

Softwareprojektbericht

Modellierung grammatischer Phänomene der deutschen Sprache mit Topologischer Dependenzgrammatik

Jochen Steigner* Ulrich Pfeiffer† Regine Bader‡
Christine Foeldes§

29. September 2004

Softwareprojekt unter der Leitung von Ralph Debusmann

*jockel@coli.uni-sb.de
†uli@coli.uni-sb.de
‡regine@coli.uni-sb.de
§cfoeldes@yahoo.de

Inhaltsverzeichnis

1	Einleitung	5
2	Umstrukturierung der Verbklassen	6
2.1	Ausgangssituation	6
2.2	Umstrukturierung der Verbklassen	6
2.3	Veränderungen im Lexikon	7
2.4	Beispielbäume	7
3	Freie Relativsätze	8
3.1	Allgemeines	8
3.2	Freie Relativsätze in TDG	8
3.2.1	Neue Labels	8
3.2.2	Neue Prinzipien	8
3.2.3	Neue Klassen	9
3.2.4	Veränderungen in den schon vorhandenen Klassen	9
3.2.5	Veränderungen im Lexikon	10
3.2.6	Beispielbäume	10
4	Verbalkomplemente von Nomina	13
4.1	Allgemeines	13
4.2	Verbalkomplemente in TDG	13
4.2.1	Neue Labels und Prinzipien	13
4.2.2	Neue Klassen	13
4.2.3	Veränderungen in den schon vorhandenen Klassen	14
4.2.4	Veränderungen im Lexikon	14
4.2.5	Beispielbäume	14
5	Komparativ	16
5.1	Allgemeines	16
5.2	Komparativ in TDG	16
5.2.1	Neue Labels	16
5.2.2	Neues Prinzip	16
5.2.3	Neue Klassen	17
5.2.4	Veränderungen in den schon vorhandenen Klassen	17
5.2.5	Veränderungen im Lexikon	17

5.2.6	Beispielbaum	17
6	Besetzung des Vor-Vorfelds	19
6.1	Vor-Vorfeld-Besetzung in TDG	19
6.1.1	Neue Edge-Constraints	19
6.1.2	Neue Labels	19
6.1.3	Neue Klassen und Veränderungen	20
6.1.4	Beispielbäume	20
7	Passivkonstruktionen	22
7.1	Passivkonstruktionen in TDG	23
7.1.1	Neue Labels	23
7.1.2	Neue Klassen und Veränderungen	23
7.1.3	Besonderheiten im Lexikon	24
7.1.4	Beispielbäume	24
8	Imperative	27
8.1	Imperative in TDG	27
8.1.1	Neue Labels	27
8.1.2	Neue Klassen und Veränderungen	27
8.1.3	Besonderheiten im Lexikon	28
8.1.4	Beispielbäume	28
9	Nomina	29
9.1	Nomina in TDG	29
9.1.1	Neue Klassen und Veränderungen	29
9.1.2	Beispielbäume	29
10	Adverbien	31
10.1	Allgemeines	31
10.2	Ausgangssituation	31
10.3	Umsetzung in TDG	31
10.3.1	Ebenen	32
10.3.2	Lexikon	32
10.4	Ableitungen	32
11	Konjunktionen	34
11.1	Allgemeines	34
11.2	Ausgangssituation	34
11.3	Umsetzung in TDG	34
11.3.1	Ebenen	34
11.3.2	Lexikon	35
11.4	Ableitungen	35

12 Belebte vs. unbelebte Nomen	36
12.1 Allgemeines	36
12.2 Umsetzung in TDG	36
12.2.1 Ebenen	36
12.2.2 Veränderungen in den bestehenden Klassen	37
12.2.3 Lexikon	37
12.3 Ableitungen	37
13 Kopula & Prädikative	39
13.1 Allgemeines	39
13.2 Umsetzung in TDG	39
13.2.1 Ebenen	39
13.2.2 Lexikon	40
13.3 Ableitungen	40
14 Fragesätze	42
14.1 Allgemeines	42
14.2 Ausgangssituation	42
14.3 Umsetzung in TDG	42
14.3.1 Ebenen	43
14.3.2 Lexikon	43
14.4 Ableitungen	43
15 Das Phänomen Es	46
15.1 Explikatives es:	46
15.2 Platzhalter-es:	46
15.3 TDG-Darstellung:	47
15.3.1 Explikatives es:	47
15.3.2 Vorfeld-es:	48
15.3.3 Korrelat-es:	49
16 Koordination	51
16.1 Koordinierte Objekte und Adverbien	51
16.2 Koordinierte Genitivattribute	52
16.3 Koordinierte Subjekte	53
16.4 Koordinierte Sätze	54
16.5 Koordinierte Präpositionalphrasen	54
16.6 Koordinierte Adjektive	56
17 Schluss	57
18 Literaturverzeichnis	58

1 Einleitung

Dieser Arbeitsbericht beschreibt, wie im Laufe eines Softwareprojekts die existierende TDG-Basisgrammatik von Ralph Debusmann (2001) für das Deutsche um einige Phänomene erweitert wurde, wobei diese mit dem bestehenden Lexikon realisiert wurden.

Topologische Dependenzgrammatik (TDG) ist eine neue Grammatiktheorie, mit der auch Sprachen, die über eine relativ freie Wortstellung verfügen, gut dargestellt werden können. Hierzu werden die syntaktischen Relationen wie in der Dependenzgrammatik ausgedrückt, die topologischen Abhängigkeiten wie in der Theorie der topologischen Felder. Eine TDG-Analyse beinhaltet somit zwei Baumstrukturen: einen ungeordneten ID-Baum, dessen Kanten mit syntaktischen Rollen (subj, obj, iobj) beschriftet sind und einen geordneten, projektiven LP-Baum, der sozusagen eine flachere Version des ID-Baumes ist und die Wortabfolge des Satzes beachtet. Zur Bestimmung der Wortordnung benutzt TDG eine Menge von externen (vf, mf, nf) und internen (v,n) Feldern.

In der Praxis besteht also eine TDG besteht aus einer Auflistung der benutzten syntaktischen und topologischen Labels, aus einer Beschreibung der ID- und LP-Beziehungen der Wörter im Satz, aus einem Lexikon, und aus grammatikalischen Prinzipien, die zur Wohlgeformtheit der LP-Bäume beitragen.

2 Umstrukturierung der Verbklassen

2.1 Ausgangssituation

Die Verbklassen waren in der Ausgangsgrammatik sehr mächtig und dadurch kompliziert und schwer zu durchschauen. Um die Behandlung der einzelnen Phänomene und den deshalb notwendigen Umgang mit den mit den Verbklassen einfacher zu gestalten, wurden diese ausgedünnt und neu strukturiert.

2.2 Umstrukturierung der Verbklassen

Die Klassen *vvinf*, *vminf* und *vainf* wurden zunächst abgeschafft und durch eine Klasse für Infinitive, *vinf*, ersetzt, da sie sich abgesehen von ihrer Bezeichnung nicht unterscheiden. Ebenso konnte mit *vvizu*, *vmizu* und *vaizu* sowie *vvpp*, *vmpp* und *vapp* verfahren werden, sie wurden als Klasse für Infinitive mit zu, *vizu*, und als Klasse für Partizip Perfekt, *vpp*, zusammengefasst. Die Klassen *vofin*, *vmfin* und *vafin* sind aus dem gleichen Grund wie oben ebenfalls ausgemustert, die Klasse *vaf* für Auxiliare wurde jedoch später für die Handhabung von Passivkonstruktionen modifiziert wieder eingeführt (s. Abschnitt 6.1.2).

Die Klasse *verb_nonfin* stand für zu viele verschiedene Verbformen. Sie wurde aufgeteilt in *vvpp* und *vvinf/vvizu*. Letztere blieb eine „Doppelklasse“, weil die Verbformen für Infinitive und Infinitive mit zu sich in ihrer äußeren Gestalt nicht unterscheiden. *verb_nvc_fin* wurde der Klarheit wegen in *vofin* umbenannt. In *vofin_v12_vpref* und *vofin_v_vpref* teilte sich die Klasse *verb_nvc_fin_vpref* auf. Die beiden kleineren, übersichtlicheren Klassen stehen für Präfixverben in erster oder zweiter Position bzw. in letzter Position. Die „Überklassen“ *verb_nvc*, *verb_vc_optco*, *verb_vc_oblinco*, *verb_vc_oblco*, *ve! rb_nvc_vpref* und *verb_vc_vpref_oblinco* wurden komplett herausgenommen. Die Informationen, die sie beinhalteten, konnten ins Lexikon verschoben werden.

Analog zu den Vollverben wurde auch bei den Modalverben die Klasse *verb_modal_nonfin* in *vminf/vmizu* und *vmpp* aufgesplittet. *verb_modal_fin* wird jetzt mit *vmfin* bezeichnet und *verb_modal* konnte wegen der Verlagerung der darin enthaltenen Angaben ganz wegfallen.

Die Klassen *verb_aux_nonfin*, *verb_aux_fin*, *verb_aux_vppc* und *verb_aux_vinf* gibt es in ihrer ursprünglichen Form nicht mehr. Die beiden letzteren legten fest, ob das Auxiliar ein Partizip Perfekt oder ein Infinitiv als Komplement nimmt. Diese Information wird nun gleich durch die Infinitivklassen *vainf/vaizu_vppc* und *vainf/vaizu_vinf* bzw. die Klassen der finiten Auxiliare *vafin_vinf* und *vafin_vppc* weitergegeben.

2.3 Veränderungen im Lexikon

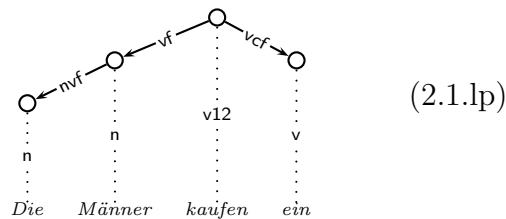
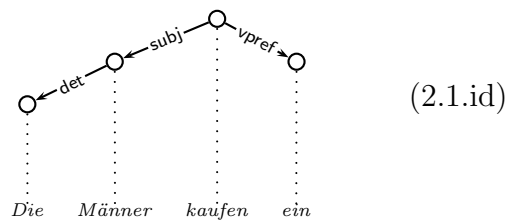
Bei der Neugestaltung der Verbklassen wurden auch die Verbformen für 3. Person Plural eingeführt. Daher sind die Angaben zum Agreement nicht mehr in den Klassen, sondern im Lexikon zu finden. Bisher gab es pro Lexem nur einen Eintrag, inzwischen gibt es durch die Abschaffung der „Überklassen“ für jede Wortform einen eigenen. Deshalb und wegen der Erweiterung durch Pluralformen ist das Lexikon stark gewachsen. Die Vereinfachung in den Verbklassen sorgt jedoch für eine unkompliziertere Behandlung der Phänomene.

2.4 Beispielbäume

Für neuen Verbklassen gibt es eigentlich unendlich viele Beispiele. Deshalb soll es hier genügen zu sagen, dass alle Beispiele aus (Debusmann, 2001) noch immer die gleichen Analysen erhalten. Beispiel (2.1) zeigt den Umgang mit Verben im Plural.

Die Männer kaufen ein.

(2.1)



3 Freie Relativsätze

3.1 Allgemeines

Relativsätze können nicht nur Nomina modifizieren, sondern auch direkt als Komplement des Kopfs fungieren. In diesem Fall werden sie freie Relativsätze genannt. Freie Relativsätze treten in fast allen syntaktischen Umgebungen auf, als Komplemente von Nomina jedoch nicht (s.(3.3)). So ist in (3.1) *was er essen soll* das Akkusativobjekt von *isst* und *wem er verpflichtet ist* dient in (3.2) als dativische Ergänzung des Adjektivs *treu*.

Er kauft, was er lesen will. (3.1)

Er ist, wem er ein buch gibt, böse. (3.2)

* Die Frau, wessen er harrt. (3.3)

3.2 Freie Relativsätze in TDG

3.2.1 Neue Labels

In der TDG-Grammatik können freie Relativsätze als Subjekt, direktes oder indirektes Objekt auftreten. Um festlegen zu können, welcher Satz in welcher syntaktischen Umgebung vorkommen darf, wurden auf der id-Ebene drei verschiedene Labels eingeführt: der Subjektsatz *subjs*, der direkte Objektsatz *objs* und der indirekte Objektsatz *iobjs*.

Auf der lp-Ebene bedurfte es nur eines neuen Labels. Um freie Relativsätze von den anderen unterscheiden zu können, gibt es das „freie relative Vorfeld“ *frvf*, in dem ausschließlich die freien Relativpronomen *wer*, *wen*, *wem* und *was* stehen dürfen. Das *frvf* steht in der lp-Ordnung zwischen *ruf* und *if*.

3.2.2 Neue Prinzipien

Relativsatz-Agreement Freie Relativsätze stehen sozusagen im Singular, weshalb ein Verb, dessen Subjekt ein Relativsatz ist, ebenfalls im Singular stehen muss. Um dies zu garantieren, wurde ein Prinzip eingeführt, das das Agreement von *subjs* und dessen Mutterknoten überprüft (parallel zum Prinzip für *subj*). Das Relativsatz-Agreement wurde mithilfe eines Kantenconstraints realisiert (vgl. Debusmann, 2001).

Relativsatz-Kasuskongruenz Tritt ein Relativsatz als Komplement an Stelle einer NP, so unterliegt der Kasus des Relativpronomens bestimmten Einschränkungen (s. Abschnitt 3.2.3). Um deren Einhaltung möglich zu machen, musste ein neues Prinzip für Kantenlabels spezifiziert werden, das sowohl auf der id- als auch auf der lp-Ebene wirkt. Es besagt, dass der Kasus des Mutter- und des Tochterknotens an einer Kante mit dem Label *frvf* gleich sein müssen. D.h. der Kasus des Relativpronomens wird zum Verb hoch gereicht. Das Prinzip für die Relativsatz-Kasuskongruenz ist ein Kantenconstraint (vgl. Debusmann, 2001).

3.2.3 Neue Klassen

Pronomen Die Klasse *freipro_id* erbt von der Klasse *noun_id* und schränkt die eingehenden Kanten auf *subj*, *obj* und *iobj* ein. *freipro_lp* übernimmt die Eigenschaften der Klasse *noun_lp* und legt als eingehende lp-Kante für freie Relativpronomen *frvf* fest. Zuletzt vereinigt *freipro* die beiden zuerst beschriebenen und bildet durch die Bereitstellung der Variablen *Word*, *Agrs* und *Cases* die Schnittstelle zum Lexikon.

Verben Der Kasus des Relativpronomens muss folgender Hierarchie genügen: *nom* > *acc* > *dat*. Dies bedeutet, dass Sätze mit einem Relativpronomen im Nominativ nur Subjektsätze sein können, solche mit einem Relativpronomen im Dativ hingegen als Subjekt, Objekt oder indirektes Objekt fungieren können. Deshalb wurden in den drei neuen Verbklassen dem Attribut *cases* unterschiedliche Werte übergeben.

* Er verspricht, wen er liebt, zu lachen. (3.4)

Er liebt, wem er verspricht, zu lesen. (3.5)

Alle Klassen erben gleichermaßen von der Klasse der finiten Verben *fin_lp*. Außerdem wird auf der lp-Ebene für alle festgelegt, dass die Verben im Vorfeld, im Mittelfeld oder im Nachfeld stehen dürfen und dass sie in Verbletztposition stehen müssen. Als zusätzliche ausgehende Kante ist *frvf* angegeben.

In *subjs_lp* wurde auf der id-Ebene als eingehende Kante *subjs* bestimmt und *cases* die Werte *nom*, *acc* sowie *dat* übergeben. Außerdem wurde definiert, dass das Verb immer in der 3. Person Singular steht, da freie Relativsätze nie im Plural stehen. Für die Klasse *objs_lp* wurde als eingehende Kante auf der id-Ebene *objs* festgelegt. Akkusativ und Dativ sind die zulässigen Kasuswerte. Der Wert für das id-Attribut *in* in *iojbs_lp* ist *iobj* und für *cases* ist es *dat*.

