Universität Zürich
Computerlinguistik
Seminar "Syntaxtheorien und computerlinguistische Praxis"
Sommersemester 2000

Einführung in die (funktionale) Dependenzgrammatik anhand des FDG-Parsers für das Englische

Regula Heckmann
Allmendstrasse 13
8952 Schlieren
Phone/Ans./Fax: 01 731 07 28
e-mail: heckmann.regula@bluewin.ch

Inhaltsverzeichnis:

0. Einleitung

Die folgenden Ausführungen sind in zwei grosse Teile unterteilt. Im ersten Kapitel wird Grundsätzliches zur Dependenzgrammatik vorgestellt; der zweite Teil befasst sich mit einer ihrer Spezifikationen, der funktionalen Dependenzgrammatik, und deren Anwendung im FDG-(functional dependency grammar)Parser von Tapanainen und Järvi.

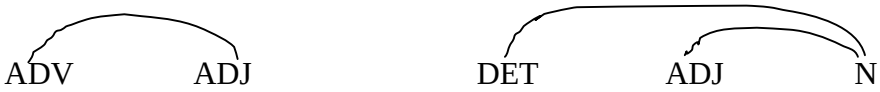
1. Was ist Dependenzgrammatik? Grundlegende Konzepte und das klassische Modell

Die folgenden Ausführungen basieren hauptsächlich auf dem Aufsatz „Dependency Syntax“ und dem Buch „Deutsche Syntax dependentiell“, beide von Hans Jürgen Heringer.

1.1. Konstituenz versus Dependenz

In der traditionellen europäischen Syntax-Forschung unterscheidet man zwei grundsätzliche Methoden: Kategorisierung und Relationalisierung. Kategorisierung bringt Elemente mit analogem syntaktischen Verhalten zusammen, Relationalisierung legt Relationen zwischen Elementen fest. Zwei Teile können zusammen gehören, weil beide Teile eines grösseren Ganzen sind. Sie können aber auch zusammen gehören, weil ein Teil vom anderen Teil abhängt. Es folgen daraus zwei unterschiedliche Prinzipien: Konstituenz und Dependenz.
Dependenzgrammatiken (DG) beruhen auf Dependenzrelationen (Abhängigkeitsbeziehungen). Bsp. einer entsprechenden Regel ist z.B.: „Ein modifizierendes Adverb hängt ab vom modifizierenden Adjektiv“ oder „Determinativ und Adjektiv sind abhängig vom Nomen“ (Bsp. 1).

Bsp. 1



Eine Dependenzgrammatik beschreibt die Syntax natürlicher Sprachen in Form von Abhängigkeitsverhältnissen unter den Elementen eines Satzes. Sie beschreibt einzelne Wörter oder ihre Kategorien und repräsentiert Sätze relativ nahe an der Oberflächenstruktur. Es existieren keine phrasalen Konstituenten wie NP, VP etc.

Der Begriff der Dependenzgrammatik ist untrennbar verbunden mit dem französischen Sprachwissenschaftler Lucien Tesnière, der als Begründer der Dependenzgrammatik gilt. Er lieferte mit seinem Hauptwerk „Eléments de syntaxe structurale“ (posthum 1959 erschienen) eine umfassende Dependenzgrammatik mit Anspruch auf Universalität.
Im weiteren wird Tesnières Dependenzmodell vorgestellt.

1.2. Konnexion und Dependenz

Die Grundideen von Tesnière können in einigen wenigen Punkten zusammengefasst werden:

1. Ein Satz ist eine strukturierte Einheit, deren Elemente Wörter sind. Die Elemente im Satz sind nicht isoliert wie ein Lexikoneintrag zu verstehen, sondern sie sind durch mehrfache Verbindungen miteinander verknüpft.

2. Jeder Satz hat eine kohärente syntaktische Struktur. In der Einheit „Satz“ tritt jedes Element in Beziehung mit anderen Elementen, keines bleibt isoliert.

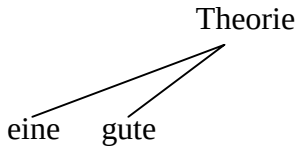
3. Die Struktur des Satzes ist nicht linear, sie geht über die eindimensionale Struktur der gesprochenen Zeichenkette hinaus.

4. Die Satzelemente sind mehr oder weniger stark miteinander verbunden. Wie die Elemente der gesprochenen Zeichenketten direkte oder indirekte Nachbarn sind, sind sie auch struktural mehr oder weniger stark miteinander verbunden. Es gibt direkte und indirekte Dependenzen (Abhängigkeiten). So ist in Bsp. 3 *Romam* eine indirekte Dependenz von *erat* und *molis* eine direkte Dependenz von *tantae*.

5. Die Satzstruktur ist determiniert durch die Konnexion. Die Konnexion ist die mentale Verbindung, der syntaktische Link. Konnexion ermöglicht es, einen kohärenten Gedanken durch einen Satz auszudrücken.

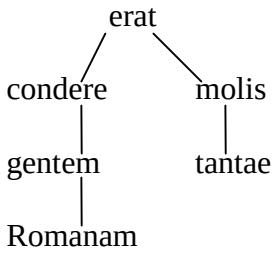
6. Konnexion erzeugt die Dependenzen der Elemente eines Satzes. Somit gibt es nur Konnexion zwischen Elementen, bei denen das eine Element vom anderen abhängt. Das abhängige Element wird Dependens (frz. subordonné) genannt, das übergeordnete Regens (frz. régissant). Die Dependenz wird durch Konnexions-Kanten repräsentiert:

Bsp. 2



7. Die syntaktische Struktur des Satzes ist hierarchisch. Dependenz ist eine asymmetrische Relation, die Hierarchien erzeugt. Da ein Element Regens und Dependens zugleich sein kann und da der ganze Satz kohärent ist, hat jeder Satz eine Hierarchie von Konnexionen. Die graphische Darstellung von Dependenzen ist das Stemma, der Dependenzbaum. Hier ein Satzstrukturbeispiel:

Bsp. 3



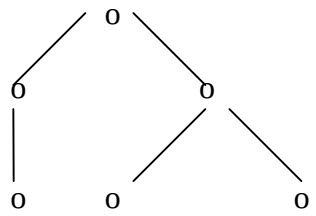
Das Zentrum des Satzes ist der Zentralknoten, von dem alle andern Elemente direkt oder indirekt regiert werden.

