Abteilung Computerlinguistik
Universität Zürich

Sommersemester 2000

Seminar „Syntaxtheorien und computerlinguistische Praxis“
Prof. Dr. Michael Hess, lic. phil. Simon Clematide, lic. phil. Gerold Schneider

Einführung in die Tree Adjoining Grammars anhand des XTAG-Parsers

Wenzel Doppler
Militärstrasse 85
8004 Zürich
wenzel@pop.agri.ch

Esther Kaufmann
Schwyzerstrasse 64c
8832 Wollerau
e.kaufmann@access.unizh.ch

Juni 2000

Inhaltsverzeichnis

1. Einleitung

Die vorliegende Arbeit entstand im Rahmen des Seminars "Syntaxtheorien und computerlinguistische Praxis" in der Abteilung Computerlinguistik an der Universität Zürich. Sie soll eine Einführung in die von Joshi, Levy und Takahashi im Jahr 1975 präsentierte Tree Adjoining Grammar geben und eine praktische Anwendung davon, den an der Universität Pennsylvania entwickelten XTAG-Parser, vorstellen.

Die Arbeit ist in zwei grössere Teile geteilt. Im ersten Teil wird der Grammatikformalismus mit den grundlegenden Strukturen sowie den möglichen Erweiterungen beschrieben. Der zweite Teil widmet sich dem XTAG-Parser, der Sätze in Relation zu einer TAG fürs Englische analysiert, wobei eine Menge von ausgewählten Beispielsätzen durch den XTAG-Parser syntaktisch analysiert und die Ergebnisse ausgewertet werden.

Der theoretische Teil mit der Einführung in die Tree Adjoining Grammars stützt sich hauptsächlich auf die Aufsätze von Aravind K. Joshi (1987) sowie Aravind K. Joshi und Yves Schabes (1997). Für die Benutzung des XTAG-Parsers stand eine umfangreiche Grammatikbeschreibung der XTAG Research Group von der Universität Pennsylvania zur Verfügung. Sie kann übers Internet unter "http://www.cis.upenn.edu/~xtag" eingesehen werden.

2. Tree Adjoining Grammars

Tree Adjoining Grammar (**TAG**) ist ein Grammatikformalismus. Er wurde 1975 von Joshi, Levy und Takahashi vorgestellt.[1] Die Sprachen (**TALs**), die mit TAGs generiert werden, weisen einige strikt kontextsensitiven Konstruktionen auf und gehören in die Klassen der sogenannten schwachen (*mildly*) kontextsensitiven Sprachen.[2] Sie beinhalten kontextfreie Sprachen. Der TAG-Formalismus wurde anfänglich nur aufgrund seiner mathematischen Eigenschaften studiert, allerdings entpuppte er sich als interessanter Kandidat für die adäquate Notation von Universalgrammatiken.[3] Die grundlegenden Elemente einer TAG sind **Bäume** (*trees*), d.h. strukturierte Objekte und nicht Zeichenketten (*strings*). Dadurch kann man Formalismen konstruieren, die grosse generative Fähigkeiten haben. Obwohl die Terminalbäume, die durch eine TAG abgeleitet werden, Strings sind, sind TAGs in erster Linie Baumgenerierungssysteme (im Gegensatz zu den Stringgenerierungssystemen der kontextfreien Grammatiken). Vijay-Schanker , Weir und Joshi (1987) haben einen Automaten entwickelt, „Embedded Pushdown Automaton“ genannt, der genau die Klasse der Tree Adjoining Languages erkennt.[4]

Was neu ist am TAG-Formalismus, ist der Modus, wie Rekursion verwendet wird und Abhängigkeiten deklariert werden. Die grundlegende Idee dahinter ist, dass das Festhalten von lokalen Kookkurenzbeziehungen innerhalb einfacher syntaktischer Strukturen von der Beschreibung der Rekursion sowie den unbegrenzten syntaktischen Abhängigkeiten getrennt wird. Als Konsequenz dieser Behandlungsart von Rekursion (auch „factor recursion“ genannt[5]) und Abhängigkeiten sind TAGs mächtiger als kontextfreie Grammatiken, jedoch nur geringfügig mächtiger.[6]

TAGs erlauben es, diverse syntaktische Phänomene wie Subkategorisierung oder Rekursivität auf eine sehr natürliche und intuitive Weise einzufangen. Selbst idiomatische Ausdrücke können mit TAGs einfach erfasst werden[7] oder linguistische Prinzipien, die Fernabhängig-

[1] Joshi et al.1975
[2] Joshi 1987, S. 87; Joshi / Schabes 1997, S. 70
[3] Kroch 1987, S. 143
[4] Vijay-Shanker et al. 1987, S. 393
[5] Joshi 1987, S. 87
[6] Harbusch 1997, S. 1
[7] Parobek / Schabes 1997, S. 5

keiten enthalten (z.B. „Who$_i$ did John tell Sam that Bill invited e_i?“).[8] Für linguistische Anwendungen wird der TAG-Formalismus ganz allgemein und lexikalisierte TAGs im Besonderen als gut geeignet beurteilt.[9]

2.1 Formale Definition der TAG

Eine Tree Adjoining Grammar G ist ein Quintuple (Σ, NT, I, A, S), wobei gilt:

- Σ ist die endliche Menge der Terminalsymbole,
- NT ist die endliche Menge der Nichtteminalsymbole,
- S ist das ausgezeichnete Nichterminalsymbol (das Startsymbol) aus der Menge NT,
- I ist die endliche Menge der Initialbäume von G,
- A ist die endliche Menge der Auxiliarbäume von G.

Die Menge der Bäume in I ∪ A nennt man die Menge der Elementarbäume (*elementary trees*). Ein Baum, der durch eine Komposition von zwei anderen Bäumen gebildet wird, nennt man Ableitungsbaum (*derivation tree*).[10] Unter Blattwort versteht man die Beschriftungen der Blattknoten in einem Ableitungsbaum von links nach rechts gelesen. Die Baummenge T(G) einer TAG G besteht aus allen initialen Bäumen sowie denen, die sich ausgehend von allen initialen Bäumen mittels Kompositionsoperation bilden lassen. Die von einer TAG G beschriebene Sprache L(G) ist definiert als Menge aller Terminalzeichenketten (oder Blattwörter) der Bäume in T(G).

Alle diese knappen formalen Definitionen werden im Folgenden ausführlicher erklärt.

[8] Kroch 1987, S. 144
[9] TAG Research Group 1999, S. 5
[10] Joshi / Schabes 1997, S. 70f.

2.2 Lexikalisierte TAGs

Die in diesem Abschnitt angebrachten Ausführungen sind im Moment vielleicht etwas unverständlich. Dennoch erscheint es uns als sinnvoll, hier auf einige Unterscheidungen und Inkonsistenzen in der Literatur zu TAGs hinzuweisen. Die Ausführungen werden nach den Erklärungen zu den grundlegenden Strukturen einer TAG im Abschnitt 2.3 klarer.

Es ist wichtig, eine grundsätzliche Differenzierung in Bezug auf TAGs hervorzuheben. Man unterscheidet bei den TAGs **lexikalisierte TAGs** (*lexicalized TAGs*) und formale TAGs. Der Unterschied ergibt sich aus den möglichen Beschriftungen der Blattknoten in einem Elementarbaum.

In einer formalen bzw. nicht-lexikalisierten TAG dürfen an den Blattknoten der Initialbäume nur Terminalsymbole oder für Substitution markierte Nichtterminalsymbole liegen (Erläuterungen dazu im Abschnitt 2.3.2.1).[11] Gemäss Joshi (1987) können die Blattknoten von Initialbäumen nur mit Nichtterminalen beschriftet sein. In einer grafischen Darstellung jedoch bezeichnet er die Blattknoten mit Terminalen.[12] Es ist anzunehmen, dass ein Druckfehler vorliegt. Einigkeit herrscht darüber, dass alle inneren Knoten von Initialbäumen durch Nichtterminale belegt sind.

Die Blattknoten der Initialbäume einer lexikalisierten TAG (manchmal **LTAG** genannt) sind mit Terminalsymbolen (natürlichsprachlichen Wörtern), welche die Elemente einer Zielsprache bilden, oder mit Nichtterminalsymbolen besetzt. Oder genauer: In einer lexikalisierten TAG muss in den Blattfolgen jedes initialen Baums und jedes auxilaren Baums mindestens ein Terminalsymbol (natürlichsprachliches Wort) erscheinen. Damit ist in jeder Baumstruktur ein lexikalisches Element realisiert und jede elementare Baumstruktur systematisch mit einem lexikalischen Element verbunden. Das Terminalsymbol, welches für das lexikalische Element steht, nennt man **Anker** (*anchor*). Die inneren Knoten der Initialbäume sind Nichtterminale. In diese können andere Bäume eingesetzt werden, wie wir im Abschnitt 2.3.2 noch sehen werden. Die Grammatik besteht in einer lexikalisierten TAG somit aus einem Lexikon, in der jedes lexikalische Element mit einer endlichen Anzahl von Baumstrukturen verbunden ist, deren Anker das lexikalische Element ist. Es ist notwendig,

[11] Joshi / Schabes 1997, S. 71
[12] Joshi 1987, S. 88

dass die Position des Ankers nicht von einem leeren String belegt sein darf. Lexikalisierte TAGs sind endlich ambig, d.h., dass es keinen Satz von endlicher Länge gibt, der von einer lexikalisierten TAG mit einer unendlichen Anzahl Analysen verarbeitet wird.[13]

Wir werden uns in der vorliegenden Arbeit auf lexikalisierte TAGs konzentrieren, da es hier darum geht, TAGs im Hinblick auf ihre Verwendung zur Beschreibung natürlicher Sprache zu betrachten, und da die Grammatik der XTAG-Tools auch eine lexikalisierte ist.

Aus der Definition von lexikalisierten TAGs ergibt sich eine Uneinheitlichkeit bei der Verwendung der Begriffe Terminal- und Nichtterminalsymbol in den Publikationen zu TAGs. Im Zusammenhang mit formalen TAGs ist mit Terminalsymbol meistens das linguistische Präterminal, die Wortkategorieklasse (z.B. Verb, Nomen etc.), gemeint. Die Zuordnung vom eigentlichen Terminal zum Präterminal erfolgt dann über ein Lexikon, wodurch die Anzahl der Grammatikregeln drastisch reduziert wird, da nicht jedes natürlichsprachliche Wort in einer Regel auftauchen muss.[14] Bei der Beschreibung von lexikalisierten TAGs bezeichnen die Terminalsymbole die natürlichsprachlichen Wörter (z.B. *bite*, *dog* etc.).[15] Auch hier legen wir uns analog zur Grammatik des XTAG-Parsers darauf fest, dass mit Terminalsymbolen von nun an natürlichsprachliche Wörter gemeint sind.

2.3 Die grundlegenden Strukturen des TAG-Formalismus

2.3.1 Die Elementarbäume

Der TAG-Formalismus ist ein System, das Bäume generiert. Es besteht aus einem endlichen Set von **Elementarbäumen**, welche mit der Funktion der Grammatikregeln einer Phrasenstrukturgrammatik vergleichbar sind[16], und einer **Verknüpfungsoperation** (*composition operation*). Mit diesen Grammatikregeln in Form von Bäumen und den Verknüpfungsoperationen sind bereits alle fundamentalen Beschreibungsmittel einer TAG definiert.

[13] Joshi / Schabes 1997; S. 80
[14] Buschauer et al. 1991, S. 5
[15] Joshi / Schabes 1997; Kroch 1987, S. 145
[16] Harbusch 1997, S. 2

Die grundlegenden Strukturen einer TAG sind also Bäume. Da Bäume eine Tiefe haben, können in ihnen komplexe strukturelle Zusammenhänge dargestellt werden. Die Regeln einer kontextfreien Grammatik, die aus einem Nichtterminalsymbol auf der linken Seite eine Folge von Symbolen auf der rechten Seite (Terminale oder Nichtterminale) ableitet, haben im Vergleich zu einem TAG-Baum nur die Tiefe 1.[17] Die folgende Abbildung 1 zeigt, dass mit einem Elementarbaum einer TAG mehr strukturelle Information dargestellt werden kann als mit einer kontextfreien Phrasenstrukturregel.

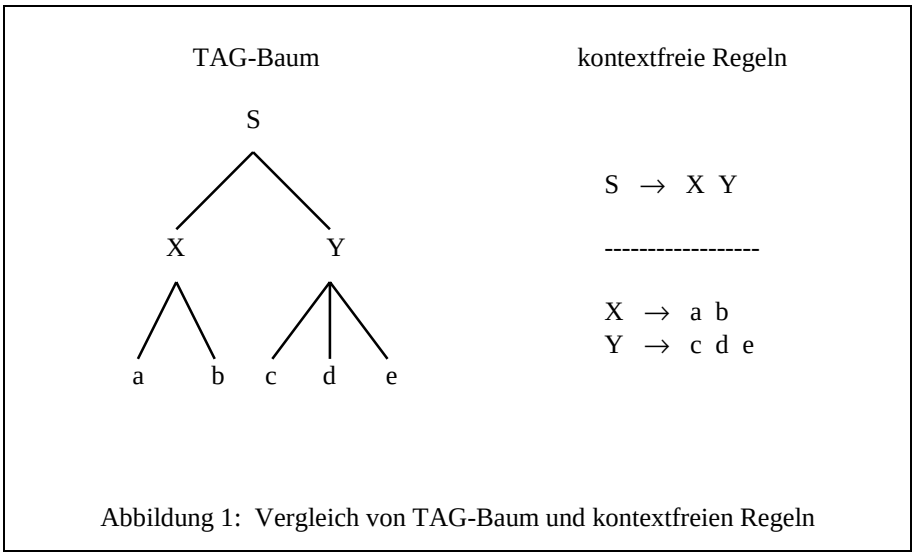


Abbildung 1: Vergleich von TAG-Baum und kontextfreien Regeln

Die Menge der elementaren Bäume bildet den Kern der Grammatik. Man unterscheidet nach der Rolle, die Bäume bei der Kompositionsoperation spielen (siehe Abschnitt 2.3.2), zwischen der Menge der initialen Bäume und der Menge der auxiliaren Bäume. Eine Tree Adjoining Grammar ist also abgekürzt G = (I, A), wobei I und A endliche Mengen von Elementarbäumen sind. (Diese Kurznotation zur Definition einer TAG wird sehr häufig verwendet.) Die Bäume in I heissen **Initialbäume** (*initial trees*) und die Bäume in A **Auxiliarbäume** (*auxiliary trees*). Die Baummenge (*tree set*) **T(G)** einer TAG G ist die Menge aller in G ableitbaren Bäume, ausgehend von den Initialbäumen in I. Und die (String)-Sprache L von G ist die Menge aller Terminalstrings bzw. **Blattwörter** aus den Bäumen von T(G).

2.3.1.1 Initialbäume

[17] Buschauer et al. 1991, S. 4f.

Initialbäume bilden die Basis der Grammatikbeschreibung. Ein Baum α ist ein Initialbaum, wenn er die folgende Form hat:

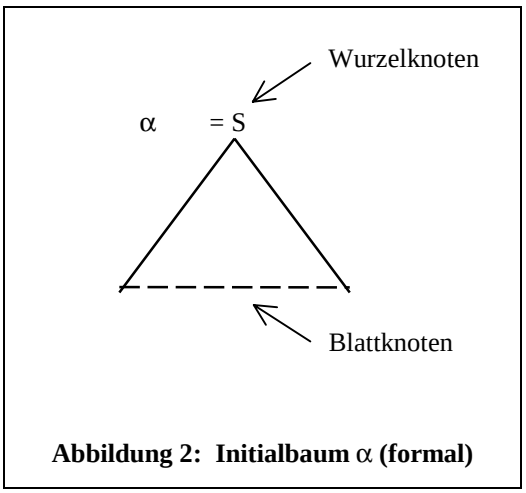


Abbildung 2: Initialbaum α (formal)

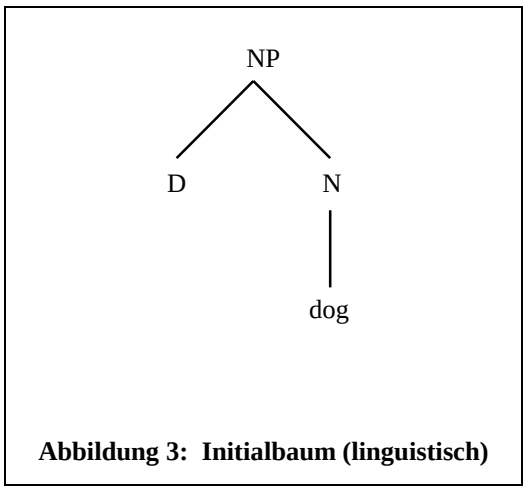


Abbildung 3: Initialbaum (linguistisch)

Der **Wurzelknoten** (*root node*) des Initialbaums α in Abbildung 2 ist mit S beschriftet. Das Gleichheitszeichen bedeutet nicht, dass α und S identifiziert werden. Vielmehr ist mit α der gesamte Initialbaum mit dem Wurzelknoten S gemeint. Es ist jedoch üblich, die Elementarbäume grafisch so darzustellen. An den **Blattknoten** (*frontier nodes*) befinden sich Terminale (mindestens eines) oder Nichtterminalsymbole. Nichtterminale Blattknoten sind für Substitutionen markiert (gekennzeichnet durch ↓). Alle inneren Knoten sind Nicht-terminalsymbole. Die Initialbäume einer lexikalisierten TAG dürfen nicht durch die reine Adjunktionsoperation (siehe Abschnitt 2.3.2) in andere Bäume eingehängt werden, und sie können für sich stehen, da sie mindestens ein vollständiges terminales Blattwort besitzen.