3.2.4 Veränderungen in den schon vorhandenen Klassen

Fast überall dort, wo bisher als ausgehende Kante *subj* bzw. *obj* oder *iobj* definiert war, wurden entsprechend die Kanten *subjs*, *objs* und *iobj* hinzugefügt. Nur bei den Klassen für Präpositionen fanden in diesem Punkt keine Änderungen statt, da Rattenfängerkonstruktionen nicht berücksichtigt wurden.

Die bisherigen Nominalklassen *noun_*, *cnoun_lp*, *pname_* und *perpro_lp* unterscheiden sich hinsichtlich der eingehenden lp-Kanten nun in folgender Weise: Alle vier lassen *vf*, *mf*, *nvf* und *nxf* zu, im *rvf* dürfen jedoch nur *noun_lp* und *cnoun_lp* stehen und einzig *noun_lp* bietet *frvf* an, um es später an *frelpro_lp* vererben zu können.

Bei *fin_id* und *vafin_id* wurde ergänzt, dass finite Verben nicht nur die eingehenden Kanten *v12*, *rel* und *sub*, sondern auch *subjs*, *objs* und *iobjs* haben können.

In den Klassen *vfin*, *vfin_vvpref*, *vmfin*, *vafin_vppc*, *vafin*, *pseudo_vafin* und *vafin_vinf* wurden die drei „neuen Satzarten“ als weitere Option zu den bereits bestehenden Möglichkeiten *v12_lp*, *sub_lp* und *rel_lp* hinzugefügt.

3.2.5 Veränderungen im Lexikon

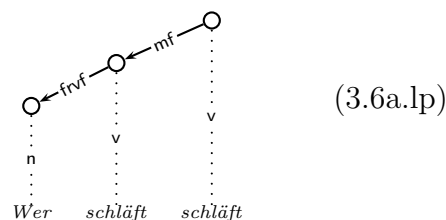
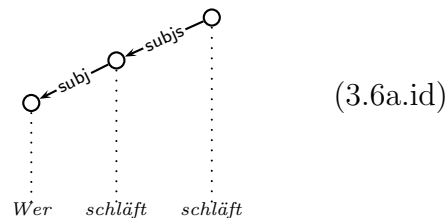
Im Lexikon kamen lediglich vier neue Einträge für die freien Relativpronomen *was*, *wer*, *wen* und *wem* hinzu.

3.2.6 Beispielbäume

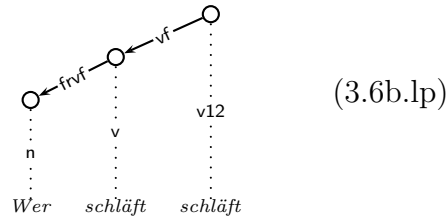
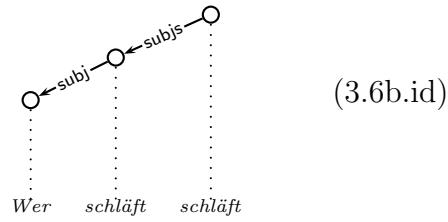
Nun soll anhand einiger Beispiele gezeigt werden, welche Analysen sich für die freien Relativsätze ergeben. In (3.6) hat der Relativsatz die Funktion des Subjekts inne.

Wer schlaeft, schlaeft. (3.6)

Im Folgenden die korrespondierende ID/LP-Analysen, wobei (3.6a.id/lp) die Analyse eines untergeordneten Nebensatzes mit Verbletzstellung zeigt und (3.6b.id/lp) die Struktur eines Hauptsatzes mit Verbzweitstellung darstellt.



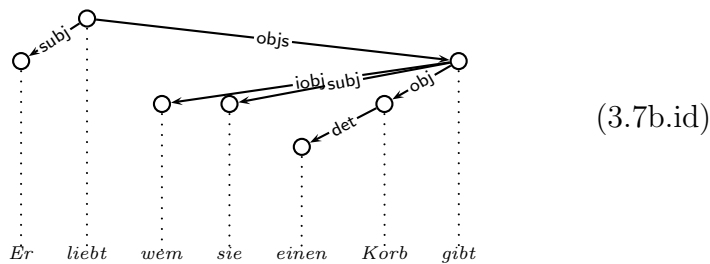
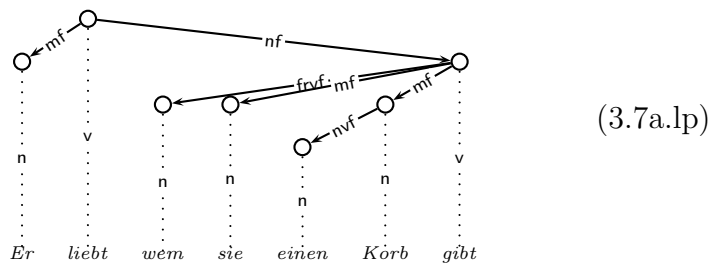
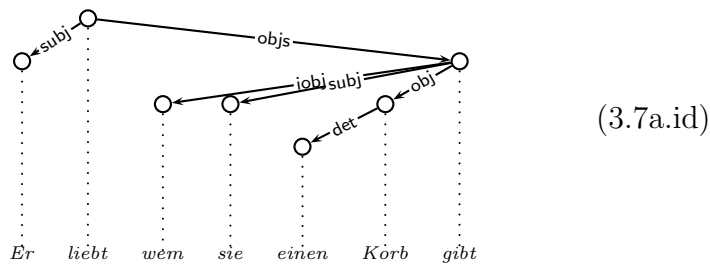
3 Freie Relativsätze



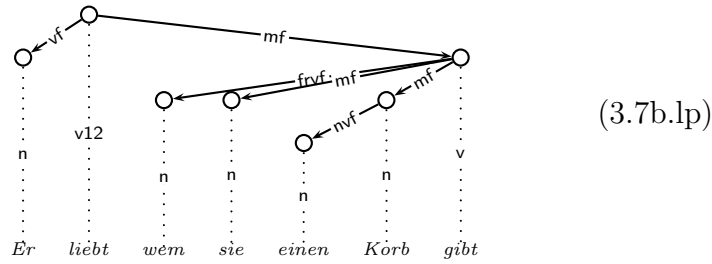
Im nächsten Beispiel sieht man, dass der Kasus des Relativpronomens nicht unbedingt zu der syntaktischen Funktion des Relativsatzes passen muss.

Er liebt, wem sie einen Korb gibt. (3.7)

Auch in (3.7) besteht der Unterschied der zwei ID/LP-Analysen lediglich in der Position des Verbs, weshalb wie bei (3.6) die ID-Analysen identisch sind.



3 Freie Relativsätze



Anhand der folgenden vier Beispiele soll gezeigt werden, dass ein freier Relativsatz, dessen Relativpronomen im Akkusativ steht, sowohl als Subjekt als auch als direktes Objekt fungieren kann. Einem im Nominativ stehendem Satz ist dies jedoch nicht möglich, er kann nur als Subjekt dienen. So ergeben die Sätze (3.8) - (3.9) korrekte Analysen, wohingegen Beispiel (3.11) nicht gepasst wird.

Er liebt, wen Maria liebt. (3.8)

Ihn liebt, wen Maria liebt. (3.9)

Ihn liebt, wer Maria liebt. (3.10)

* Er liebt, wer Maria liebt. (3.11)

4 Verbalkomplemente von Nomina

4.1 Allgemeines

Nomina können durch viele verschiedene Phrasen modifiziert werden. Neben Präpositionalphrasen, Genetivattributen und Relativsätzen können auch Verbalkomplemente Bestandteil einer Nominalphrase sein, jedoch nicht bei allen Nomina. Verbalkomplemente treten dann entweder in Form eines Infinitivs mit *zu* (4.1) oder als dass-Satz (4.2) auf.

Er gibt ihm das Versprechen, ein Buch zu kaufen. (4.1)

Es gibt ihm das Versprechen, dass er schläft. (4.2)

4.2 Verbalkomplemente in TDG

4.2.1 Neue Labels und Prinzipien

Verbalkomplemente konnten in TDG ohne das Hinzufügen neuer Kantenlabels und Prinzipien realisiert werden. Die bereits vorhandenen Labels für Infinitive mit *zu* und dass-Sätze *vzu* und *comp* konnten hier erneut Verwendung finden.

4.2.2 Neue Klassen

Nomina Da nur bestimmte Nomina ein Verbalkomplement nehmen können, musste eine Klasse für diese eingeführt werden. Somit wird verhindert, dass ein beliebiges Nomen durch ein Verbalkomplement erweitert wird. In (4.3) ist ein solcher ungrammatischer Fall zu sehen.

Er hat ein Buch, Maria zu lieben. (4.3)

Die neue Klasse *cnoun_vc_id* lässt als ausgehende Kanten optional *vzu* und *comp* zu. Eine zusätzliche Klasse auf der *lp*-Ebene war nicht notwendig, da die Verbalkomplemente genauso wie Relativsätze im Nachfeld des Nomens landen.

Verben Für die Infinitive entstand die Klasse *vc_nouns_lp*. Sie bestimmt eine eingehende Kante, *nf*, und als ausgehende Kanten beliebig oft *mf* bzw. optional *nf*. Zur Behandlung des subordinierten dass-Sätze als Komplemente von Nomina mussten keine neuen Klassen eingeführt werden, da sie auch als Komplement des Verbs im Nachfeld stehen.

4.2.3 Veränderungen in den schon vorhandenen Klassen

Die neue Klasse für die Infinitive kann nun alternativ zu *can_lp* oder *noncan_lp* von den Klassen *vvinf/vvizu*, *vminf/vmizu* und *vainf/vaizu_vppc* benutzt werden.

4.2.4 Veränderungen im Lexikon

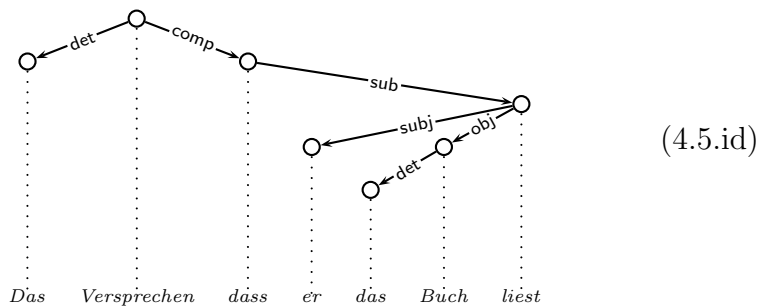
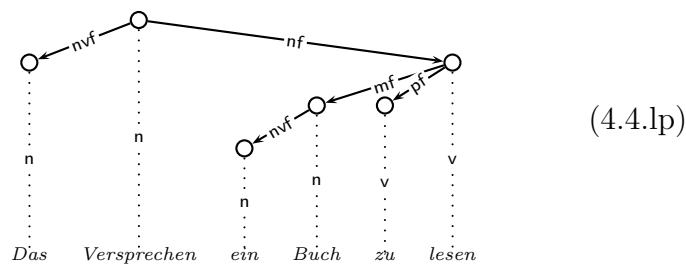
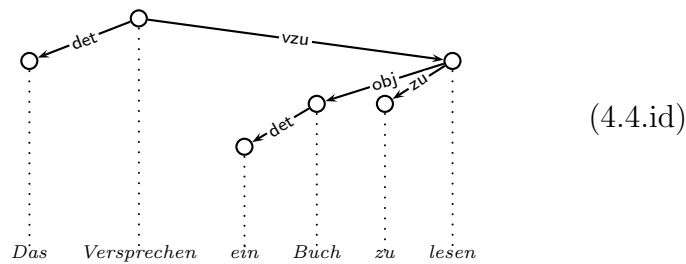
Als beispielhaftes Nomen, das ein Verbalkomplement nehmen darf, wurde *versprechen* neu eingeführt. Außer den üblichen Angaben für Nomina benutzen die Einträge für *versprechen* die Klasse *cnoun_vc_id*, um das Anschließen eines Verbalkomplements zu ermöglichen.

4.2.5 Beispielbäume

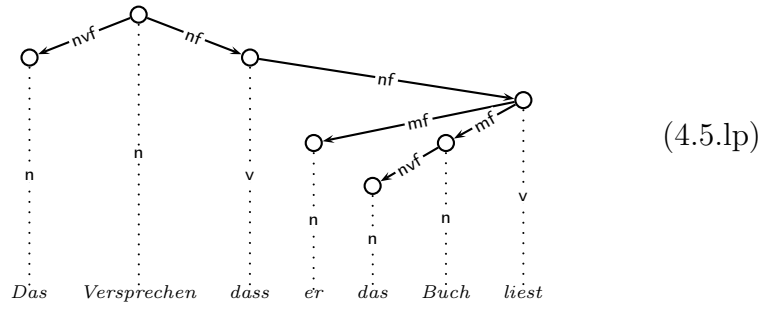
Es folgen Beispiele für die TDG-Analyse von Verbalkomplementen, in (4.4) mit Infinitiv und in (4.5) mit dass-Satz.

Das Versprechen, ein Buch zu lesen. (4.4)

Das Versprechen, dass er das Buch liest. (4.5)



4 Verbalkomplemente von Nomina



5 Komparativ

5.1 Allgemeines

Um auszudrücken, dass zwei Objekte oder Parteien in Bezug auf ein Merkmal ungleich sind, verwendet man den Komparativ. Dem Komparativ folgt in der Regel die Vergleichspartikel *als*, mit der das Vergleichskomplement angehängt wird. Dabei unterliegt das gesteigerte Adjektiv den gleichen Konkruenzbedingungen wie der Positiv. Die beiden Vergleichselemente müssen nicht in Genus und Numerus übereinstimmen, wie die Beispiele (5.2) und (5.3) zeigen, auch wenn Sätze wie (5.4) für die meisten Deutschsprecher eher unakzeptabel klingen. Den gleichen Kasus müssen sie jedoch aufweisen (s. (5.5) und (5.6)).

Ein kleinerer Mann als er kauft das Buch. (5.1)

Ein kleinerer Mann als sie lacht. (5.2)

Ein kleinerer Mann als die Richter liest ein Buch. (5.3)

? Der kleinere Mann als die Frauen lacht oft. (5.4)

Einer kleineren Frau als ihr gibt er den Korb. (5.5)

* Sie liebt einen kleineren Mann als er. (5.6)

5.2 Komparativ in TDG

5.2.1 Neue Labels

Das Vergleichskomplement hängt auf der id-Ebene vom Vergleichspartikel *als* ab. Um diese Beziehung auszudrücken, wurde das neue Kantenlabel *cc*, wie „comparative complement“, eingeführt. Das *als* wiederum ist ein Dependent des Adjektivs, hierfür gibt es nun das Kantenlabel für die *als*-Phrase *alsp*. Auf der lp-Ebene waren keine neuen Labels erforderlich, da die *als*-Phase im nominalen Extrapositionsfeld *nxf* landet.

5.2.2 Neues Prinzip

Kasuskongruenz Zur Einhaltung der Kasuskongruenz wird der Kasus vom Adjektiv über die Kanten *alsp* und *cc* nach unten weitergegeben. Dies geschieht mit Hilfe der beiden neuen Einschränkungen für Kanten, die bestimmen, dass zwei Knoten, die mit *alsp* bzw. *cc* verbunden sind, den selben Kasus haben müssen. Diese Einschränkungen sind mithilfe eines Kantenconstraints implementiert (vgl. Debusmann, 2001).

5.2.3 Neue Klassen

Es war ausreichend, eine neue Klasse einzuführen, die für gesteigerte Adjektive als ausgehende Kante auf der id-Ebene zusätzlich *alsp* festlegt. Diese Klasse heißt *adj_comp_id*. Da nur eine Vergleichspartikel eingeführt wurde, sind alle Informationen dies bezüglich direkt in den Lexikoneintrag für *als* gesteckt worden.

5.2.4 Veränderungen in den schon vorhandenen Klassen

Um Nomina und Pronomina als Vergleichskomplemente zuzulassen, wurde in der Klasse *noun_id* als eingehende id-Kante *cc* spezifiziert.

