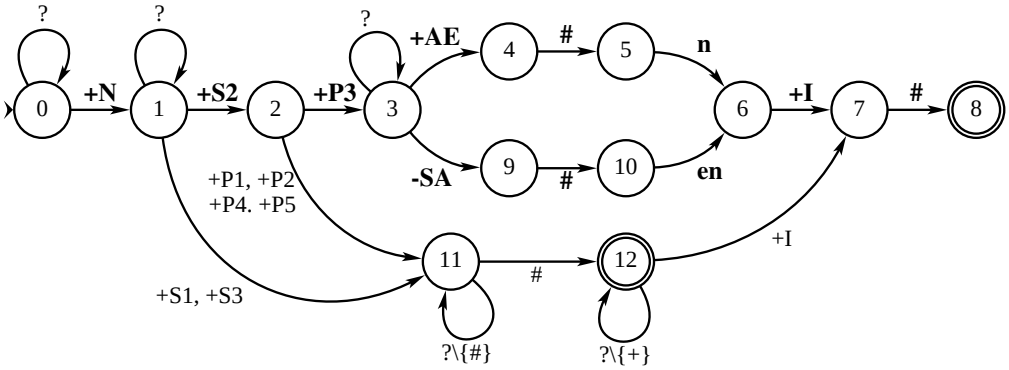


Abbildung 3.16: Kompositionsstammformen schwacher Substantive – Regel NK_2

Gruppe ausmachen, im Tagset enthalten sind. Gegebenenfalls muss auch geprüft werden, ob nicht Tagsets enthalten sind, die zu einer Differenzierung führen. Dies ist bei der Regel NK_2 der Fall.

Die auf Abbildung 3.16 fett gedruckten Übergänge beschreiben die Struktur der schwachen Substantive. Liest man diese Regeln gemäß der Pfeilrichtungen, ergibt sich die folgende Funktionsweise:

Die graphematische Form des Segments wird ignoriert, was bei allen tag-basierten Regeln der Fall ist. Ein Zustandswechsel zu „1" findet erst statt, wenn die Wortart des Segments durch +N erkannt wird. Nach der Wortart steht im Tagset der Substantive das grammatische Geschlecht. Dieses Tag wird hier durch eine Schleife ignoriert. Erst die Zugehörigkeit zu einer bestimmten Flexionsklasse im Singular verursacht einen Zustandswechsel. Hier gibt es zwei Möglichkeiten. Ist das eingelesene Substantiv ein schwaches Substantiv, muss es mit +S2 markiert sein und die Regel wechselt zu „2". In allen anderen Fällen handelt es sich offensichtlich um kein schwaches Substantiv und der allgemeine Teil der Regel übernimmt ab Zustand „11". Nach der Flexionsklasse im Singular folgt im Tagset die Flexionsklasse im Plural. Es müssen keine Zeichen ignoriert werden, deshalb ist an dieser Stelle keine Schleife im Transitionsdiagramm vorhanden. Vom Zustand „2" gibt es wieder eine Entscheidungsmöglichkeit. Um ein schwaches Substantiv eindeutig festzustellen, reicht die Eigenschaft +S2 nicht aus. Es muss auch mit +P3 markiert sein. Ist dies der Fall, wechselt die Regel zum Zustand „3", sonst wieder zum Zustand „11".

In diesem Augenblick ist das schwache Substantiv von der Regel eindeutig bestimmt worden. Es muss jedoch differenziert werden zwischen schwachen Substantiven ohne charakteristischen Auslaut und solchen mit *Schwa*. Beide Fälle sind mit entsprechenden Tags markiert, die durch die Erstgliedanalyse hinzugefügt wurden. Die beiden Fälle könnten auch mithilfe von zwei eigenständigen Regeln beschrieben werden, aber in diesem Fall bietet sich eine Modellierung als Einheit an, da sie sich nur durch ein Unterkriterium unterscheiden und eine logische Einheit bilden. Außerdem herrscht an der Fuge der schwachen Substantive Allomorphie vor, was ebenfalls ein Argument für eine einzelne Regel ist.

Im Zustand „3" findet nach der Identifizierung eines schwachen Substantivs die Differenzierung bezüglich des Auslautes statt. Von dieser Stelle an, ist die Regel eindeutig und wird nicht mehr in den allgemeinen Fall wechseln. Jedes schwache Substantiv hat entweder *Schwa* im Auslaut und folgt dann dem mit +AE gekennzeichneten Pfad, oder hat keinen spezifischen Auslaut und folgt dem Pfad mit –SA. Ist ein Pfad eingeschlagen, ergibt sich daraus die obligatorische Fugengestaltung *-e* bzw. *-en*. Andere Möglichkeiten gibt es für die schwachen Substantive nicht. Das Fugenelement wird erkannt, indem es als Segment in seiner graphematischen Form eingelesen wird und anschließend als Fugenelement durch das Tag +I identifiziert wird. Betrachtet man den allgemeinen Teil dieser Regel, der alle nicht-schwachen Substantive abdeckt, sieht man keine Übergänge, die mit konkreten Fugenelementen versehen sind.

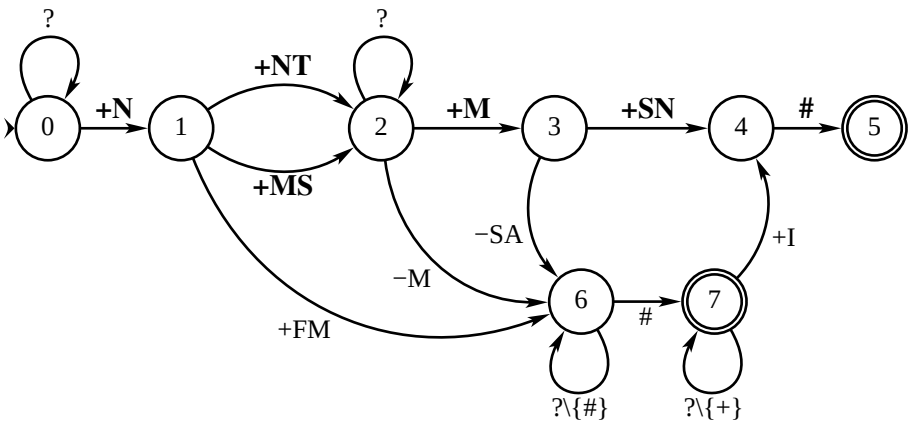


Abbildung 3.17: Kompositionsstammformen mit spezifischem Suffix – Regel NK_9

Zum Vergleich folgt in Abbildung 3.17 die Regel NK_9, die für die beschriebene Kompostionsstammform eine Nullfuge verlangt. So haben alle neutralen oder maskulinen Substantive mit z. B. dem Suffix -*bold* kein Fugenelement in ihrer Kompositionsstammform. Die nötigen Tags, um diese Substantive möglichst genau zu identifizieren, sind +N, +NT oder +MS, +M und +SN. Erfasst werden also alle Substantive, die maskulinen oder neutralen Geschlechts sind, die mehr als eine Silbe haben und am Ende ein Suffix aufweisen, das die Nullfuge fordert.