Initialbäume repräsentieren minimale linguistische Strukuren ohne Rekursion, z.B. Phrasenstrukturen oder einfache Sätze. Man nennt einen initialen Baum **X-Typ-Initialbaum**, wenn sein Wurzelknoten mit X beschriftet ist. In Abbildung 3 handelt es sich um einen NP-Typ-Initialbaum. Alle Kategorien oder Konstituenten, die als Argumente für mehrere Initialbäume oder Auxiliarbäume (siehe Abschnitt 2.3.1.2) stehen können, sind X-Typ-Initialbäume. Der S-Typ-Initialbaum ist ein Spezialfall. Sein Wurzelknoten ist mit S (dem Startsymbol) etikettiert, und es wird von der Grammatik verlangt, dass jedes gültige Blattwort von mindestens einem S-Typ-Initialbaum abgeleitet werden muss.

2.3.1.2 Auxiliarbäume

In den TAGs dienen Auxiliarbäume zur Beschreibung der rekursiven Strukturen. Ein Auxiliarbaum β hat die folgende Struktur:

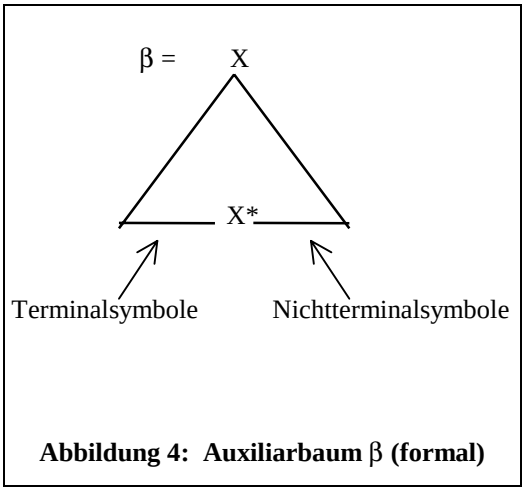


Abbildung 4: Auxiliarbaum β (formal)

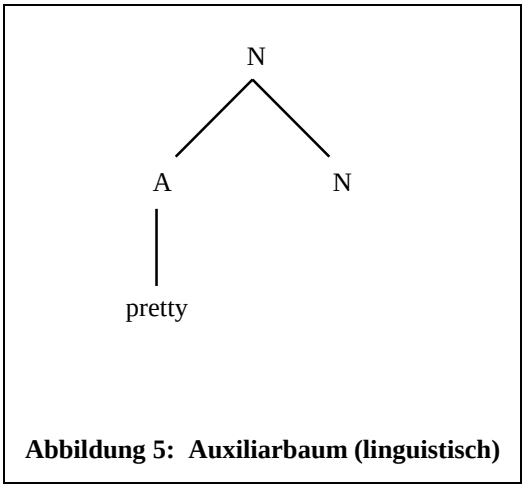


Abbildung 5: Auxiliarbaum (linguistisch)

Der Wurzelknoten des Auxiliarbaums β ist mit X beschriftet, wobei X ein Nichtterminalsymbol ist. Der mit X beschriftete Blattknoten wird als **Fussknoten** (*foot node*) von β bezeichnet (gekennzeichnet mit ∗). Alle inneren Knoten sind mit Nichtterminalsymbolen beschriftet; alle Blattknoten sind mit Nichtterminal- oder Terminalsymbolen etikettiert. Nichtterminalsymbole an den Blattknoten sind für Substitutionen markiert, ausser der sogenannte Fussknoten, welcher für die Adjunktion des Baums in einen anderen gebraucht wird. Die Beschriftung des Fussknotens eines Auxiliarbaums muss identisch sein mit der des Wurzelknotens. Dem Fussknoten eines Auxiliarbaums wird systematisch eine Null-Adjunktions-Beschränkung auferlegt (siehe Abschnitt 2.4.2), die eine Adjunktion des Baums im eigenen Fussknoten aus linguistischer Motivation verhindert, um z.B. eine unmittelbare Wiederholung vom gleichen natürlichsprachlichen Wort zu vermeiden. Auxiliare Bäume können ansonsten ineinander und in initiale Bäume sowie in bereits teilweise abgeleitete Baumstrukturen adjungiert werden.

Auxiliarbäume repräsentieren linguistische Konstituenten, die entweder als Adjunkte oder als Modifikatoren zu den anderen Grundkonstituenten hinzukommen können (z.B. Adverbiale).

Sie können auch grundlegende Satzstrukturen für Verben oder Konstituenten, die Sätze als Komplemente haben, repräsentieren.

Es gibt für die Initialbäume und die Auxiliarbäume keine weiteren Beschränkungen. Beide müssen jedoch minimal sein, d.h., dass ein Initialbaum einem minimalen Satz ohne Rekursionsanwendung auf irgendeinem Nichtterminalsymbol entspricht und ein Auxiliarbaum, dessen Wurzelknoten und Fussknoten mit X etikettiert sind, mit einer Struktur, die eine einmalige Rekursionsanwendung aufweist, korrespondiert.

2.3.2 Die Verknüpfungsoperation

2.3.2.1 Adjunktion

Der Name der Tree Adjoining Grammar gibt selbst Aufschluss über die Art der im Formalismus verwendeten Verknüpfungsoperation, die **Adjunktion** (*adjoining*). Darunter kann man sich „ein Ineinander-Einhängen von Bäumen“[18] vorstellen. Operiert werden darf mit den Elementarbäumen sowie den durch Adjunktion daraus abgeleiteten Baumstrukturen. Die Auxiliarbäume können ineinander und in Initialbäume oder Ableitungen davon eingesetzt werden. Wie wir bereits in Abschnitt 2.3.1 gesehen haben, basiert die Unterteilung der Elementarbäume in initiale Bäume und auxiliare Bäume auf der Funktion, welche die Bäume bei der Adjunktionsoperation ausüben.

Grundsätzlich wird bei der Adjunktionsoperation ein auxiliarer Baum β mit Wurzel- und Fussknoten X an einem Knoten X in einen initialen Baum α oder durch Adjunktion bereits modifizierten Baum γ hineingehängt.

Nehmen wir an, wir hätten einen Baum γ, der einen Knoten k mit der Beschriftung X enthält, und einen Auxiliarbaum β, dessen Wurzel auch mit X etikettiert ist (aufgrund der Definition von Auxiliarbäumen in Abschnitt 2.3.1.2 muss β einen einzigen Blattknoten mit X beschriftet haben). Das Hineinhängen von β in γ am Knoten k resultiert in einem Baum γ' unter der Voraussetzung, dass die folgenden Operationen ausgeführt worden sind:

1. Der **Unterbaum** (*sub tree*) b von γ, der von k dominiert wird, wird herausgeschnitten und hinterlässt eine Kopie von k.
2. Der Auxiliarbaum β wird bei k angefügt und sein Wurzelknoten mit k gleichgesetzt.
3. Der Unterbaum b wird an den Fussknoten von β angefügt, wobei der Wurzelknoten k von b mit dem Fussknoten von β identifiziert wird.

Die folgende Darstellung illustriert die Adjunktion vom Auxiliarbaum β am Knoten X des Initialbaums α mit dem Wurzelknoten S.

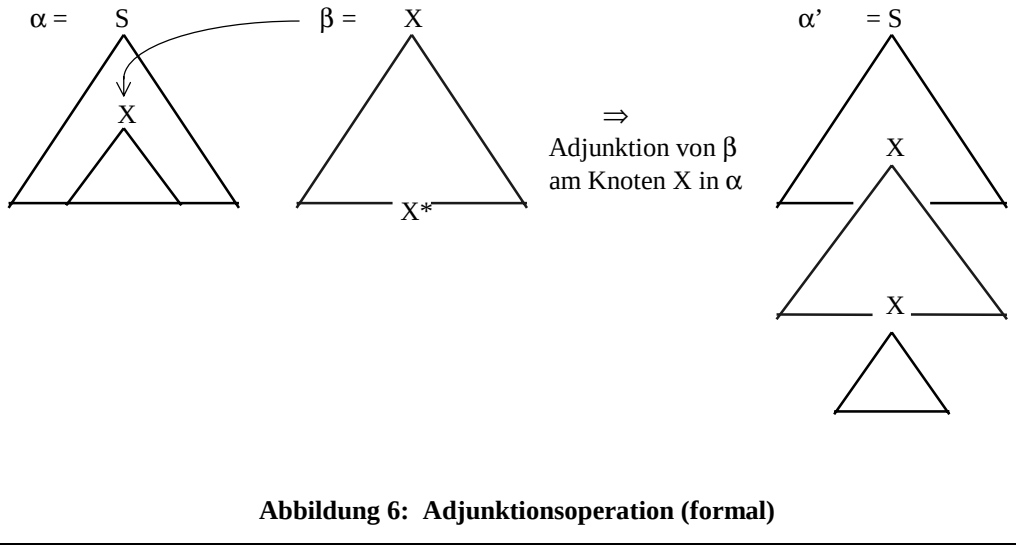


Abbildung 6: Adjunktionsoperation (formal)

In Abbildung 7 wird die Adjunktion anhand eines natürlichsprachlichen Beispiels nochmals veranschaulicht. Der ↓ kennzeichnet einen Nichtterminalknoten, der mit einer Substitutionsoperation (siehe Abschnitt 2.3.2.2) ersetzt werden kann; der ∗ markiert Fuss-knoten. (Diese Notation wird im Folgenden fortgesetzt.)

[18] Buschauer et al. 1991, S. 4

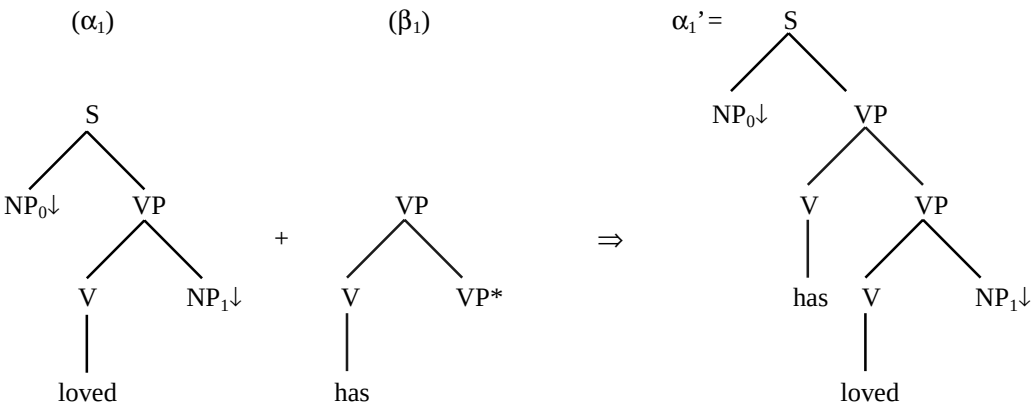


Abbildung 7: Adjunktionsoperation (linguistisch)

Es ist wichtig zu bemerken, dass die Idee hinter der Adjunktion nicht einfach ein Ersetzen ist, sondern ein Ineinanderhängen von Bäumen. Laut Kroch / Joshi (1987) erinnert die Adjunktionsoperation an die Transformationen von Chomsky. Sie konstatieren jedoch, dass sie anderer Natur ist.[19] Bedauerlicherweise gehen sie nicht näher darauf ein, von welcher Natur die Operation denn ist. Wir betrachten die Feststellung trotzdem als interessante Grundlage für Diskussionen.

2.3.2.2 Substitution als Spezialfall der Adjunktion

Ein Spezialfall der Adjunktionsoperation wird **Substitution** (*substitution*) genannt, nämlich dann, wenn ein Initialbaum an einen Wurzelknoten eines anderen Baumes angehängt wird, sodass der angehängte Baum gewissermassen auf dem andern Baum draufsitzt. Oder andersrum formuliert, jedoch mit dem gleichen Effekt: Ein Initialbaum wird an seinem Wurzelknoten durch den Fussknoten eines anderen Initialbaums substituiert.

Die Substitutionsoperation findet nur an Blattknoten von Bäumen statt, die mit Nichtterminalsymbolen beschriftet sind. Bei einer Substitution wird der Wurzelknoten eines Initialbaums in ein Nichtterminal-Blattknoten, der für Substitution markiert ist, eines anderen Initialbaums (oder eines durch Komposition schon modifizierten Baums) hineingehängt und

[19] Kroch / Joshi 1987, S. 111

ein neuer Baum produziert. Dabei müssen der Wurzelknoten und der Substitutionsknoten dieselbe Beschriftung haben. Wenn an einem Knoten k eine Substitution gemacht wird, so wird der Knoten k von dem Baum, der durch Substitution eingehängt wird, vollständig ersetzt.

Die Illustrationen 8 und 9 zeigen je zwei Initialbäume, von denen der eine durch Substitution am Knoten X in den anderen eingefügt wird, und den daraus resultierenden Baum. (Wiederum sind die Knoten, an denen die Substition erlaubt ist, mit einem ↓ gekennzeichnet.)

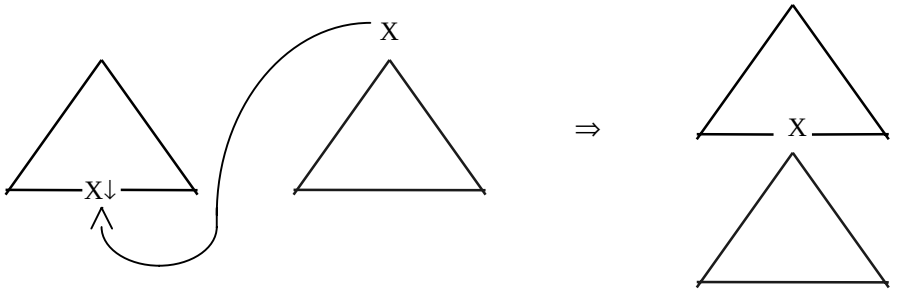


Abbildung 8: Substitutionsoperation (formal)

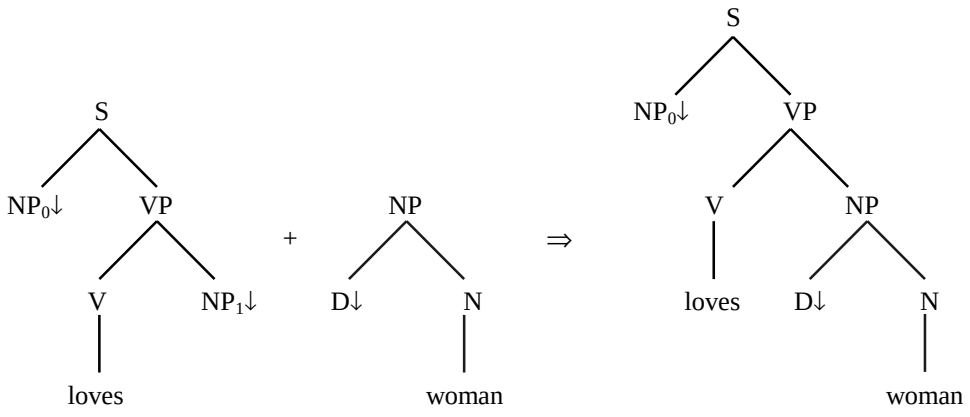


Abbildung 9: Substitutionsoperation (linguistisch)

Obwohl die Substitution technisch gesehen nur eine spezialisierte Version der Adjunktion ist, so ist es doch wichtig, eine Differenzierung zu machen.[20] Denn an einem für Substitution markierten Knoten darf keine Adjunktion vorgenommen werden. Der Ausbau der Adjunktionsoperation durch Substitution hat keinen Einfluss auf die formalen Eigenschaften von TAGs. Er beruht ebenso wie die Einführung von lokalen Beschränkungen für die Adjunktion (siehe Abschnitt 2.4.2) auf linguistischen Überlegungen.[21]

TAGs sind mit ihrer Idee vom Einhängen von Bäumen in Bäume sehr gut dazu geeignet, natürlichsprachliche Strukturen zu erfassen.[22] Die Idee der Adjunktion entspricht bsw. der Erweiterbarkeit von Konstituenten, z.B. die Erweiterung einer Nominalphrase um einen Relativsatz. Durch wiederholtes Ineinandersetzen lässt sich beliebig oft angewendete Rekursion erreichen. Die folgende Abbildung zeigt die Adjunktion eines natürlich-sprachlichen Satzes um einen Relativsatz.