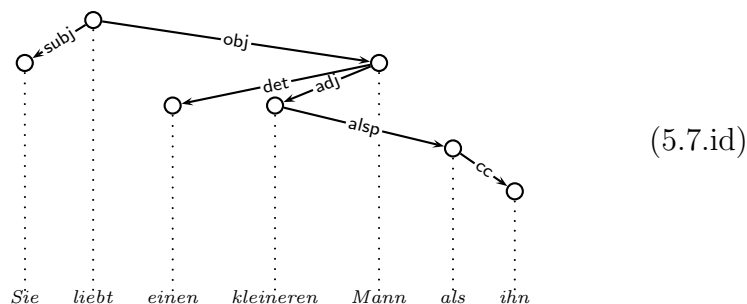
5.2.5 Veränderungen im Lexikon

Exemplarisch wurden für das Adjektiv *klein* die Komparativformen hinzugefügt, und zwar die undefinierten, die definiten und die indefiniten Formen. Sie entsprechen dabei ihren Positivformen, benutzen nur zusätzlich die Klasse *adj_comp_id*. Im Eintrag für *als* wird auf der id-Ebene festgelegt, dass *als* alle Kasus aufweisen kann. Die einzige eingehende Kante soll *alsp* sein und die einzige ausgehende Kante *cc*. Des Weiteren wird spezifiziert, dass die *als*-Phrase im nominalen Extrapositionsfeld und ihr Dependet wiederum in selbigen landet. Außerdem wird vorgeschrieben, dass das Vergleichskomplement *cc* von *als* geblockt wird. Es darf niemals höher als *als* klettern.

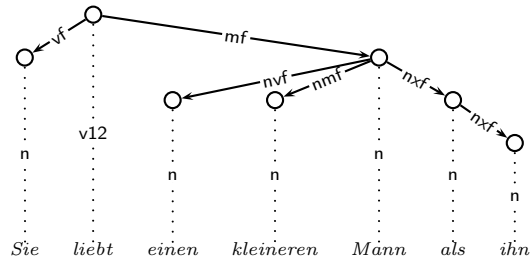
5.2.6 Beispielbaum

Da alle Beispiele die selbe Analyse erhalten, reicht an dieser Stelle eine Beispielableitung. Der Satz (5.6) wird korrekterweise von der Grammatik ausgeschlossen.

Sie liebt einen kleineren Mann als ihn. (5.7)



5 Komparativ



(5.7.lp)

6 Besetzung des Vor-Vorfelds

Bei der topologischen Felderanalyse des Satzes (6.1) steht *Die Frau* im Vorfeld, *lacht* in der linken Satzklammer. Es gibt jedoch auch Sätze, bei denen nicht nur das Vorfeld besetzt ist, sondern auch noch ein weiteres Feld, das Vor-Vorfeld. Da beispielsweise in (6.2) das Vorfeld schon durch *die* besetzt ist, kommt es zu einer Linksversetzung der NP-Konstituente *Die Frau*. Das Vor-Vorfeld ist damit besetzt. Es können dabei auch kompliziertere Konstruktionen wie (6.3) im Vorfeld stehen. Dabei muss die Konstituente im Vor-Vorfeld von einem in Kasus und Agreement übereinstimmenden Resumptivpronomen wieder aufgenommen werden, beispielsweise *der* in (6.3). Sätze wie (6.6) sind dadurch ausgeschlossen. Das Resumptivpronomen übernimmt sozusagen die Rolle des Subjekts oder Objekts, je nach Kasus der NP im Vorfeld. In (6.4) übernimmt es beispielsweise die Rolle des indirekten Objekts. Im folgenden wird das eigentliche Subjekt/Objekt als Resumptiv! um des Pronomens bezeichnet. Ein Resumptivpronomen steht üblicherweise im Vorfeld, kann jedoch in Fragesätzen wie (6.5) auch im Mittelfeld stehen.

Die Frau lacht. (6.1)

Die Frau, die lacht. (6.2)

Ein kleinerer Mann als er, der kauft das Buch. (6.3)

Dem Richter, dem ist Hans böse. (6.4)

Der Mann, kauft der der Frau das Buch? (6.5)

* Peter, bekommt die das Teleskop gegeben? (6.6)

6.1 Vor-Vorfeld-Besetzung in TDG

6.1.1 Neue Edge-Constraints

Um zu gewährleisten, dass die Konstituente im Vor-Vorfeld und das Resumptivpronomen im Vor- oder Mittelfeld in Kasus und Agreement übereinstimmen, wurden zwei entsprechende Edgeconstraints hinzugefügt. Diese legen fest, dass Kasus und Agreement der NP im Vorfeld mit Kasus und Agreement des Resumptivpronomens identisch sind.

6.1.2 Neue Labels

id-Ebene Auf der id-Ebene wurde das Label *res* eingeführt. Dieses Label bezeichnet Pronomen, die bei Besetzung des Vor-Vorfeldes im Vor- oder Mittelfeld korrekterweise

als Resumptivpronomen identifiziert wurden.

lp-Ebene Auf der lp-Ebene gibt es auch eine Veränderung: Das Label *vvf* bezeichnet das Vor-Vorfeld. In der Reihenfolge der lp-Labels steht es zwischen den nominalen Labels und dem Label *vf*, da es immer vor dem Vorfeld steht und da es auch komplexere nominale Konstituenten beinhalten darf.

6.1.3 Neue Klassen und Veränderungen

id-Ebene In der Definition der Klasse *noun_id* wurde *res* als eingehende Kante definiert, da ein Nomen, welches im Vor-Vorfeld steht, das Resumptivum des Resumptivpronomens sein kann. Dies wurde durch die neue Klasse *respro_id* festgelegt. In dieser Klasse steht, dass das Resumptivpronomen die eingehenden Kanten *subj*, *obj*, *iobj* oder *genobj* bekommen darf und dass *res* eine ausgehende Kante ist.

lp- und id/lp- Ebene Um das Vor-Vorfeld auf der lp-Ebene zu etablieren wurde die Klasse *vvf_lp* eingeführt. Das Label für Verben in Sätzen mit besetztem Vor-Vorfeld wird durch diese Klasse als *v12* definiert, da solche Sätze immer Verb-Zweit-Sätze sind. Als ausgehende Kanten stellt die Klasse eine obligatorische *vvf*-Kante zur Verfügung, sowie eine optionale *mf*-Kante, damit wie in (6.4) *Hans* und *einen Gefallen* auch Konstituenten im Mittelfeld stehen können. Die Klasse *v12_lp* kann von nun an auch *vvf_lp* benutzen anstatt wie bisher nur *fin_lp*.

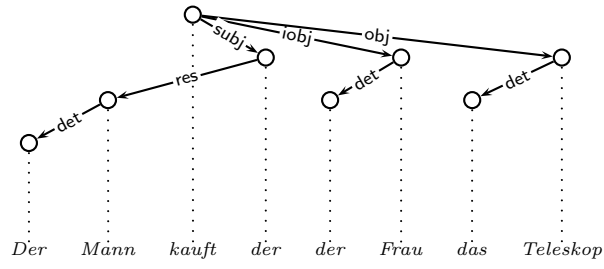
Desweiteren gibt es eine neue Klasse *noun_vvf_lp*, für Nomen, die im Vorfeld stehen, also für die sogenannten Resumptiva. Die Klasse legt fest, dass ein Resumptivum auf der id-Ebene *res* als eingehende Kante bekommt und auf der lp-Ebene *vvf*. Dazu kommt, dass die Klasse *cnoun_lp* nun entweder *noun_lp* oder *noun_vvf_lp* benutzen kann. Auch die Klasse *pnoun_lp* kann von nun an wahlweise auch *noun_vvf_lp* benutzen, da auch Eigennamen im Vor-Vorfeld stehen können.

Es wurde auch die Klasse *respro* für Resumptivpronomen eingeführt. Diese können, wie weiter oben schon beschrieben, sowohl im Vor- als bei Fragesätzen auch im Mittelfeld stehen. Sie werden, wie alle anderen Pronomen auch, als *n* klassifiziert. Als Schnittstelle zum Lexikon erbt die Klasse *respro* von *respro_id* und *respro_lp* und benutzt die Variablen *Word*, *Agrs* und *Cases*.

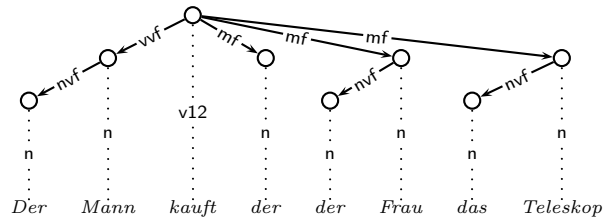
6.1.4 Beispielbäume

Die folgenden Bäume zeigen die aus den Veränderungen resultierenden Analysen für den Beispielsatz (6.5):

6 Besetzung des Vor-Vorfelds



(6.5.id)



(6.5.lp)

7 Passivkonstruktionen

In einem normalen, transitiven Aktiv-Satz ist das Agens das Subjekt und das Patiens Objekt. In (7.1) fungiert *Maria* als Agens und *Hans* als Patiens. Es gibt jedoch auch die Möglichkeit, den in (7.1) ausgedrückten Sachverhalt sozusagen in einem Passiv-Satz auszudrücken. Im Deutschen gibt es zwei verschiedene Formen des Passivs. Die am häufigsten gebrauchte Form ist das Vorgangspassiv, das mit *werden* gebildet wird. (7.2) bietet ein Beispiel für einen Satz mit einer transitiven Vorgangspassiv-Konstruktion. Die Besonderheit solcher Konstruktionen ist, dass Agens und Patiens die grammatikalischen Rollen tauschen. *Hans* wird zum Subjekt und *Maria* wird zur indirekten Objekt-PP, da es in transitiven Passivkonstruktionen kein direktes Objekt gibt. Bei ditransitiven Passivkonstruktionen gibt es sowohl ein Subjekt und zwei indirekte Objekte. Wenn man den von einem ditransitiven Aktiv-Satz wie (7.3) ausgedrückten Sachverhalt in einer Passivkonstruktion (7.4) darstellt, wird das Agens *Maria* zur indirekten Objekt-PP, das Patiens *ein Teleskop* zum Subjekt, wogegen das Beneficiens *dem Mann* auch als indirektes Objekt realisiert wird.

Oft wird das Vorgangspassiv auch dazu benutzt, einen eigentlich transitiven Sachverhalt wie in (7.5) auszudrücken, wenn das Agens entweder nicht bekannt oder für die Aussage nicht relevant ist.

Maria liebt Hans. (7.1)

Hans wird von Maria geliebt. (7.2)

Maria kauft dem Mann ein Teleskop. (7.3)

Dem Mann wird von Maria ein Teleskop gekauft. (7.4)

Hans wird geliebt. (7.5)

* Maria wird von der Mann ein Teleskop gekauft. (7.6)

Die zweite Form des Passivs ist das Zustandspassiv. Diese Konstruktion wird mit *sein* gebildet. Wie der Name impliziert, wird es benutzt um gegenwärtige Zustände wie in (7.7) oder (7.8) auszudrücken.

Das Buch ist gekauft. (7.7)

Die Teleskope sind von dem Mann gekauft. (7.8)

Wenn man *bekommen* und *kriegen* wie Auxiliare verwendet (im folgenden: Pseudo-Auxiliare), kann man noch auf eine weitere Art Passivkonstruktionen bilden. Diese werden jedoch selten in einem anderen Kontext als in (7.9) oder (7.10) benutzt. Sie können

im Gegensatz zu den Passivkonstruktionen mit *werden* und *sein* auch nur in Verbindung mit einigen wenigen Verben benutzt werden, weshalb sie auch in der Grammatik anders als die Passiv-Auxiliare behandelt werden.

Hans bekommt das Teleskop von einer Frau gekauft. (7.9)

Maria kriegt ein Buch versprochen. (7.10)

* Der Mann bekommt geliebt. (7.11)

7.1 Passivkonstruktionen in TDG

7.1.1 Neue Labels

id-Ebene Auf der id-Ebene wurden die Labels *vpass1* und *vpass2* eingeführt. *vpass1* bezeichnet die eingehenden Kanten der Partizipien in Passivkonstruktionen mit *werden* und *sein*, *vpass2* die der Partizipien in Passivkonstruktionen mit den Pseudo-Auxiliaren *bekommen* und *kriegen*.

7.1.2 Neue Klassen und Veränderungen

id-Ebene Als neue Klassen für Passiv-Auxiliare wurden *vppp1* und *vppp2* eingeführt. Diese Klassen bestimmen dass die von den Auxiliaren zu den passivisch verwendeten Perfekt-Partizipien wie *geliebt* in (7.2) verlaufende Kante das Label *vpass1* bzw. *vpass2* erhält.

Um zu erreichen, dass von den Passiv-Auxiliaren *werden* und *sein* ein Subjekt gefordert wird, wurde außerdem die Klasse *vafin_id* eingeführt. Die beiden neuen Klassen *vpass1_id* und *vpass2_id* benutzen die Klasse *nonfin_id*, damit das in der Passivkonstruktion verwendete Partizip Perfekt (gekennzeichnet durch die eingehenden Kanten *vpass1* und *vpass2*) als non-finite VP behandelt wird.

lp- und id/lp- Ebene Bei der Entschlackung der Verben wurde die Unterteilung der finiten Verben aufgehoben, da es bei der damaligen Version der Grammatik keinen Unterschied machte, ob ein finites Verb ein Vollverb, Modalverb oder Auxiliarverb war. Für die Behandlung von Passivkonstruktionen musste jedoch wieder eine eigene Klasse für die finiten Passiv-Auxiliare eingeführt werden, die Klasse *vaf*, welche die Schnittstelle zum Lexikon bietet. Diese benützt die im letzten Abschnitt beschriebene Klasse *vafin_id*. Die Klasse *vafin* integriert die Klassen *vaf* und *vppp1*, und sorgt damit dafür, dass das Auxiliar als finites Verb benutzt wird und dass eine Kante mit dem Label *vpass1* zu einem Partizip Perfekt geht. Die Klasse *vafin* wird wahlweise auch benutzt wenn das Auxiliar für eine Konstruktion mit einem Verbalkomplex wie in (7.12) benötigt wird.

Maria wird heute lachen. (7.12)

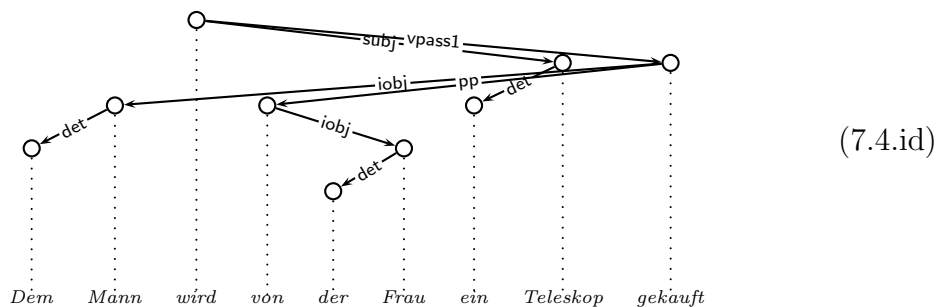
Eine weitere neue Klasse *pseudo_vafin* wurde nur aus dem Zweck eingeführt, die Passivkonstruktion mit Pseudo-Auxiliaren zu ermöglichen. Dabei integriert sie analog zur Klasse *vafin* die Klassen *vaf* und *vpass2*. Im Unterschied zu *vafin* wird sie nur zur Passivbildung verwendet.