Folgt man diesen Tags durch das Transitionsdiagramm der Regel NK_9, erreicht man das Ende des Segments, dem offensichtlich keine Fugenbeschreibung folgt. Das heißt, dass jedes Substantiv, das diesen Kriterien entspricht, aber wider Erwarten doch ein Fugenelement aufweist, nicht von dieser Regel akzeptiert werden kann und als falsch verworfen wird. Auch diese Regel ist wieder so gestaltet, dass Substantive, die den geforderten Kriterien nicht entsprechen, bezüglich einer allgemeinen Kompositionsstammform akzeptiert werden. Ist z. B. ein Substantiv femininen Geschlechts oder besteht nur aus einer Silbe oder weist kein entsprechendes Suffix auf, dann kann ihm jedes beliebe Fugenelement einschließlich der Nullfuge folgen. Erst in einer anderen Regel wird hier weiter eingeschränkt.

Da die Fugengestaltung nur bei den nominalen Kompositionsstammformen kompliziert ist, wurden diese hier besonders ausführlich besprochen. Natürlich lassen sich anhand der vorhandenen Tags auch einfache Regeln für die Bestimmung der verbalen oder adjektivischen Kompositionsstammformen erstellen, die der Vollständigkeit wegen beschrieben werden. Deren Struktur ist vergleichsweise einfach. Nach den Adjektiven stehen keine Fugenelemente, die Anbindung erfolgt immer nahtlos. Nach den Verben, die hier in Form von Verbalstämmen im Lexikon enthalten sind, steht in der meisten Fällen kein Fugenelement, und nach einigen wenigen schwierig erfassbaren Fällen -*e*. Andere Fugenelement kommen bei den Verben nicht vor. So wird für die

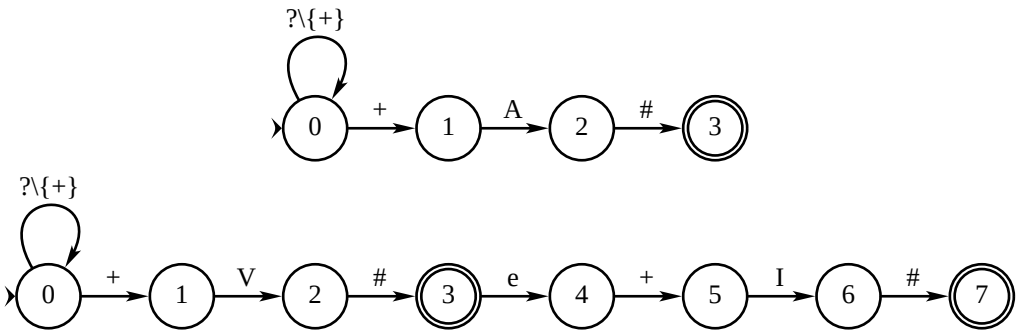


Abbildung 3.18: Adj. und verb. Kompositionsstammformen – Regeln AK und VK

Adjektive und Verben nur jeweils eine Regel entworfen, die nur von der Wortart und keinen weiteren Kriterien abhängt.

Abbildung 3.18 zeigt diese Kompositionsstammformen. Das obere Transitionsdiagramm beschreibt die Regel AK, für die adjektivischen Erstglieder. Wie man sieht, wird hier kein Fugenelement berücksichtigt und jede Segmentierung, die ein falsches Fugenelement an dieser Stelle aufweist, wird verworfen. Ähnlich wird bei den Verben durch VK entweder nur die Form ohne Fugenelement oder nur mit *-e* als Fugenelement akzeptiert. Andere Regeln werden für die Adjektive und Verben nicht definiert.

3.3.3 Lexikalisierte Regeln

Die im vorigen Unterkapitel beschriebenen Transducer fußen auf allgemeinen Regeln, die aufgrund von spezifischen Merkmalen bestimmte Substantive zu Gruppen bzw. zu Mengen zusammenfassen. Nun gibt es Fälle, die sich nicht aufgrund ihrer morphologischen Merkmale in bestimmte Gruppen einteilen lassen. In Unterkapitel 1.6.2 wurden diese Elemente unter dem Punkt d) von den übrigen Kriterien a)–c) abgehoben. Um diese Ausnahmen zu erfassen, wird an dieser Stelle die Verwendung von lexikalisierten Regeln vorgeschlagen. Während die bisherigen Distributionsregeln die graphematische Form der Segmente ignorierten und nur deren linguistischen Merkmale als Kriterien zur Mengenbildung verwendeten, werden hier konkrete Einzelfälle modelliert, die umgekehrt die morphologischen Merkmale ignorieren und ganz von der graphematischen Form des Segments abhängig gemacht werden. Es wird höchstens überprüft, ob es sich bei dem Segment um ein Substantiv handelt. So werden keine Mengen erfasst, sondern Ausnahmen, die eigentlich aufgrund ihrer Merkmale einer Menge zugeordnet werden müssten, sich dieser Zuordnung aber aus verschiedenen uneinheitlichen Gründen entziehen.

