

Syntax natürlicher Sprachen

CIS, WS 2010/11

Hans Leiß

Universität München
Centrum für Informations- und Sprachverarbeitung

17. Februar 2011

Plan

Wortart

Distribution

CFG

Konstituente

Valenz

DCG

Funktion

CG

MG

Felder

GF

Plan

- ▶ Historischer Überblick
- ▶ Strukturalismus: Bloomfield, Harris, Chomsky
- ▶ Phrasenstrukturgrammatiken: CFG, TAG, MCFG
- ▶ Kategorialgrammatik: CG (Ajdukiewicz, Bar-Hillel, Lambek)
- ▶ Verallgemeinerte Phrasenstrukturgrammatik: GPSG, HPSG
- ▶ Typentheoretische Grammatik: Montague, GF

Organisatorisches:

- ▶ Kursseite: <http://www.cis.uni-muenchen.de/~leiss/syntax-10-11/>
- ▶ Tutorin: Katharina Groth (katharina-groth@web.de)
- ▶ Fragestunde: Di 16-18, Hauptgebäude M 210
- ▶ Klausur: Do 3.2.2011, 14-16 Uhr

Probleme

- ▶ Wozu brauchen wir Grammatik?
- ▶ Zweck und Form der Sprachbeschreibung
- ▶ Wie (un)abhängig ist die Form der Ausdrücke von der Bedeutung und dem Äußerungszweck?
- ▶ Was sind syntaktische Einheiten?
- ▶ Was sind schlüssige Argumente im Bereich der Syntax?
- ▶ Allgemeinheit oder Genauigkeit der Sprachbeschreibung?
- ▶ Was sind „Gesetze“ des Sprachaufbaus?
- ▶ Gibt es eine den Sprachen gemeinsame „Universalgrammatik“ ?

Grammatik des Deutschen

- ▶ Rektion und Kongruenz
- ▶ Valenz
- ▶ Nominalphrasen
- ▶ Feldstruktur der einfachen Sätze
- ▶ Adjektive und Vergleichskonstruktionen
- ▶ Infinitivkonstruktionen
- ▶ Koordinierungen

Zeitplan (?)

1. Woche: Aristoteles: Satz (Subj+Präd), Ausdruck (Kategorien)
2. Woche: Wortarten und -Formen (Beispiel: Thrax)
3. Woche: Satz schemata: Kategorienfolgen
4. Woche: Konstituentenbegriff (Bloomfield) + -Tests
5. Woche: Phrasenstruktur (Chomsky) + Transformationen; CFG
6. Woche: Struktur- vs. Funktionsbegriffe
7. Woche: Felderstruktur des einfachen Satzes im Deutschen
8. Woche: Anhebungs-/Kontrollverben, Koordination, Vergleiche
9. Woche: CG 1: Funktor+Argument, Beweiskalküle, Anhebung
10. Woche: CG 2: Beweisterme, Semantik, Montague-Grammatik
11. Woche: HPSG: Merkmalstrukturen (SynSems), Regelschemata
12. Woche: GF 1: Abstrakte+konkrete Syntax + Übersetzung
13. Woche: GF 2: Unzusammenhängende Konstituenten
14. Woche: Klausur

Literatur

GPSG

1. Gerald Gazdar, Ewan Klein, Geoffrey Pullum, Ivan Sag: Generalized Phrase Structure Grammar. Blackwell 1985 (Eine Erweiterung der CFGs um Merkmalstrukturen, Typen, Merkmalvererbungsprinzipien, Metaregeln.)

HPSG

1. Stefan Müller: Head-Driven Phrase Structure Grammar. Eine Einführung. Stauffenburg Verlag, 2007 (HPSG angepaßt an das Deutsche, mit subtilen Argumenten, aber weniger klar/übersichtlich im Formalismus.)
2. Ivan Sag, Thomas Wasow, Emily Bender: Syntactic Theory, 2nd ed. CSLI-LN, Stanford, 2003. (Bringt die Grundstruktur der HPSG gut, ist aber auf das Englische zugeschnitten.)

CG (Categorial Grammar)

1. Michael Moortgat: (s. Handbook of Logic and Language)
2. Mark Steedman: Combinatory Categorial Grammar
3. Glynn Morrill: Type Logical Grammar. 1994

GF (Grammatical Framework)

1. Aarne Ranta: Type-Theoretical Grammar. Oxford UP, 1994.
2. <http://www.cs.chalmers.se/~aarne/GF/> (Programme)
3. Aarne Ranta: Grammatical Framework. Stanford 2010

Allgemein:

1. Ursula Klenk: Generative Syntax. Narr, Tübingen 2003. (Guter Überblick über G.formalisten TG, GPSG, HPSG, LFG)
2. Christa Dürscheid: Syntax. Grundlagen und Theorien. 4.Auflage. Vandenhoeck & Ruprecht, 2007.
3. Robert Borsley: Syntax-Theorie. Ein zusammengefaßter Zugang. Niemeyer, Tübingen 1997.
4. Stefan Müller: Grammatiktheorie. Stauffenburg, Tübingen 2009.