1.3. Das Stemma

Ein Stemma repräsentiert die Struktur eines Satzes. Seine Knoten repräsentieren Wörter, seine Kanten Konnexionen oder Dependenzen. Das Stemma unterliegt folgenden Bedingungen:

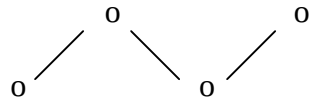
1. Ein Stemma ist ein gerichteter Graph, da die Dependenzen und somit auch die Kanten asymmetrisch sind. Die Hierarchie beginnt am Zentralknoten und läuft bis zum untersten Knoten durch, wobei jeweils der Knoten der höheren Ebene denjenigen der tiefer liegendenden Ebene regiert, der mit ihm direkt durch eine Kante verbunden ist.
2. Ein Stemma ist verzweigt, denn ein Knoten kann nur von einem einzigen anderen Knoten regiert werden, während er selber mehrere andere regieren kann. Folglich kann von jedem Knoten nur eine einzige Kante nach oben führen, aber mehrere nach unten. Folgendes Stemma ist also erlaubt:

   Bsp. 4



3. Das folgende Stemma ist nicht erlaubt, weil der Knoten rechts unten von zwei Knoten regiert wird:

   Bsp. 5



4. Ein Stemma hat keine Schleifen, Kreise oder mehrfachen Kanten.
5. Ein Stemma hat genau eine Wurzel (engl. root). D.h. es gibt einen Knoten, der von allen anderen tiefer liegenden Knoten erreicht werden kann. Dieser Knoten ist der Zentralknoten.
6. Ein Stemma ist zusammenhängend. Es enthält keine isolierten Knoten. Dies ist die graphische Repräsentation der Tatsache, dass ein Satz eine kohärente Struktur hat.

Da es in der Dependenzgrammatik um Dependenzrelationen zwischen Satzelementen geht, ist eine Einteilung der Elemente (und der Funktionen), eine Kategorisierung, unerlässlich.

1.4. Kategorien

Jeder Satz und jedes Stemma ist nur eine Instantiierung von generellen Strukturen. Die Repräsentation der Struktur eines einzelnen Satzes ist nur der erste Abstraktionsschritt in der Dependenzgrammatik. Der zweite Schritt besteht in der Verbindung der Lexeme an den Knoten mit lexikalischen Kategorien. Tesnière geht aus von der Unterscheidung zwischen statischen Kategorien und dynamischen Funktionen (entsprechend von statischer Syntax und funktionaler oder strukturaler Syntax). Die Funktionen (im Stemma: Kanten) sind der Nexus, der die kategorialen Elemente (im Stemma: Knoten) verbindet und den Satz lebendig macht. Die Kategorien sind ein Netzwerk, das das Gehirn produziert und der Wirklichkeit "überstülpt", um sie wahrzunehmen. Die Kategorien sind sprachabhängig und unterscheiden sich von einer Sprache zur anderen. Ihre Funktionen hingegen sind universal.
Kategorien stellen eine statische Ordnung dar, die im Gehirn des Sprechers existiert, bevor sie in der Performanz realisiert und dynamisiert wird. Die Kategorisierung ist ein Potential, das durch das Sprechen aktiviert werden muss.
Bei der Kategorisierung unterscheidet Tesnière zwei Wortgruppen: Voll-Wörter und Leer-Wörter. Voll-Wörter haben einen lexikalischen Gehalt und sind semantisch konstitutiv. Leer-Wörter haben

eine grammatische Funktion und sind nicht semantisch konstitutiv. Sie geben die Kategorien der Voll-Wörter an, ändern ihre grammatische Kategorie und regulieren so ihre Funktion im Satz.

Voll-Wörter werden aufgrund ihres semantischen Gehalts klassifiziert. Nach Tesnière gibt es vier kategorial-semantische Klassen:

- Substanz
- Ereignis
- Abstrakte Eigenschaften von Substanzen
- Abstrakte Eigenschaften von Ereignissen

Damit gibt es folgende vier grammatischen (Haupt-)Kategorien[1]:

- Substantive beschreiben Substanzen
- Verben beschreiben Ereignisse
- Adjektive beschreiben Eigenschaften von Substanzen
- Adverbien beschreiben Eigenschaften von Ereignissen

Diese grob anmutend Kategorisierung wird in Tesnières DS verfeinert und modifiziert durch die folgenden Beschreibungswerkzeuge:

- Funktionswörter stellen den Mörtel zwischen den groben Kategorien in den Sätzen dar.
- Funktionswörter können den kategorialen Wert von Wörtern ändern

Die Leer-Wörter werden gemäss ihrer grammatischen Funktion eingeteilt in:

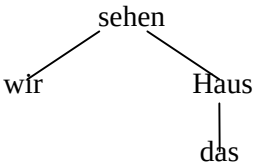
- Junktoren, die Nuclei der selben Art miteinander verbinden (und, oder, etc.)
- Translatoren, die eine Art Nuclei in eine andere Art umwandeln (mit, von, ob, etc.)

Andere Dependenzgrammatiker haben andere Kategorisierungen vorgeschlagen. Im zweiten Teil der Arbeit werden wir eine davon, die Kategorisierung (lexikalische und funktionale) der FDG genauer betrachten.

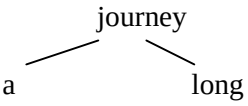
Durch Ersetzen der Wörter eines einzelnen konkreten Satzes durch ihre Wortklasse im Stemma erhält man das sog. virtuelle Stemma und damit eine Generalisierung, die die Struktur von ganzen Satztypen darstellt. Konkrete Sätze sind somit nur Instantiierungen von generelleren syntaktischen Strukturen. Ausgehend von der Kategorie des Zentralknotens kann man vier Typen unterscheiden:

Bsp. 6

1. Verbalausdruck Verb ist Zentralknoten *Wir sehen das Haus*

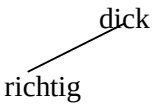


2. Nominalausdruck Nomen ist Zentralknoten *a long journey*

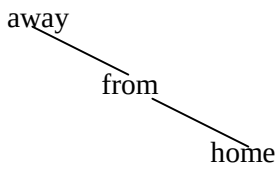


[1] Tesnière teilt diese vier Kategorien gemäss ihres Inhalts in Subkategorien auf.