Als Beispiel wurde bereits mehrfach das Substantiv *Liebe* genannt, das seine Kompositionsstammform unparadigmisch mit *-s* bildet und von keiner Regel erfasst wird. Dafür ist diese Kompositionsstammform die einzige mögliche und kann damit als

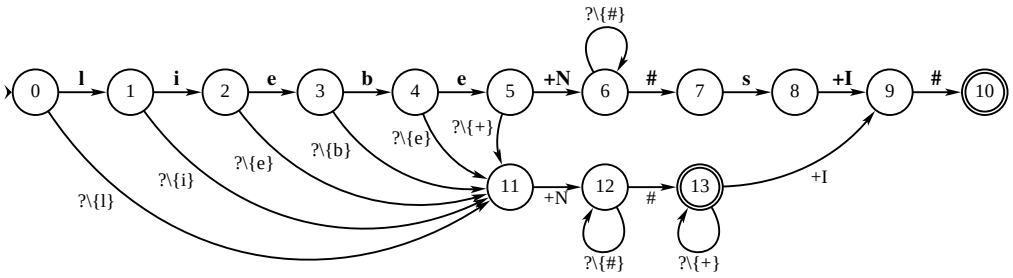


Abbildung 3.19: Lexikalisierte Kompositionsstammform

lexikalisiert gewertet werden. Letztendlich unterscheidet sich diese Regel im Hinblick auf die Struktur des implementierten Automaten nicht so sehr von den tag-basierten Regeln, da immer Zeichenketten verarbeitet werden. Auch hier muss für alle Substantive, die nicht genau *Liebe* lauten, die allgemeine Kompostionsstammformstruktur berücksichtigt werden, damit die Regel alle anderen Substantive akzeptieren kann.

Abbildung 3.19 zeigt einen Automaten, der für das Erstglied *Liebe* nur die Kompositionsstammform *liebes* akzeptiert. Ähnlich können auch andere lexikalisierte Kompositionsstammformen beschrieben werden. Problematisch ist deren Einbindung durch die Anzahl der Ausnahmen. Jede unparadigmische und dazu nicht-regelhafte Kompositionsstammform müsste einzeln eingebunden werden. Da es sich gerade um Einzelfälle handelt, lässt sich dies wohl kaum einfacher gestalten. Eine andere Lösung wäre eine entsprechende Markierung auf der Ebene des Lexikons durch spezielle Tags. Aber auch dies führt zu einer Auseinandersetzung mit jedem einzelnem Fall.

Aus diesem Grund wird in dieser Arbeit auf die Einbindung der lexikalisierten Regeln in das Modell verzichtet. Dabei ist diese Einbindung prinzipiell möglich, denn die lexikalisierten Regeln können vom Standpunkt der Implementation aus genauso behandelt werden, wie die tag-basierten Regeln. Sie können ebenso über die Operation des Durchschnitts mit dem allgemeinen Modell verbunden werden. Schließlich handelt es sich auch bei den tag-basierten Regeln um Ausnahmen vom allgemeinen Modell. Der Unterschied beruht darauf, dass die tag-basierten Regeln, Mengen von Ausnahmen modellieren, während die lexikalisierten Regeln einzelne Ausnahmen beschreiben, oder wenn man so will, Mengen, die nur ein Element enthalten.

Die lexikalisierten Regeln können aber nach der Implementation des gesamten Systems zu Verfeinerungszwecken genutzt werden. Eine Implementierung der tag-basierten Regeln kann auf einen Korpus von Komposita angewendet werden. Fälle diese grammatisch richtig oder zumindest üblich sind, aber von den implementierten Regeln verworfen werden, können diese Einzelfälle mithilfe der lexikalisierten Regeln eingebunden werden, sofern sich für diese verworfenen Regeln keine Mengen aufgrund ihrer Merkmale erstellen lassen. Somit kann durch die Einbindung einer immer größerer Zahl von Ausnahmen ein zunehmend genaueres Distributionsmodell erstellt werden, dessen Toleranz immer weiter eingeschränkt wird.

3.3.4 Die Distributionsregeln als Ganzes

Die einzelnen Regeln zur Bildung der Kompostionsstammformen sind für sich genommen wenig aussagekräftig. Es wurde bereits mehrfach darauf hingewiesen, dass die nominalen Regeln durch die Anwendung der Abschlusseigenschaft des Durchschnitts zu einem einzigen Automaten zusammengestellt werden können, der dann alle Regeln beinhaltet. Wird der Durchschnitt zweier oder mehrerer Automaten gebildet, entsteht ein Automat, der nur die Fälle akzeptiert, die von allen Automaten akzeptiert werden. Alle Regeln sind für sich genommen viel zu tolerant, weil sie nur eine bestimmte Art von Kompositionsstammform bestimmen und für die übrigen Kompositionsstammformen alle Möglichkeiten akzeptieren, auch die vielen ungrammatischen. Als ungrammatisch werden nur die Kompositionsstammformen erkannt, die der Menge, die durch diese Regeln definiert werden, andere nicht erfasste Fugenelemente zuordnen. Bildet man nun den Durchschnitt dieser Regeln, werden die Kompositionsstammformen der Substantive, die keiner der gebildeten Mengen entsprechen, durch die allgemeine Regel bestimmt. Wie zuvor beschrieben wurde, lässt sich deren Zahl durch die Anwendung lexikalisierter Regeln verringern. Auch die lexikalisierten Regeln können durch die Durchschnittsbildung hinzugefügt werden.

An dieser Stelle wird ein Automat NK definiert, der gegenüber dem in Unterkapitel 3.3.1 definierten KS, deutlich intoleranter ist, und alle beschriebenen tagbasierten Regeln für nominale Kompositionsstammformen beinhaltet. NK ist der Durchschnitt der Regeln NK_1 bis NK_{13}.

$$NK = \bigcap_{i=1}^{13} NK_i = NK_1 \cap NK_2 \cap \ldots \cap NK_{13}$$

Damit sind die nominalen Kompositionsstammformen in NK erfasst. Die adjektivischen und verbalen Kompositionsstammformen werden in AK beziehungsweise VK modelliert.

Bisher wurden im Rahmen der spezifischen Regeln zur Bildung der Kompositionsstammformen der lokale Kontext innerhalb dieser Formen besprochen, damit wurde die Paradigmatik der Erstglieder modelliert. Die Regeln sind aber erst funktionsfähig, wenn sie auch in den globalen Kontext eingebettet werden, der die Syntagmatik zwischen den Gliedern beschreibt.

Im Zusammenhang mit dem allgemeinen Modell wurde mit Hilfe der Abschlusseigenschaften bereits ein einfacher globaler Kontext für nominale Kompositionsglieder erstellt. Nominale Komposita sind demnach potentiell unendliche Sequenzen aus nominalen Kompositionsstammformen, die mit einem Substantiv in der Grundform abgeschlossen werden. Diese Zusammenhänge kann man auch auf andere Wortarten erweitern. Dann ist ein Kompositum eine potenziell unendliche Sequenz aus Kompositionsstammformen verschiedener Wortarten, die mit einem Wort abgeschlossen

wird, das in der Grundform der entsprechenden Wortart steht. Hier wird angenommen, dass die Art der Kompositionsstammform und die Wortart beliebig sind, sofern jedes Glied für sich genommen grammatisch richtig gebildet wird.