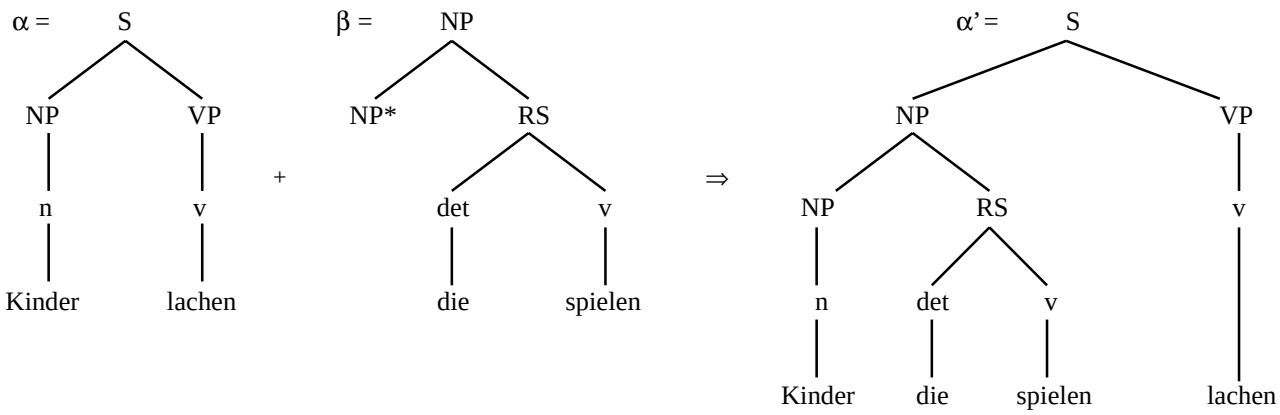


Abbildung 10: Relativsatz als Beispiel einer Adjunktion

Die Idee des Spezialfalls Substitution korrespondiert unter anderem mit der Forderung nach Komplementen von Verben. Ein Beispiel dafür ist in Abbildung 8 illustriert.

[20] XTAG Research Group 1999, S. 6
[21] Joshi / Schabes 1997, S. 74
[22] Buschauer et al. 1991, S. 6

2.4 Erweiterungen im TAG-Formalismus

2.4.1 Abhängigkeiten

Elementarbäume (Initial- und Auxiliarbäume) bilden in einer TAG den Bereich, in welchem Abhängigkeiten (z.B. Subkategorisierungsabhängigkeiten oder Lücke-Füller-Abhängigkeiten wie im Satz „Who$_i$ does this dog belong to e$_i$?“) zwischen Elementen von Bäumen deklariert werden können. Zur Beschreibung gewisser Abhängigkeiten, vorallem der Lücke-Füller-Abhängigkeit, wird eine spezielle Beziehung zwischen den Knoten eines Elementarbaums hergestellt und in die Grammatik eingeführt. Man nennt die Beziehung „**Linking**“.[23] Ein Knoten kann eine oder mehrere Verbindungen (*links*) zu anderen Knoten im gleichen Baum haben. In der nachfolgenden Abbildung sind die Knoten B und Q miteinander verlinkt, was durch eine gepunktete Bogenlinie angezeigt ist. Abhängigkeiten in TAGs werden üblicherweise mit gepunkteten Bogenlinien oder der Einfachheit halber manchmal durch Koindizierung gekennzeichnet.

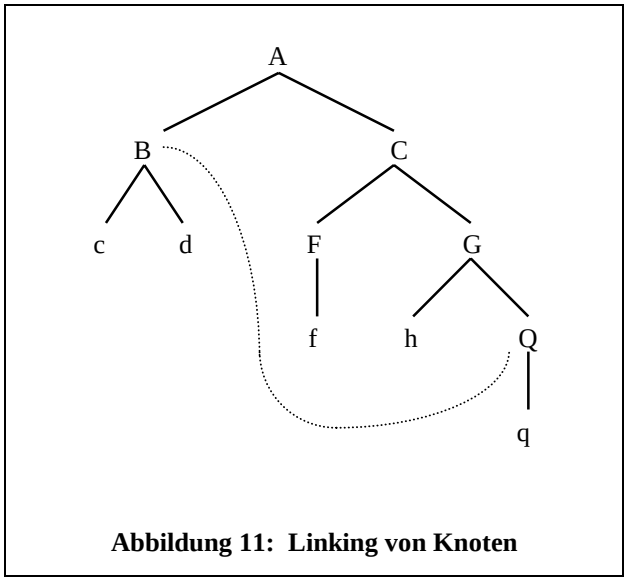


Abbildung 11: Linking von Knoten

Im Prinzip sind alle Elemente eines Elementarbaums irgendwie miteinander verbunden, weil sie ganz einfach zum selben Elementarbaum gehören.[24] Deshalb könnten eigentlich zwei beliebige Knoten eines Elementarbaumes durch Linking miteinander verknüpft werden.

[23] Joshi 1987, S. 95-99
[24] Kroch / Joshi 1987, S. 113

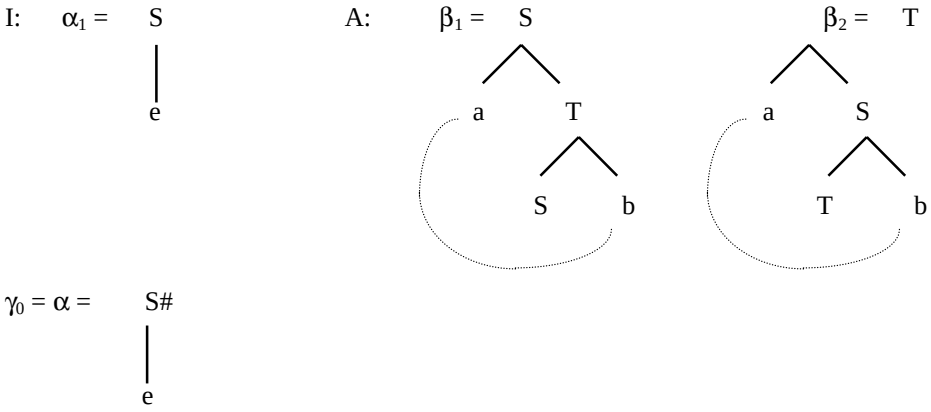
Aufgrund linguistischer Überlegungen ergibt sich jedoch für das Verlinken von zwei Elementen in einem Elementarbaum die Bedingung, dass die eine Konstituente die andere c-kommandieren muss. Im Falle von Lücke-Füller-Abhängigkeiten gelten sogar die folgenden Bedingungen:[25]

Ein Knoten k1 darf an einen Knoten k2 gelinkt werden, wenn

1. k2 k1 c-kommandiert (d.h., k2 dominiert nicht k1, und es existiert ein Knoten m, der k2 unmittelbar dominiert und der auch k1 dominiert),
2. k1 und k2 die gleiche Beschriftung haben,
3. k1 einen Nullstring oder ein Terminalsymbol dominiert.

Wichtig ist, dass die Abhängigkeitsbeziehungen bei der Verknüpfung von Elementarbäumen bewahrt werden. Die Links können im Verlaufe einer Ableitung durch Adjunktion aber gedehnt werden, wie aus den Abbildungen 12 bis 15 ersichtlich ist.

Gegeben sei eine TAG G = (I, A), wobei



(Der Gartenhag markiert den Knoten, an den adjungiert wird)

Abbildung 12: TAG mit Links

Wenn wir β_1 im Knoten S# von γ_0 adjungieren, so erhalten wir γ_1. Der Link bleibt bestehen.

[25] Kroch / Joshi 1987, S. 113

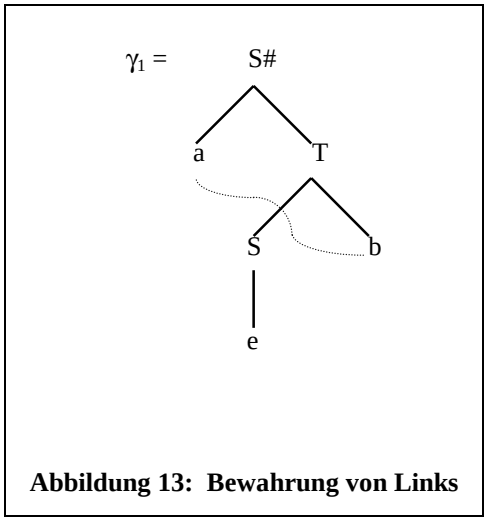


Abbildung 13: Bewahrung von Links

In einem zweiten Adjunktionsschritt wird β_1 nochmals im Wurzelknoten S# von γ_1 eingehängt. Es entsteht der Baum γ_2 mit eingebetteten (oder geschachtelten) Abhängigkeiten in Abbildung 14.

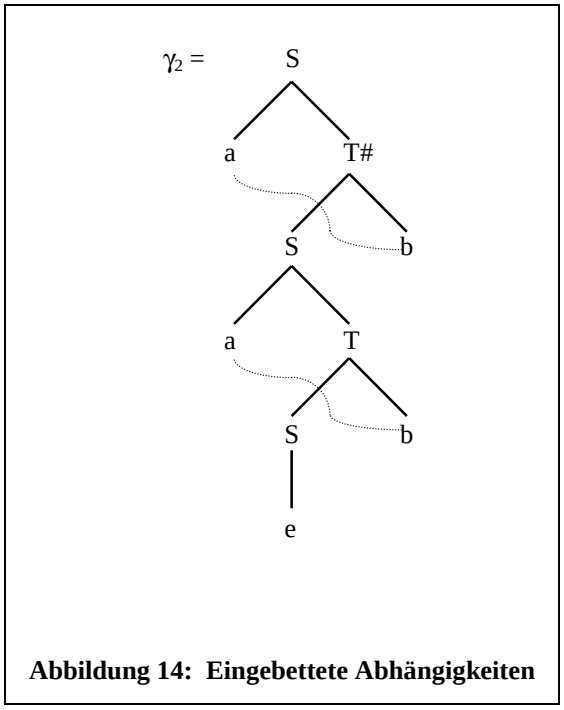


Abbildung 14: Eingebettete Abhängigkeiten

Wird nun β_2 am Knoten T# von γ_2 adjungiert, resultiert daraus γ_3 mit gekreuzten Abhängigkeiten.

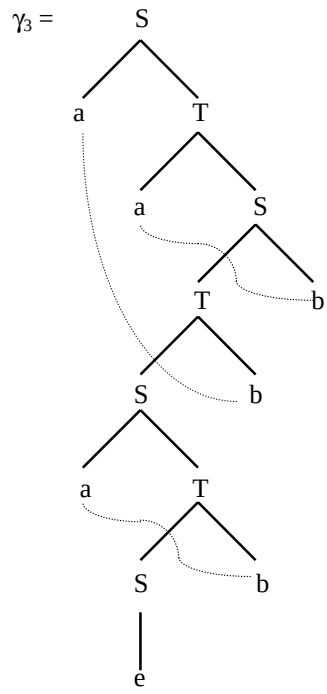


Abbildung 15: Gekreuzte Abhängigkeiten

In diesem Baum γ_3 sind **eingebettete** (*nested*) Abhängigkeiten sowie auch **überlappende oder gekreuzte** (*cross-serial*) Abhängigkeiten zu erkennen. Das bedeutet, dass TAGs mit Links gewisse gekreuzte Abhängigkeiten (sowie eingebettete Abhängigkeiten) charakterisieren können. Diese Abhängigkeiten sind entweder das Resultat der Komposition von Bäumen, oder sie sind bereits in den Elementarbäumen definiert. Schachtelung entspricht der Eigenschaft, dass man Unterbäume nicht vorne oder hinten anhängen muss, sondern durch Adjunktion an entsprechende Knoten in der Mitte und bis zu einem gewissen Grad ortsunabhängig einsetzen kann.

Der Baumstruktur in Abbildung 15 entspricht das in Abbildung 16 als Zeichenkette dargestellte Blattwort „aaaebbb“.

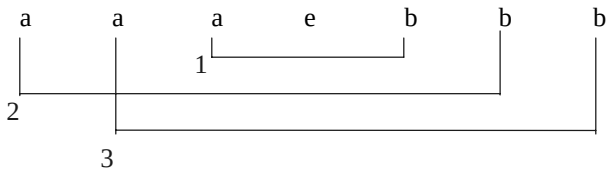


Abbildung 16: Abhängigkeiten in String dargestellt

In dieser Darstellung ist besser zu erkennen, welche Abhängigkeiten geschachtelt (Links 1 und 2, Links 1 und 3) und welche gekreuzt (Links 2 und 3) sind.

Durch die Beibehaltung von Abhängigkeiten bei der Komposition von Elementarbäumen ergibt sich ein **erweiterter Lokalitätsbereich**. (Ein lokaler Baum ist ein Teil eines Stukturbaums, der aus nur einem verzweigenden Knoten und seinen Töchtern besteht.[26]) Der erweiterte Lokalitätsbereich und die Anwendung von Rekursion erlauben einer TAG, nicht nur lokale Abhängigkeiten, sondern auch Fernabhängigkeiten zu beschreiben (z.B. „Who_i did John tell Sam that Bill invited e_i?“). Die wh-Bewegungen und -Abhängigkeiten des Englischen lassen sich mit dieser Maschinerie in einer TAG einfach unterbringen.[27]

Obwohl gezeigt wurde, dass sich mit TAGs überlappende Abhängigkeiten darstellen lassen, ist zu bemerken, dass TAGs solche nur zwischen zwei Symbolen, jedoch nicht zwischen drei Symbolen einfangen können. Für mehr Informationen dazu sei auf Vijay-Shanker et al. (1987) verwiesen.[28]

2.4.2 Lokale Beschränkungen

In einer TAG können **lokale Beschränkungen** (*local constraints*) für das Ineinanderhängen von Elementarbäumen vorkommen, welche die Adjunktionsoperation, die bisher kontextfrei war, schwach kontextsensitiv machen.
In einer TAG G = (I, A) mit lokalen Beschränkungen ist für jeden Knoten k eine und nur genau eine der folgenden Beschränkungen spezifiziert:

[26] Bussmann 1990, S. 464
[27] Kroch 1987, S. 166
[28] Vijay-Shanker et al. 1987, S. 397

1. **Selektive Adjunktion** (*selective adjunction*): Nur eine spezifizierte Teilmenge von der Menge aller Auxiliarbäume darf am Knoten k adjungiert werden. Die durch selektive Adjunktion an einen Knoten anfügbare Teilmenge von Auxiliarbäumen wird mit $\overline{\beta}$ bezeichnet.
2. **Null-Adjunktion** (*null adjoining*): Am Knoten k darf kein Auxiliarbaum angefügt werden. Null-Adjunktion wird durch ϕ angezeigt.
3. **Obligatorische Adjunktion** (*obligatory adjoining*): An einem Knoten k muss mindestens einer der Auxiliarbäume, die an k angefügt werden können, adjungiert werden. Man schreibt O($\overline{\beta}$) für die obligatorische Adjunktion, wobei $\overline{\beta}$ die Menge aller Auxiliarbäume darstellt, die an k adjungiert werden können.

Jede Adjunktion an einem Knoten k ist nur den lokalen Beschränkungen in Bezug auf k unterworfen. Sie findet immer nur an einem Knoten in einem Elementarbaum statt. Es darf aber mehr als nur ein Auxiliarbaum an einen Elementarbaum adjungiert werden, solange jeder Auxiliarbaum an unterschiedlichen Knoten des Elementarbaums hineingefügt wird. Auch die Knoten der eingefügten Auxiliarbäume sind für weitere Adjunktionen wieder verfügbar.

Die folgende Abbildung zeigt eine kleine TAG mit lokalen Beschränkungen:

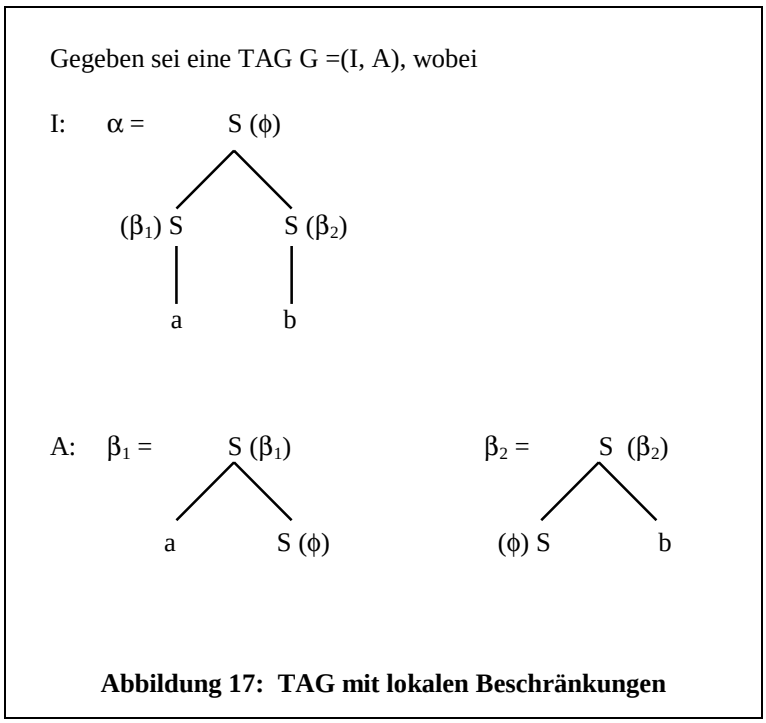


Abbildung 17: TAG mit lokalen Beschränkungen

Es ist daraus ersichtlich, dass am Wurzelknoten von α kein Auxiliarbaum angehängt werden darf. Nur β_1 ist an den linken S-Knoten und nur β_2 ist an den rechten S-Knoten auf der Tiefe 1 adjungierbar. In β_1 darf nur β_1 an den Wurzelknoten hineingehängt werden, und am Blattknoten sind keine Auxiliarbäume anfügbar. Das Gleiche gilt bei β_2.