3. Adjektivausdruck Adjektiv ist Zentralknoten *richtig dick*



4. Adverbialausdruck Adverb ist Zentralknoten *away from home*



Schon bei Tesnière hat das Verb die zentrale Rolle inne. Er verglich den Verbalsatz mit einem kleinen Schauspiel, in dessen Zentrum das Verb als Verkörperung eines Ereignisses steht. Die beteiligten Schauspieler kommen dazu und werden mit Komplementen versehen (frz. actants, dt. Aktanten). Weitere Details, vor allem Ort und Zeit einer Handlung, sind durch Adverbiale realisiert (frz. circonstants, dt. Circumstanten). Dieser Vergleich beruht auf der Beobachtung, dass in der Syntax das Verb als strukturales Zentrum die Satzstruktur definiert. Das Verb regiert die anderen Satzelemente. Es strukturiert den Satz durch seine Valenz.

1.5. Valenz

Die Eigenschaft von Verben, bestimmte Arten und Formen von Elementen im Satz zu verlangen oder zuzulassen, nennt man Valenz („Wertigkeit"). Das Verb eröffnet Leerstellen, die durch Komplemente (in Form von Argumenten) besetzt werden. Diese Verbeigenschaft wirkt sich aber nur auf bestimmte Teile des Satzes aus, nämlich auf die Aktanten (frz. actants, engl. complements). Im Unterschied dazu stehen die Circumstanten (frz. circonstants, engl. adjuncts), die weitere Details zur Handlung beschreiben und nicht verbspezifisch sind (siehe weiter unten).

Valenz beinhaltet:
- quantitative Valenz: Anzahl der Aktanten
- qualitative Valenz: Form der Aktanten
- selektionale Valenz: Bedeutung der Aktanten

Quantitative Valenz:

1. avalente (non-valente) Verben	Es regnete.	0V	v.a. meteorologische Verben (regnen, schneien, hageln)
2. monovalente	Der junge Mann hustet.	NV	1 obligatorischer Subjekt-Aktant (husten blinzeln, bellen)
3. bivalente (di-)	Sie sah den Apfel.	NVN	1 oblig. Subjekt-Aktant + 1 weiterer Aktant (glauben, suchen, öffnen)
4. trivalente	Er gab der Frau das Buch.	NVNN	1 oblig. Subjekt-Aktant + 2 weitere Aktanten (geben, anvertrauen, leihen)
5. tetravalente (sehr selten)	Er fuhr das Auto aus der Garage vor das Haus. NVNNN (wobei sich hier die Frage stellt, ob die PP ein Aktant ist)		

Die qualitative Valenz besteht darin, dass ein Verb als Lexem bestimmte Formen oder Anschlüsse des abhängigen Knotens verlangt, also z.B.
- bestimmte Kasus (Nominativ, Genitiv etc.) den Baum; them
- bestimmte Präpositionen sich wundern über, hoffen auf
- eine bestimmte Position (Stellung im Satz) die erste NP ist üblicherweise das Subjekt

Selektionale Valenz bezieht sich darauf, dass ein Verb nur bestimmte lexikalische Belegungen seiner Leerstellen zulässt; die Leerstellen sind semantisch beschränkt. Bsp. dafür sind die Merkmale HUMAN, ANIMATE, PROPOSITION. Dies sind nur grobe Bestimmungen, da jedes Verb prinzipiell eine individuelle Selektion hat, die von seiner Bedeutung abhängt.

Wie die Frage der Valenz in der praktischen Anwendung, im FDG-Parser, gelöst ist, wird später erforscht.

Direkt verbunden mit der Verbvalenz ist das Problem der Einteilung von Aktanten und Circumstanten.

1.6. Aktanten und Circumstanten

Die Valenztheorie basiert auf der Verschiedenheit von nominalen Satzteilen. Man unterscheidet Aktanten (A) und Circumstanten (C), und die Aktanten stehen in einem besonderen Verhältnis zum Zentralverb. Es geht um Fragen wie: Was ist das Spezielle an diesem Verhältnis? Wie unterscheiden sich Aktanten von Circumstanten?

Das Grundproblem besteht darin, dass es keine kategorialen oder formalen Unterschiede zwischen A und C gibt. So könnte, wird behauptet, eine Akkusativ-NP oder –PP beides, A oder C, sein:

Aktant:
They enjoyed the year.
Wir verzichten auf Verdacht.

Circumstant:
They married this year.
Sie verhafteten ihn auf Verdacht.

Es liegt nahe, dass A und C nicht kategoriale oder formale Phänomene sind, sondern funktionale und relationale. Immer wieder hat man versucht, diese intuitiv einleuchtende Unterscheidung zu erklären[2]. Tesnière verwendet ein semantisches Unterscheidungskriterium: A beschreibt Lebewesen und Dinge, die an Ereignissen teilnehmen können, C beschreiben die näheren Umstände. Auch dies lässt viele Fragen offen. Weitere vorgeschlagene Unterscheidungskriterien sind syntaktischer Art (Notwendigkeit,Verteilung) oder semantischer Art (logische Funktion, semantische Nähe). Bisher liegt aber weder ein zufriedenstellender Kriterienkatalog noch eine verlässliche Theorie vor[3].

Das Problem von Aktanten und Circumstanten wird im FDG-Parser durch die Funktionen der Kanten gelöst.

Neben der Konnexion, der Abhängigkeitsrelation zwischen zwei Elementen, wird auch die Relation der Junktion und diejenige der Translation berücksichtigt. Junktionen erfassen Koordinationen wie in *Syntax und Semantik sind wichtige Gebiete,* Translationen beschreiben die Tatsache, dass einige Funktionswörter (Translative) die syntaktische Kategorie eines Ausdrucks verändern und derart seine Konnexion zum nächsthöheren Knoten ermöglichen. So wird z.B. das Nomen *Markus* in *das Buch von Markus* erst mit Hilfe des Translativs *von* (funktional) zu einem "Adjektiv", das von *Buch* regiert werden kann.

Damit sind die Grundlagen der Dependenzgrammatik dargelegt. Im nächsten Teil wird auf den FDG-Parser und die ihm zugrunde liegende Funktionale Dependenzgrammatik eingegangen.

---

[2] Eine systematische Kategorisierung ist z.B. zu finden bei Tarvainen 1981 oder Engel 1988.
[3] Genaueres in Heringer 1995, S.305 ff.

2. English Functional Dependency Grammar (Funktionale Dependenz-grammatik)

2.1. Einführung

Zur Veranschaulichung dessen, was Dependenzgrammatik ist, soll der FDG-Parser dienen, den Pasi Tapanainen und Timo Järvinen von der Universität Helsinki[4] gebaut haben und der online zur Verfügung steht (Homepage: http://www.conexor.fi/analysers.html#testing).
Ihr Parsing-System besteht aus einem Lexikon, der morphologischen CG-2-Disambiguierung (CG steht für Constraint Grammar und wird später genauer erklärt) und der Functional Dependency Grammar. Der springende Punkt an diesem System ist die Tatsache, dass hier alle Kanten mit Funktionsbezeichnungen versehen sind. Durch diese Funktionalitätsbeschreibung können Probleme wie dasjenige von Aktanten/Circumstanten gelöst werden.