Grammatiken: (ggf. neuere Ausgaben nehmen)

1. Gerhard Helbig, Joachim Buscha: Deutsche Grammatik. Ein Handbuch für den Ausländerunterricht. 17.Auflage. Langenscheidt, 1996
2. Peter Eisenberg: Grundriß der Deutschen Grammatik. Metzlersche Verlagsbuchhandlung, Stuttgart 1986.
3. Günter Drosdowski. DUDEN: Grammatik der deutschen Gegenwartssprache. 4.Auflage. Bibl.Institut, Mannheim 1984.
4. Uwe Engel: Deutsche Grammatik. Groos, Heidelberg, 1991.
5. E.Hentschel, H.Weydt: Handbuch der deutschen Grammatik. de Gruyter, 1990.

Prüfungsvorbereitung:

1. Hans Altmann, Suzan Hahnemann: Prüfungswissen Syntax. Vandenhoeck und Ruprecht, 4.Auflage, 2010
2. Karin Pittner, Judidth Berman: Deutsche Syntax. Ein Arbeitsbuch. narr studienbücher, 3.Auflage 2008,

Einführung: Syntax in der Computerlinguistik

Grundfragen:

1. Wie dürfen sprachliche Ausdrücke zu größeren Ausdrücken zusammengesetzt bzw. in kleinere zerlegt werden?
2. Was sind die „Einheiten“, die zusammengesetzt werden?
Welche Kriterien entscheiden darüber, ob mehrere Ausdrücke eine syntaktische (grammatische) Einheit bilden?

Abgrenzung von der Semantik:

- ▶ Syntax betrachtet den Aufbau von Ausdrücken nach „formalen“ Kriterien, „abgesehen“ von der Bedeutung.

Abgrenzung von der Pragmatik:

- ▶ Syntax betrachtet hauptsächlich, ob eine Zusammensetzung (im geeigneten Kontext) überhaupt „möglich“ ist, nicht, wie üblich sie ist.

Grammatik

Ziel der Grammatik ist, die Regelmäßigkeiten (und Ausnahmen) des Aufbaus von Ausdrücken genau zu beschreiben

Grammatik =

$\left\{ \begin{array}{l} \text{Gesamtheit der Regeln zum Aufbau von Ausdrücken } \oplus \\ \text{Gesamtheit der Regeln zur Umwandlung von Ausdrücken} \end{array} \right.$

Nutzen einer Grammatik:

1. Angabe eines Verfahrens, den Aufbau von Ausdrücken zu bestimmen
2. am Aufbau eines Ausdrucks soll (mit Zusatzinformation aus dem „Kontext“) seine Bedeutung berechnet werden

In der Computerlinguistik:

Benutzereingabe → *Syntaktische Analyse*
→ *Fehlermeldung*
| *Semantische Auswertung* → *Bedeutung*

Wörter

Eine (deskriptive vs. normative) Grammatik beschreibt die (vorkommende vs. erlaubte) Zusammenstellung (*Syntax*) von Wörtern zu Ausdrücken (oder „syntaktischen Einheiten“).

Wörter sind die kleinsten in der Syntax betrachteten „Einheiten“.

Aber: Wörter können selbst aus Wörtern zusammengesetzt sein (Komposita), und daher ist die Grenze etwas willkürlich:

Autofahrer = Fahrer eines Autos

Wörter (Lexeme) sind Abstrakta: dasselbe Wort kann (endlich viele) verschiedene *Wortformen* haben, und eine Wortform hat beliebig viele *Vorkommen* (Token).

Wörter sind i.a. *bedeutungstragende* Einheiten.

Wortformen und Wortarten

Man unterscheidet einige wenige *Wortarten* (oder *Redeteile*, *parts of speech* (POS)). Die Einteilung erfolgt i.a. sowohl nach den Formen als auch nach der Bedeutung und dem Verwendungszweck.

Jede Wortart hat eine endliche Menge von *abstrakten Formen*. Jedes Wort der Wortart hat entsprechende *konkrete Formen*, die das *Paradigma* des Worts bilden:

Paradigma : *abstrakte Formen* → *konkrete Formen*

Die abstrakten Formen sind oft nach *Formdimensionen* gegliedert.

Beispiel: Formen der Nomen im Deutschen = Kasus × Numerus

abstrakte Formen			sg	pl
(nom,sg)	(nom,pl)	nom	Haus	Häuser
(gen,sg)	(gen,pl)	gen	Hauses	Häuser
(dat,sg)	(dat,pl)	dat	Haus	Häusern
(akk,sg)	(akk,pl)	akk	Haus	Häuser

Aristoteles: logische Analyse von Ausdrücken

Aristoteles (384 -322 v.Chr.) unternimmt in den Kategorien (κατηγορίαι) und der Hermeneutik (περι ἑρμηνείας) [1] eine Untersuchung der sprachlichen Ausdrücke nach logischen Gesichtspunkten.

Er unterscheidet explizit zwischen Nomina und Verben, wobei nur der Nominativ des Nomens und das Präsens des Verbs als Benennung (ὄνομα) und Tätigkeitswort (ῥήμα) gelten.