Die Sprache L, die sich mit einer TAG mit lokalen Beschränkungen darstellen lässt, wird definiert als Menge von Terminalzeichenketten von allen aus G ableitbaren Bäumen, die keine obligatorischen Adjunktionen beinhalten. Wenn eine TAG keine Adjunktionsbeschränkungen hat, handelt es sich um eine „reine“ (*pure*) TAG (wie beschrieben bei Joshi et al. 1975) mit einer kontextfreien Adjunktionsoperation.[29]

2.4.3 Multikomponenten-Adjunktion

Joshi, Levi und Takahashi (1975) haben in ihrer ursprünglichen Version von TAGs bereits gezeigt, dass man mit der Adjunktionsoperation nicht nur einzelne Auxiliarbäume adjungieren darf, sondern sogar ein Set von solchen Bäumen in einen gegebenen Elementarbaum einfügen

[29] Kroch / Joshi 1987, S. 114

kann.[30] Eine solche **Multikomponenten-Adjunktion** ist die simultane Adjunktion von jeder Baumkomponente eines auxiliaren Baumsets an einen bestimmten Knoten in einem Elementarbaum (deshalb bisweilen auch „simultaneous adjunction“ genannt).[31] Die Bedingung für die Adjunktion eines auxiliaren Baumsets ist, dass alle Komponenten des Sets adjungiert werden müssen.[32] Die Adjunktion findet an verschiedenen Adressknoten statt. Selbstverständlich müssen dabei alle lokalen Beschränkungen, wenn vorhanden, in Bezug auf alle Knoten des Elementarbaums erfüllt werden. Die Abbildungen 18 und 19 zeigen eine Multikomponenten-Adjunktion.

[30] Joshi et al. 1975, S. 157f.
[31] Joshi 1987, S. 110
[32] Kroch 1987, S. 166

Gegeben sei eine TAG G =(I, A), wobei

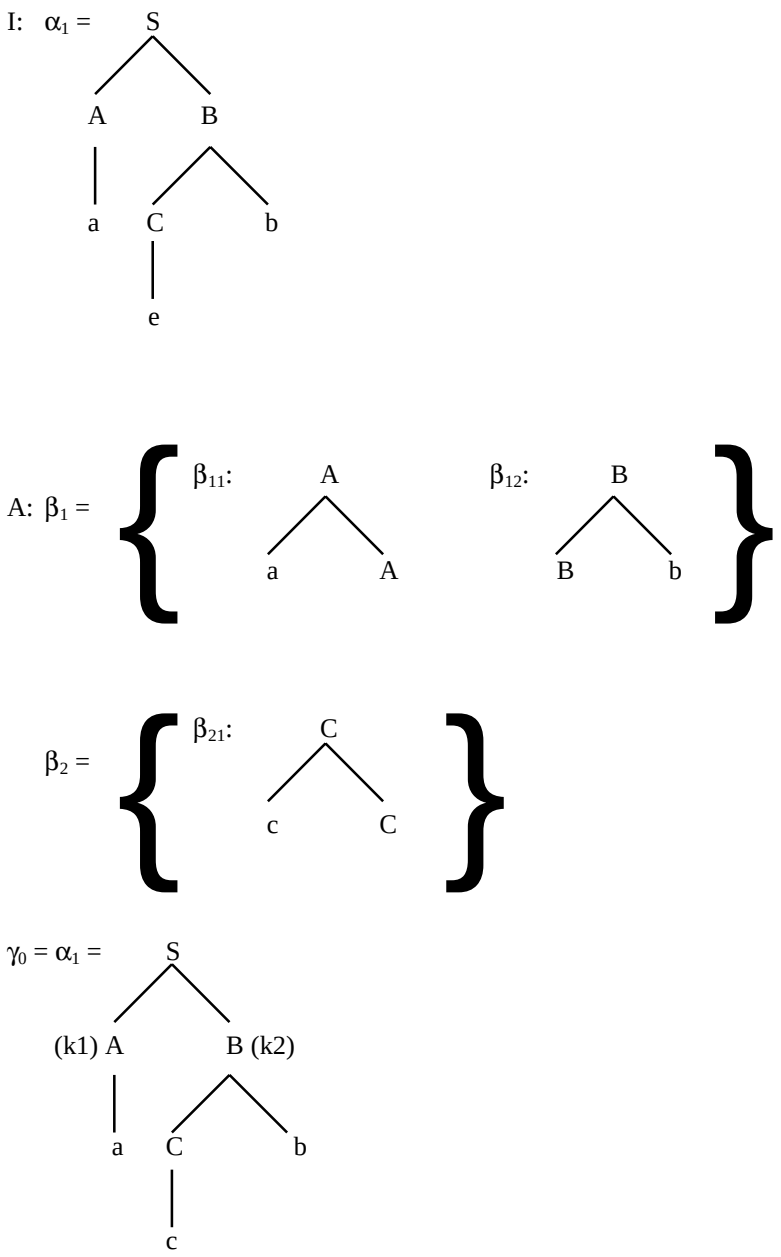


Abbildung 18: TAG mit auxiliarem Baumset

Wenn die Komponenten β_{11} und β_{12} der Auxiliarmenge β_1 in γ_0 an den Knoten k1 und k2 adjungiert werden, so entsteht der abgeleitete Baum γ_1 mit dem Blattwort „aaebb“.

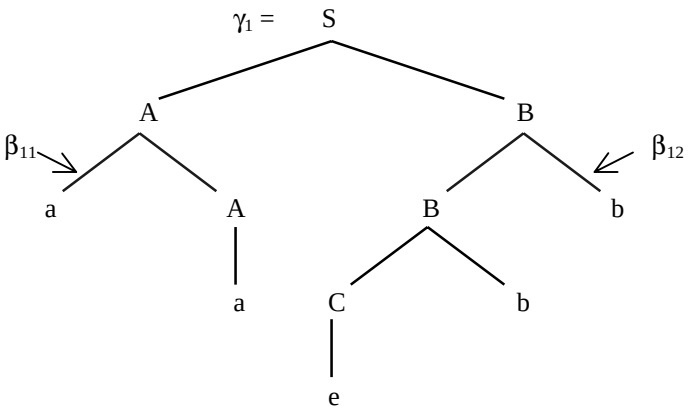


Abbildung 19: Multikomponenten-Adjunktion

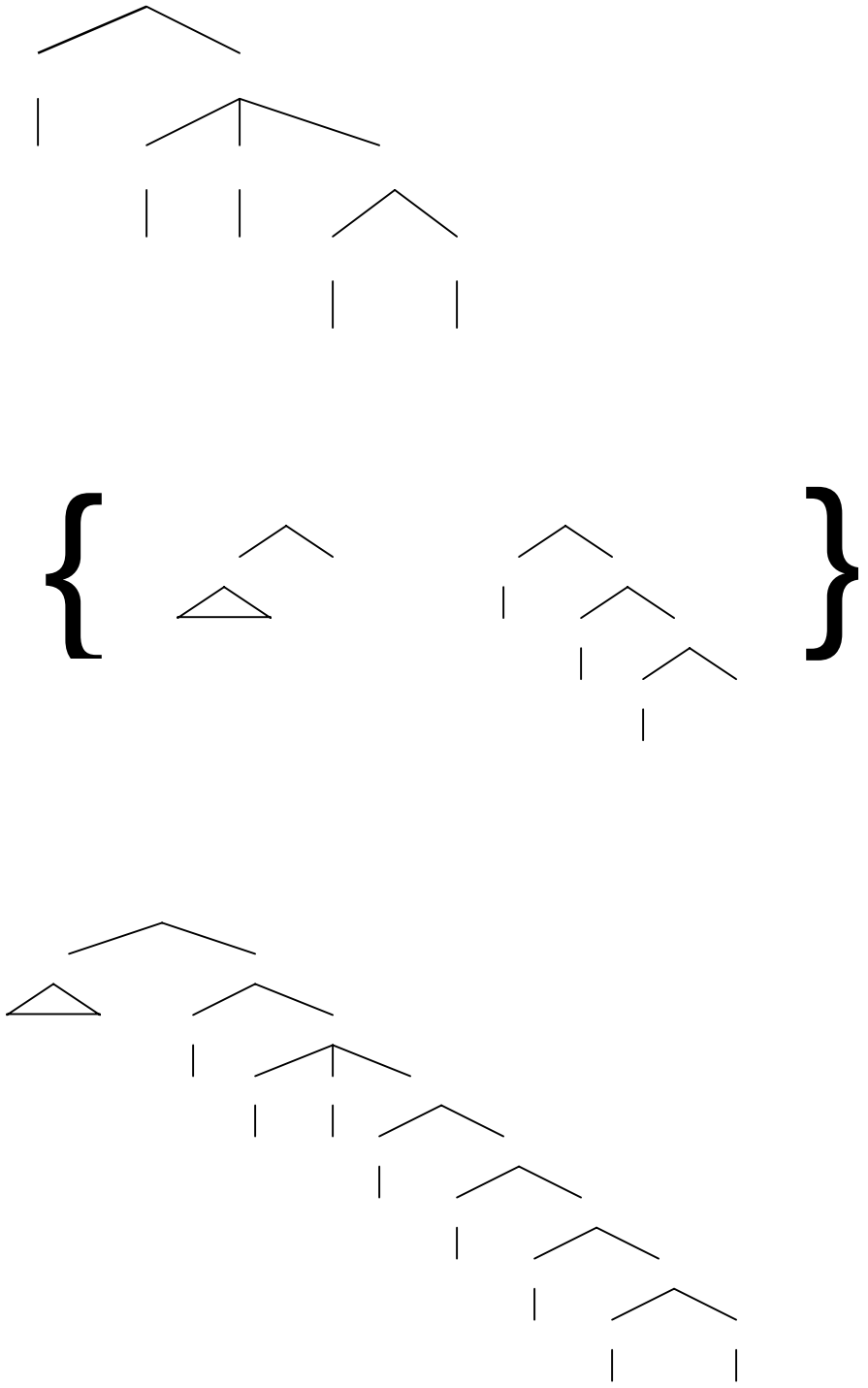
Die aus TAGs mit Multikomponenten-Adjunktion resultierenden Sprachen gehören zu den schwach kontextsensitiven Sprachen.

Für zwei beliebige Knoten, die den Bäumen des gleichen auxiliaren Baumsets angehören, kann man Abhängigkeiten definieren. Wenn zwei Knoten, die nicht zum gleichen Baum eines auxiliaren Baumsets gehören, verlinkt sind, so ist es in einem linguistischen Kontext das C-Kommando, das die Abhängigkeit zwischen den beiden Knoten aufrecht erhält, wenn das auxiliare Baumset an einen Elementarbaum adjungiert wird. Beispielsweise muss die leere Kategorie in einer Lücke-Füller-Abhängigkeit von ihrem koindizierten Antezedens c-kommandiert werden.[33] Abbildung 20 demonstriert eine Multikomponenten-Adjunktion, die den Satz „Which painting did you see a copy of?“ ableitet. Die Abhängigkeit zwischen den koindizierten Kategorien ist zulässig, da das leere Element e_i von seiner Füllerkategorie Np_i c-kommandiert wird.

[33] Kroch 1987, S. 166

S#

COMP S



Das Linking wird auch im Falle einer Adjunktionsoperation mit einem auxiliaren Baumset bewahrt.

2.4.4 Ableitungen in TAGs

Im Gegensatz zu den kontextfreien Grammatiken enthalten durch Ableitung (*derivation*) erzielte Baumstrukturen in einer TAG, wie wir sie bis anhin gesehen haben, nicht genügend Informationen darüber, wie die abgeleitete Struktur konstruiert wurde. Es gibt deshalb **Ableitungsbäume** (*derivation trees*), die über die Konstruktion der **abgeleiteten Bäume** (*derived trees*) Auskunft geben.[34] Teilbäume, die durch Adjunktion in einen Baum gehängt wurden, werden in den Ableitungsbäumen mit durchgezogenen Linien angezeigt; substituierte Teilbäume sind mit einer gestrichelten Linie markiert. Abbildung 22 zeigt einen Ableitungsbaum zum abgeleiteten Baum in Abbildung 21. In Abbildung 23 sind die zugrundeliegenden Elementarbäume dargestellt.

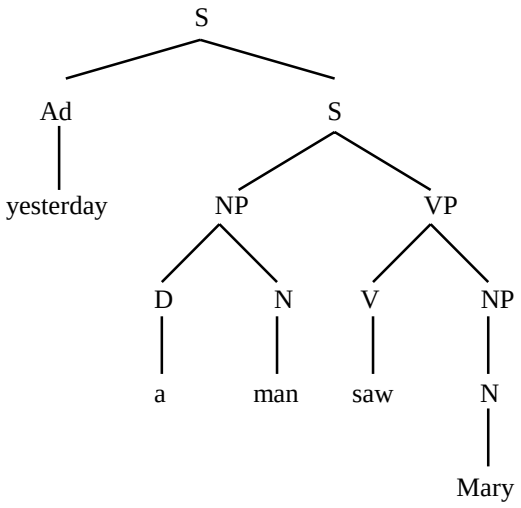


Abbildung 21: Abgeleiteter Baum

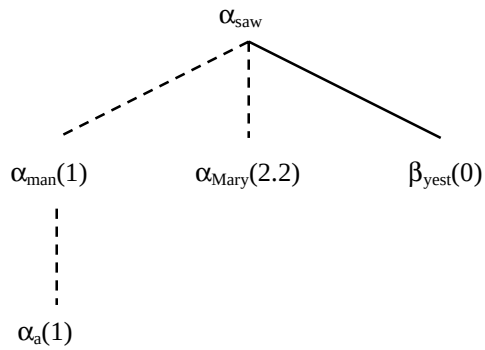


Abbildung 22: Ableitungsbaum

[34] Joshi / Schabes 1997, S. 74f.

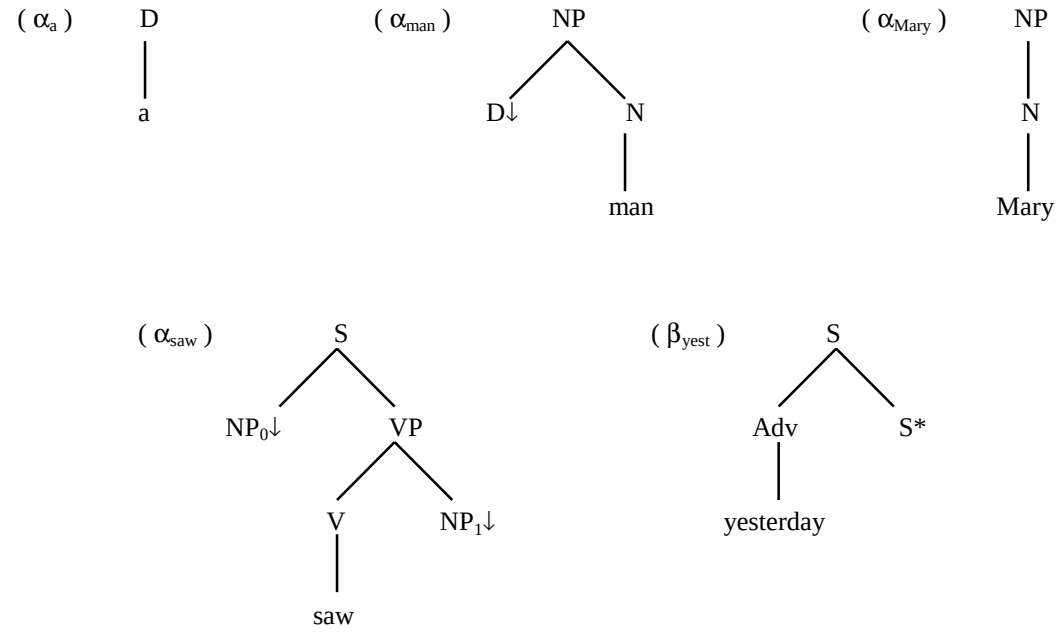


Abbildung 23: Zugrundeliegende Elementarbäume

Der Ableitungsbaum in Abbildung 22 besagt, dass α_a durch Substitution in den Baum α_{man} am Adressknoten 1 (D) eingefügt wurde, dass α_{man} substituiert wurde in den Baum α_{saw} am Adressknoten 1 (NP_0), dass α_{Mary} substituiert wurde in den Baum α_{saw} am Adressknoten 2.2 (NP_1) und dass β_{yest} am Adressknoten 0 (S) des Baums α_{saw} adjungiert wurde. Die Reihenfolge der Interpretation vom Ableitungsbaum hat keine Einwirkung auf den resultierenden abgeleiteten Baum.

Wir werden diese Unterscheidung von Ableitungsbaum und abgeleitetem Baum auch im XTAG-Parser antreffen (siehe Kapitel 3.2).

2.5 Varianten von TAGs

Seit der Vorstellung des ursprünglichen TAG-Formalismus im Jahr 1975 sind einige Varianten von TAGs entwickelt worden. Wir möchten drei davon mit kurzen Erklärungen dazu präsentieren.

2.5.1 TAGs mit Merkmalstrukturen (*Feature structure based TAGs*)

Eine auf Merkmalstrukturen basierende TAG ist eine TAG, in der die Knoten jedes Elementarbaums mit Merkmalstrukturen assoziert sind[35] (die XTAG-Grammatik ist ein Beispiel dafür). Die Merkmalstrukturen enthalten Informationen darüber, wie sich die Knoten im Baum gegenseitig beeinflussen. Die Kompositionsoperation ist definiert im Sinne von Unifikation von Merkmalstrukturen. Adjunktionsbeschränkungen basieren auf dem Erfolg oder Misserfolg dieser Unifikationen. Eine Merkmalstruktur eines Knotens besteht aus einem **oberen Teil** (*top part*), der gewöhnlich Informationen zum übergeordneten Knoten, und einem **unteren Teil** (*bottom part*), der Informationen in Bezug zu den untergeordneten Knoten einschliesst. Substitutionsknoten beinhalten nur Merkmale des oberen Teils, da der durch Substitution eingefügte Baum die Merkmale des unteren Bereichs trägt.