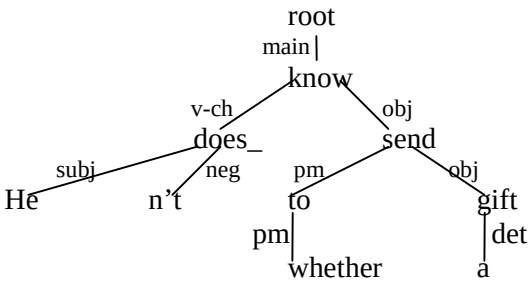
2.2. Syntaktische Strukturen

Die Dependenzfunktionen in der aktuellen Grammatik, die im Parser-Output als Kanten-bezeichnungen erscheinen, beinhalten ungefähr 30 Funktionen (link functions). Von den Autoren werden die Funktionen eingeteilt in folgende Gruppen: Hauptfunktionen, intranukleare Verbindungen, Verbkomplementation, Adverbialfunktionen, Determinativfunktionen, Modifikatoren und Junktive.

Hauptfunktionen

Die Funktion **main** (= main element) steht für das Hauptelement eines Teilsatzes. Normalerweise handelt es sich um ein Verb. In verblosen Teilsätzen können auch andere Elemente als Head fungieren.
Bsp. *He doesn't **know** whether to send a gift.*

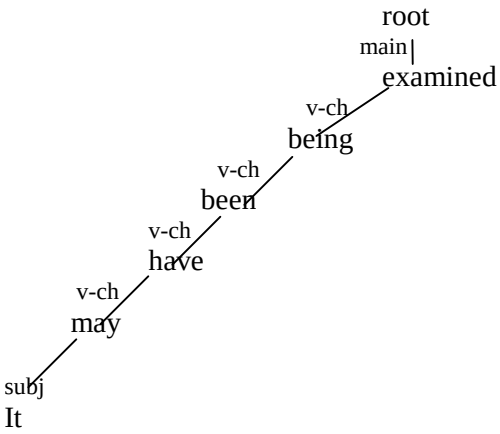
[4] Siehe Järvinen 1997.



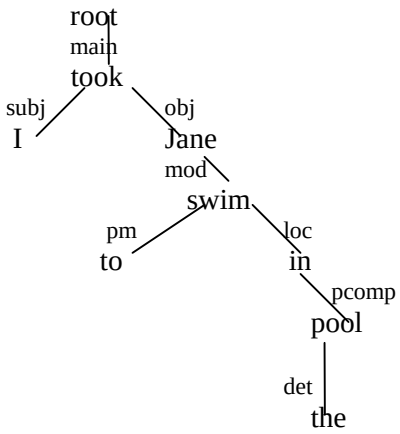
Intranukleare Verbindungen

Intranukleare Verbindungen kombinieren Wörter oder Wortteile, die zu einem Nucleus[5] gehören. Darunter fallen **v-ch** (Verbalketten), **pm** (Präpositions-Marker), **pcomp** (Präpositonal-Komplement) und **phr** (Verbpartikel).

Verbalkette:



Präpositons-Marker , Präpositonal-Komplement:



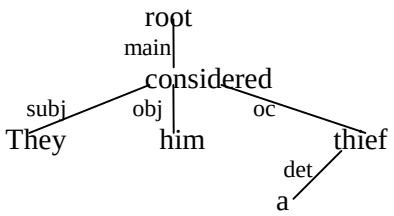
Die Autoren verstehen die Verbalkette als **einen** Nucleus. Deshalb wird das Subjekt nicht ans Hauptverb gebunden, sondern ans letzte Glied der Verbalkette.

Verb-Komplementation

In dieser Kategorie finden sich die Funktionen **subj** (Subjekt), **obj** (Objekt), **comp** (Subjekt-Komplement), **dat** (indirektes Objekt), **oc** (Objekt-Komplement), **copred** (Koprädikativ), **voc** (Vokativ).

[5] Der Begriff Nucleus wurde schon von Tesnière verwendet und wird meist dem Begriff Knoten gleichgesetzt. Wichtig ist, dass Nucleus nicht ein einzelnes Element (meistens ein Wort) beschreiben muss, sondern dass es auch mehrteilige Nuclei gibt (wie eben die Verbalkette).

Subjekt, Objekt, Objekt-Komplement:



Adverbialfunktionen

In diese Kategorie gehören, wie der Name schon sagt, Adverbiale: **tmp** (time), **dur** (duration), **frq** (frequency), **qua** (quantity), **man** (manner), **loc** (location), **sou** (source), **goa** (goal), **cnt** (contingency), **cnd** (condition), **meta** (clause adverbial), **cla** (clause initial element) und **ha** (heuristic high attachment).

Determinativfunktionen

qn (quantifier), **det** (determiner), **neg** (negator).

Modifikatoren

attr (attributive nominal), **mod** (other postmodifiers), **ad** (attributive adverbial).

Junktive

cc (coordination).

Beruhend auf der Constraint Grammar nach Karlsson gibt es auch ein Set von morphosyntaktischen Funktionen[6] (functional tags). Obwohl die meiste Information aus den Dependenzverbindungen und den PS-(part-of-speech)Tags erfolgt (und diese Funktionen somit einen redundanten Output liefern), sind sie für bestimmte syntaktische Unterscheidungen unerlässlich, z.B. für die Apposition.

Auch diese Kategorien sind im Parser-Output ablesbar (durch Klicken auf die Knoten). Auf ein Beispiel angewendet (diesmal nicht als Stemma, sondern als Text-Output):

Zeichen-kette	Wort	Lemma	Abhängigkeit von	Bezeichnung
0	(Root)			
1	He	he	subj>2	**@SUBJ** PRON PERS MASC NOM SG3
2	does	do	v-ch>4	**@+FAUXV** PRES SG3
3	n't	not	neg>2	**@ADVL** NEG-PART
4	know	know	main>0	**@-FMAINV** V INF
5	whether	whether	pm>6	**@CS** CS
6	to	to	pm>7	**@INFMARK**> INFMARK>
7	send	send	obj>4	**@-FMAINV** V INF
8	a	a	det>9	**@DN**> DET SG
9	gift	gift	obj>7	**@OBJ** N NOM SG

[6] Eine Auflistung aller Funktionen steht in Anhang 1.