7.1.3 Besonderheiten im Lexikon

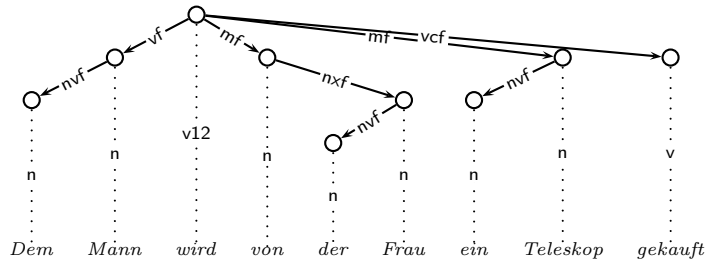
Ein Lexikoneintrag für ein ditransitives Partizip Perfekt das nicht in einer Passivkonstruktion verwendet wird benutzt die Klassen *vvpp*, *ditr_id* und *optiobjc*. Da es, wie in der Beschreibung der verschiedenen Passivkonstruktionen erwähnt, bei der Passivbildung zur Kasusveränderung von Agens und Patiens kommt, können in den Einträgen für ditransitive Partizipien nicht die Klassen *ditr_id* und *optiobjc* gemeinsam benutzt werden. Stattdessen werden die Klassen *intr_id* und *optiobjc* zusammen verwendet. Damit wird beispielsweise erreicht, dass in (7.4) *das Teleskop* als Subjekt und *Hans* als indirektes Objekt erkannt wird. *Maria* wird als indirektes Objekt in einer PP analysiert. Bei denjenigen ditransitiven Partizipien, die zusammen mit den Pseudo-Auxiliaren eine Passiv-Konstruktion bilden können, werden in einem zweiten Eintrag die Klassen *intr_id* und *optobjc* benutzt, da es in diesen Fälle! n immer ein direktes Objekt gibt. Bei transitiven Partizipien, die zur Passivbildung verwendet werden können, wird statt der Klasse *tr_id* die Klasse *intr_id* verwendet. So wird beispielsweise eine korrekte Analyse des Satzes (7.2) gewährleistet.

7.1.4 Beispielbäume

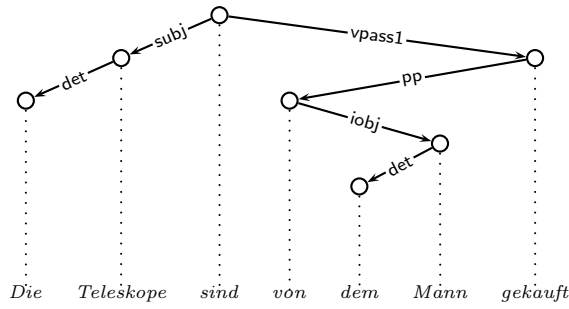
In diesem Abschnitt sollen einige beispielhafte TDG-Analysen von verschiedenen Passivkonstruktionen dargestellt werden. Die Schaubilder (7.4.id) und (7.4.lp) zeigen die id- und die lp-Analyse von Beispiel (7.4), Schaubilder (7.8.id) und (7.8.lp) die von Satz (7.8).



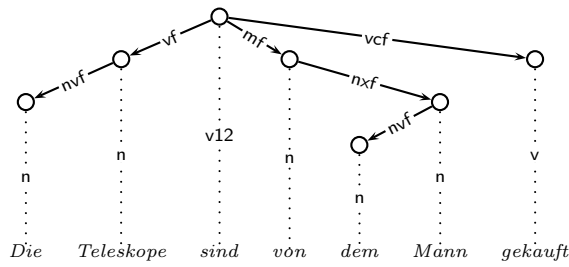
7 Passivkonstruktionen



(7.4.lp)

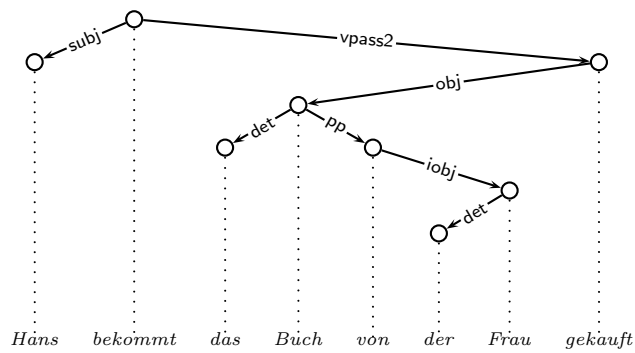


(7.8.id)



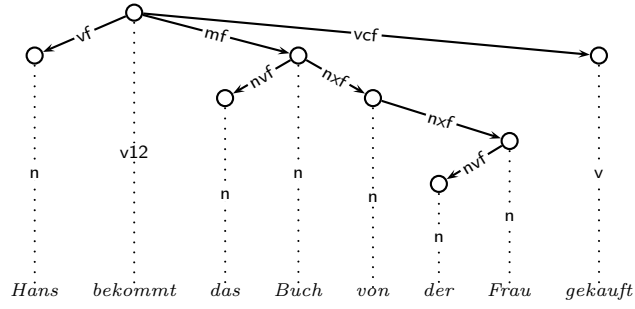
(7.8.lp)

Zum Abschluss dieses Kapitels folgen nun noch beispielhafte Analysen der Sätze (7.9) und (7.10), jeweils mit Darstellung des id- und des lp-Baums.

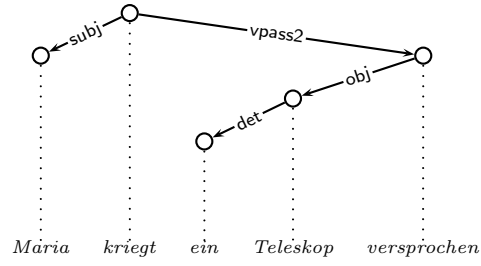


(7.9.id)

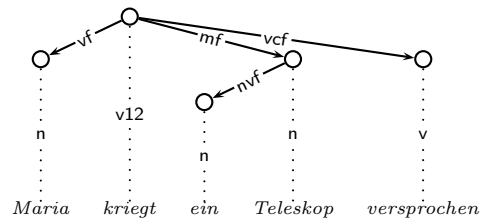
7 Passivkonstruktionen



(7.9.lp)



(7.10.id)



(7.10.lp)

8 Imperative

In der bisherigen Grammatik war es nicht möglich, Imperative korrekt zu analysieren. In diesem Kapitel wird beschrieben, wie einfache Imperative, so wie in den Beispielen (8.1), (8.2) und (8.4), in die Grammatik integriert wurden. Satz (8.4) bietet ein ungrammatikalisches Beispiel.

Glaub mir! (8.1)

Kauf das Buch von der Frau mit dem Mann. (8.2)

Versprecht, das Buch zu lesen. (8.3)

* Glaub der Mann. (8.4)

8.1 Imperative in TDG

8.1.1 Neue Labels

Da ein Imperativ-Satz ein normaler Verb-Erst-Satz ist, mussten keine neuen Labels für die Imperativ-Bildung eingeführt werden.

8.1.2 Neue Klassen und Veränderungen

id-Ebene Auf der id-Ebene wurde die Klasse *vimp_id* eingeführt, um Sätze mit Präpositionalphrasen wie *mit dem Mann* in (8.2) und mehreren Adverbien korrekt zu analysieren.

lp- und id/lp-Ebene Auf der lp-Ebene wurde die Klasse *vimp_lp* eingeführt. Durch die Benutzung dieser Klasse bekommen die imperativen Verben das Label *v12*. Außerdem legt diese Klasse fest, dass das Mittelfeld mehrfach und das Nachfeld optional besetzt werden kann. Natürlich lässt die Klasse auch die Möglichkeit, einen Befehl wie (8.5) korrekt zu parsen.

Schlaf! (8.5)

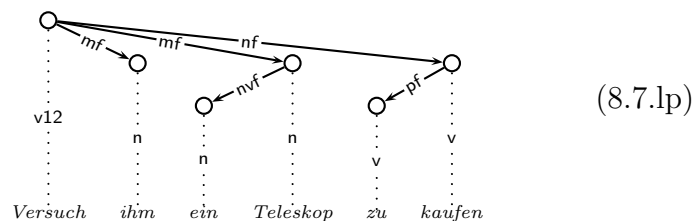
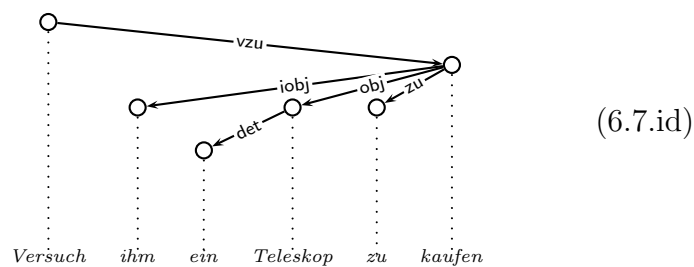
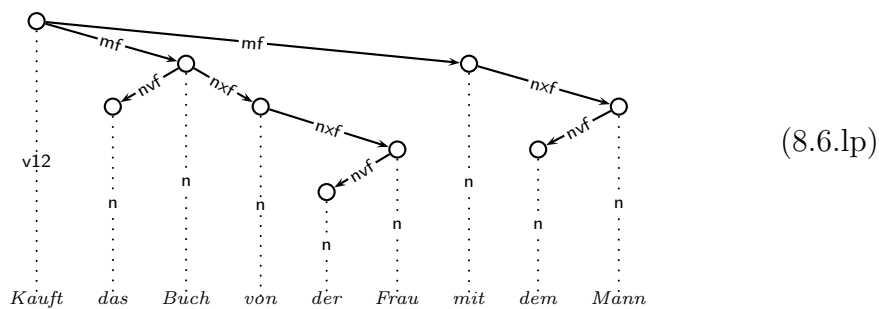
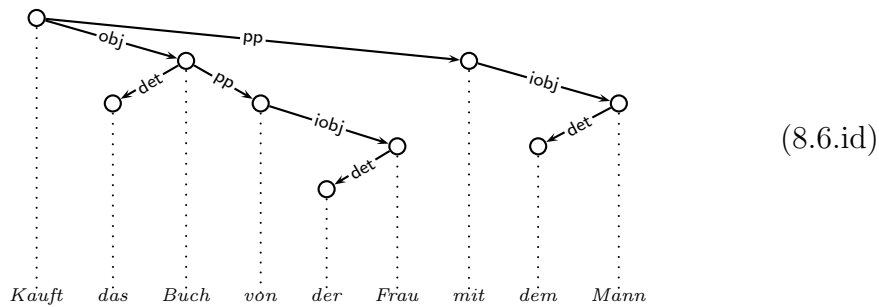
Die Schnittstelle zum Lexikon wird durch die Klasse *vimp* bereitgestellt. Diese Klasse benutzt die Klasse *vimp_id*. Schließlich gibt es noch die Klasse *vfin_imp*, welche sowohl die Klasse *vimp* als auch die Klasse *v12_lp* benutzt. Durch die Benutzung von *v12_lp* werden Imperativ-Sätze als korrekte Verb-Erst-Sätze geparkt.

8.1.3 Besonderheiten im Lexikon

Imperative im Lexikon wie beispielsweise *schlaf* oder *verspricht* benutzen die Klasse *vfn_imp* als auch die passende Transitivitäts-Klasse: *intr_id*, *tr_id* oder *ditr_id*.

8.1.4 Beispieltäume

In diesem Abschnitt sollen einige beispielhafte TDG-Analysen von Imperativ-Sätzen dargestellt werden.



9 Nomina

Eine kleinere Änderung gab es bei den Nomina. Ein Problem der ursprünglichen Grammatik bestand darin, dass Satz (9.1) im Gegensatz zu (9.2) beim Parsen keine Lösung erzeugte. Die Klasse *cnoun_id*, die von allen Nomina im Lexikon benutzt wurde, verlangte generell einen Determiner vor dem Nomen. Dies ist im Singular zwar richtig, wie man an den Beispielen (9.3) und (9.4) sehen kann, gilt jedoch nicht für Nomina im Plural. Im Plural ist der Artikel vor Nomina optional.

Nazis lesen keine Bücher. (9.1)

Die Nazis lesen keine Bücher. (9.2)

Ein Teleskop liest keine Bücher. (9.3)

* Teleskop liest keine Bücher. (9.4)

9.1 Nomina in TDG

9.1.1 Neue Klassen und Veränderungen

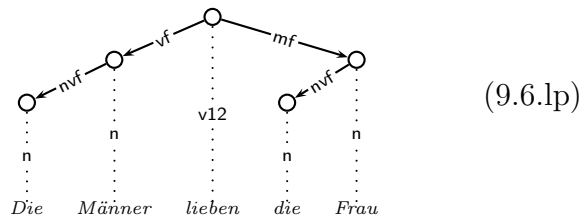
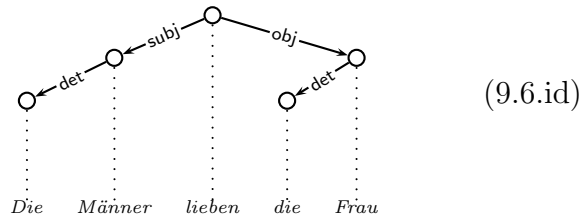
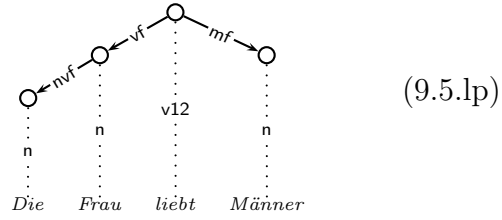
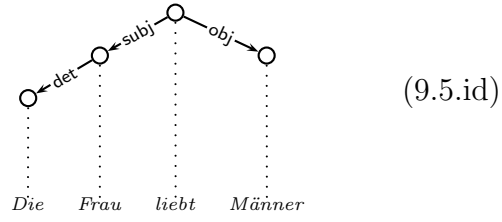
id-Ebene Die Klasse *cnoun_id* wurde durch die beiden Klassen *cnoun_sg_id* für Singular-Nomen und *cnoun_pl_id* für Plural-Nomina ersetzt. Beide erben von der Klasse *noun_id* und bestimmen die gleichen ausgehenden Kanten wie vorher *cnoun_id*. Der Unterschied besteht darin, dass *cnoun_sg_id* die ausgehende Kante *det* als obligatorisch festlegt, während sie bei *cnoun_pl_id* optional ist.

lp- und id/lp- Ebene Auf der lp-Ebene hat der Unterschied zwischen Singular- und Plural-Nomina keine Auswirkung. Auf der id-/lp-Ebene, also der schnittstelle zum Lexikon, wurde *cnoun* wiederum durch die Klassen *cnoun_sg* und *cnoun_pl* ersetzt. *cnoun_sg* erbt dabei logischerweise von *cnoun_sg_id* und *cnoun_pl* von *cnoun_pl_id*. Im Lexikon benutzen Singular- und Plural-Einträge dann die entsprechenden Klassen.

9.1.2 Beispielbäume

Dieser Abschnitt zeigt einige beispielhafte TDG-Analysen von Sätzen, in denen Nomina im Singular und im Plural vorkommen.

9 Nomina



10 Adverbien

10.1 Allgemeines

Das Adverb illustriert einen Umstand, unter dem eine Handlung verläuft. Im Deutschen gibt es verschiedene Adverbtypen, die sich in der Art der Modifikation unterscheiden. Man kennt beispielsweise Lokaladverbien (*hier, da*), Temporaladverbien (*oft, heute*), Modaladverbien (*gern, nicht*) und Kausaladverbien (*also, demnach*). Ferner gibt es noch Satzadverbien, welche sich nicht auf einen Teil, sondern auf die Gesamtaussage eines Satzes beziehen (*leider, dummerweise*).

10.2 Ausgangssituation

In der Ausgangsgrammatik (Debusmann, 2001) konnte ein Verb von allen im Lexikon verzeichneten Adverbien modifiziert werden. Da es hierbei zu unpassenden Anwendungen kommen konnte (10.1 – 10.3), bestand die Aufgabe darin, die Adverbien zu klassifizieren und ihre Anwendungsdomäne sinnvoll einzugrenzen.