Abbildung 24 demonstriert eine Substitution in einer TAG mit Merkmalstrukturen.

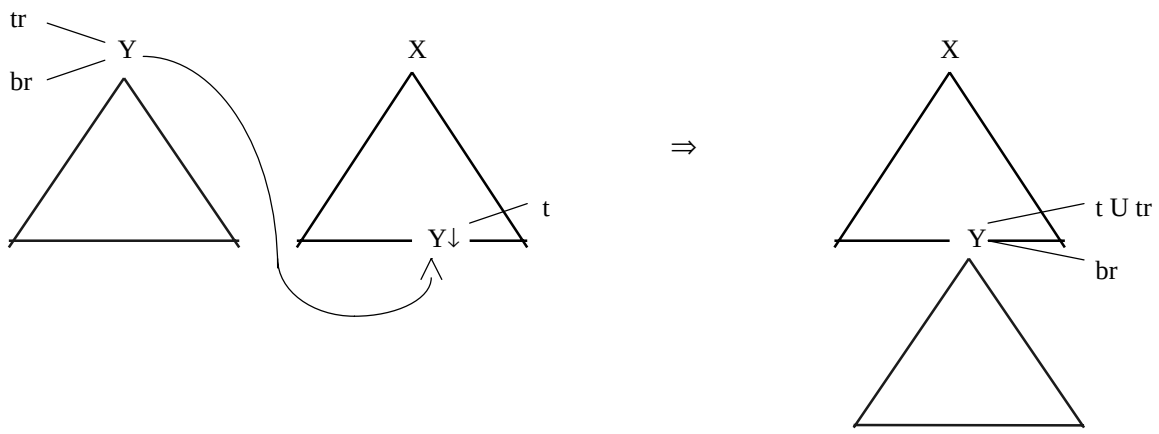


Abbildung 24: Substitution in Merkmaltruktur-TAG

Die Adjunktionsoperation ist ein wenig komplizierter. Die oberen Merkmale des Knotens, der durch die Adjunktion gesplittet wird, unifizieren mit den oberen Merkmalen des adjungierten Wurzelknotens, während die unteren Merkmale mit den unteren Merkmalen des adjungierten Fussknotens unifizieren. Die grafische Veranschaulichung in Abbildung 25 ist leichter verständlich.

[35] Joshi / Schabes 1997, S. 101

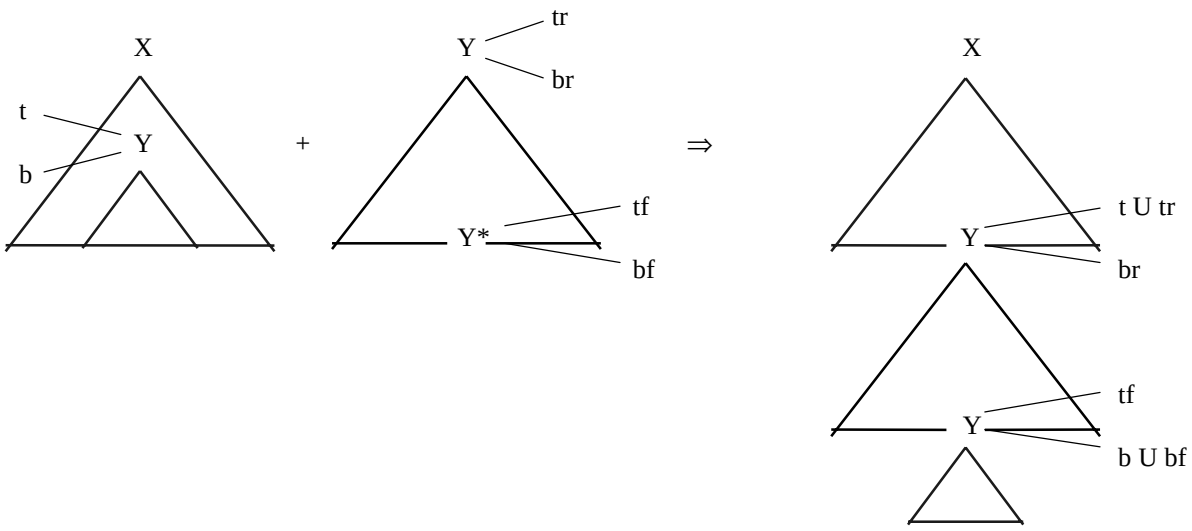


Abbildung 25: Adjunktion in einer Merkmalstruktur-TAG

Die Zusammenführung von TAGs und Unifikation erlaubt es, lokale Beschränkungen dynamisch zu spezifizieren, z.B. die Einschränkungen, die Verben ihren Komplementen auferlegen. Diese müssen ohne die Einbettung der Unifikation statisch in den Elementarbäumen deklariert werden.[36]

2.5.2 Synchrone TAGs (*Synchronous TAGs*)

Synchrone TAGs sind eine Variante von TAGs, die Korrespondenzen zwischen Sprachen beschreiben. Sie erlauben eine Anwendung von TAGs ausserhalb der Syntax bsw. mit dem Ziel einer semantischen Interpretation, Sprachgenerierung oder automatischen Übersetzung. Für die semantische Interpretation wird die Syntaxanalyse eines Satzes mit einer anderen Struktur, einer logischen Repräsentation, verbunden. Sowohl die Originalsprache wie auch die Zielsprache oder -struktur werden durch Grammatiken im TAG-Formalismus festgehalten. Für genauere Informatioen ist Joshi / Schabes (1997) zu konsultieren.[37]

[36] XTAG Research Group 1999, S. 8

2.5.3 Probabilistische TAGs (*Probabilistic TAGs*)

Bei einer probabilistischen TAG wird jede Möglichkeit für eine Adjunktionsoperation in einem Elementarbaum mit einer Wahrscheinlichkeit verknüpft. Dann wird der Wert der Wahrscheinlichkeit bei einer Ableitung ermittelt. Auch dazu kann Joshi / Schabes (1997) für weitere Erklärungen hinzugezogen werden.[38]

2.6 TAGs im Vergleich mit anderen Grammatiken

2.6.1. TAGs und kontextfreie Grammatiken

Es sind zwei Fähigkeiten, die TAGs den kontextfreien Grammatiken gegenüber mächtiger machen.

Erstens: TAGs haben einen grösseren Bereich von Lokalität als kontextfreie Grammatiken (z.B. HPSG oder LFG).[39] Dadurch lassen sich komplexe Phänomene wie Fernabhängigkeiten in einer einzigen Regel bzw. in einem einzigen Baum repräsentieren.[40] Die Abhängigkeit zwischen dem Verb *likes* und seinen zwei Argumenten, Subjekt und Objekt, wird in Abbildung 26 über die ersten zwei Regeln der kleinen kontextfreien Grammatik spezifiziert. Es ist nicht möglich, diese Abhängigkeit innerhalb einer einzigen Regel zu beschreiben, ohne die Konstituente VP aufzugeben. Die Abhängigkeit kann also nicht lokal beschrieben werden.

S → NP VP	NP → peanuts
VP → VP ADV	V → likes
VP → V NP	ADV → passionately
NP → Harry	

Abbildung 26: Abhängigkeit in kontextfreier Grammatik

[37] Joshi / Schabes 1997, S. 101
[38] Joshi / Schabes 1997, S. 101f.
[39] Joshi / Schabes 1997, S. 95
[40] Harbusch1997, S. 1

In einer TAG sind solche Abhängigkeiten durch den erweiterten Lokalitätsbereich lokal realisierbar.

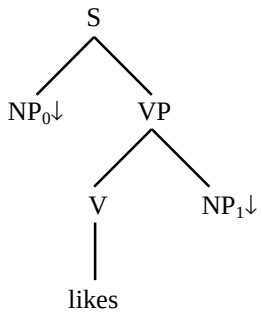


Abbildung 27: Abhängigkeit in einem TAG-Baum

Zweitens: Die Elementarbäume sind der Bereich einer TAG, wo Abhängigkeiten wie z.B. Kongruenz, Subkategorisierung, Lücke-Füller-Beziehungen festgehalten werden. Das Bestehen solcher Abhängigkeiten über grosse Distanzen, also die einfache Darstellung von Fernabhängigkeiten, ergibt sich aus der Kompositionsoperation sowie der faktorweisen Anwendung von Rekursion auf Bereiche, in denen Abhängigkeiten ursprünglich definiert wurden.

2.6.1 TAGs und Constraint Dependency Grammars

Sowohl für die TAGs wie auch für die Constraint Dependency Grammars gilt, dass sie beide schwach kontextsensitiv sind. Mit beiden Formalismen können natürlichsprachliche Phänomene angemessen spezifiziert werden.[41] Jede TAG kann auch in eine äquivalente Constraint Dependency Grammar übersetzt werden. In der umgekehrten Richtung funktioniert

[41] Harbusch 1997, S. 1

die Übersetzung nicht, da Constraint Dependency Grammars noch etwas mächtiger sind als TAGs.[42] Gerade dadurch aber, dass TAGs nur „slightly“ mächtiger sind als kontextfreie Grammatiken, sind sie für die Beschreibung von natürlicher Sprache sehr interessant.[43] Die erweiterte Mächtigkeit ist allerdings stark eingeschränkt, sie bezieht sich nur auf gewisse linguistische Strukturen.

[42] Harbusch 1997, S. 8
[43] Kroch 1987, S. 143

3. Der XTAG-Parser

XTAG ist eine Implementation des TAG-Grammatikformalismus im Sinne einer lexikalisierten TAG (LTAG). Das System wird seit 1988 an der University of Pennsylvania entwickelt. Mittlerweile beinhaltet es eine umfangreiche englische Grammatik, es gibt aber auch eine Grammatik für Französisch sowie kleinere für Koreanisch, Chinesisch und Hindi. Für unsere Arbeit war nur die englische Grammatik massgeblich.

Die Entwickler stellen in ihrer Dokumentation das System als Grammatikentwicklungsumgebung vor, das aus verschiedensten Modulen und Interfaces besteht. Wie sich aber gezeigt hat, ist die Installation des gesamten Systems mit vertretbarem Aufwand unmöglich. Es stand uns deshalb nur ein Teil der Funktionen zur Verfügung. Diese beinhalten das Parsen und Visualisieren von Sätzen. Dabei werden jeweils zwei Bäume pro Parsing-Analyse ausgegeben: ein Ableitungsbaum und ein abgeleiteter Baum (siehe Abschnitt 3.2). Ein Feature-Checker sortiert nicht konsistente Lösungen aus und liefert genaue Merkmalsstrukturen für die Analysen. Wir konnten zusätzlich über ein separates Tool auf das Lexikon und die Tree-Datenbank zugreifen.

Die englische XTAG-Grammatik ist primär für Akzeptorsysteme und nicht für Generatorsysteme konzipiert. Laut den Entwicklern soll sie eine grosse Anzahl syntaktischer Phänomene behandeln können. Diese sollen unter anderem Hilfsverbkonstruktionen (inklusive Inversion), Kopula, Raising-Konstruktionen, Topikalisierung, Relativsätze, Infinitive, Gerundien, Passivkonstruktionen, Adjunkte, it-Konstruktionen, wh-Fragesätze, PRO-Konstruktionen, Nomen-Nomen-Modifikation, Genitive, Negation, Nomen-Verb-Kontraktionen und Imperative beinhalten.[44] Wie unsere Untersuchung zeigen wird, konnten diese Ergebnisse nur zum Teil reproduziert werden.

3.1 Die Bestandteile des XTAG-Systems

Die wichtigsten Bestandteile von XTAG sind die morphologische Analyse, die Syntaxdatenbank und die Datenbank der TAG-Bäume. Die morphologische Analyse kann auf eine Datenbank von ca. 317'000 flektierten Wortformen zurückgreifen, die von 90'000

Stämmen abgeleitet sind. Die Syntaxdatenbank hat mehr als 30'000 Einträge. Davon enthält jede die unflektierte Form des Wortes, sein POS (*part of speech*) und eine Liste von Bäumen oder Baumfamilien, die mit dem Wort verbunden sind. Das Herzstück von XTAG ist die Baumdatenbank. In ihr sind die Elementarbäume der TAG enthalten. Die Baumdatenbank der englischen Grammatik beinhaltet 1004 Bäume, die in 53 Baumfamilien und 221 individuelle Bäume unterteilt sind. Ein lexikalisches Element, das nicht in den Datenbanken enthalten ist, erhält default-mässig eine Auswahl von Bäumen und Merkmalen zugeteilt.

3.1.1 Baumfamilien

Die Baumfamilien gruppieren eine Anzahl von Bäumen, die zum selben Subkategorisierungsrahmen gehören und somit starke syntaktische Ähnlichkeiten aufweisen. So rufen zum Beispiel alle transitiven Verben die Familie der transitiven Verben auf. Eine Baumfamilie ist also eine Gruppe von Bäumen, die aufgrund gewisser syntaktischer Gemeinsamkeiten „verwandt“ sind.

3.1.2 Redundanzen

Die Entwickler weisen selbst auf das Problem der Redundanzen hin: „Generell entsteht bei der Lexikalisierung einer TAG Redundanz, weil die selben Bäume, modulo ihre Ankerbeschriftungen mit vielen verschiedenen lexikalischen Wörtern verbunden werden können. [...] Eine weitere Quelle für Redundanzen ist aber auch das wiederholte Anwenden der Substrukturen von Bäumen in vielen verschiedenen Baummasken [*tree templates*]. Zum Beispiel beinhalten die meisten Satz-Baummasken ein strukturelles Fragment, das der Phrasenstrukturregel S → NP VP entspricht“.[45]

[44] XTAG Research Group 1999, S. 19
[45] Doran et al. 1996, S. 32 (übersetzt)

3.2 Die Ausgabeform von XTAG

Wie bereits erwähnt wurde, liefert der XTAG-Parser jeweils zwei Bäume pro Analyse, wobei der erste eine Repräsentation der Verknüpfung der Elementarbäume zeigt, genannt Ableitungsbaum, und der zweite den daraus resultierenden eigentlichen Syntaxbaum, genannt abgeleiteter Baum (siehe auch Abschnitt 2.4.4). Im Folgenden wird versucht, die beiden Repräsentationsformen und insbesondere den Übergang vom Ableitungsbaum zum abgeleiteten Baum zu erklären. Dies stellt eine Voraussetzung dar für die eigentliche Untersuchung der Beispielsätze, bei der die Syntaxbäume im Zentrum stehen werden und die Ableitungsbäume nur am Rande zur Erklärung beigezogen werden. Das Konzept wird anhand eines einfachen Beispielsatzes erklärt.[46]

Beispielsatz: *The man sleeps.*

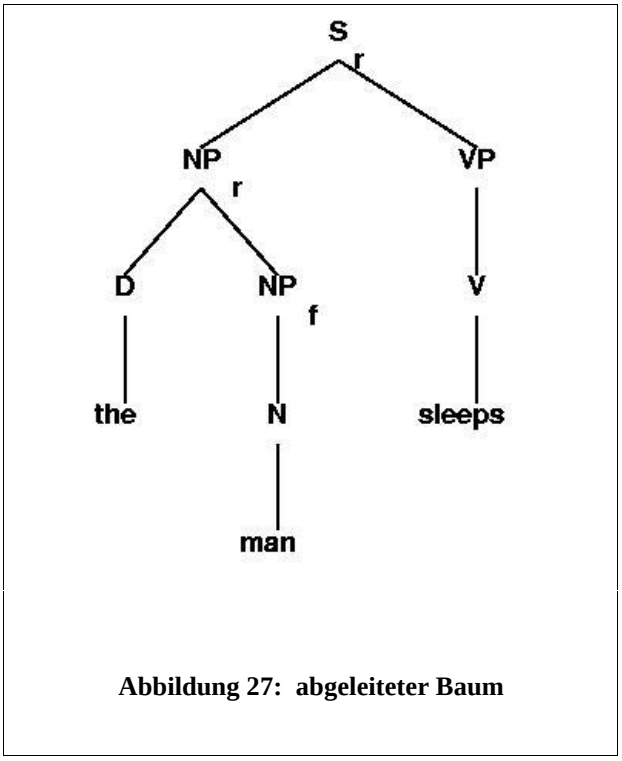


Abbildung 27: abgeleiteter Baum

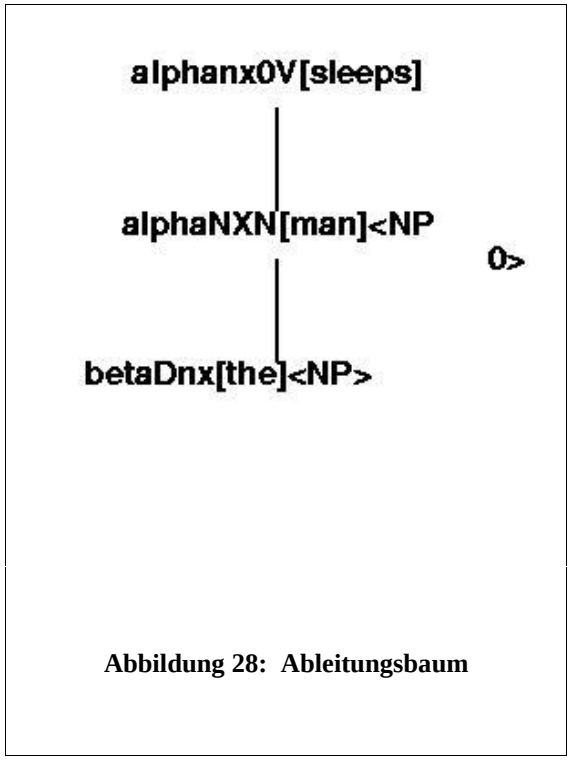


Abbildung 28: Ableitungsbaum

Der Ableitungsbaum in Abbildung 28 zeigt, dass für den Satz „The man sleeps.“ drei verschiedene Elementarbäume aus der Baumdatenbank von XTAG verwendet wurden. Im Gegensatz zum theoretischen Teil (Kapitel 2) sind hier die Bezeichnungen für α und β

[46] Bei der Eingabe von Sätzen werden Satzendpunkte ebenso ignoriert wie Grossschreibung (alles wird in

ausgeschrieben (alpha, beta). Nach wie vor stellt α einen Initialbaum dar, während β einen Auxiliarbaum bezeichnet. Die Unterscheidung zwischen durchgezogenen und gestrichelten Linien zur Identifikation, ob eine Beziehung der Art Adjunktion respektive Substitution besteht, wird nicht angezeigt. Dies stellt uns aber nicht vor unlösbare Probleme, da wir die Art der Beziehung auch auf andere Weise eruieren können.