Die Beliebigkeit der Wortart kann durch die Abschlusseigenschaft der Vereinigung modelliert werden. Es reicht einen Automaten zu bilden, der alle möglichen Kompositionsstammformen der in der vorliegenden Arbeit berücksichtigten Wortarten integriert.

$$NK \cup VK \cup AK$$

Das Ergebnis der obigen Vereinigung ist ein solcher Automat. Werden zwei oder mehrere Automaten vereinigt, akzeptiert der vereinigte Automat alle Eingaben, die von wenigstens einem der zu vereinigenden Automaten erfasst werden. Beim Durchschnitt ist dies anders, da werden nur die Eingaben akzeptiert, die von allen Automaten gleichzeitig angenommen werden. Verwirft einer, verwirft der Ergebnisautomat. So wird durch die Vereinigung ein alternatives Verhältnis erfasst, der Automat akzeptiert damit nominale und verbale und adjektivische Kompositionsstammformen. Die Schwierigkeiten beim Verständnis ergeben sich durch verschiedenen Auffassungen von „oder" sowie „und" in der Mathematik und der Alltagssprache. Der Durchschnitt dieser Automaten würde überhaupt keine Kompositionsstammformen beschreiben, da es keine Eingaben gibt, die gleichzeitig nominal, adjektivisch und verbal sind.

Mit dem Kleene-Abschluss kann nun ein Automat berechnet werden, der eine Sequenz von mindestens einer Kompositionsstammform einer beliebigen Wortart akzeptiert. Die Länge der Sequenz dieser Kompositionsstammformen ist unbeschränkt.

$$\Big(NK \cup VK \cup AK\Big)^{+}$$

Ähnlich wie Kompositionsstammformen kann das Grundwort des Kompositums aus einem Substantiv, einem Verb oder einem Adjektiv bestehen, wobei hier nur die Wortart entscheidend ist und andere Merkmale nicht beachtet werden. Ein Automat, der diese Möglichkeiten für ein einzelnes Grundwort beschreibt, hätte die folgende Form, wobei auch hier durch die Abschlusseigenschaft der Vereinigung das alternative Verhältnis der Bestandteile gekennzeichnet wird:

$$\Big(NG \cup VG \cup AG\Big)$$

Es reicht nun, dieses einzelne Grundwort an den Automaten anzuhängen, der die Kompositionsstammformsequenzen beschreibt; dies geschieht über die Abschlusseigenschaft der Konkatenation. Das Ergebnis ist der Automat T_{dis}, der alle hier beschriebenen Distributionsregeln zur Bildung von Kompositionsstammformen enthält

und diese durch die Einbettung in einen globalen Kontext auf ganze Komposita anwenden kann.

$$T_{\text{dis}} = \Big(NK \cup VK \cup AK\Big)^{+} \cdot \Big(NG \cup VG \cup AG\Big)$$

Sollen die Regeln durch das Hinzufügen zusätzlicher Regeln noch verfeinert werden, reicht es, nur einen bestimmten Bestandteil zu modifizieren und die Reihenfolge der verwendeten Abschlusseigenschaften beizubehalten. Anhand dieser modularen Bauweise werden die Stärken der endlichen Automaten offensichtlich. Wie genau die Operationen im Hintergrund ablaufen, ist nicht weiter wichtig. Nimmt man z. B. eine zusätzliche Regel NK_n zur nominalen Kompositionsstammformbildung an, sei es eine tag-basierte Regel oder eine lexikalisierte Regel, genügt es wieder, den Durchschnitt aus NK und NK_n zu bilden und das Ergebnis ins Modell einzubinden:

$$NK' = NK \cap NK_n$$
$$T_{\text{dis}} = \Big(NK' \cup VK \cup AK\Big)^{+} \cdot \Big(NG \cup VG \cup AG\Big)$$

Damit wurde mit T_{dis} der letzte Bestandteil des Finite-State-Modells beschrieben. Im Anschluss wird gezeigt, wie alle Komponenten zu einem Mechanismus zusammengesetzt werden, der aufgrund eines als Zeichenkette gegebenen Kompositums eine Menge von möglichen und disambiguierten Segmentierungen zurück gibt.

3.4 Zusammenspiel der Komponenten

In den vorigen Unterkapiteln sind alle Elemente des Finite-State-Modells vorgestellt und formal beschrieben worden. An dieser Stelle wird gezeigt, wie die vier Transducer T_{seg}, T_{sil}, T_{suf} und T_{dis} hinter einander geschaltet werden können, um bei der Eingabe eines einzelnen deutschen Kompositums als Ergebnis eine Menge von grammatisch korrekten Segmentierungen zu erhalten.

Diese Hintereinanderschaltung wird mithilfe der Abschlusseigenschaft der Komposition[8] realisiert und als Transducer-Kaskade bezeichnet. Damit zwei Transducer zu einer Kaskade verbunden werden können, muss das Ergebnis des ersten Transducers als Eingabe des zweiten Transducers angenommen werden. Transducer werden an dieser Stelle als Transduktionen aufgefasst bzw. als Abbildungen einer Menge auf eine andere. Die Komposition von Transduktionen setzt voraus, dass der Wertebereich der ersten Transduktion dem Definitionsbereich der zweiten Transduktion entspricht. Dies muss für die genannten Transducer gewährleistet werden.

[8] Hierbei handelt es sich um die mathematische Komposition von Funktionen, die nicht mit der Komposition im Sinne der Wortbildung verwechselt werden darf.