10.3 Umsetzung in TDG

Hierzu wurde ein Edgeconstraint für Adverbien eingeführt, welches in analoger Weise zu dem Constraint für Verbpräfixe funktioniert (Debusmann, 2001). Die Idee dahinter besteht darin, dass ein Edgeconstraint bestimmte Eigenschaften einer ausgehenden Kante eines Knotens festlegt. Hierbei muss ein Tochterknoten, welcher an diese Kante anschliesst, in einem Feld von Eigenschaften bestimmte Features aufweisen, die eine Teilmenge dessen darstellen, was im Lexikoneintrag des Mutterknotens festgelegt ist. In diesem Fall stellt das Adverb den Tochterknoten und das Verb den Mutterknoten dar. Bei der Implementierung weist man jedem Adverb einen bestimmten Typ zu, wobei die oben erläuterte Einteilung der Adverbien im Deutschen eine Richtlinie darstellen kann. Man legt gleichzeitig fest, mit welchen Adverbtypen jedes einzelne Verb modifiziert werden darf. Dadurch werden unpassende Anwendungen eines Adverbs verhindert, wie in folgenden Beispielen leicht nachzuvollziehen ist:

* Heute kauft Maria nicht sehr ein. (10.1)

? Maria hat sorgfaeltig versprochen den Mann nicht zu lieben. (10.2)

? Maria hat Peter sorgfaeltig geliebt. (10.3)

10.3.1 Ebenen

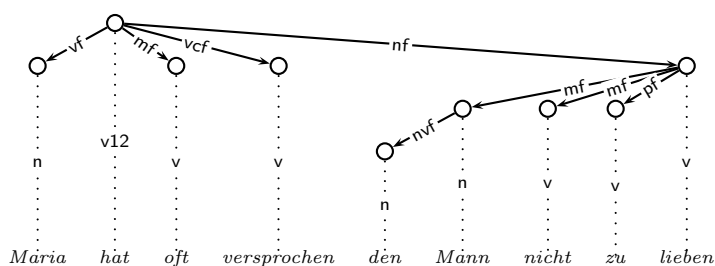
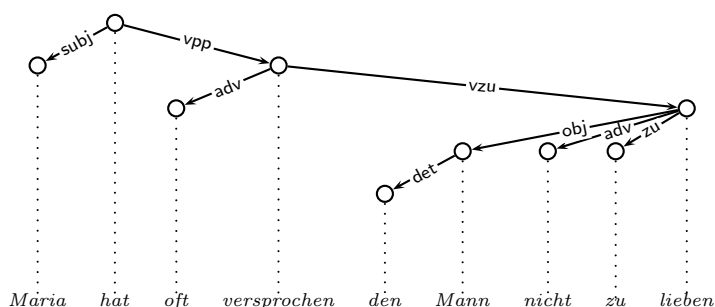
In der *dimension ID* im Header der Grammatik wurde ein Typ *id.adv* spezifiziert, dessen Domäne die fünf Konstanten *adv_1*, *adv_2*, *adv_3*, *adv_4* und *adv_5* beinhaltet. Diese Konstanten stellen die Klassen von Adverbien dar.

In der Grammatik zeigte sich, dass die in Abschnitt 10.1 vorgeschlagene Einteilung nicht exakt wiedergegeben werden konnte, da sich die implementierte Klassifizierung sehr stark an den im Lexikon befindlichen Verben orientierte. Mit dem lexikalischen Attribut *advReq* wird einem Verb eine Menge von Adverbklassen zugewiesen, welche dieses als Modifikatoren akzeptieren kann. Das Attribut *adv* dient dazu, das Adverb in eine der fünf Klassen einzuordnen.

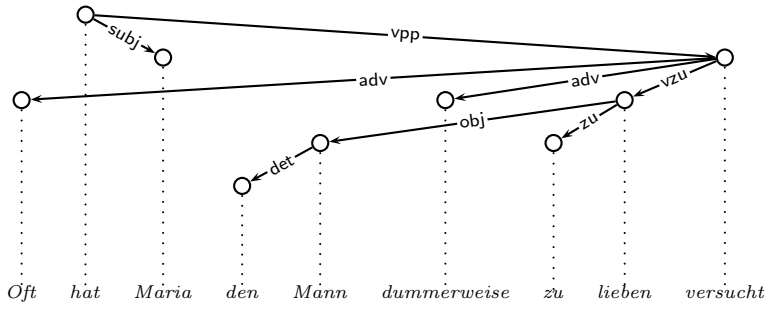
10.3.2 Lexikon

Das Lexikon wurde um die neuen Adverbien *leider*, *dummerweise*, *rasch*, *wieder* und *sorgfältig* erweitert. Allen Adverbien wurde unter dem Attribut *adv* ein Typ zugeteilt. Ferner wurden allen Verben im Lexikon unter dem Attribut *advReq* eine Reihung von Adverbtypen zugeteilt, welche das betreffende Verb als Modifikatoren akzeptieren darf.

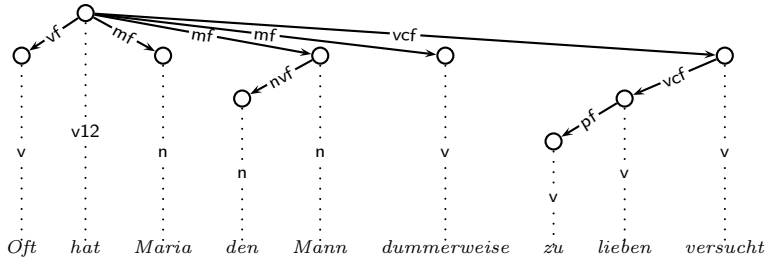
10.4 Ableitungen



10 Adverbien



(10.5.id)



(10.5.lp)

11 Konjunktionen

11.1 Allgemeines

Im Deutschen werden Konjunktionen in zwei Klassen unterteilt, es gibt nebenordnende und unterordnende Konjunktionen. Nebenordnende Konjunktionen (*und, oder*) verbinden zwei Hauptsätze, zwei Nebensätze oder zwei Satzteile miteinander. Unterordnende Konjunktionen dienen als Bindewort zwischen einem Haupt- und einem Nebensatz (*dass, weil*).

11.2 Ausgangssituation

Ursprünglich konnten mit der Grammatik die Konjunktionen *dass, weil* und *obwohl* behandelt werden, jedoch ohne zwischen diesen eine Unterscheidung zu treffen.

11.3 Umsetzung in TDG

11.3.1 Ebenen

Um kausale Konjunktionen wie *weil* und *obwohl* von einfachen unterordnenden Konjunktionen wie *dass* zu unterscheiden, wurde eine Klasse *compcausal* eingeführt, der ausschliesslich die kausalen Konjunktionen angehören. Mit der bereits existierenden Klasse *comp*, welcher von nun an nur noch die einfache Konjunktion *dass* angehört, teilt sich die neue Klasse *compcausal* die LP-Klasse *comp_lp*. In dieser wird festgelegt, dass alle eingehenden Kanten zu Konjunktionen entweder Vorfeldkanten (*vf*) oder Nachfeldkanten (*nf*) sein müssen. Desweiteren stellt man den Konjunktionen über diese Klasse eine ausgehende *nf*-Kante zur Verfügung. Auf der ID-Ebene wurde eine Klasse *compcausal_id* eingeführt, welche den kausalen Konjunktionen eine eingehende Kante bereitstellt, die mit dem neuen Label *compcausal* versehen ist. Analog zur Klasse *comp_id* wird in der Klasse *compcausal_id* für kausale Konjunktionen eine ausgehende Kante *sub* eröffnet, welche einen untergeordneten Satz einleitet.

Letztendlich steuert man den Zugriff eines bestimmten Verbes auf die beiden Typen von Konjunktionen über die Klassen *compc* und *compcausalc*. Dadurch wird dem Verb entweder eine optionale ausgehende Kante für eine kausale Konjunktion (*compcausalc*) oder eine optionale Kante für eine einfache Konjunktion (*compc*) oder beides vererbt.

11.3.2 Lexikon

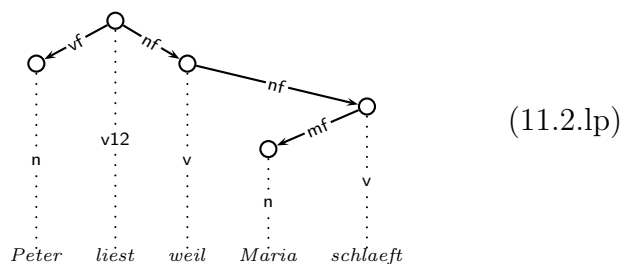
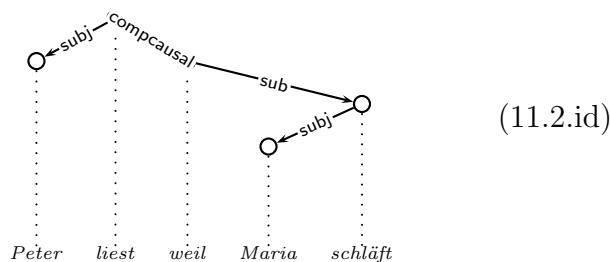
Die Konjunktionen *dass*, *weil* und *obwohl* waren schon vorher im Lexikon. Sie mussten lediglich in die in Abschnitt (11.3.1) beschriebenen Klassen eingeteilt werden. Darüberhinaus musste für die Verben bestimmt werden, von welcher der beiden Klassen jedes einzelne erben sollte.

11.4 Ableitungen

Es ist beispielsweise grammatikalisch nicht korrekt, dass das deutsche Verb *schlafen* einen Nebensatz mit der Konjunktion *dass* einleiten kann:

* Maria schläft dass Peter liest. (11.1)

In Folge dessen stellt man sicher, dem Verb *schlafen* keine ausgehende Kante für diese Konjunktion mitzugeben. Im Gegensatz dazu sind kausale Konjunktionen ausgehend vom Verb *schlafen* möglich, wodurch beim Lexikoneintrag dieses Verbs von der Klasse *compcausal* geerbt werden kann (Abb. 11.2.id/lp).



Man erkennt in (11.2.id/lp), dass das Verb *schläft* auf der ID-Ebene eine ausgehende Kante für eine kausale Konjunktion (*compcausal*) erhält. In der Abbildung wird ersichtlich, dass der mit *weil* eingeleitete Nebensatz gemäss der Satztopologie im Nachfeld des Hauptsatzes positioniert wird.

12 Belebte vs. unbelebte Nomen

12.1 Allgemeines

Im Zuge der Arbeit mit der TDG-Grammatik sollte es sich als praktisch herausstellen, belebte von unbelebten Nominalphrasen unterscheiden zu können. Auf diese Weise kann die Besetzung der Argumentposition eines Verbes gesteuert werden. Dadurch werden sinnlose Ableitungen, beispielsweise die Besetzung des Agens durch eine unbelebte Nominalphrase, ausgeschlossen.

12.2 Umsetzung in TDG

Als Beispiel wurde hier das Verb *geben* gewählt, welches bis zu drei Nominalargumente verlangt. Es leuchtet ein, dass es sich beim Subjekt um eine belebte Nominalphrase handeln muss, wohingegen das direkte Objekt eine unbelebte NP verlangt. In den vorgestellten Beispielen wurde für das indirekte Objekt eine belebte NP gewählt, da die Grammatik für diese Position eine belebte NP verlangt. Es ist jedoch zu beachten, dass auch eine unbelebte Nominalphrase für dieses Argument von *geben* denkbar ist, wie folgendes Beispiel zeigt, das von der Grammatik nicht akzeptiert wird.

Maria gibt dem Haus einen Anstrich. (12.1)

Die Unterscheidung zwischen belebter und unbelebter Nominalphrase konnte in der Grammatik nur für Vollverben realisiert werden. Bei Auxiliarkonstruktionen weiß das Auxiliarverb nicht, ob seine nominalen Argumente belebt oder unbelebt sein sollen. Zur vollständigen Umsetzung des Belebtheitsphänomens müsste in der Grammatik eine neue Ebene verwendet werden, welche semantische Abhängigkeiten im Satz widerspiegelt.

12.2.1 Ebenen

Zur Unterscheidung zwischen belebten und unbelebten Nominalphrasen wurde auf der ID-Ebene zunächst der Typ *anim* definiert, welcher die Werte *anim* (belebt) und *nonanim* (unbelebt) annehmen kann. Um zu gewährleisten, dass die Belebtheitsanforderung des Verbes mit der des sich in Argumentposition befindlichen Substantivs übereinstimmt, wurde ein Nodeconstraint (Debusmann und Duchier, 2004) für *anim* implementiert. Von nun an muss jedem Nomen im Lexikon analog zu den Werten für *Agrs* und *Cases* ein Wert für das Attribut *animated* übertragen werden. Dieser kann nur aus den beiden Werten des Typs *anim* bestehen.

Um die Belebtheitsanforderungen für ein Verb zu spezifizieren, wurden exemplarisch die Klassen *ditranim* und *subjanim* ins Leben gerufen. Die Klasse *subjanim* stellt hierbei die Forderung des Verbs nach einem belebten Subjekt dar. Die Klasse *ditranim* spezifiziert exakt die Belebtheitsanforderungen, wie sie beispielhaft für das Verb *geben* im Abschnitt 12.2 beschrieben wurden.

12.2.2 Veränderungen in den bestehenden Klassen

Die bestehenden Nomenklassen *cnoun_sg*, *cnoun_pl*, *pname* und *pred_noun* mussten um das Attribut *animated* erweitert werden.

12.2.3 Lexikon

Im Lexikon wurden alle Nomen um das Attribut *animated* erweitert. Das Attribut wurde mit dem zum Nomen passenden Wert für belebt oder unbelebt belegt. Exemplarisch erben das Partizip des Verbs *geben* (*gegeben*) und die dritte Person Singular dieses Verbs (*gibt*) von der Klasse *ditranim*.

12.3 Ableitungen

Im Folgenden seien Beispiele aufgeführt, welche die Grammatik gemäß der oben beschriebenen Subkategorisierungsanforderungen des Verbs *geben* als ungrammatisch identifiziert hat:

* Das Teleskop gibt Maria das Buch. (12.2)

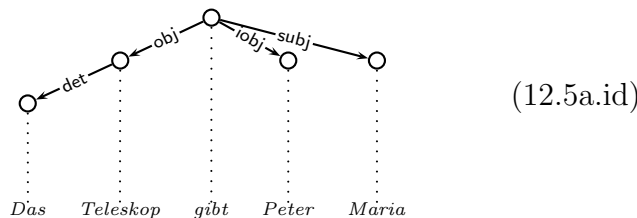
* Maria gibt dem Teleskop das Buch. (12.3)

* Maria gibt Peter dem Mann. (12.4)

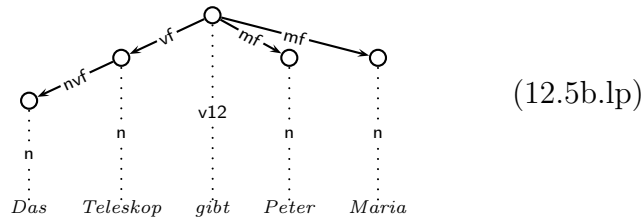
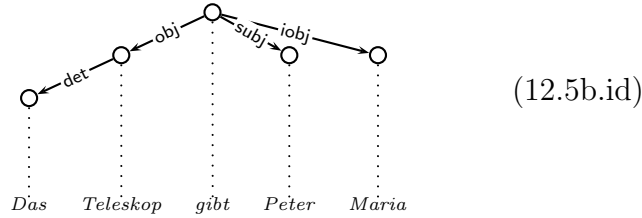
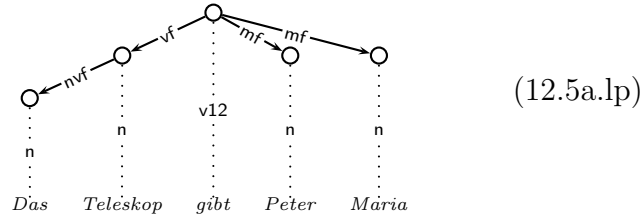
Für den Satz

Das Teleskop gibt Peter Maria. (12.5)

bietet die Grammatik zwei Lösungen an (12.5a.id/lp und 12.5b.id/lp).