Bei der Morphologieanalyse werden dem Zielwort all seine möglichen Funktionen (alle Ambiguitäten) zugeordnet. Also ergeben sich z.B. im Satz Joan said whatever John likes to decide suits her. folgende Zuordnungen:

"<Joan>"
    "joan" N NOM SG @NH @SUBJ @OBJ @I-OBJ @PCOMPL-S @PCOMPL-O @APP @A> @<P @O-ADVL
"<said>"
    "say" PCP2 @<P-FMAINV @-FMAINV
    "say" V PAST VFIN @+FMAINV
    "say" A ABS @PCOMPL-S @ PCOMPL-O @A> @APP @SUBJ @OBJ @I-OBJ @<P @<NOM
u.s.w.

Subkategorisierungs- und Valenzinformationen sind hier wichtig, werden aber von den Autoren nicht angegeben.

Zur Eliminierung von falschen und zum Auswählen der richtigen Lesart des Zielworts sind weitere Schritte nötig.

2.3. Regeln und Tests

Wie schon erwähnt, basiert der Parser auf der Constraint Grammar nach Fred Karlsson[7]. Die grundlegenden Regeln dieser Grammatik sind REMOVE und SELECT, wenn es darum geht, eine mögliche Leseart eines Wortes abzulehnen oder auszuwählen. Solche Regeln führen kontextuelle Tests mit sich, welche die Bedingungen beschreiben, unter denen die Regeln anwendbar sind. Z.B.verwirft die Regel

REMOVE (V) IF (-1C DET);

die Bezeichnung eines Wortes als Verb (V), wenn das vorausgehende Wort (-1) ohne Ambiguität (C) ein Determinativ ist. Es können mehrere Tests in dieser Weise an eine Regel angefügt werden.

Die oben genannte Regel repräsentiert eine lokale Regel, denn der Test überprüft nur die Nachbarwörter in der Position vor und nach dem Zielwort. Ein Test kann sich aber auch auf Positionen irgendwo im Satz beziehen, ohne dass die genaue Position bekannt ist. Z.B. sagt der Test

SELECT (IMP) IF (NOT *-1 NOM-HEAD);

dass ein nominaler Head (NOM-HEAD ist ein Set bestehend aus Part-of-speech-tags, die einen Nominal-Head repräsentieren können) nicht auf der linken Seite des Zielwortes vorkommen kann (NOT *-1), egal in welchem Abstand. Der Abstand kann eingeschränkt werden durch Barrieren (BARRIERs), welche das Testgebiet einschränken. Die Barriere kann den Test auf den aktuelle Teilsatz beschränken (durch Teilsatz-End-Marker und „Stop-Wörter“) oder auf eine Konstituente (durch „Stop-Kategorien“). Zudem kann ein Test zur Relativierung der unbeschränkten Kontextposition durch das Wort LINK hinzugefügt werden. Z.B.verwirft die folgende Regel die syntaktische Funktion @I-OBJ[8] (indirektes Objekt):

[7] Siehe Karlsson 1990.
[8] @ ist das Zeichen für Funktionalität.

```
REMOVE (@I-OBJ)
IF (*-1C VFIN BARRIER SVOO
        LINK NOT 0 SVOO);
```

Die Regel hat Bestand, wenn das finite Verb, das links am nächsten steht, ohne Ambiguität (C) ein finites Verb ist (VFIN) und es kein ditransitives Verb oder Partizip (Subkategorisierung SVOO) zwischen Verb und indirektem Objekt gibt. Wenn zusätzlich das Verb kein direktes Objekt verlangt, d.h. wenn es kein SVOO im gleichen Verb gibt (LINK NOT 0 SVOO), wird die @I-OBJ-Leseart verworfen.

Dieser Formalismus wird von den Autoren für die syntaktische Analyse übernommen. Nach der morphologischen Disambiguierung werden alle legitimen oberflächensyntaktischen Labels zu den morphologischen Lesarten hinzugefügt. Dann verwerfen die syntaktischen Regeln kontextuell illegitime Alternativen oder wählen die legitimen aus.

Die syntaktischen Tags der Constraint Grammar stellen eine unterspezifizierte Dependenzbeschreibung zur Verfügung. Z.B.markieren Labels für funktionale Heads (wie @SUBJ, @OBJ, @I-OBJ) das Wort, das der Head einer Nominalphrase ist und diese Funktion im Teilsatz hat, jedoch wird das Regens nicht angegeben. Zudem ist die Repräsentation flach, d.h. Objekte von Infinitiven und Partizipien erhalten das gleiche Label wie Objekte von finiten Verben. Andererseits erhalten nicht-finite Verbformen, welche die Funktion eines Objekts haben, nur Verb-Labels.

Wenn man den Grammatikformalismus benützt, der oben beschrieben wurde, kann eine relativ grosse Anzahl von Ambiguitäten nicht verlässlich aufgelöst werden und bleibt deshalb im Parse-Vorgang hängen. Als Folge davon ist der Output in vielen Fällen nicht optimal. Z.B. ist es unmöglich, verlässlich Head-Modifier-Paare oder Argumente von Verben aus dem Parser-Output herauszulesen.

Um diese Probleme zu lösen, haben Tapanainen und Järvinen einen stärkeren Regelformalismus entwickelt, der eine explizite Dependenz-Repräsentation verwendet. Die Grundidee der Constraint Grammar, die Information "gestückelt" einzuführen, wird beibehalten, aber die Integration von verschiedenen Informationsteilen ist im neuen System effizienter.

Die Notation folgt dem klassischen Dependenzmodell von Lucien Tesnière, das von Igor Mel’cuk weitergeführt wurde.

2.3.1. Einzigartigkeit und Projektivität

Bei Tesnières und Mel’cuks Dependenznotation hat jedes Element des Dependenzbaums einen einzigen Head. Das Verb dient als Head eines Teilsatzes, und oberstes Element eines Satzes ist demnach das Hauptverb eines Hauptsatzes. In anderen Theorien sind mehrere Heads erlaubt.

Projektivität (projectivity, adjacency) war bei Tesnière kein Punkt, da er dachte, die lineare Wortfolge gehöre nicht zur syntaktischen Repräsentationsebene, die ausschliesslich die strukturale Ordnung beinhaltet.
Einige frühe Formalisierungen führten die Bedingung der strikten Projektivität (kontextfrei) ins Dependenzmodell ein. Diese Art Beschränkung findet man in vielen dependenzbasierten Parsern. Offensichtlich sollte sich jede erkennende Grammatik idealerweise mit nicht-projektiven Phänomenen in dem Grad befassen, wie sie in natürlichen Sprachen vorkommen. Das aktuelle System hat keine eingebauten Beschränkungen bezüglich Projektivität, obwohl uns der Formalismus erlaubt zu bestimmen, wann überkreuzende Verbindungen verboten sind.