Der Transducer T_{seg}, der für die naive Segmentierung der Komposita verantwortlich ist, nimmt als Eingabewörter alle Folgen von Zeichen an, die aus dem deutschen Alphabet gebildet werden können. Das deutsche Alphabet ist die Menge Σ_D. Die Menge aller Wörter, die aus Σ_D gebildet werden können, ist Σ_D^*. Damit ist der Definitionsbereich für T_{seg} bestimmt. Das Alphabet der im Laufe der Arbeit vorgestellten Tags, wird als Σ_T bezeichnet. Tagsets sind Folgen von Symbolen aus Σ_T und sind dementsprechend in der Menge Σ_T^* enthalten. Nach der Segmentierung besteht ein Segment aus der graphematischen Form des Segments gefolgt von einem Tagset. Es besteht also aus einer Folge von Symbolen aus Σ_D, auf die eine Folge von Symbolen aus Σ_T folgt. Alle möglichen Segmente mit allen möglichen Tagsets bilden eine Untermenge von $\Sigma_D \cdot \Sigma_T$. Eine Segmentierung ist eine Folge von Segmenten, die Menge dieser Segmentfolgen wird durch eine Untermenge von $(\Sigma_D \cdot \Sigma_T)^*$ beschrieben. Da die Segmentierungen prinzipiell mehrdeutig sind, wird einem Eingabewort eine Menge von Ausgabewörtern zugeordnet. Die Menge dieser Mengen ist $2^{(\Sigma_D \cdot \Sigma_T)^*}$ und entspricht der Wertebereich von $|T_{\text{seg}}|$. Die Transduktion $|T_{\text{seg}}|$ ist durch die folgende Abbildungsvorschrift definiert:

$$|T_{\text{seg}}| : \Sigma_D^* \rightarrow 2^{(\Sigma_D^* \cdot \Sigma_T^*)^*}$$

Der Definitionsbereich ist weiter gewählt als die Menge der deutschen Komposita. Dies ist nötig, da erst durch die Segmentierung untersucht werden kann, ob ein Eingabewort ein Kompositum ist. Ist die Ergebnismenge der Transduktion für ein Eingabewort leer, dann ist das Eingabewort kein Kompositum oder kein Wort, dass im Lexikon enthalten ist.

Nun sind die Transducer T_{sil}, T_{suf} und T_{dis} aber nur für einzelne Eingabewörter definiert, und nicht für Mengen von Wörtern. Die Definitions- und Wertebereiche können an den folgenden Abbildungsvorschriften der jeweiligen Transduktionen abgelesen werden:

$$|T_{\text{sil}}| : (\Sigma_D^* \cdot \Sigma_T^*)^* \rightarrow (\Sigma_D^* \cdot \Sigma_T^*)^*$$
$$|T_{\text{suf}}| : (\Sigma_D^* \cdot \Sigma_T^*)^* \rightarrow (\Sigma_D^* \cdot \Sigma_T^*)^*$$
$$|T_{\text{dis}}| : (\Sigma_D^* \cdot \Sigma_T^*)^* \rightarrow (\Sigma_D^* \cdot \Sigma_T^*)^*$$

Um die Definitionsbereiche dieser Transducer dem Wertebereich von T_{seg} anzupassen, reicht es, die Transduktion über Mengen von Eingabewörtern zu erweitern, indem für jedes Eingabewort aus dieser Menge die Ausgabewörter der Transduktion zu einer Ausgabemenge vereinigt werden. Dies kann für jeden der obigen Transducer auf dieselbe Weise vollzogen werden.

$$|T_{\text{sil}}|(W) = \bigcup_{w \in W} |T_{\text{sil}}|(w)$$

$$|T_{\text{suf}}|(W) = \bigcup_{w \in W} |T_{\text{suf}}|(w)$$
$$|T_{\text{dis}}|(W) = \bigcup_{w \in W} |T_{\text{dis}}|(w)$$

Durch die Erweiterung über die Mengen von Eingabewörtern verändern sich die Abbildungsvorschriften der einzelnen Transduktionen zu den nachfolgenden Definitionen, wobei sich neben dem Definitionsbereich auch der Wertebereich entsprechend erweitert:

$$|T_{\text{sil}}| : 2^{(\Sigma_D^* \cdot \Sigma_T^*)^*} \rightarrow 2^{(\Sigma_D^* \cdot \Sigma_T^*)^*}$$
$$|T_{\text{suf}}| : 2^{(\Sigma_D^* \cdot \Sigma_T^*)^*} \rightarrow 2^{(\Sigma_D^* \cdot \Sigma_T^*)^*}$$
$$|T_{\text{dis}}| : 2^{(\Sigma_D^* \cdot \Sigma_T^*)^*} \rightarrow 2^{(\Sigma_D^* \cdot \Sigma_T^*)^*}$$

Damit kann die Komposition der Transduktionen vollzogen werden. Komponiert man mehrere Transduktionen, ergibt sich eine Folge von Abbildungen, bei der das Ergebnis einer Transduktion die Eingabe der nächsten ist usw. Für die vorliegenden Transduktionen sieht die Komposition als Abbildungsvorschrift und als Gleichung wie folgt aus:

$$|T_{\text{seg}} \circ T_{\text{sil}} \circ T_{\text{suf}} \circ T_{\text{dis}}| : \Sigma_D^* \rightarrow 2^{(\Sigma_D^* \cdot \Sigma_T^*)^*} \rightarrow 2^{(\Sigma_D^* \cdot \Sigma_T^*)^*} \rightarrow 2^{(\Sigma_D^* \cdot \Sigma_T^*)^*} \rightarrow 2^{(\Sigma_D^* \cdot \Sigma_T^*)^*}$$

$$|T_{\text{seg}} \circ T_{\text{sil}} \circ T_{\text{suf}} \circ T_{\text{dis}}|(w) = |T_{\text{dis}}|\Big(|T_{\text{suf}}|\Big(|T_{\text{sil}}|(|T_{\text{seg}}|(w))\Big)\Big)$$

Begreift man die Komposition als Abschlusseigenschaft, lässt sich auch ein einzelner Transducer T_{Parser} berechnen, der mithilfe nur einer Abbildung die Verkettung der Teiltransducer wiedergibt. Welche Möglichkeit gewählt wird, hängt von der Art der Implementierung der Transducer ab. Beide Methoden sind in Hinsicht auf die Ergebnisse äquivalent.

$$T_{\text{Parser}} = T_{\text{seg}} \circ T_{\text{sil}} \circ T_{\text{suf}} \circ T_{\text{dis}}$$
$$|T_{\text{Parser}}| : \Sigma_D^* \rightarrow 2^{(\Sigma_D^* \cdot \Sigma_T^*)^*}$$

Die Funktionsweise dieser Transduktionen wird wieder an dem Beispiel des Kompositums *Druckerwartung* illustriert. Jedes Zwischenergebnis wird hier als eigene Ebene der Analyse verstanden. Teilweise sind diese Beispiele bereits an anderer Stelle in der vorliegenden Arbeit verwendet worden, werden nun aber in einem größeren Zusammenhang dargestellt.