12 Belebte vs. unbelebte Nomen



Hierbei handelt es sich um die beiden einzig denkbaren Möglichkeiten, bei denen entweder Peter die Rolle des Subjekts und Maria die Rolle des indirekten Objekts einnimmt oder aber Peter das indirekte Objekt und Maria das Subjekt darstellt.

13 Kopula & Prädikative

13.1 Allgemeines

Als Kopulaverb bezeichnet man ein Verb, welches das Subjekt des Satzes mit dem sogenannten Prädikativ verknüpft. Es bildet zusammen mit dem Prädikativ das mehrteilige Prädikat des Satzes und ordnet dem Substantiv eine bestimmte Eigenschaft zu. Die wichtigsten Kopulaverben sind *sein* und *werden*. Das Prädikativ wird in der Regel durch ein Adjektiv oder ein Nomen repräsentiert:

Maria wird Ärztin. (13.1)

Hans ist treu. (13.2)

13.2 Umsetzung in TDG

13.2.1 Ebenen

Auf der ID-Ebene erbt die neue Klasse für das Kopulaverb (*verb_cop*) von der Klasse *verb_cop_id*. Darin wird festgelegt, dass das Kopulaverb als eingehende Kanten *sub* und *rel* annehmen darf. Als ausgehende Kanten werden dem Kopulaverb ein obligatorisches Subjekt (*subj*) und optional ein prädikatives Adjektiv *adj_cop* sowie optional ein prädikatives Nomen (*pred_noun*) mitgegeben. Über *outcases* legt man den Kasus für das Subjekt und das Prädikativnomen auf nominativ fest.

Auf der LP-Ebene erbt *verb_cop* von der Klasse *verb_cop_lp*. Darin werden die eingehende Kante *nf* und die ausgehenden Kanten *rvf*, *vf* und *mf* bestimmt. Je nach Position im Satz erhält die Kante des Kopulaverbs das Label *v* oder *v12*.

Um vom Kopulaverb ausgehende Kanten zum Prädikativ zu etablieren, wurden auf der ID-Ebene die Labels *pred_noun* für das prädikative Nomen und *adj_cop* für das prädikative Adjektiv integriert. Die neue Klasse *adj_cop* erbt von den Klassen *adj_cop_id* und *adj_cop_lp*. Da in Sätzen wie

Maria ist des Mordes schuldig. (13.3)

Hans ist dem Mann boese. (13.4)

das prädikative Adjektiv für ein Objekt subkategorisieren kann, legt man dies in *adj_cop_id* als optional ausgehende Kanten *genobj* und *iobj* fest.

Prädikativ verwendete Nomen erben von der Klasse *pred_noun*. In der ID-Klasse *pred_noun_id*, von der *pred_noun* erbt, legt man die Kante *pred_noun* als eingehende Kante für das Prädikativnomen fest.

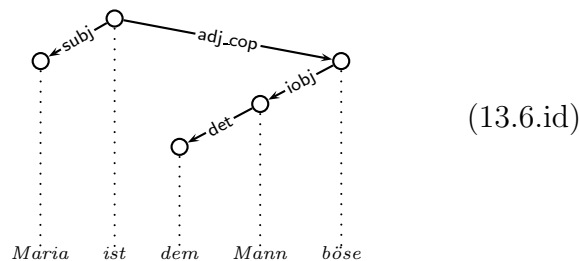
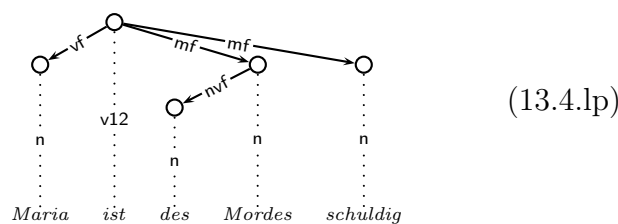
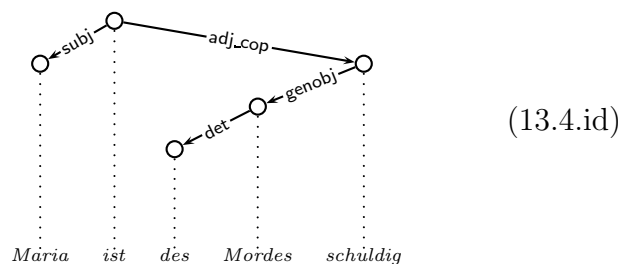
Da auch dieses von einem Substantiv modifiziert werden kann, integriert man in *pred_noun_id* eine entsprechende ausgehende Kante. In unserem Fall wurde beispielhaft ein Kante für einen Modifizierer im Genitiv (*genmod*) implementiert. In der LP-Klasse *pred_noun_lp* wird sicher gestellt, dass das Prädikativnomen im Mittelfeld steht.

13.2.2 Lexikon

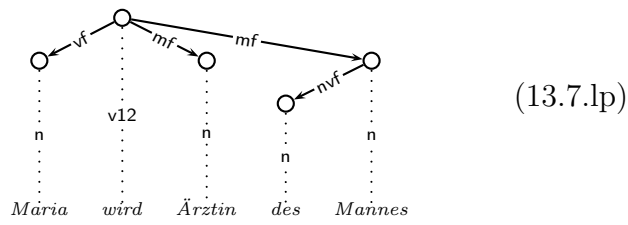
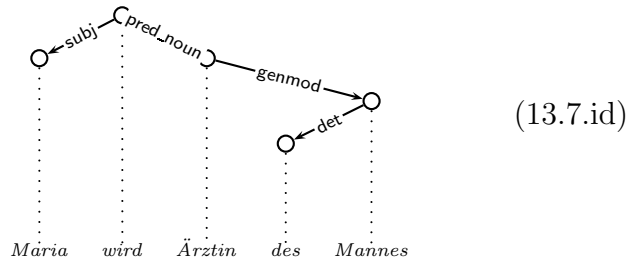
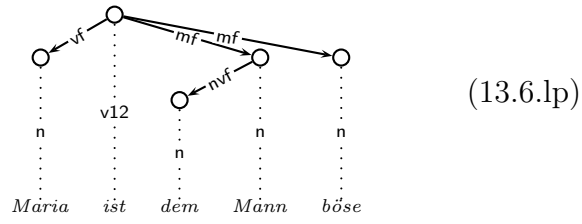
Als prädikative Adjektive wurden *schuldig* und *böse* ins Lexikon aufgenommen. Die Substantive *Ärztin* und *Arzt* stellen die prädikativen Nomen dar. Als Kopulaverben wurden beispielhaft die Flektionen *ist* vom Verb *sein* und *wird* vom Verb *werden* implementiert.

13.3 Ableitungen

Die folgenden Ableitungen sollen zeigen, wie das Phänomen der Kopulaverben und Prädikative vom TDG-Formalismus behandelt wird. Beispiel (13.4.id/lp) zeigt die Ableitung des prädikativen Adjektivs *schuldig* mit einem Genitivobjekt, Abbildung (13.6.id/lp) das Adjektiv *böse* mit einem Dativobjekt. In der Ableitung (13.7.id/lp) wurde ein prädikatives Nomen mit einem Genitivmodifizier geparkt.



13 Kopula & Prädikative



14 Fragesätze

14.1 Allgemeines

Man unterscheidet allgemein bei Fragesätzen zwischen Entscheidungsfragen und Ergänzungsfragen. Entscheidungsfragesätze, auch Satzfragesätze genannt, können in der Regel mit *ja* oder *nein* beantwortet werden. Hierbei wird nach dem Wahrheitsgehalt des in der Äußerung erwähnten Sachverhaltes gefragt (Drosdowski et al., 1995).

Kaufen Hans und Maria ein? (14.1)

Versucht Peter heute den Roman zu lesen? (14.2)

Ergänzungsfragen oder Wortfragen richten sich nach einem Aspekt des ausgedrückten Sachverhaltes, nicht der gesamte im Satz ausgedrückte Sachverhalt ist fraglich. Sie werden durch ein Interrogativpronomen (*wer, wen*) oder ein Interrogativadverb (*wo, warum, wann*) eingeleitet.

Wer gibt Maria ein Versprechen? (14.3)

Warum liebt Maria Hans? (14.4)

Wann versucht Peter heute den Roman zu lesen? (14.5)

14.2 Ausgangssituation

In der Ausgangsgrammatik konnten Entscheidungsfragen behandelt werden, da die hierzu benötigte Verb-Erst-Stellung schon in der Verbklasse (*v12_lp*) auf der LP-Ebene integriert war.

14.3 Umsetzung in TDG

In der Grammatik wurden Ergänzungsfragen mit Interrogativpronomen von Ergänzungsfragen mit Interrogativadverb unterschieden. Die unterschiedliche Behandlung der beiden Fälle wird dadurch notwendig, dass Interrogativpronomen eine Argumentposition des Verbs besetzen, also entweder als Subjekt, direktes Objekt oder indirektes Objekt fungieren. Interrogativadverbien hingegen stellen eine zusätzliche Argumentkante dar, die vom Verb zur Verfügung gestellt werden muss.

14.3.1 Ebenen

Hierzu wurde auf der ID-Ebene das Label *wh_quest* eingeführt, welches eine Kante zu einem Interrogativadverb ausweist. Desweiteren wurde die Klasse *optquest* implementiert, die für Verben eine optionale Kante für ein Interrogativadverb bereitstellt. Die Interrogativadverbien finden sich in der Klasse *quest_oa*, die von der Klasse *quest_oa_lp* die Eigenschaft erbt, dass die Interrogativadverbien im Vorfeld stehen.

Auf der ID-Ebene erbt die Klasse *quest_oa* von *quest_oa_id*. Darin wird festgelegt, dass eine eingehende Kante vom Typ *wh_quest* sein muss.

Für die Interrogativpronomen wurde die Klasse *quest* eröffnet. Diese legt durch Erbung von *quest_lp* das Interrogativpronomen topologisch auf das Vorfeld fest.

Auf der ID-Ebene werden über die Klasse *quest_id* die Kanten *subj*, *obj* und *iobj* als eingehende Kanten für das Pronomen an *quest* vererbt. Weiterhin stellt die Klasse *quest* sicher, dass das Interrogativpronomen seine Agreement-Features über die Werte aus den Parametern *Agrs* und *Cases* seines Lexikoneintrages erhält.

14.3.2 Lexikon

Ins Lexikon wurden die Interrogativadverbien *warum* und *wo* eingetragen. Desweiteren wurden die Interrogativpronomen *wer*, *wem*, *wen* und *was* integriert. Diese reichen jeweils über die Parameter *Agrs* und *Cases* ihre Agreement-Features weiter.

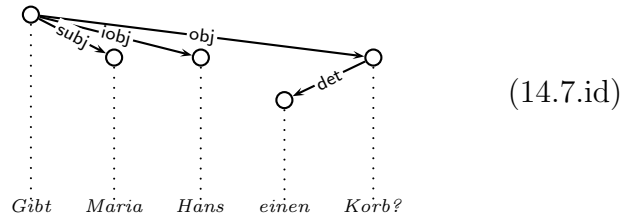
14.4 Ableitungen

Abbildung (14.7.id/lp) zeigt die Ableitung einer Entscheidungsfrage, wie sie schon in der Ausgangsgrammatik möglich war. Die Ableitung einer Ergänzungsfrage mit einem Interrogativpronomen lässt sich in Abbildung (14.8.id/lp) nachvollziehen. Für die Ableitung des Satzes

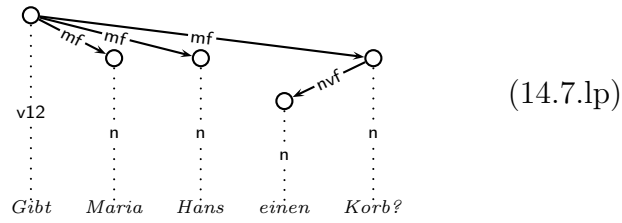
Warum gibt Hans Maria ein Buch? (14.6)

zeigt sich wiederum der praktische Nutzen des in Kapitel 12 eingeführten Belebtheitsconstraints. Die Grammatik bietet für diesen Satz vier Lösungen an, wobei sich bei zwei von ihnen zeigt, dass hier *Buch* als Subjekt fungiert. Nach der Einführung des Belebtheitsconstraints fallen diese beiden Lösungen weg, da das Verb *geben* kein unbelebtes Nomen in Agensposition binden kann. Die beiden nun verbliebenen Lösungen können in (14.6a.id/lp) und (14.6b.id/lp) miteinander verglichen werden. Wie sich herausstellt, wird *Buch* in beiden Ableitungen als direktes Objekt eingesetzt, wohingegen *Hans* und *Maria* jeweils als Subjekt oder indirektes Objekt abgeleitet werden können.

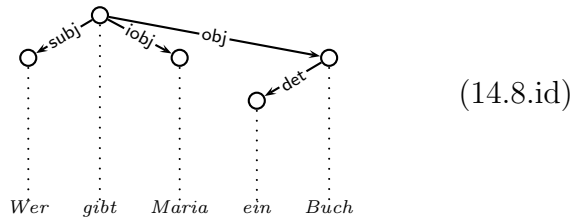
14 Fragesätze



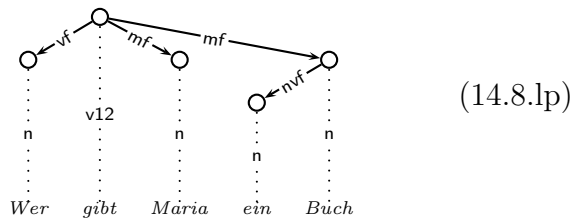
(14.7.id)



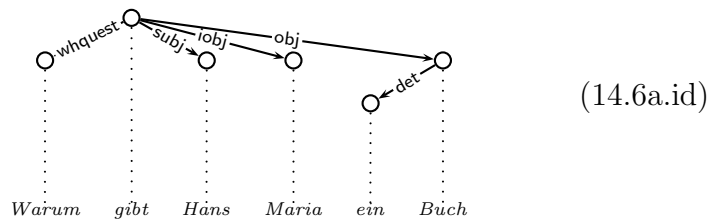
(14.7.lp)



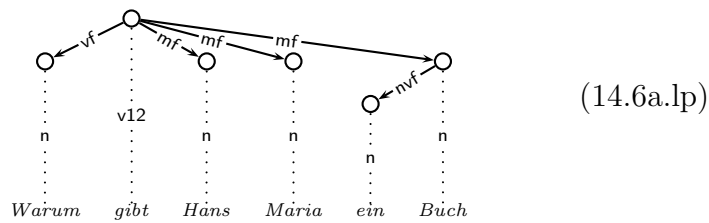
(14.8.id)



(14.8.lp)

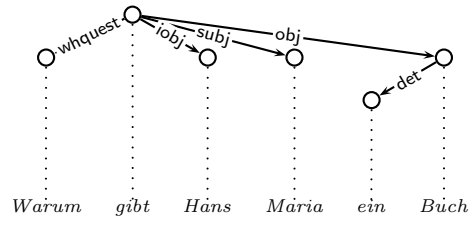


(14.6a.id)

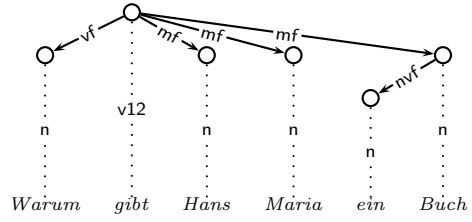


(14.6a.lp)

14 Fragesätze



(14.6a.id)



(14.6b.lp)

15 Das Phänomen Es

Es lassen sich neben dem *es* als sächliches Pronomen noch zwei weitere Arten von *es* unterscheiden: Explikativa und Platzhalter. Die beiden letzten wurden neu in die Diplomgrammatik implementiert.

15.1 Explikatives es:

Im Deutschen erfordert eine kleine Gruppe von Verben obligatorisch ein *es*, welches frei in der Satzstellung ist.