3.2.1 Die Beziehungen in einem XTAG-Ableitungsbaum

Aus der Definition des TAG-Formalismus wissen wir, dass nur β-Bäume (Auxiliarbäume) adjungiert werden dürfen, nicht aber α-Bäume (Initialbäume). Somit ist der am obersten α-Baum (alphanx0V[sleeps]) in Abbildung 28 angehängte α-Baum (alphaNXN[man]) durch Substitution an diese Position gelangt. Für β-Bäume gilt entsprechend: Wenn an einem Knoten substituiert werden kann, so dürfen nur solche Bäume angehängt werden, die von α-Bäumen abgeleitet sind, i.e. es dürfen keine β-Bäume substituiert werden, was bedeutet, dass β-Bäume nur durch Adjunktion in einen Ableitungsbaum eingehängt werden können, in Abbildung 28 betrifft dies den β-Baum betaDnx[the].[47]

3.2.2 Die Bezeichnungen im XTAG-Ableitungsbaum

Abgesehen von den Bezeichnungen alpha, beta und der Baumstruktur selbst wirken die Benennungen im Ableitungsbaum in Abbildung 28 etwas kryptisch. Die XTAG Grammatikbeschreibung erklärt diese ausführlich. Mit Hilfe dieser Angaben lässt sich auch der Übergang zur Notation im abgeleiteten Syntaxbaum (Abbildung 27) nachvollziehen (siehe auch XTAG Research Group 1999, S. 266ff. für detaillierte Informationen).

alphanx0V[sleeps]	alphanxOV beschreibt den Initialbaum für intransitive Aussagesätze.

Kleinschreibung umgeformt). Dies gilt nicht für andere Interpunktionszeichen.

[47] Joshi / Schabes 1997, S.70

	Das Verb V ist dabei der Anker des Baumes [sleeps], es ist deshalb grossgeschrieben. nx0 bedeutet, dass dieser Baum eine NP (repräsentiert durch 'nx') in Subjektsposition (angezeigt durch die Markierung '0' verlangt. ***'0' bezeichnet die Subjektsposition, '1' das erste Argument (d.h. das direkte Objekt), '2' das zweite Argument (d.h. das indirekte Objekt).***
alphaNXN[man] <NP_0>	In diesem Baum bezeichnet NX den Wurzelknoten (eine NP), das nachfolgende N bezeichnet den Anker des Baumes (das Nomen). Ausserdem zeigt <NP_0> die Benennung des im abgeleiteten Baum darüberliegenden Knotens an.
betaDnx[the]<NP>	D zeigt, dass der Anker dieses Baumes der Determiner *the* ist. Das rechts von D stehende nx zeigt an, dass sowohl der Wurzel- als auch der Fussknoten des Baumes (die ja bei β-Bäumen identisch sein müssen) eine NP ist. Der Determiner kommt dabei auf der linken Seite der NP zu stehen. <NP> zeigt die Benennung des im abgeleiteten Baum darüberliegenden Knotens an.

Diesen drei Bäumen entsprechen die folgenden drei grafisch dargestellten Elementarbäume.

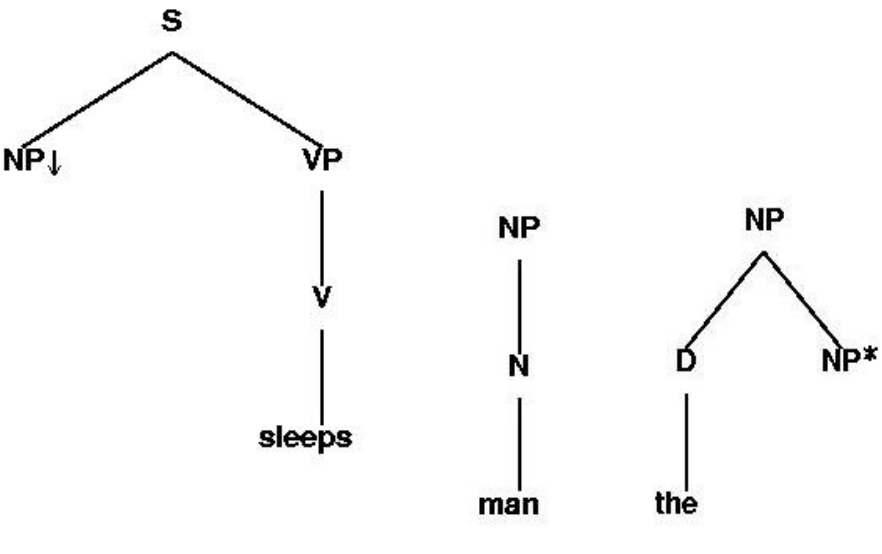


Abbildung 29: Elementarbäume für *The man sleeps.*

In Abbildung 29 wird erkennbar, welche Elementarbäume dem abgeleiteten Baum in Abbildung 27 zugrunde liegen und wie dieser zusammengesetzt ist.

Nachdem nun die Arbeitsweise des Programms und seine Notation erklärt worden sind, können wir uns im nächsten Teil der Auswertung von Beispielsätzen zuwenden.

3.3 Die Beispielsätze

Im Rahmen dieses Seminars, in dem Implementationen unterschiedlicher Grammatiktheorien auf ihre Funktion und vor allem auch auf ihre Tauglichkeit hin untersucht werden, steht ein Katalog von 14 englischen Testsätzen zur Verfügung, die mit dem jeweiligen Programm getestet werden sollen. Die Sätze sind nach den Anforderungen, die sie ans System stellen, in Gruppen gegliedert. Sie lauten:

Satztypen (Fragesätze [y/n, wh], Aussagesätze, Nebensätze)

1. Do dogs bite?
2. Who bites postmen?
3. Who do dogs bite?
4. Who does this dog belong to?
5. Dogs that bark bite.
6. If a dog barks it bites.

Unterscheidung Komplement/Adjunkt

1. The student of English with long hair bites the dog.
2. The postman gives a bone to the dog every day.
3. The dog bites the postman on the street.

Raising-Konstruktionen, Infinitive, Hilfsverben

1. The dog seems to bite.
2. The dog wants the postman to give him a bone.
3. The postman promises the dog to bring a bone.

4. The dog has already eaten the bone.
5. The dog must have eaten the bone.

Wir erinnern uns vom Anfang dieses Kapitels, dass laut Aussage der Entwickler XTAG von diesen Themen ausdrücklich die Behandlung von wh-Fragesätzen, Inversion, Raising-Konstruktionen, Infinitiven und Hilfsverben behandeln kann. Inwiefern unsere Untersuchung dies bestätigen kann, werden die folgenden Abschnitte zeigen.[48]

3.3.1 Satztypen (Fragesätze [y/n, wh], Aussagesätze, Nebensätze)

3.3.1.1 Do dogs bite?

Für diesen Satz findet XTAG neun verschiedene Syntaxstrukturen. Es ist aber keine korrekte Analyse des Satzes dabei. Zur Illustration sei hier eine der Lösungen wiedergegeben.

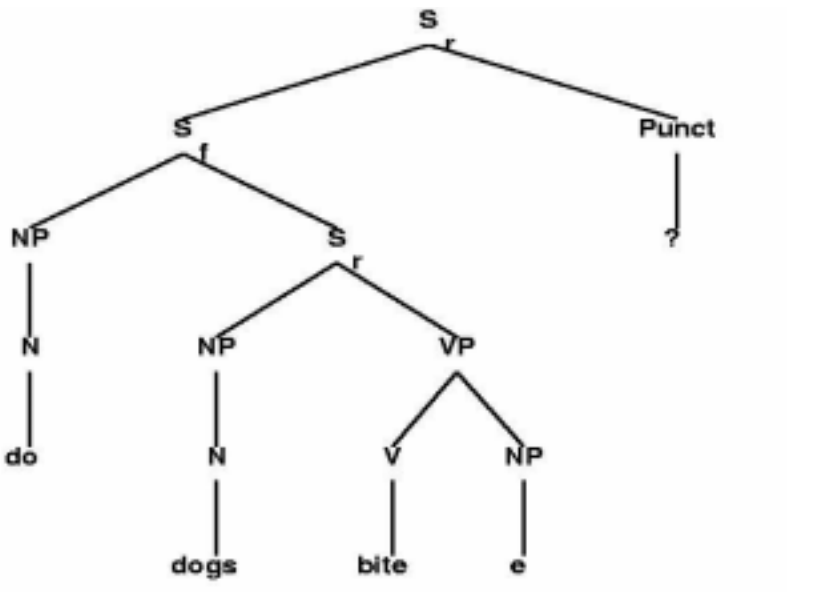


Abbildung 30: ***Do dogs bite?***

[48] Wir werden hier jeweils nur den Syntaxbaum bzw. abgeleiteten Baum zeigen und den Ableitungsbaum des Satzes falls nötig erwähnen.

Wie zu sehen ist, wurde *do* fälschlicherweise als NP analysiert. Wir führen dieses Beispiel gerade deswegen an, um auf ein paar Probleme hinzuweisen. Zum Bespiel zeigt sich schon hier die grosse NP-Lastigkeit von XTAG, der wir auch bei den folgenden Sätzen immer wieder begegnen werden. Was hier speziell erstaunt, ist, dass XTAG anscheinend nicht in der Lage ist, *do*-Fragesätze aufzulösen. Dies legt auch die Bedienungsanleitung nahe, die das Thema nicht speziell behandelt. Anders als bei anderen, im Seminar schon behandelten Anwendungen es ist hier nicht der fehlende Lexikoneintrag für *bite* ohne direktes Objekt, der eine korrekte Lösung verhindert[49]. Vielmehr ist das Programm schlicht nicht in der Lage, *do* und *bite* als Bestandteile eines Fragesatzes zu identifizieren. Stattdessen analysiert es nacheinander *do*, *dogs*, *bite*, *do dogs*, *dogs bite* als NPs (in den hier nicht gezeigten Lösungen). Dabei taucht schon ein erstes Mal die Auswirkung der immer akzeptierten Nomen-Nomen-Koordinationen auf: Da fast jedes Verb auch als Nomen im Lexikon erfasst zu sein scheint, werden sehr häufig Verb und Nomen zu einer NP zusammengefasst, die aus zwei Nomen besteht. Noch weiter verstärkt wird dieser Effekt durch die Tatsache, dass fast überall leere Kategorien erlaubt sind: XTAG stört sich nicht daran, wenn ein Satz nur aus NPs und leeren VPs besteht[50].

3.3.1.2 Who bites postmen?

XTAG findet zwei verschiedene Ableitungsbäume, sie resultieren aber in denselbem abgeleiteten Baum.

[49] Dieses Problem wird durch das Leerlassen der NP-Position umgangen.
[50] siehe dazu Abbildung 39

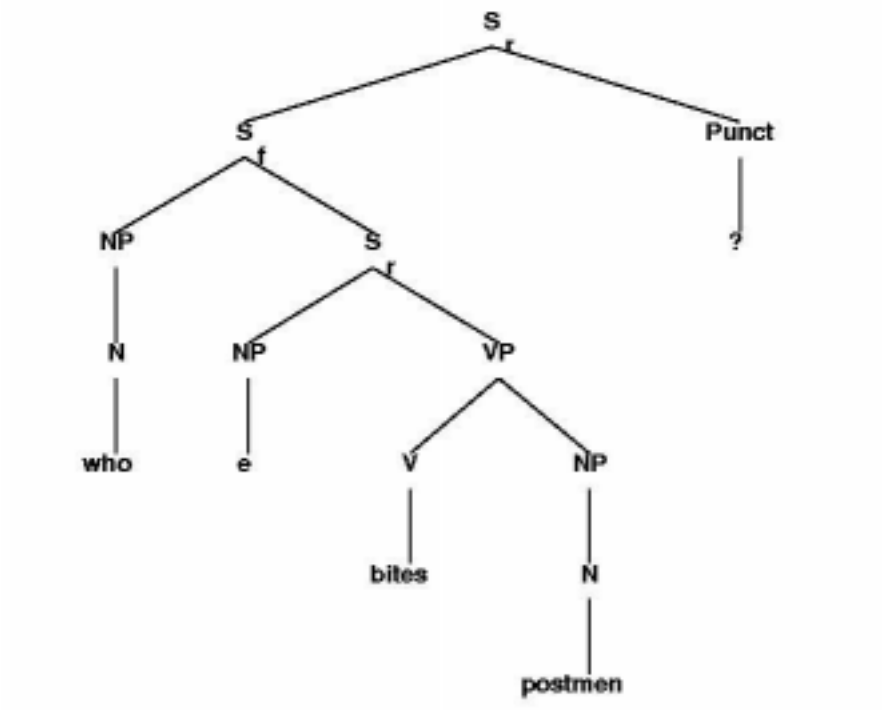


Abbildung 31: ***Who bites postmen?***

Diese Lösung kann als korrekt erachtet werden. Besonders interessant ist die Auflösung des wh-Fragepronomens. Die eigentliche Subjektsposition vor dem Verb ist leer (e), während das Fragepronomen in einem übergeordneten S die Subjekts-NP ersetzt.

3.3.1.3 Who do dogs bite?

XTAG findet eine einzige Lösung für diesen Satz, diese ist allerdings falsch.

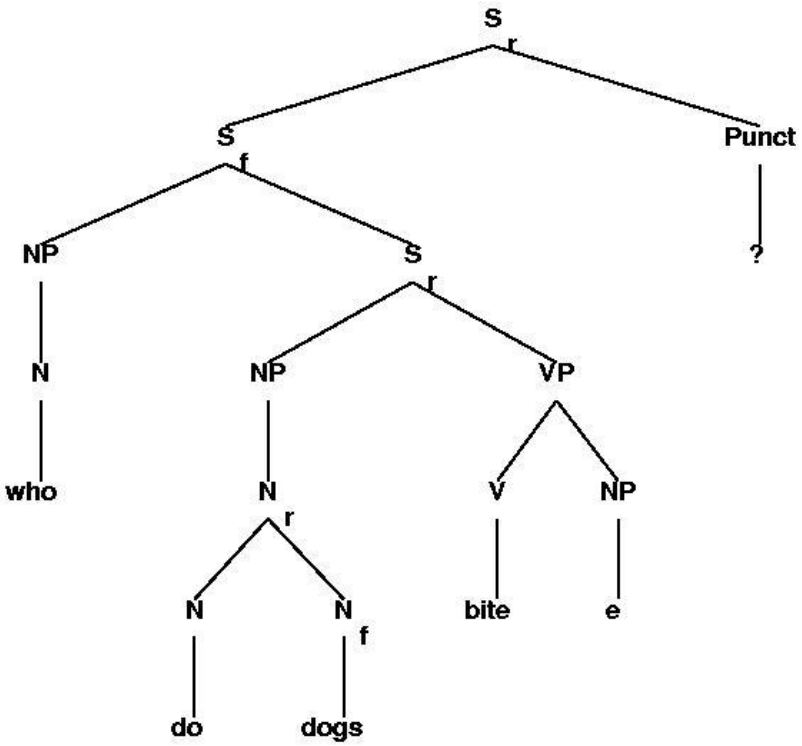


Abbildung 32: ***Who do dogs bite?***

Wieder zeigt sich die Unfähigkeit von XTAG, *do* zusammen mit *bite* als Verbalkonstruktion zu erkennen. Es ist jedoch fraglich, ob der Satz nicht schon auf einer anderen Ebene zuvor falsch analysiert wurde. Die Analyse von *do dogs* als Nomen-Nomen-Koordination legt nahe, dass bei der Erfassung von wh-Konstruktionen in der Grammatik wichtige Elementarbäume ausser Acht gelassen wurden und nun in der Datenbank fehlen. Es ist auch fraglich, bis zu welchem Grad der Satz als wh-Konstruktion erkannt wurde, zumal wir bei einer solchen - wie wir im vorangehenden Beispiel gesehen haben - eine leere NP direkt vor dem Verb erwarten dürften.