Sei *druckerwartung* das zu analysierende Eingabewort. Die Transduktion $|T_{\text{seg}}|$ liefert für *druckerwartung* $\in \Sigma_D^*$ das Zwischenergebnis $W_1 \in 2^{(\Sigma_D^* \cdot \Sigma_T^*)^*}$.

$$|T_{\text{seg}}|(\text{druckerwartung}) = W_1$$

$W_1 = \{$
(1) $\text{drucker}_{+\text{N+MS+S1+P1\#}}\text{wartung}_{+\text{N+FM+S3+P3\#}}$,
(2) $\text{druck}_{+\text{N+MS+S1–P\#}}\text{erwartung}_{+\text{N+FM+S3+P3\#}}$,
(3) $\text{druck}_{+\text{V\#}}\text{erwartung}_{+\text{N+FM+S3+P3\#}}$,
(4) $\text{druck}_{+\text{N+MS+S1–P\#}}\text{er}_{+\text{I\#}}\text{wartung}_{+\text{N+FM+S3+P3\#}}$
(5) $\text{druck}_{+\text{V\#}}\text{er}_{+\text{I\#}}\text{wartung}_{+\text{N+FM+S3+P3\#}}$,
$\}$

Das Zwischenergebnis W_1 enthält alle möglichen naiven Segmentierungen, wie sie schon zuvor vorgestellt wurden. Jedes Wort dieser Ergebnismenge muss nun in Bezug auf die Gestalt der Segmente hin untersucht werden. Die Transduktion $|T_{\text{sil}}|$ überprüft zunächst alle Segmente eines Wortes hinsichtlich ihrer Mehrsilbigkeit. Anschließend werden die Ergebnisse wieder zu einer Menge $W_2 \in 2^{(\Sigma_D^* \cdot \Sigma_T^*)^*}$ zusammengefasst.

$$|T_{\text{sil}}|(W_1) = \bigcup_{w \in W_1} |T_{\text{sil}}|(w) = W_2$$

$W_2 = \{$
(1) $\text{drucker}_{+\text{N+MS+S1+P1+M\#}}\text{wartung}_{+\text{N+FM+S3+P3+M\#}}$
(2) $\text{druck}_{+\text{N+MS+S1–P–M\#}}\text{erwartung}_{+\text{N+FM+S3+P3+M\#}}$,
(3) $\text{druck}_{+\text{V–P\#}}\text{erwartung}_{+\text{N+FM+S3+P3+M\#}}$,
(4) $\text{druck}_{+\text{N+MS+S1–P–M\#}}\text{er}_{+\text{I\#}}\text{wartung}_{+\text{N+FM+S3+P3+M\#}}$,
(5) $\text{druck}_{+\text{V–M\#}}\text{er}_{+\text{I\#}}\text{wartung}_{+\text{N+FM+S3+P3+M\#}}$,
$\}$

Die Tagsets aller Segmente außer der Interfixe werden mit den zusätzlichen Tags +M und –M versehen, je nach dem, ob ein Segment mehr- oder einsilbig ist. Anschließend werden die Segmente auf spezifische Suffixe oder Auslaute hin untersucht. Dies wird durch die Transduktion $|T_{\text{suf}}|$ realisiert. Ähnlich wie zuvor wird bei der Analyse jedes Wort der Eingabemenge W_2 einzeln untersucht. Die Ergebnisse werden zu der neuen Ergebnismenge $W_3 \in 2^{(\Sigma_D^* \cdot \Sigma_T^*)^*}$ vereinigt.

$$|T_{\text{suf}}|(W_2) = \bigcup_{w \in W_2} |T_{\text{suf}}|(w) = W_3$$

$W_3 = \{$
(1) $\text{drucker}_{+\text{N+MS+S1+P1+M–SA\#}}\text{wartung}_{+\text{N+FM+S3+P3+M+SS\#}}$
(2) $\text{druck}_{+\text{N+MS+S1–P–M–SA\#}}\text{erwartung}_{+\text{N+FM+S3+P3+M+SS\#}}$,
(3) $\text{druck}_{+\text{V–P–SA\#}}\text{erwartung}_{+\text{N+FM+S3+P3+M+SS\#}}$,
(4) $\text{druck}_{+\text{N+MS+S1–P–M–SA\#}}\text{er}_{+\text{I\#}}\text{wartung}_{+\text{N+FM+S3+P3+M+SS\#}}$,
(5) $\text{druck}_{+\text{V–M–SA\#}}\text{er}_{+\text{I\#}}\text{wartung}_{+\text{N+FM+S3+P3+M+SS\#}}$,
$\}$

Die Analyse der Segmente ist mit den letzten beiden Transduktionen abgeschlossen, was bleibt, ist die Anwendung der tag-basierten Distributionsregeln, die durch die Transduktion $|T_{\text{dis}}|$ beschrieben werden. Diese Transduktion nimmt keine Veränderungen an den Eingabewörtern selbst vor, sondern gibt sie unverändert zurück, wenn sie grammatisch richtig sind, oder es wird das leere Wort zurückgegeben, wenn sie als grammatisch falsch erkannt werden. Durch die Erweiterung der Transduktion auf eine Menge von Eingabewörtern enthält man als Endergebnismenge die Menge $W \in 2^{(\Sigma_D^* \cdot \Sigma_T^*)^*}$, die nur noch die grammatisch korrekten Segmentierungsarten enthält.

$$|T_{\text{dis}}|(W_3) = \bigcup_{w \in W_3} |T_{\text{dis}}|(w) = W$$

$W = \{$
(1) $\text{druck}_{\text{+N+MS+S1-P-M-SA\#}}\text{erwartung}_{\text{+N+FM+S3+P3+M+SS\#}}$,
(2) $\text{druck}_{\text{+V-P-SA\#}}\text{erwartung}_{\text{+N+FM+S3+P3+M+SS\#}}$,
(3) $\text{drucker}_{\text{+N+MS+S1+P1+M-SA\#}}\text{wartung}_{\text{+N+FM+S3+P3+M+SS\#}}$
$\}$

Die Segmentierungsarten (4) und (5) werden verworfen, da sie aufgrund der Distributionsregeln als grammatisch falsch erkannt werden. Die Segmentierungsart (4) wird durch die Regel NK_{13} ausgeschlossen, da Substantive ohne spezifischen Wortausgang, die darüber hinaus Singulariatantum sind, keine Fugenelemente aufweisen. Es ist auch diese Regel, die die Segmentierungsart (2) akzeptiert. Bei der Segmentierungsart (5) ist für die Eliminierung die Regel VK verantwortlich, die besagt, dass mit Verbstämmen nur die Nullfuge oder -*e* stehen kann. Die Segmentierungsart (2) wiederum wird durch diese Regel als grammatisch richtig akzeptiert. Für die Segmentierungsart (1) ist keine Regel formuliert. Sie wird durch die allgemeine Kompositastruktur akzeptiert. Dies ist legitim, denn es heißt nichts anderes, als dass keine Regeln erfasst wurde, die die Segmentierungsart (1) ausschließen würde. Sie muss also akzeptiert werden.