Es regnet. (15.1)

* Regnet. (15.2)

Heute regnet es. (15.3)

* Heute regnet. (15.4)

15.2 Platzhalter-es:

Ausserdem erfordern bestimmte Satzkonstruktionen ein *es*. Das *es* ist in diesem Fall unabhängig vom Verb oder Adjektiv, wenn jedoch die Satzglieder umgestellt werden, fällt das *es* weg.

Diese Gruppe kann noch einmal unterteilt werden in Vorfeld-es und Korrelat-es. Beim Vorfeld-es hat das *es* nur die Aufgabe, im Satz die erste freie Stelle vor dem finiten Verb zu besetzen.

Es liebt ein Mann eine Frau. (15.5)

Es haben keine Frauen gelacht. (15.6)

Beim Umstellen der Satzglieder wie in (15.7a/b) fällt das *es* weg.

Es schläft heute eine Frau. (15.7a)

Heute schläft eine Frau. (15.7b)

Beim Korrelat-es steht diese Partikel als Platzhalter eines anderen Satzgliedes. Dieses Satzglied kann u.a. ein zu-Infinitivsatz (15.8) oder ein dass-Nebensatz (15.9) sein.

Es ist notwendig, zu schlafen. (15.8)

Ist es notwendig, dass peter heute schläft? (15.9)

Es ist notwendig, dass er rasch das buch liest. (15.10)

Das *es* dient in diesem Fall als Scheinsubjekt und ist relativ frei in der Satzstellung. Das Satzglied, wofür das *es* steht wird *repeated element* genannt. Wenn das *repeated element* seine "vorgesehene Funktion" im Satz einnimmt, fällt das *es* weg, siehe Beispiel (15.11a/b).

Es ist notwendig, das Buch zu lesen. (15.11a)

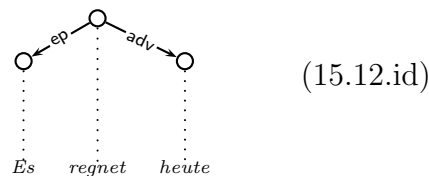
Das Buch zu lesen, ist notwendig. (15.11b)

Obwohl das explikative *es* fast inhaltlich leer ist, wird es im Duden als Subjekt geführt und kann somit dependenzgrammatisch ohne Probleme dargestellt werden. Anders liegt es bei Platzhalterkonstruktionen. Hier ist das *es* semantisch leer, es hat nur eine syntaktische Funktion. Dessenungeachtet wird in unserer Analyse auch das Platzhalter-*es* als syntaktischer *Dependent* des Verbs dargestellt.

15.3 TDG-Darstellung:

15.3.1 Explikatives *es*:

Für das explikative *es* wurde auf der *id*-Ebene das neue Label *ep* eingeführt. Desweiteren wurde die Grammatik mit einer neuen Verbklasse *fin_expl* und der *es*-Klasse *expl_id* ergänzt. Die Verbklasse *fin_expl* beinhaltet eine kleine Gruppe von Verben wie *regnet*, *hagelt* oder *schneit*, wovon *regnet* als neuer Lexikoneintrag eingeführt wurde. Da diese Verben obligatorisch ein *es* erfordern, haben sie auf der *id*-Ebene genau ein *ep* auf der ausgehenden Kante stehen. Natürlich erlaubt die Verbklasse auch optional Adverbien oder Präpositionalphrasen. Umgekehrt akzeptiert das explikative *es* nur explikative Verben und da das explikative *es* keine *Dependents* haben kann, ist die ausgehende Kante leer. Der *id*-Baum für den Satz *es regnet heute* (15.12) sieht also folgendermaßen aus:



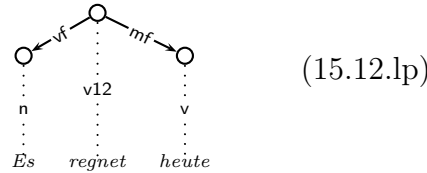
Auf der *id*-/*lp*-Ebene wurde die Verbklasse *expl_verb* durch eine Kombination vorhandener Verbklassen definiert, wodurch das *es* theoretisch fast überall im Satz erscheinen kann. Diese Vorgehensart war notwendig, um explikative *es*-Konstruktionen auch in untergeordneten Sätzen oder Fragesätzen zu ermöglichen wie in (15.13) und (15.14).

Peter glaubt, dass es heute regnet. (15.13)

Regnet es heute? (15.14)

Die Einschränkung der möglichen Felderbestzungen für das explikative es erfolgt schließlich dadurch, dass die es-Klasse *expl_lp* nur die Felder *vf* und *mf* akzeptiert.

Der folgende lp-Baum ergibt sich für den Satz *es regnet heute*:



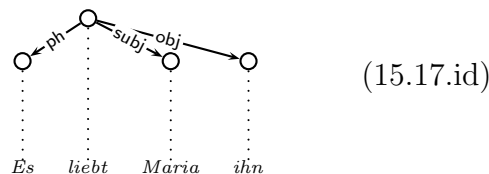
15.3.2 Vorfeld-es:

Für das Vorfeld-es und Korrelat-es wurde ein gemeinsames id-Label *ph* bereitgestellt, da es sich in beiden Fällen um Platzhalterkonstruktionen handelt. Auf der id-Ebene wurden zwei neue Klassen eingeführt: die Verbklasse *fin_es_id* und die es-Klasse *ph_vf_id*. Die beiden Klassen greifen ineinander, da die Verbklasse obligatorisch ein *ph* fordert und die es-Klasse nur ein *fin_es_id*-Verb akzeptiert. Da es sich beim Vorfeld-es eigentlich um eine stilistische Variante handelt (s. Beispiel 15.15, 15.18), kann es theoretisch mit jedem Verb konstruiert werden.

Ein Mann lacht. (15.15)

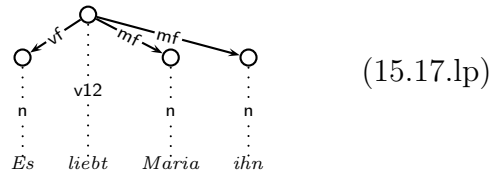
Es lacht ein Mann. (15.16)

Damit aber im Lexikon nicht jedes Verb noch einmal als mögliches *fin_es*-Verb definiert werden muss, wurde auf der id-/lp-Ebene, also der Schnittstelle zum Lexikon, bei der Definierung der Verbklasse *vvfin* eine Disjunktion eingeführt. Die Klasse *vvfin* benutzt nun entweder die Klasse *fin_id* oder *fin_es_id*, wobei *fin_es_id* sich nur durch das *ph* auf der ausgehenden Kante von *fin_id* unterscheidet. Der id-Baum für den Satz *Es liebt Maria ihn* (15.17) sieht folgendermaßen aus:



Auf der lp-Ebene wurde festgelegt, dass die Klasse *ph_vf_lp* (Vorfeld-es) nur das Vorfeld, also die erste Position im Satz, besetzen kann. Eine neue Verbklasse war auf der lp-Ebene nicht nötig, da die existierenden ohne Veränderung alle nötigen Felder für Vorfeld-es-Konstruktionen bereitstellen.

Für den Satz *Es liebt Maria ihn* ergibt sich folgender lp-Baum:



15.3.3 Korrelat-es:

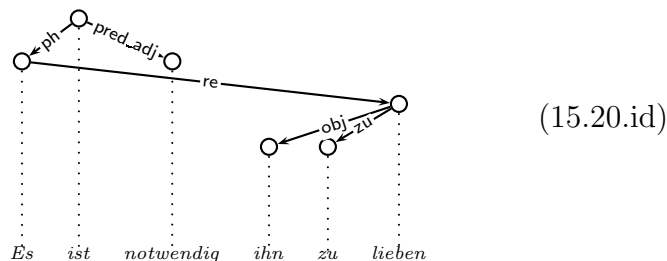
Hierfür wurden neben dem id-Label *ph* noch zwei neue id-Labels eingeführt: *pred_adj* und *re*. Auf der id-Ebene kamen als neue Klassen *aux_es_id*, *pred_adj_id* und *ph_korr_id* hinzu. Möglicherweise ließe sich die Einführung der Klassen *aux_es_id* und *pred_adj_id* vermeiden, da sie jeweils eine Untermenge der Auxiliärverben bzw. Adjektiven darstellen, aber für einen besseren Überblick wurden sie zunächst getrennt definiert. Die Klasse *aux_es_id* besteht aus Auxiliärverben (*ist*, *wird*), die als syntaktische Dependents ein Korrelat-es und ein prädikatives Adjektiv fordern. Im Gegensatz zu den "normalen" Auxiliärverben erlauben sie aber kein Subjekt. Die Klasse *pred_adj_id* steht für Adjektive, die im prädikativen Gebrauch mit einem Korrelat-es vorkommen können, wie in (15.18).

Es ist notwendig, dass er einen Mord begeht. (15.18)

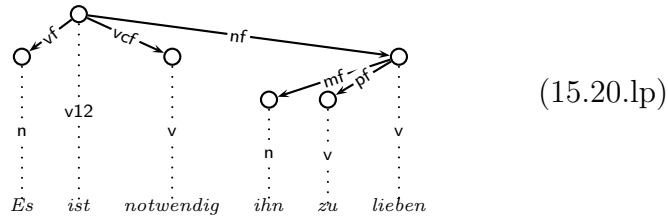
* Es ist treu, dass er einen Mord begeht. (15.19)

Der syntaktische Kopf der Satzkonstruktion (15.20) ist das Auxiliärverb *ist*. Seine syntaktischen Dependents sind das Platzhalter-es (*ph_korr_id*) und das prädikative Adjektiv *pred_adj_id*. Diese erscheinen als obligatorische Funktionen auf der ausgehenden Kante des Auxiliärverbs. Der optionale Nebensatz in (15.20) wird in der Literatur *repeated element* genannt, da es den semantischen Inhalt vom Platzhalter-es trägt, welches nur eine syntaktische Funktion im Satz hat. Dependenzgrammatisch ist das *repeated element* mit dem Label *re* der Dependent vom Platzhalter-es *ph_korr_id*, also erscheint das Label *re* optional auf der ausgehenden Kante vom Platzhalter-es (*ph_korr_id*). Es wurde auch eine Klasse *rep_element* definiert. Auf der id-Ebene akzeptiert das *repeated element* ein *re*, und fordert optional ein *zu* oder ein *sub*, d.h. einen zu-Infinitivsatz oder einen dass-Nebensatz. Dafür wurde die eingehende Kante der Klasse *nonfin_id* mit *re* ergänzt und die unterordnende Konjunktion *dass* als *repeated element* im Lexikon definiert.

Somit entsteht der folgende id-Baum für den Satz *Es ist notwendig, ihn zu lieben* (15.20):



Auf der lp-Ebene waren die Veränderungen vergleichsweise gering. Das Korrelat-es (*korr_es_lp*) akzeptiert *vf* und *mf*. Die Klasse *pred_adj_lp* akzeptiert *vcf* und *aux_es_lp* bietet *vf* *mf* *vcf* und *nf* an. Da die optionalen Nebensätze immer im Nachfeld stehen, und dies nur von *aux_es_lp* bereitgestellt wird, rutscht der Nebensatz nach oben zum Auxiliarverb. Der lp-Baum für den Satz *Es ist notwendig, ihn zu lieben* sieht so aus:



Zuletzt wurden im Lexikon die Einträge *notwendig*, *ist* und *wird* neu definiert.

16 Koordination

Die Implementierung von Koordinationen ist aus zwei Gründen problematisch. Erstens gibt es eine Fülle von möglichen Koordinationen: es können zwei bis unendlich viele Phrasen, ineinander geschachtelte Phrasen oder Phrasen mit verschiedenen Kategorien koordiniert werden. Zweitens ist die interne Struktur der Koordinationen in der Dependenzgrammatik sehr umstritten: eigentlich kann keine der Phrasen aber auch kein Konjunkt der Kopf sein. Für die Diplomgrammatik wurden also zunächst nur Koordinationen mit 2-3 gleichen Phrasen bzw. Sätzen eingebaut und die Konjunktion wird als syntaktischer Kopf der Koordination analysiert. Es wurden folgende id-Labels in die Diplomgrammatik eingebaut: *kon* für die Konjunktion, *cadj*, *cadv*, *cgenmod*, *cgenobj*, *ciobj*, *cobj cp*, *csent* und *csubj* für die einzelnen Konjunkte. Als lp-Labels kommen *coord_sent1*, *coord_sent_2* für Sätze, *coord1*, *coord2* für Objekte und Adverbien, *nmxf* und *cnmf* für Adjektive hinzu. In die Diplomgrammatik wurde nur die Konjunktion *und* als Lexikoneintrag eingeführt.

Die Koordination verläuft prinzipiell immer nach dem gleichen Prinzip. Auf der id-Ebene wird die ursprüngliche syntaktische Kategorie auf der Kante zur Koordination beibehalten. Die Konjunktion wird zum syntaktischen Kopf der Koordination und die Konjunkte nennen sich *c* gefolgt vom ursprünglichen syntaktischen Namen (das *c* steht für koordiniert). Die einzige Ausnahme davon ist die Koordination von Sätzen, da hier die Konjunktion keine eingehende Kante hat. Auf der lp-Ebene steht die Koordination im Vorfeld oder im Mittelfeld, die Konjunktion ist der topologische Kopf der Koordination und die Konjunkte vor der Konjunktion nennen sich *coord1*, das Konjunkt nach der Konjunktion *coord2*. Ausgenommen davon sind die Koordination der Genitivattribute und die Konstruktion P NP KON NP, wo die Koordination im Feld *nmf* steht. Ausgenommen ist auch die Adjektivkoordination, hier steht die Koordination im *nmf* Feld. Noch eine Ausnahme bilden die Satzkoordinationen, wo die Konjunktion gar keine eingehende Kante hat.

Zuerst wird die Koordination bei Objekten und Adverbien detailliert beschrieben. Bei den anderen Koordinationen wird nur noch auf die Unterschiede dazu eingegangen.