3.3.1.4 Who does this dog belong to?

Für diesen Satz findet XTAG keine Lösung.

3.3.1.5 Dogs that bark bite.

XTAG findet 3 identische Lösungen für diesen Satz.

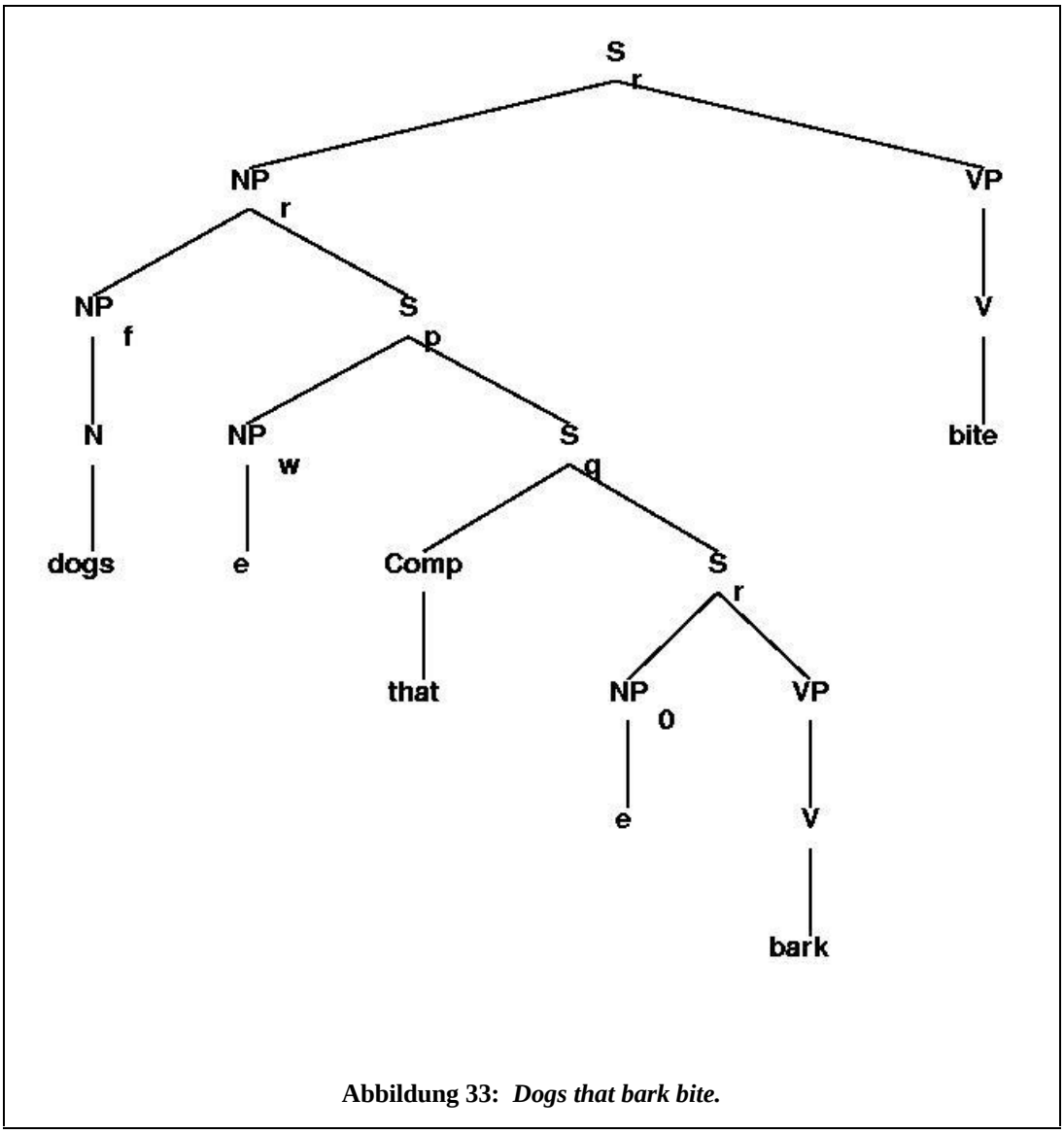


Abbildung 33: ***Dogs that bark bite.***

Die Lösung ist korrekt. Die beiden leeren NPs entstehen, da die Entwickler von XTAG - wie sie in der Bedienungsanleitung erklären - für alle Arten von Relativsätzen zwei unabhängige Beziehungen annehmen, d.h., dass die Kopf-NP nicht direkt mit dem extrahierten Element im Relativsatz zusammenhängt[51]. Dies kommt auch hier zum Tragen, obwohl die Beziehung zwischen Relativsatz und Subjekt direkter ist.

[51] Die XTAG Research Group erklärt dieses System anhand des Satzes: *the person whose mother likes Chris. the person* wird nicht als NP des Relativsatzes (*whose mother ε likes Chris*) interpretiert, dem Relativsatz fehlt

3.3.1.6 If a dog barks it bites.

XTAG findet 8 verschiedene Syntaxanalysen für diesen Satz, wovon die vierte Lösung noch am ehesten als richtig eingestuft werden kann, auch wenn die Behandlung von *if* als Präposition etwas seltsam wirkt.

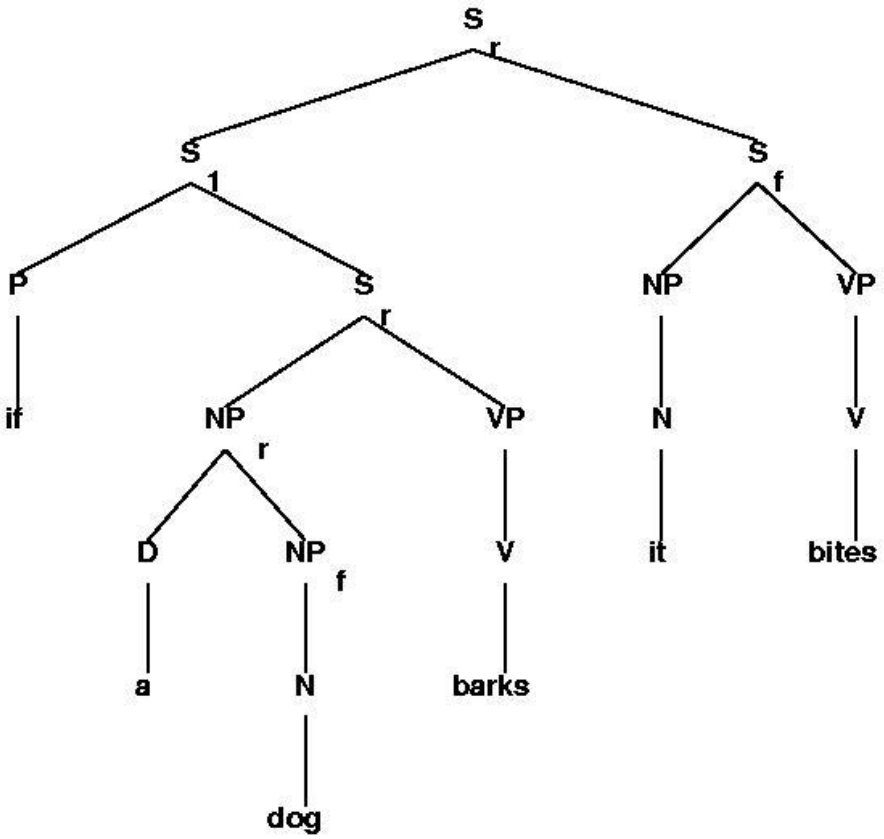


Abbildung 34: ***If a dog barks it bites.***

Es lässt sich kaum feststellen, ob es sich bei dieser Satzkonstruktion um einen für solche Fälle definierten Spezialfall der Anwendung einer Präposition handelt (immerhin ist doch *if* als Präposition im Lexikon erfasst) oder ob es sich um ein Zufallsprodukt aus verschiedenen, eigentlich für andere Konstruktionen bestimmten Bäumen handelt. Wenn der vorliegende

eine eigentliche NP (ersetzt durch leere NP) (XTAG Research Group 1999, S. 131ff.).

Baum auch von seiner Struktur her einsichtig erscheint, so bleibt doch anzumerken, dass mit drei dem eigentlichen Satz S untergeordneten S nicht sehr viel Information gewonnen wurde.

3.3.2 Unterscheidung Komplement - Adjunkt

3.3.2.1 The student of English with long hair bites the dog.

Für diesen Satz findet XTAG die erstaunlich hohe Anzahl von 351 Lösungen. Davon entsprechen 6 aufeinanderfolgende identische Lösungen der von uns gesuchten Syntax-struktur.

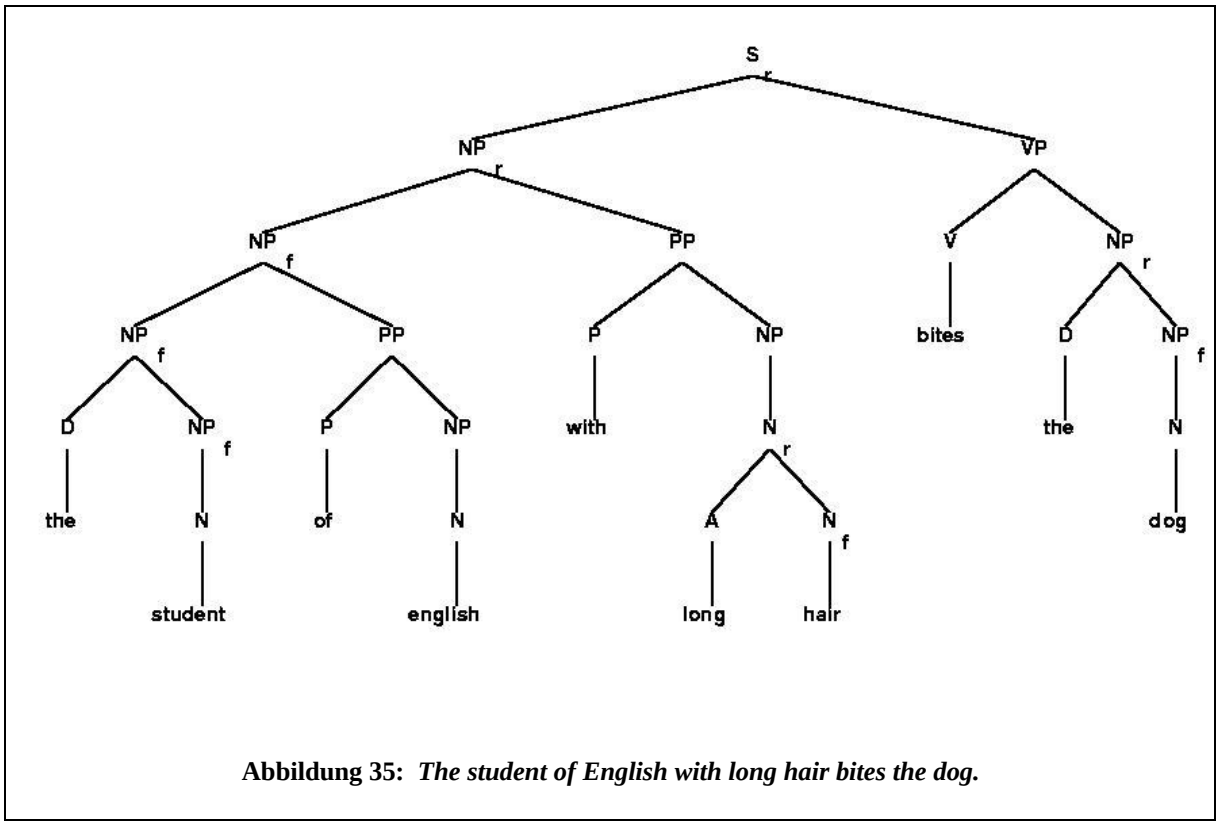


Abbildung 35: ***The student of English with long hair bites the dog.***

Das Hauptproblem bei diesem Satz liegt sicherlich in der unüberschaubar grossen Anzahl von Parsing-Resultaten. Es gestaltete sich recht schwierig, die richtigen Lösungen überhaupt zu finden. Die fehlerhaften Lösungen sind von so zahlreicher Art, dass wir hier nur auf einige hauptsächlichen Problempunkte hinweisen wollen. Sicher hat sich auch hier die Fähigkeit von

XTAG, den Grossteil aller Wörter auch als Nomen erkennen zu können, verbunden mit seiner Vorliebe, N-N-Koordinationen zu bilden, nachteilig für ein einsichtiges Parsing-Resultat ausgewirkt. Hinzu kommt, dass XTAG an den verschiedensten Orten leere Kategorien akzeptiert bzw. einführt. Auch dies geschieht, ohne dass wir uns ein Bild über das zugrundeliegende System hätten machen können. Alle diese Möglichkeiten lassen die Anzahl Parsinganalysen mit zunehmender Satzlänge ins Unüberblickbare anwachsen. Da erscheinen die klassischen, von uns erwarteten falschen Lösungen, bei denen die PP *with long hair* die NP *English* modifiziert, nicht mehr als zentrales Problem der Analyse.

3.3.2.2 The postman gives a bone to the dog every day.

XTAG findet 216 Lösungen für diesen Satz. Keine einzige zeigt den korrekten Syntaxbaum. Trotzdem zeigen wir hier eine Lösung, bei der *every day* als Adjunkt erkannt wurde. Diese Lösung entspricht noch am ehesten dem von uns erwarteten Baum, auch wenn die VP nicht ganz richtig aufgelöst wurde.

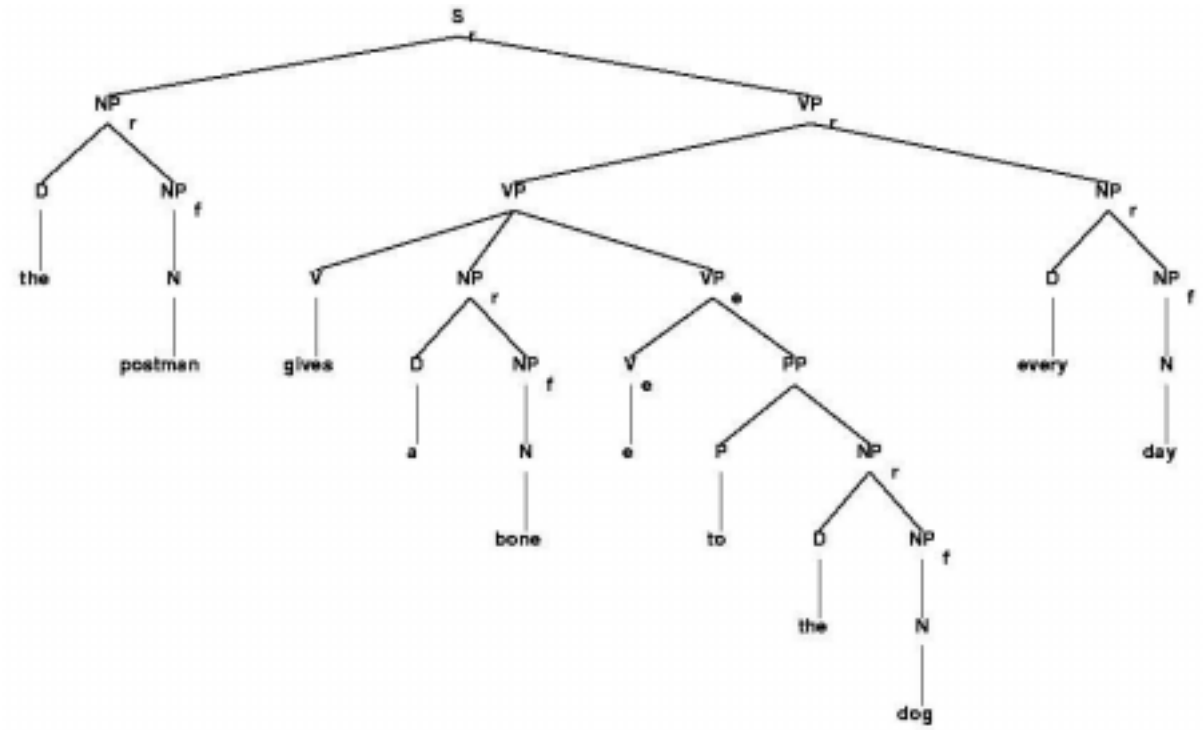


Abbildung 36: ***The postman gives a bone to the dog every day.***

Erstaunlich ist, dass offenbar *to give something to someone* in diesem Satz nicht richtig erkannt wird. Stattdessen erscheint an unerwarteter Stelle eine VP als Knoten in einer VP, in der dann das Verb fehlt, was die Auflösung des Satzes in dieser Weise überhaupt erst ermöglicht (d.h., hier führt die Freiheit, die XTAG gewährt, ob ein Präterminal auch ein lexikalisches Terminal haben muss, einmal mehr zu einer falschen Lösung).

3.3.2.3 The dog bites the postman on the street.

XTAG findet 90 Lösungen für diesen Satz, darunter sind auch beide möglichen korrekten Syntaxbäume, wobei beim hier zuerst gezeigten die PP *on the street* die VP *bites the postman* modifiziert, beim zweiten die PP lediglich die NP *the postman* modifiziert.