Die Analyse aller Komposita läuft in dieser Reihenfolge ab. Ein Transducer T_{Parser}, wie er zuvor beschrieben wurde, würde das Endergebnis W direkt ohne Zwischenschritte liefern, müsste aber erst sehr aufwendig, wenn auch automatisch, berechnet werden. Eine Modifizierung des Modells an beliebiger Stelle hätte eine Neuberechnung des ganzen Modells zu Folge. Daher bleibt man in der vorliegenden Arbeit bei der modularen Form.

Hiermit sind in Kapitel 3 alle Bestandteile des Modells beschrieben worden. Die Funktionsweise wurde im Einzelnen sowie im Hinblick auf das gesamte Modell dargestellt.

Zusammenfassung

In der vorliegenden Arbeit wird ein Finite-State-Modell zur morphologischen Analyse der deutschen Komposita vorgeschlagen.

In Kapitel 1 werden zunächst die linguistischen Anforderungen an ein Wortbildungsmodell erfasst und nach ihrer Modellierungsmöglichkeit mit Finite-State-Mitteln bewertet, wobei die Wortbildungsart der Komposition besonders hervorgehoben wird. Die bereits an dieser Stelle erwähnte syntagmainterne Paradigmatik und Syntagmatik der Kompositionsglieder spielt später eine bedeutende Rolle bei der Modellierung der Finite-State-Regeln. Anschließend werden die Eigenschaften der deutschen Komposita in Hinsicht auf ihre Oberflächenstruktur besprochen (1.2 u. 1.3). Sonderfälle, die sich nicht klar in das Wortbildungsschema der Komposition einbinden lassen, werden von der Modellierung ausgeschlossen, diese Entscheidungen werden an Beispielen begründet (1.4).

Die Bedeutung der Segmentierung sowie des Taggings für eine automatische Analyse der deutschen Komposita wird im Anschluss beschrieben (1.5). Der Segmentbegriff wird definiert und auf lexikalisierte Konstituenten der Komposita beschränkt (1.5.1). Die zentrale Bedeutung des Lexikons für die Prozesse der Segmentierung und des Taggings wird besprochen, gleichzeitig wird das Lexikon zur einzigen Instanz bei der Bestimmung lexikalisierter Segmente erhoben (1.5.1 u. 1.5.2). Das Hauptproblem bei der Segmentierung der Komposita, die Ambiguität auf struktureller und lexikaler Ebene, wird dargestellt und mit den Teilprozessen der reinen Segmentierung und des Taggings in Verbindung gesetzt (1.5.3).

Als Erscheinung an der Kompositaoberfläche sind die Fugenelemente und deren Verteilung ein wichtiger Faktor bei der computerbasierten Segmentierung. Deren Wesen wird gezeigt und in Verbindung mit dem zugehörigen Erstglied wird der Begriff der Kompositionsstammform eingeführt (1.6), die später eine wichtige Rolle bei der Modellierung der Regeln erfüllt. Die Bildung der Kompositionsstammformen mit den verschiedenen Fugenelementen wird detailliert beschrieben und anschließend im Hinblick auf die Implementierungsmöglichkeiten im Modell zusammengefasst (1.6.1 u. 1.6.2). Dargestellt wird auch die Bedeutung dieser Regulariäten an der Kompositionsfuge für den Segmentierungsprozess und eine folgende Disambiguierung (1.6.4).

Kapitel 2 führt die in der vorliegenden Arbeit genutzten Mittel aus der Automatentheorie ein. Endliche Automaten, Transducer und Transducer mit Endausgabefunktion werden vorgestellt und ihrer Verwendungsart entsprechend interpretiert (2.1.1,

2.2.1 u. 2.2.3). Die Abschlusseigenschaften werden für beide Konzepte dargelegt und als einer der größten Vorteile der Finite-State-Modellierung hervorgestellt (2.1.3 bzw. 2.2.2). Weitere Vorteile wie Determinierungs- und Minimierungsmöglichkeiten der endlichen Automaten werden illustriert (2.1.2). Abgeschlossen wird dieses Kapitel durch einige weniger formale, meta-linguistische Überlegungen zur Analogie der menschlichen Sprachverarbeitung und Finite-State-Methoden und zu den Vor- und Nachteilen der Finite-State-Modellierung, wobei die bekanntesten und auch einige weniger bekannte Argumente vorgestellt werden.

In Kapitel 3 wird das Finite-State-Modell konzipiert und besprochen. Es werden insgesamt vier Transducer modelliert, deren Zusammenwirken am Ende eine erfolgreiche Komposita-Analyse liefern soll.

Zunächst wird das Referenzlexikon des Modells aufgebaut. Die Struktur der Einträge und die verwendeten Tags bzw. Tagsets werden detailliert dargestellt (3.1.1). Anschließend wird ein deterministischer, azyklischer Lexikon-Transducer mit Endausgabefunktion konstruuiert (3.1.2) und schrittweise durch einfache Definitionsänderungen der Übergangs-, Ausgabe-, und Endausgabefunktion zu einem Segmentierungsmechanismus für Komposita modifiziert, der gleichzeitig auch das Tagging der gefundenen Segmente übernimmt (3.1.3). Der Einfluss der Kompositastruktur auf die Determinierungsmöglichkeit eines so enstandenen Transducers wird diskutiert.