16.1 Koordinierte Objekte und Adverbien

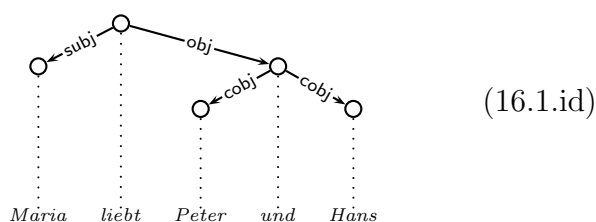
Maria liebt Peter und Hans. (16.1)

Maria gibt Peter und Hans einen Korb. (16.2)

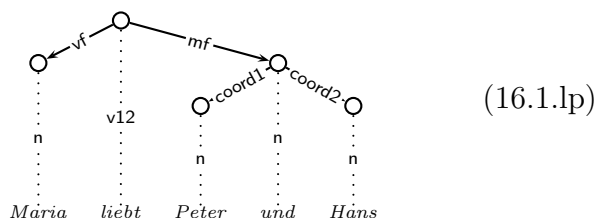
Peter harrt der Männer und Frauen. (16.3)

Heute und morgen kaufen sie keine Teleskope. (16.4)

Die Koordination für Akkusativ-Dativ-Genitivobjekte und für Adverbien verläuft nach dem gleichen Prinzip. Auf der id-Ebene wird die Konjunktion *kon* zum syntaktischen Kopf. Die Konjunkte sind die syntaktischen Dependents und nennen sich jeweils *cobj*, *ciobj*, *cgenobj* oder *cadv*. Auf der id-Ebene wurden die und-Klassen *con_obj_id*, *con_iobj_id*, *con_genobj_id* und *con_adv_id* definiert. Diese haben jeweils das Label *obj*, *iobj*, *genobj* oder *adv* auf der eingehenden Kante und *cobj*#{2 3}, *ciobj*#{2 3}, *cgenobj*#{2 3}, *cadv*#{2 3} auf der ausgehenden Kante. Dies ermöglicht die Koordination von wahlweise 2 oder 3 Elementen. Die Klassen *cnoun_sg_id*, *cnoun_pl_id*, *pnoun_id* und *perpro_id* wurden jeweils dahingehend verändert, dass sie nun auch die Konjunkte *cobj*, *ciobj*, *cgenobj* und *cadv* akzeptieren. Der id-Baum von Satz (16.1) sieht so aus:



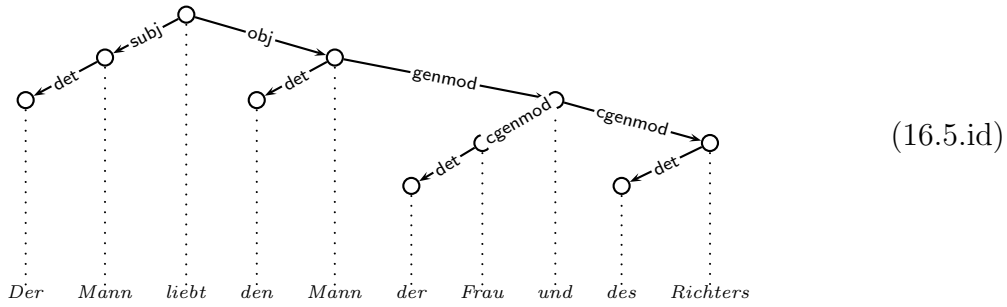
Auf der lp-Ebene wird die Koordination der Objekte und Adverbien durch die Klasse *con_general_lp* bewirkt. Das *con_general_lp* hat auf der eingehenden Kante *vf* und *mf* stehen, ausserdem fordert es 1 oder 2 *coord1* und genau ein *coord2*. Die eingehenden Kanten von *cnoun_lp*, *pnoun_lp* und *perpro_lp* werden mit den Labels *coord1* und *coord2* ergänzt. Hier folgt der lp-Baum zu Satz (16.1):



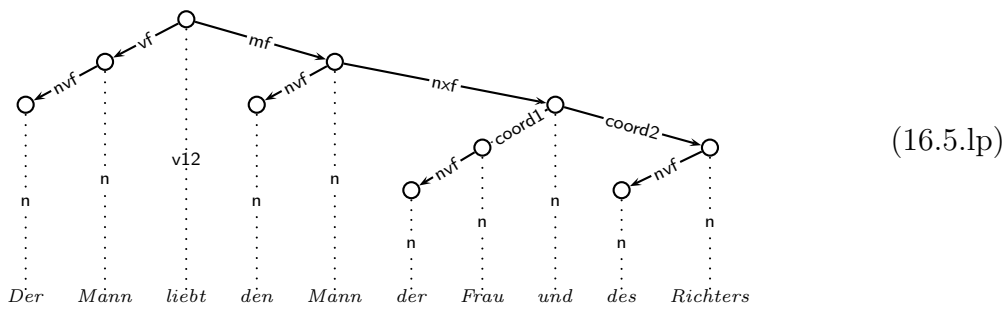
16.2 Koordinierte Genitivattribute

Der Mann liebt den Mann der Frau und des Richters. (16.5)

Die Koordination des *genitive modifiers* verläuft auf der id-Ebene genauso wie die Objekte. Der einzige Unterschied ist, dass auf der lp-Ebene das Genitivattribut im Feld *nxf* steht. Aus diesem Grund wird auf der lp-Ebene eine eigene und-Klasse *con_genmod_lp* definiert, die als eingehende Kante *nxf* akzeptiert. Hier folgt der id-Baum zu Satz (16.5):



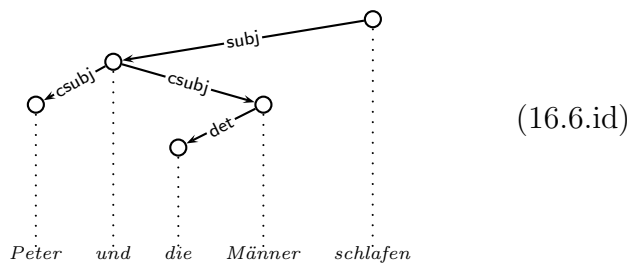
Der dazugehörige lp-Baum sieht so aus:



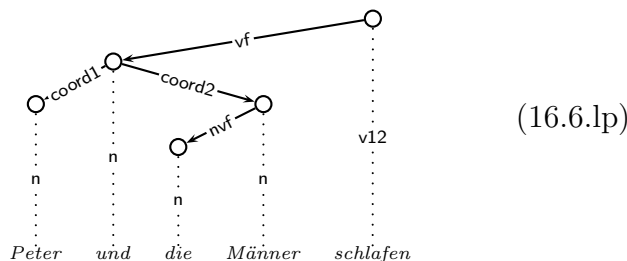
16.3 Koordinierte Subjekte

Peter und die Männer schlafen. (16.6)

Die Koordination der Subjekte geschieht sowohl auf der id- wie auch auf der lp-Ebenen wie bei den Objekten. Es muss aber zusätzlich sichergestellt werden, dass bei koordinierten Subjekten das Verb immer im Plural steht. Hierfür wurde auf der id- /lp- Ebene bei der Definition der Klasse *con_subj* ein agreement constraint eingeführt. Hier folgt der id-Baum zu Satz (16.6)



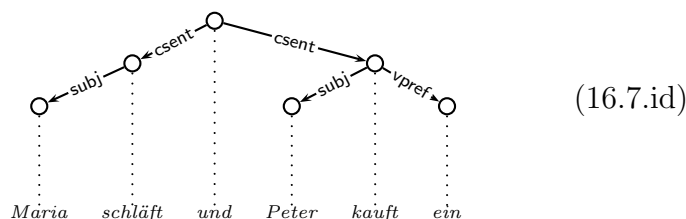
Der lp-Baum zu Satz (16.6):



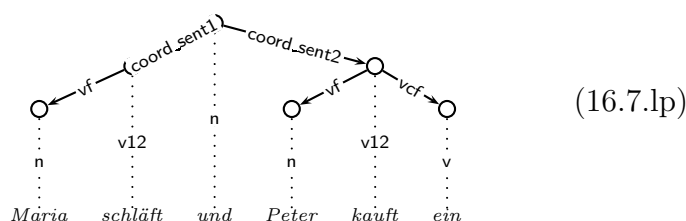
16.4 Koordinierte Sätze

Maria schläft und Peter kauft ein. (16.7)

Das Problem bei der Satzkoordination liegt darin, dass normalerweise das finite Verb der Kopf des Satzes ist. Liegt eine Satzkoordination vor, wird aber die Konjunktion zum Kopf des zusammengesetzten Satzes. Aus diesem Grund wird auf der id-Ebene die eingehende Kante der Klasse *fin_id* mit *csent* ergänzt und die eingehende Kante der Konjunktion stellt nur ein *nf* bereit, womit auch Sätze wie (16.8) analysiert werden können. Der id-Baum zu Satz (16.7) sieht so aus:



Auf der lp-Ebene wurden die lp-Label *coord_sent1* und *coord_sent2* statt *coord1* und *coord2* wie bei den Objekten benutzt. Und hier der lp-Baum zu Satz (16.7):



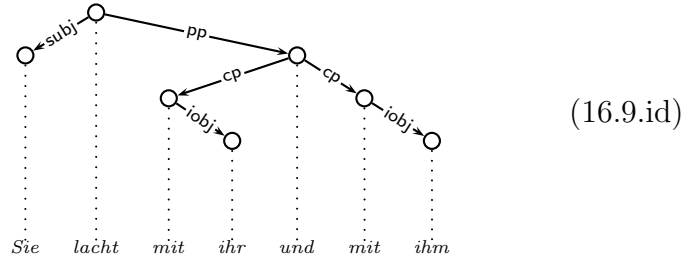
16.5 Koordinierte Präpositionalphrasen

Es wurden in der Diplomgrammatik zwei Arten von Koordinationen mit Präpositionalphrasen implementiert, siehe (16.9) und (16.10). Der Satz (16.9) entspricht der Konstruktion PP KON PP und wird auf der id- und lp-Ebene genauso geparkt wie koordinierte Objekte. Der Satz (16.10) entspricht der Konstruktion P NP KON NP. Auf der id-Ebene ist auch dieser Satztyp genauso zu analysieren, wie die Koordination der Objekte. Auf der lp-Ebene wird der Satz wiederum so analysiert wie die Koordination der Genitivattribute, d.h. das *und* hat auf der eingehenden Kante ein *nof*.

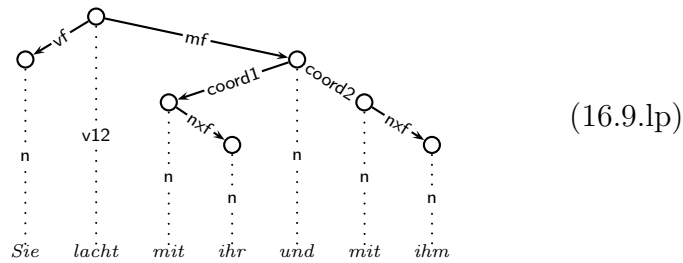
Sie lacht mit ihr und mit ihm. (16.9)

Maria lacht mit dem Mann und der Frau. (16.10)

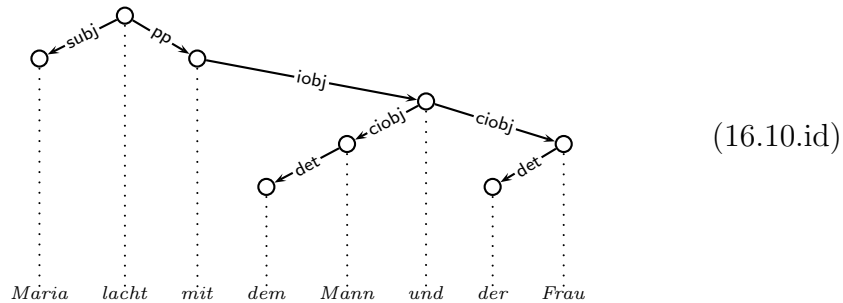
Hier der entsprechende id-Baum zu (16.9):



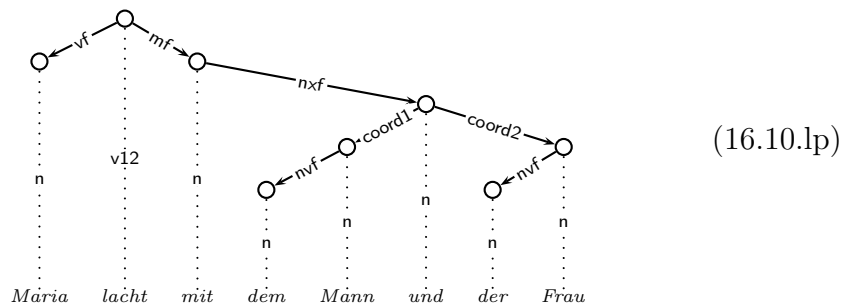
Der lp-Baum zu (16.9):



Der id-Baum zu (16.10):



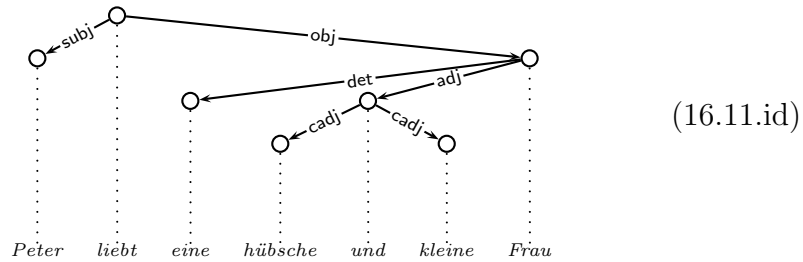
Der lp-Baum zu (16.10):



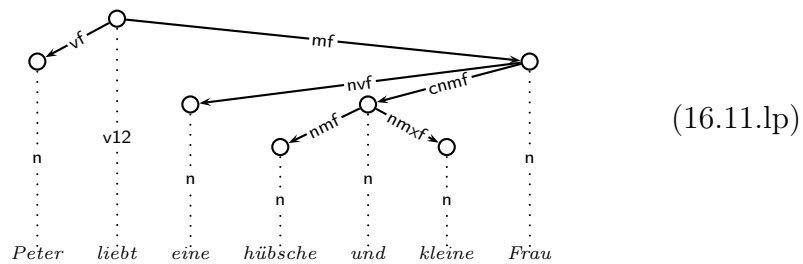
16.6 Koordinierte Adjektive

Peter liebt eine hübsche und kleine Frau. (16.11)

Auf der id-Ebene verläuft die Implementierung genauso wie bei der Koordination der Objekte, auf der lp-Ebene wurden jedoch neue lp-Label (*cnmf*, *nmxf*) eingeführt, um die Ähnlichkeit zu den ursprünglichen Namen zu gewährleisten. Siehe den id-Baum zu Satz(16.11):



Dies ist der lp-Baum zu Satz (16.11):



17 Schluss

In der vorliegenden Arbeit wurde eine Auswahl von syntaktischen Phänomenen im Deutschen mit dem TDG-Grammatiksystem realisiert. Ausgehend von der Diplomgrammatik von Ralph Debusmann (2001) wurden gezielt Verbesserungen und Erweiterungen gesucht und implementiert. Das System wurde hierbei um grundlegende syntaktische Problemstellungen wie dem Passiv ausgebaut. Auch wurden wichtige funktionale Verbesserungen in der Grammatik erzielt, beispielhaft sei hier die Vereinfachung und Umstrukturierung der Verbklassen erwähnt.

Im Zuge der Arbeit stellte sich das TDG-Grammatiksystem als relativ leicht zu erlernendes syntaktisches Werkzeug heraus. Auch wenn die zugrundeliegenden Constraints, welche die Grammatik benutzt, ein erhöhtes mengentheoretisches sowie grammatikalisches Verständnis erfordern, so kann die intuitive Syntax der Notation schnell erlernt werden, sodass der Benutzer in einer existierenden Grammatik sehr schnell Erweiterungen implementieren und Verbesserungen erzielen kann. Als weiteres Positivum stellte sich die Modularität der Grammatik heraus. Das getrennte Bearbeiten der einzelnen Phänomene und anschließende Zusammenlegen der Grammatikteile erwies.

Jedoch konnten nicht alle in dieser Arbeit vorgestellten Phänomene problemfrei implementiert werden. Vereinzelt generiert die Grammatik unerwünschte Lösungen, die sich aber wegen dem Zusammenhängen verschiedener Problemfelder und Thematiken innerhalb der Grammatik nicht unterdrücken lassen, beispielsweise die Unterscheidung zwischen belebter und unbelebter NP bei Auxiliaren.

Zu erwähnen sei an dieser Stelle noch der von Denis Duchier und Ralph Debusmann entworfene TDG-Parser, der für Windows- und Linuxsysteme verfügbar ist und neben einem benutzerfreundlichen Interface praktische Zusatzfunktionen, wie zum Beispiel der Ausgabe der Ableitungsbäume in Latex-Code, bieten kann.

Ein Themenbereich, der in dieser Arbeit nicht untersucht wurde, ist die Ableitung syntaktisch-semantischer Strukturen wie Anhebung und Kontrolle. Zur Analyse solcher Abhängigkeiten können dem TDG-System neben der ID- und LP-Ebene noch weitere Betrachtungsebenen hinzugefügt werden, welche in der Lage sind, tief im Satz eingebettete Abhängigkeitsbeziehungen abzuleiten und darzustellen. Dies würde zur Behandlung von Raising-Phänomenen oder zur schon erwähnten Lösung des Belebtheitsproblems bei Auxiliaren benötigt.

Für zukünftige Entwicklungen in TDG sei an dieser Stelle auf die weitere Arbeit von Ralph Debusmann verwiesen.

18 Literaturverzeichnis

Debusmann, R.(2001). *A declarative grammar formalism for dependency grammar*. Diplomarbeit, Saarbrücken.

Debusmann, R. & Duchier, D.(2004). *Manual of the XDG Parser*.

<http://www.canoo.net>.

Kathol, A.(2000). *Linear Syntax*. Oxford Linguistics. Oxford University Press, Oxford.

Müller, S.(1999). *Deutsche Syntax deklarativ*. Max Niemeyer Verlag, Tübingen.