2.3.2.Valenz und Kategorien

Verben (wie auch andere Elemente) haben eine Valenz, welche die Anzahl und die Art der Modifikatoren beschreibt, die mit ihnen vorkommen können. In der Valenztheorie wird normalerweise unterschieden zwischen Aktanten und Circumstanten.

Hier wird unterschieden zwischen Valenz (regelbasiert) und Subkategorisierung (lexikalisch): Die Valenz sagt aus, welche Argumente erwartet werden, die Subkategorisierung sagt aus, welche Kombinationen legitim sind. Die Valenz zeigt lediglich die Möglichkeit auf, ein Argument zu haben. Somit kann ein Verb mit drei Valenzstellen Subkategorisierung SVOO oder SVOC haben. Das erstere bezeichnet: Subjekt, Verb, indirektes Objekt, Objekt, das letztere: Subjekt, Verb, Objekt und Objekt-Komplement. Das Standard-Komplement ist ein nominales Komplement, aber es kann auch zusätzliche Informationen bezüglich der Art von möglichen Komplementen geben. Z.B. kann das engl. Verb *say* ein Objekt (SVO) haben, das auch in Form eines to-infinitive clause, einer WH-clause einer that-clause oder einer Struktur der direkten Rede realisiert sein kann.

Die Circumstanten sind normalerweise in den Verben nicht markiert, da die meisten Verben z.B. räumlich-zeitliche Argumente haben können. Andererseits sind adverbiale Aktanten und Circumstanten angegeben, die für ein bestimmtes Verb typisch sind. Z.B. hat das engl. Verb *decide* das Tag <P/on>, das bedeutet, dass die Präpositionalphrase *on* typischerweise mit dem Verb vorkommt.

2.3.3. Dependenzen

Normalerweise sind das abhängige Element und sein Head implizit (und ambig) in den Constraint-Grammar-Regeln vorhanden. Hier werden diese Dependenzbeziehungen explizit gemacht, und zwar durch die Bezeichnung von Heads und Dependentien in den Kontext-Tests.

Z.B. wird das Subjekt-Label (@SUBJ) ausgewählt und markiert als Dependens des direkt folgenden Hilfsverbs (AUXMOD) in der folgenden Regel:

SELECT (@SUBJ) IF (1C AUXMOD HEAD);

Um den Parser voll auszunutzen, ist es auch hilfreich, die Valenzstelle in der Regel zu nennen. Dies hat zwei Effekte: 1. Es gibt nur eine Valenzstelle, d.h. es kann nicht mehr als ein Subjekt an ein finites Verb gebunden sein, und 2. wir können explizit in Regeln formulieren, von welcher Art von Valenzstellen wir erwarten, dass sie besetzt werden. Damit hat die Regel die Form:

SELECT (@SUBJ)
IF (1C AUXMOD HEAD = subject);

Die obige Regel funktioniert in einem nicht-ambigen Kontext gut, aber es ist dennoch nötig, tolerantere Regeln für ambige Kontexte zu spezifizieren. Die Regel

INDEX (@SUBJ)
IF (1C AUXMOD HEAD = subject);

unterscheidet sich von der vorherigen Regel insofern, als sie die anderen Lesarten des Nomens intakt lässt und nur eine (mögliche) Subjektdependenz hinzufügt, während beide vorherigen Regeln auch die Nomen-Lesart disambiguierte.

Jedoch reicht der Kontext-Test vor allem in der obigen Regel nicht aus, um verlässlich die Subjekt-Lesart auszuwählen. Er lässt lediglich die Möglichkeit offen, eine Dependenz einer anderen syntaktischen Funktion anzuhängen, d.h. die Dependenzrelationen bleiben ambig. Die Grammatik hütet sich möglichst davor, falsche Dependenzen einzuführen, aber aus offensichtlichen Gründen ist dies nicht immer möglich. Wenn mehrere syntaktische Funktionen eines Wortes Dependenzrelationen haben, bilden sie einen Dependenzwald. Deshalb kann, wenn die syntaktische Funktion nicht vorschnell disambiguiert wird, die korrekte Lesart selbst nach unerlaubtem Verbinden überleben, da globales Beschneiden (engl. pruning) später Dependenzverbindungen extrahiert, die konsistente Bäume bilden.

Verbindungen zwischen syntaktischen Labels bilden partielle Bäume, normalerweise um verbale Nuclei. Aber ein neuer Mechanismus ist nötig, um die strukturelle Information, die aus Regeln gewonnen wurde, voll auszunutzen. Wenn eine Verbindung zwischen Labels gebildet wird, kann sie von den anderen Regeln genutzt werden. Wenn z.B. der Head einer Objektphrase (@OBJ) gefunden und einem Verb zugeschrieben (indiziert) wird, ist die Nominalphrase rechts davon (wenn eine vorhanden ist) wahrscheinlich ein Objekt-Komplement (@PCOMPL-O). Es sollte den gleichen Head wie das existierende Objekt haben, wenn das Verb das richtige Subkategorisierungs-Tag hat (SVOC). Die folgende Regel legt eine Dependenzrelation eines Verbes und seines Objekt-Komplements fest, falls das Objekt schon existiert:

```
INDEX (@PCOMPL-O)
IF (*-1 @OBJ BARRIER @NPHEAD)
        LINK 0 UP object SVOC HEAD=o-compl);
```

Die Regel sagt, dass eine Dependenzrelation (o-compl) hinzugefügt werden, aber die syntaktischen Funktionen nicht disambiguiert (INDEX) werden sollten. Das Objekt-Komplement (@PCOMPL-O) ist verbunden mit den Verb-Lesarten, welche die Subkategorisierung SVOC haben. Die Relation zwischen dem Objekt-Komplement und seinem Head ist derart, dass die Nominalphrase links vom Objekt-Komplement ein Objekt ist (@OBJ), das eine Dependenzrelation (object) mit dem Verb eingegangen ist.

Natürlich können Dependenzrelationen auch nach unten verfolgt werden (DOWN). Aber es ist auch möglich, vom letzten Zeichen in einer Verbindungskette auszugehen (z.B. die Verbkette *would have been wanted*) und dazu die Schlüsselwörter TOP und BOTTOM zu verwenden (Sprung zum obersten resp. untersten Element).