Nach der Segmentierung werden die identifizierten Segmente in zwei Schritten analysiert, um weitere Kriterien und Tags für die Bildung der Kompositionsstammformen zur Verfügung zu haben. Untersucht wird im ersten Schritt (3.2.1), ob die Segmente ein- oder mehrsilbig sind. Zu diesem Zweck wird für jeden möglichen deutschen Silbenkern eine Transducerregel konzipiert. Einige Transducerregeln, deren Strukturen beispielhaft sind, werden gezeigt und ihre Funktionsweise wird erklärt. Auch wird besprochen, wie die einzelnen Regeln zu einem einzigen Transducer vereinigt und schließlich von Segmenten auf Folgen von Segmenten erweitert werden. Ähnlich wird beim zweiten Analyseschritt verfahren, der sich der Erkennung von bestimmten Suffixen und Auslauten an den Segmentenden widmet (3.2.2). Auch hier wird für jedes der 19 erfassten Suffixe und für zwei Arten von Auslauten jeweils eine Regel in Form eines Transducers entworfen. Dieser Transducer erweitert das Tagset des Segments mit Informationen, die den Analyseergebnissen entsprechen. Die einzelnen Regeln werden wieder zu einem einzigen Transducer vereinigt und dieser so modifiziert, dass er Sequenzen von Segmenten als Eingabe annimmt.

Bei der Beschreibung der Distributionsregeln (3.3) wird von zwei Kontexten ausgegangen: einmal lokal, innerhalb der Kompositionsstammform zwischen Grundform des Erstgliedes und Fugenelement und einmal global, zwischen Erstglied und Grundwort des Kompositums. Beschrieben werden zuerst die lokalen Regeln, wobei zwischen tag-basierten (3.3.2) und lexikalisierten Regeln (3.3.3) unterschieden wird. Alle Regeln werden als Transducer modelliert, wobei sie so konzipiert sind, dass sie Ausnahmen von einem allgemeinen Modell der Kompositionsstammformen erfassen. Für die Fälle, die sie genau abdecken sollen, sind die eindeutig, in allen anderen Fällen

tolerant. Werden diese Regeln dann mit der Abschlusseigenschaft des Durchschnitt zusammengefügt, ergibt sich ein falschen Kompositionsstammformen gegenüber zunehmend intoleranteres Modell. Der globale Kontext wird mithilfe von verschiedenen Abschlusseigenschaften aufgrund der Regeltransducer des lokalen Kontextes modelliert (3.3.4). So entsteht ein Transducer, der alle Kontexte beschreibt und auf die bisherigen naiven Segmentierungsergebnisse angewendet werden kann.

Schließlich werden die vier beschriebenen Transducer zu einem Modell zusammengefügt (3.4). Dazu werden noch einige überwiegend formal-mathematische Voraussetzungen besprochen und entsprechende Modifikationen durchgeführt. Die Funktionsweise des Modells sowie Zwischenstufen und Disambiguierungsvorgänge werden an Beispielen verdeutlicht. Die Richtigkeit dieser Prozesse wird linguistisch begründet.

Literaturverzeichnis

Asteroth, A. und Baier, C. (2002). *Theoretische Informatik – Eine Einführung in Berechenbarkeit, Komplexität und formale Sprachen mit 101 Beispielen.* Pearson Studium, München.

Carstensen, K.-U., Ebert, C., Endriss, C., Jekat, S., Klabunde, R. und Langer, H. (Hg.) (2001). *Computerlingusitik und Sprachtechnologie – Eine Einführung.* Spektrum Akademischer Verlag, Heidelberg.

Church, K. W. (1980). *On Memory Limitations of Natural Language Processing.* Diplomarbeit, Massachusetts Institute of Technology.

Daciuk, J., Watson, B. W. und Watson, R. E. (1998). Incremental Construction of Minimal Acyclic Finite State Automata and Transducers. In L. Karttunen (Hg.), *FSMNLP'98: International Workshop on Finite State Methods in Natural Language Processing*, S. 48–55. Association for Computational Linguistics, Somerset, New Jersey.

Dudenredaktion (Hg.) (1998). *Duden. Grammatik der deutschen Gegenwartssprache*, Bd. 4. Dudenverlag, Mannheim.

Eichinger, L. M. (2000). *Deutsche Wortbildung: Eine Einführung.* Gunter Narr Verlag, Tübingen.

Erben, J. (2000). *Einführung in die deutsche Wortbildungslehre.* Erich Schmidt Verlag, Berlin.

Fleischer, W. und Barz, I. (1995). *Wortbildung der deutschen Gegenwartssprache.* Max Niemayer Verlag, Tübingen.

Fuhrhop, N. (1998). *Grenzfälle morphologischer Einheiten.* Dissertation, Freie Universität Berlin.

Glück, H. (Hg.) (2000). *Metzler-Lexikon Sprache.* Verlag J. B. Metzler, Stuttgart.

Hopcroft, J. E. und Ullman, J. D. (2000). *Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie.* Oldenbourg Verlag, München.

Jurafsky, D. und Martin, J. H. (2000). *Speech and Language Processing – An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, New Jersey.

Kornai, A. (1985). Natural Languages and the Chomsky Hierarchy. In M. King (Hg.), *Proceedings of the 2nd European Conference of the Association for Computational Linguistics*, S. 1–7.

Langer, S. (1998). Zur Morphologie und Semantik von Nominalkomposita. In *Tagungsband KONVENS 98*, S. 83–97. Bonn.

Mohri, M. (1997). Finite-State Transducers in Language and Speech Processing. *Computational Linguistics*, Bd. 23(2): S. 269–311.

Mohri, M. und Allauzen, C. (2002). p-Subsequentiable Transducers. In *Seventh International Conference CIAA 2002*, S. 24–34.

Rackow, U., Dagan, I. und Schwall, U. (1992). Automatic Translation of Noun Compounds. In *Proceedings of COLING-92, Nantes*, S. 1249–1253.

Roche, E. und Schabes, Y. (1995). Deterministic Part-of-Speech Tagging with Finite-State Transducers. In E. Roche und Y. Schabes (Hg.), *Finite State Language Processing*, S. 205–239. MIT Press, Cambridge.

Roche, E. und Schabes, Y. (1997). Introduction to Finite-State Devices in Natural Language Processing. In E. Roche und Y. Schabes (Hg.), *Finite State Language Processing*, S. 1–66. MIT Press, Cambridge.

Sternefeld, W. (2004). Syntax – Eine merkmalbasierte generative Beschreibung des Deutschen. Kursbegleitendes Skript, Universität Tübingen.

van Noord, G. (2000). Treatment of ε-Moves in Subset Construction. *Computational Linguistics*, Bd. 26(2): S. 61–76.

Hiermit erkläre ich, dass diese Arbeit von mir weder an der Akademia Bydgoska noch an einer anderen wissenschaftlichen Einrichtung als Magisterarbeit eingereicht wurde.

Ferner erkläre ich, dass ich diese Arbeit selbständig verfasst und keine anderen als die darin angegebenen Hilfsmittel benutzt habe.