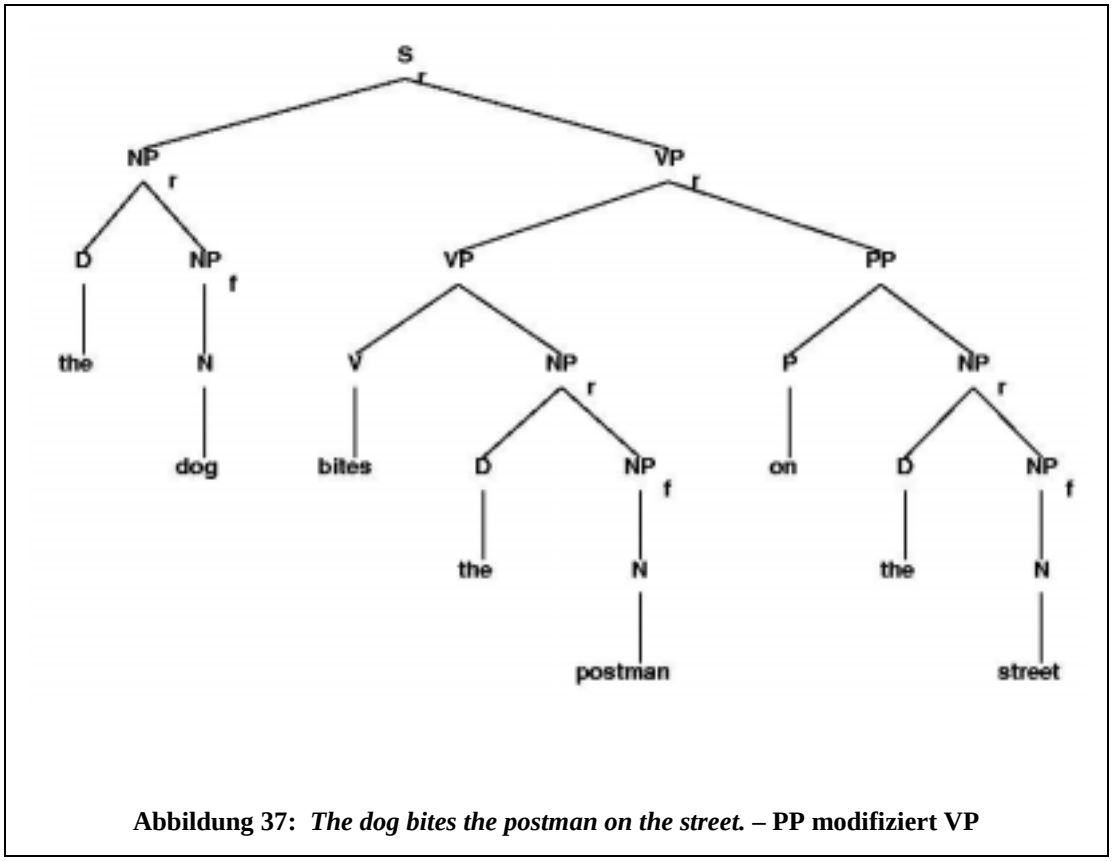


Abbildung 37: ***The dog bites the postman on the street.*** **– PP modifiziert VP**

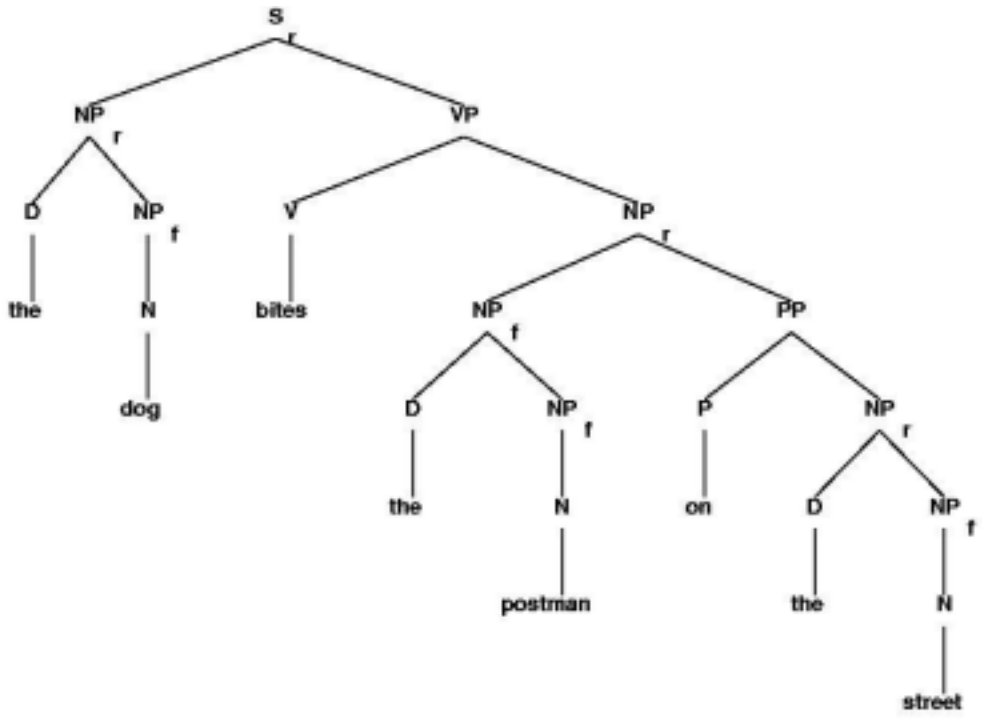


Abbildung 38: ***The dog bites the postman on the street.* – PP modifiziert NP**

Wiederum ist auch bei diesem Satz die Anzahl der Lösungen beachtlich hoch. Zudem verhält es sich in diesem Beispiel so, dass etliche Lösungen dieselbe Struktur ausgeben. Dies wirft natürlich die Frage auf, weshalb das System diese nicht aussortiert. Fairerweise sei gesagt, dass wir nicht mit dem ursprünglich konzipierten System arbeiten, da die Installation desselben - wie am Anfang des Kapitels 3 erwähnt - nicht oder zumindest äusserst schwierig zu bewerkstelligen war. Es ist also möglich, dass das ursprüngliche System identische Lösungen ausschliesst.

3.3.3 Raising-Konstruktionen, Infinitive, Hilfsverben

3.3.3.1 The dog seems to bite.

XTAG findet keine Lösung zu diesem Satz.

3.3.3.2 The dog wants the postman to give him a bone.

XTAG findet keine Lösung zu diesem Satz.

Aufgrund der vorangehenden beiden Beispiele ist zu vermuten, dass XTAG grosse Mühe hat mit Raising-Konstruktionen, obwohl diese in der Grammatikbeschreibung speziell erwähnt und erklärt werden.[52]

3.3.3.3 The postman promises the dog to bring a bone.

XTAG findet 174 Lösungen für diesen Satz. Sie sind jedoch alle falsch. Das zeigt drastisch, wie das System zwar viel zu viele Antworten liefert, wie wir das schon in anderen Beispielen gesehen haben, während es gleichzeitig nicht die richtige Lösung findet. Es hat also, überspitzt formuliert, zu viele Regeln, die sich auf diesen Satz anwenden lassen, aber leider sind es die falschen.

3.3.3.4 The dog has already eaten the bone.

XTAG findet keine Lösung zu diesem Satz.

3.3.3.5 The dog must have eaten the bone.

XTAG findet keine Lösung zu diesem Satz.

Die letzten beiden Beispiele weisen auf eine Schwäche hin, die wir mit diversen anderen Sätzen auch identifizieren konnten: XTAG kann nicht mit Hilfsverben umgehen, obwohl die Entwickler gerade damit ihr System preisen.

3.4 Die Hauptprobleme in XTAG

Die vorliegende Auswertung des XTAG-Parsers zeigt zwei Gruppen von Problemen, welche die hohe Fehlerquote von XTAG beeinflussen. Die Erste führt generell zu einer Übergenerierung, die Zweite zu einer Untergenerierung.

Die erste dieser Fehlerquellen liegt zweifelsohne in der Allianz von drei problematischen Eigenschaften: 1. XTAG ist nomenlastig. Am deutlichsten - und verheerendsten - wirkt sich das in den schon oft erwähnten N-N-Koordinationen aus, d.h. zwei Nomen werden zu einer Nominalphrase zusammengefasst. 2. Wie es scheint, ist der grösste Teil aller Verben und Adjektive, aber z.B. ebenso der Determiner *a*, in der Datenbank auch als Nomen erfasst (was auch wieder XTAGs Vorliebe für Nomen zeigt). 3. XTAG erlaubt scheinbar ohne Begrenzung Lücken in den Terminalen. Während vielleicht eine einzelne dieser Eigenschaften noch nicht allzu viel Schaden anrichten würde, lassen sie zusammen die Anzahl Parsinganalysen explodieren. Wenn beispielsweise ein Verb auch als Nomen erfasst in der Datenbank ist (2. Eigenschaft), so wird es mit dem vorangehenden Nomen als NP der Form N-N erkannt (1. Eigenschaft). Die nachfolgende Lücke in der VP wird aber nicht gefüllt (3. Eigenschaft). Zur Veranschaulichung, dass dies selbst bei harmlosen Sätzen geschieht, führen wir in Abbildung 39 ein Beispiel an.

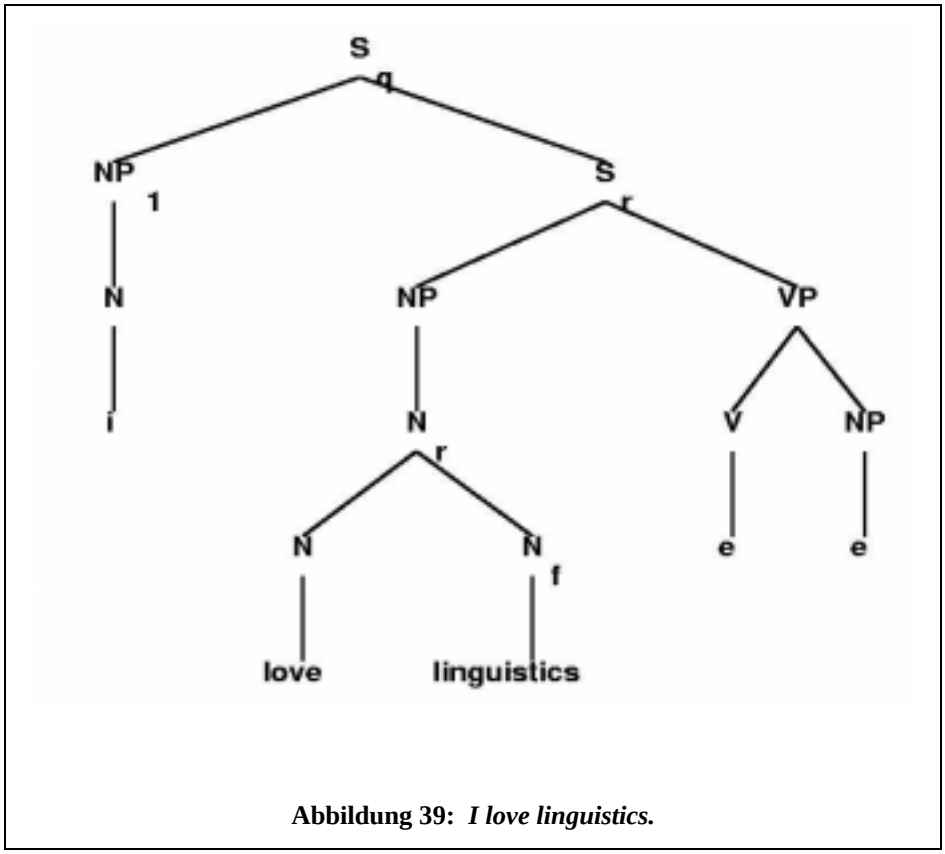


Abbildung 39: ***I love linguistics.***

[52] XTAG Research Group 1999, S. 98ff.

Der hier repräsentierte Baum ist zwar bei weitem nicht die einzige Lösung, die der XTAG-Parser liefert (unter anderem gibt das Programm auch den korrekten Syntaxbaum aus), er bestätigt aber das eben beschriebene Problem.[53]

Die zweite Fehlerquelle betrifft verschiedene, nicht oder nicht korrekt erfasste Konstruktionen. Diese umfassen Fragen mit *do*, verschiedene Hilfsverb-, Infinitiv- und Raising-Konstruktionen. Die Folge davon ist eine Untergenerierung; die relevanten Syntaxbäume werden nicht gefunden. Dies ist insbesondere dadurch störend, dass XTAG - laut der Grammatikbeschreibung - diese Konstruktionen angemessen behandeln können sollte.

[53] Ausserdem spielen bei der Übergenerierung natürlich auch die Redundanzen eine wichtige Rolle (vgl. 3.1.2).

4. Konklusion

Auf den ersten Blick scheinen die TAGs eine ganze Reihe von Vorteilen gegenüber anderen Grammatikformalismen zu haben. Hier sind einige davon:

- TAGs eignen sich dazu, natürlichsprachliche Strukturen auf einfache und intuitive Weise einzufangen.
- Selbst idiomatische Ausdrücke, Nebensätze, Koordinationen, Adjunkte, Fernabhängigkeiten, gekreuzte Abhängigkeiten, Extraposition, Topikalisierungen. können häufig durch wenige Grammatikregeln bzw. wenige Elementarbäume beschrieben werden.
- Durch die Verwendung von Bäumen als grundlegende Strukturen einer TAG werden komplexe strukturelle Zusammenhänge (im Gegensatz zu kontextfreien Phrasenstrukturregeln) in einem einzigen Elementarbaum dargestellt. Als Konsequenz davon entsteht ein erweiterter Lokalitätsbereich.
- Lokale Beschränkungen für die Adjunktionsoperation sowie die Möglichkeit der Multikomponenten-Adjunktion machen eine TAG schwach kontextsensitiv.

Nach einem genaueren Blick auf die Testergebnisse des XTAG-Parsers entsteht der Eindruck, dass bei weitem nicht alle linguistischen Phänomene, von denen behauptet wird, die XTAG sei in der Lage, diese besonders elegant zu behandeln, angemessen verarbeitet werden können. Erschwerend kommt die Tendenz zu Übergenerierung des Systems hinzu. Ob dies nun am TAG-Formalismus oder an der Umsetzung der XTAG Research Group liegt, können wir hier nicht beurteilen.

5. Literaturverzeichnis

[Buschauer et al. 1991] Buschauer, Béla / Harbusch, Karin / Poller, Peter / Schauder, Anne: Tree Adjoining Grammars mit Unifikation. Technical Memo. Kaiserslautern / Saarbrücken 1991 (= DFKI Publikationen, TM-91-10)

[Harbusch 1997] Harbusch, Karin: The Relation between Tree-Adjoining Grammars and Constraint Dependency Grammars. In: Fifth Meeting on the Mathematics of Language. Schloss Dagstuhl (Saarbrücken) 1997. S. 38-45. (= MOL5)

[Doran et al. 1996] Doran, Christine / Hockey, Beth / Hopely, Philip / Rosenzweig, Joseph / Sarkar, Annop / Srinivas, B. / Xia, Sei: Maintaining the Forest and Burning out the Underbrush in XTAG. Philadelphia, PA 1996. S. 30-37. http://www.cis.upenn.edu:80/~xtag

[Joshi et al. 1975] Joshi, Aravind K. / Levy, Leon S. / Takahashi, Masako: Tree Adjunct Grammars. Journal of Computer and System Sciences. New York / London 1975. Volume 10:1. S. 136-163.

[Joshi 1987] Joshi, Aravind K. (1987): An Introduction to Tree Adjoining Grammars. In: Manaster-Ramer, Alexis (Hg.): Mathematics of Language. Amsterdam/Philadelphia 1987. S. 87-114.

[Joshi / Schabes 1997] Joshi, Aravind K. / Schabes, Yves: Tree Adjoining Grammars. In: Rozenberg, Grzegorz / Salomaa, Arto (Hgg.): Handbook of Formal Languages. Volume 3. Beyond Words. Berlin / Heidelberg 1997. S. 69-123.

[Kroch 1987] Kroch, Anthony S.: Unbounded Dependencies and Subjacency in a Tree Adjoining Grammar. In: Manaster-Ramer, Alexis (Hg.): Mathematics of Language. Amsterdam / Philadelphia 1987. S. 143-172.

[Kroch / Joshi 1987] Kroch, Anthony S. / Joshi, Aravind K.: Analyzing Extraposition in a Tree Adjoining Grammar. In: Huck, Geoffrey J. (Hg.): Discontinuous Constituency. Orlando, FL 1987. S. 107-149. (= Syntax and Semantics, Volume 20)

[Paroubek / Schabes 1997] Paroubek, Patrick / Schabes, Yves und XTAG Research Group: XTAG User Manual. Version 1.0. An X Window Graphical Interface Tool for Manipulation of Tree-Adjoining Grammars. Philadelphia 1997. ftp://ftp.cis.upenn.edu/pub/xtag/papers/user-manual.ps.Z

[Vijay-Shanker et al. 1987] Vijay-Shanker, K. / Weir, David J. / Joshi, Aravind K.: On the Progression from Context-Free to Tree Adjoining Languages. In: Manaster-Ramer, Alexis (Hg.): Mathematics of Language. Amsterdam / Philadelphia 1987. S. 391-401.

[XTAG Research Group 1999] The XTAG Research Group: A Lexicalized Tree Adjoining Grammar for English. Philadelphia, PA, Institute for Research in Cognitive Science, 15. Januar 1999. http://www.cis.upenn.edu/~xtag