2.3.4. Ambiguität und Beschneiden

Folgende Strategie wird für das Verbinden und die Disambiguierung verfolgt:

- Im besten Fall sind ist mit Sicherheit eine Lesart im aktuellen Kontext korrekt ist. In diesem Fall kann Disambiguieren und Verbinden gleichzeitig ausgeführt werden (mit Befehl SELECT und Schlüsselwort HEAD).
- Der typischste Fall ist derjenige, in dem der Kontext Hinweise auf die korrekte Lesart gibt, aber wir wissen, dass es einige wenige Ausnahmefälle gibt, wo diese Lesart nicht korrekt ist. In so einem Fall fügen wir nur einen Link hinzu.
- Manchmal gibt der Kontext klare Hinweise, was auf keinen Fall die korrekte Lesart sein kann. In diesem Fall können wir einige Lesarten entfernen, auch wenn wir nicht wissen, was die korrekte Alternative ist. Dies ist in der Constraint Grammar ein ziemlich typischer Fall, kommt aber in der neuen Dependenzgrammatik relativ selten vor. In der Praxis sind diese Regeln, die abgesehen von ihrer linguistischen Interpretation oft sehr obskur sind, die

wahrscheinlichste Fehlerquelle. Ausserdem besteht kein Grund mehr, diese Lesarten explizit durch Regeln auszuschalten, weil das globale Beschneiden Lesarten entfernt, die keine „speziellen Beweise“ erhalten haben.

Grob gesagt, werden die REMOVE-Regeln der Constraint Grammar durch die INDEX-Regeln ersetzt. Das Ergebnis davon ist, dass die Regeln unter den neuen Rahmenbedingungen viel sorgfältiger sind.

Wie schon bemerkt, hat die Dependenzgrammmatik grosse Vorteile beim Problem der Ambiguitäten. Da die Dependenzen einen Baum formen sollen, können wir heuristisch Lesarten beschneiden, die mit nur geringer Wahrscheinlichkeit in so einem Baum vorkommen. Wir haben die folgenden Hypothesen:

1. Der Dependenzwald hat ziemich viele Lichtungen und ein ganzer Parse-Baum wird nicht immer gefunden;
2. Das Beschneiden sollte grosse (Unter-)Bäume bevorzugen;
3. Verbindungslose Lesarten von Wörtern können entfernt werden, wenn es eine Lesart mit Verbindungen unter den Alternativen gibt;
4. Nicht-ambige Unterbäume sind mit grösserer Wahrscheinlichkeit korrekt als ambige; und
5. Das Beschneiden muss nicht unbedingt Wörter zwingen, nicht-ambig zu sein. Wir können Regeln iterativ verwenden, und normalerweise sind einige der Regeln anwendbar, wenn die Ambiguität reduziert ist. Das Beschneiden wird dann nochmals angewendet u.s.w. Ausserdem beinhaltet der Beschneidungsmechanismus keine sprachspezifischen Statistiken.

Einige der heuristischsten Regeln können erst nach dem Beschneiden angewendet werden. Dies hat zwei Vorteile: Extrem heuristische Verbindungen würden den Beschneidungsmechanismus verwirren, und Wörter, die andernfalls keinen Head hätten, könnten noch immer einen bekommen.

2.4. Beispielsätze

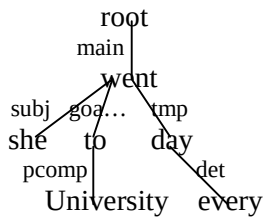
Um den Parser zu testen, soll der Output für einige Beispielsätze gezeigt werden:

Satz 1: *She went to University every day*

Text-Output:

0				
1	She	she	subj:>2	@SUBJ PRON PERS FEM NOM SG3
2	went	go	main:>0	@+FMAINV V PAST
3	to	to	goa:>2	@ADVL PREP
4	University	university	pcomp:>3	@<P N NOM SG/PL
5	every	every	det:>6	@DN> DET SG/PL
6	day	day	tmp:>2	@ADVL <ADV-N> N NOM SG

Stemma:



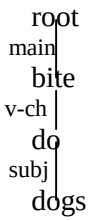
Bemerkungen: Der Parser erkennt bei *every day*, dass es sich um eine temporale Adverbialfunktion handelt (tmp).

Satz 2: *Do dogs bite?*

Text-Output:

0				
1	Do	do	v-ch:>3	@+FAUXV V PRES -SG3
2	dogs	dog	subj:>1	@SUBJ N NOM PL
3	bite	bite	main:>0	@-FMAINV V INF
	?	?		

Stemma:



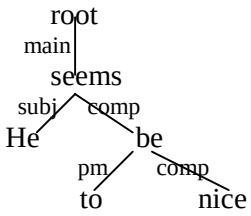
Bemerkungen: Interessant ist, dass *dogs* an das Hilfsverb *do* gebunden wird, nicht ans Hauptverb. Die Erklärung dafür findet sich in der Darstellung der Verbalketten in der FDG (siehe unter 'intranukleare Verbindungen')

Satz 3: *He seems to be nice.*

Text-Output:

0				
1	He	he	subj:>2	**@SUBJ** PRON PERS MASC NOM SG3
2	seems	seem	main:>0	**@+FMAINV** V PRES SG3
3	to	to	pm:>4	**@-FMAINV** V INF
4	be	be	comp:>2	**@OBJ** N NOM SG
5	nice	nice	comp:>4	**@PCOMPL-S** A ABS
	.	.		

Stemma:



Bemerkungen: Hier ist interessant, dass *b*e kein Subjekt hat (im Gegensatz z.B. zur GB-Theorie).

2.5. Zusammenfassung

Der FDG-Parser von Tapanainen und Järvinen kombiniert zwei Mechanismen: Mit dem ersten, der Constraint Grammar, werden ambige Lesarten entfernt. Mit dem zweiten werden Dependenzen eingefügt. D.h. während der Parser disambiguiert, erzeugt er auch einen Dependenzwald (also mehrere Dependenzbäume), der wiederum durch andere Disambiguierungsregeln und einen globalen Beschneidungsmechanismus reduziert wird. Die Autoren haben die Effizienz dieses Systems bisher nur mit einem ihrer eigenen anderen Systeme verglichen (dem ENGCG syntactic analyser)[9], wobei der FDG-Parser besser abschneidet, indem er mehr Informationen liefert, die auch genauer und expliziter ist. Die Ambiguitätsquote beträgt zudem nur ein Viertel derjenigen des anderen Systems, so die Autoren[10].

Die wichtigsten Punkte am FDG-System in Kürze:

- Es gibt keine Konstituenten, nur Dependenzen.
- Es gibt funktionale Dependenzen, also Kantenbezeichnungen.
- Es wird auf nur einer syntaktischen Ebene analysiert, der oberflächensyntaktischen.
- Das grundlegende syntaktische Element ist nicht das Wort, sondern der Nucleus.

27.4.2000

[9] Siehe Tapanainen/Järvinen: A non-projective dependency parser, letzte Seite.
[10] Auf die Vor- und Nachteile der praktischen Anwendung des FDG-Parsers wird bei der Präsentation eingegangen.

Bibliographie

Bussmann, Hadumod: Lexikon der Sprachwissenschaft. Alfred Kröner Verlag. Stuttgart 1990.

Engel, Ulrich. Syntax der deutschen Gegenwartssprache. Reihe Grundlagen der Germanistik. Erich Schmidt Verlag. Berlin 1982.

Heringer, Hans Jürgen: Approaches to a Theory of Syntax III: Dependency Syntax. In: Syntax. Ein internationales Handbuch zeitgenössischer Forschung. Walter de Gruyter. Berlin, New York 1995 (1. Halbband, S. 298-316).

Heringer, Hans Jürgen: Deutsche Syntax dependentiell. Stauffenburg Verlag 1996.

Järvinen, Timo und Tpanainen, Pasi: A Dependency Parser of English. Technical Reports, no. TR-1. Department of General Linguistics. University of Helsinki. March 1997. (Online auf http://www.ling.helsinki.fi/~tapanainen/dg/doc/TR-1/TR-1.html)

Järvinen, Timo und Tapanainen, Pasi: Towards an implementable dependency grammar. Research Unit for Multilingual Language Technology. University of Finland.

Karlsson, Fred: Constraint grammar as a framework of parsing running text. In: Karlgren, Hans (Ed.): Papers presented to the 13th International Conference on Computational Linguistics. Helsinki 1990 (Vol.3, p 168-173).

Mel'cuk, Igor A.: Dependency Syntax. Theory and Practice. SUNY Series in Linguistics. State University of New York Press. Albany 1988.

Tapanainen, Pasi und Järvinen, Timo: A non-projective dependency parser. University of Helsinki. Department of General Linguistice. Research Unit for Multilingual Language Technology.

Tarvainen, Kalevi: Einführung in die Dependenzgrammatik. Reihe Germanistische Linguistik. Max Niemeyer Verlag. Tübingen 1981.

Weber, Heinz J.: Dependenzgrammatik. Ein interaktives Arbeitsbuch. Gunter Narr Verlag. Tübingen 1997.

Anhang 1: Liste der Dependenzfunktionen[11]

Link functions

Tag	Explanation
main	main element
qtag	tag question
v-ch	verb chain
pm	preposed marker
pcomp	prepositional complement
phr	verb particle
subj	subject
obj	object
comp	subject complement
dat	indirect object
oc	object complement
copred	copredicative
voc	vocative
tmp	time
dur	duration
frq	frequency
qua	quantity
man	manner
loc	location
sou	source
goa	goal
cnt	contingency
cnd	condition
meta	clause adverbial
cla	clause initial element
ha	heuristic high attachment
qn	quantifier
det	determiner
neg	negator
attr	attributive nominal
mod	other postmodifiers
ad	attributive adverbial

[11] Eine explizitere Auflistung findet sich in Järvinen 1997.

Functional tags

Tag	Explanation
@+FAUXV	Finite auxiliary predicator
@-FAUXV	Nonfinite auxiliary predicator
@+FMAINV	Finite main predicator
@-FMAINV	Nonfinite main predicator
@NP	Stray NP
@VOC	Vocative
@SUBJ	Subject
@F-SUBJ	Formal subject
@OBJ	Object
@I-OBJ	Indirect object
@PCOMPL-S	Subject complement
@PCOMPL-O	Object complement
@ADVL	Adverbial
@O-ADVL	Object adverbial
@APP	Apposition
@DN>	Determiner
@A>	Premodifier of a nominal
@QN>	Premodifying quantifier
@AD-A>	Intensifier
@<NOM-OF	Postmodifying OF-phrase
@<AD-A	Postmodifying intensifier
@<NOM	Postmodifier of a nominal
@INFMARK>	Infinitive marker TO
@<P-FMAINV	Nonfinite clause as preposition complement
@<P	Other preposition complement
@CC	Coordinating conjunction
@CS	Subordinating conjunction

Anhang 2: Glossar

Dependens	Dependenzgrammatik: unmittelbar abhängiges Element
Junktion	Koordinationsrelation zwischen gleichartigen Elementen im Satz
Kante	Eine K. ist eine Verbindungslinie zw. zwei Knoten im Stemma. K. verdeutlichen die Struktur des Stemmas.
Kategorie	Menge von Elementen, die durch gleiche Merkmale bestimmt sind oder Familienähnlichkeit zeigen
Knoten	Schnittpunkt im Stemma, der mit einem Lexem oder einem Kategorienzeichen etikettiert ist
Konnexion	syntaktische Relation, der die abstrakte Abhängigkeitsbeziehung zw. syntaktischen Elementen bezeichnet (unabhängig von ihrer linearen Oberflächenordnung)
Link	Verbindung zw. zwei Knoten, Kante
Nucleus	meist mit Knoten gleichgesetzt (siehe dort), wobei wichtig ist, dass ein N. mehrteilig sein kann (z.B. ist die Verbalkette *may have been* **ein** Nucleus).
Projektivität	Bezeichnung für die deskriptiv äquivalente Darstellung von Satzstrukturen durch Stemma und durch indizierte Klammerung, die eindeutig aufeinander abbildbar sind. Einfacher ausgedrückt: Das Stemma entspricht der in linearer Anordnung.darunter geschriebenen Zeichenkette
Regens	Dependenzgrammatik: unmittelbar übergeordnetes Element
Root	Wurzel, Zentralknoten
Stemma	Strukturbaum; in der Dependenzgrammatik Repräsentation der Satzstruktur durch Knoten und Kanten
Translation	Dependenzrelation; Funktionswörter (Translative) können die syntaktische Kategorie eines Ausdrucks verändern und derart seine Konnexion zum nächsthöheren Knoten ermöglichen
Wurzel	Root, Zentralknoten
Zentralknoten	Das Zentrum des Satzes, von dem alle andern Elemente direkt oder indirekt regiert werden; Root, Wurzel