Von der *bedeutungsvollen Rede* (λόγος, φωνή σημαντική) wird die (wahre) *Aussage* (λόγος ἀποφαντικός) ausgesondert und allein untersucht. Behauptung und Verneinung (κατάφασις und ἀπόφασις) werden unterschieden und als einfachste Aussagen von den zusammengesetzten unterschieden.

(Kap.5) Der Hauptbestandteil einer Aussage ist das Verb; Namen allein bilden keine aussagende Rede.

Wortarten bei Aristoteles

Def (16a,19) Ein (*einfaches* (ἁπλως)) *Nennwort* (ὄνομα, Nomen) ist ein Ausdruck, der

1. eine (durch Konvention festgelegte) Bedeutung hat,
2. keine Zeitmarkierung trägt,
3. in Verbindung mit „ist“, „war“ und „wird sein“ etwas Wahres oder Falsches bedeutet, und
4. dessen Teile keine Bedeutung haben.

Def (16b,6) Ein *Aussagewort* (ῥῆμα, Verb) ist ein Ausdruck, der

1. eine Bedeutung hat,
2. eine Zeitmarkierung trägt, die das Jetzt-Vorliegen angibt,
3. ein Zeichen für das ist, was von etwas anderem ausgesagt wird, und
4. dessen Teile keine Bedeutung haben.

Nur der Nominativ bei Nomina und nur das Präsens bei (finiten) Verben gelten als Nennwort bzw. Aussagewort; jede andere Form ist eine Abwandlung (πτῶσις) davon.

Für andere Wortarten gibt Aristoteles keine Definitionen an.

Aus 10.20 geht hervor, daß **Adjektive** in Verbindung mit ἔστω ein Verb bilden, in Sätzen wie (ἔστι) δίκαιος ἄνθρωπος.

In Kap. 12 (21b,5-10) wird sogar die finite Form des Verbs mit Hilfe des Partizips auf diesen Fall zurückgeführt: οὐδὲν γὰρ διαφέρει εἰπεῖν ἄνθρωπον βαδίζειν ἢ ἄνθρωπον βαδίζοντα εἶναι.

(Es macht keinen Unterschied zu sagen, „der Mensch geht“ und „der Mensch ist ein Gehender“.)

Ein *unbestimmtes* (ἄοριστον) Nenn- bzw. Aussagewort ist ein Nenn- bzw. Aussagewort mit vorangestelltem „nicht“ (οὐκ).

Notation: Ich schreibe N für ein Nennwort, V für ein Aussagewort, \overline{N} für unbestimmte Nennwörter, \overline{V} für unbestimmte Aussagewörter, $e = \xi\sigma\tau\iota\nu$.

Beispiel: Gesundheit : N , gesundet : V , ist weiß : V

Def (16b,26) Eine *Phrase* (λόγος) ist ein Ausdruck,

1. der etwas bedeutet (besagt, nicht: behauptet), und
2. dessen Teile auch getrennt etwas bedeuten.

Def (17a,20) Ein (*Aussage-*)*Satz* (λόγος ἀποφαντικός) ist eine Phrase, die wahr oder falsch sein kann.

Nicht jede Phrase kann wahr oder falsch sein, z.B. eine Bitte kann es nicht.

Einfache Aussagen (Kap 5, Kap 8, Kap 11)

Ein *Basissatz* ($\acute{\alpha}\pi\lambda\eta\acute{\alpha}\pi\acute{o}\phi\alpha\nu\sigma\iota\varsigma$) ist ein atomarer oder verneinter atomarer Aussagesatz.

Def (17a,15) Ein *einfacher* ($\acute{\epsilon}\iota\varsigma$, einheitlicher) Satz ist

1. ein atomarer Aussagesatz ($\kappa\alpha\tau\acute{\alpha}\phi\alpha\sigma\iota\varsigma$), oder
2. ein verneinter atomarer Aussagesatz ($\acute{\alpha}\pi\acute{o}\phi\alpha\sigma\iota\varsigma$), oder
3. ein in bestimmter Weise (s.u.) zusammengesetzter Aussagesatz.

Bem. (18a,12) Eine Einfachaussage macht *eine* Aussage über *einen* Gegenstand (der allerdings allgemein, d.h. N oder \bar{N} , sein kann). Eine Mehrfachaussage macht eine Aussage über mehrere Gegenstände, oder besteht aus mehreren Einfachaussagen.

Was ist hier *ein* Gegenstand und *eine* Aussage?

1. (Kap. 8) Eine Koordination von Benennungen bezeichnet keinen *einheitlichen* Gegenstand, sondern dient zur Abkürzung einer Koordination von Aussagen: man expandiere gemäß

$$(NP_1 \wedge NP_2)VP \implies (NP_1VP) \wedge (NP_2VP).$$

Aussagen der Form $(NP_1 \wedge NP_2)VP$ sind also *nicht* einfach: sie haben implizite Teilsätze.

2. (Kap 11, 21a,6) Eine Koordination von Aussagewörtern bzw. prädikativ gebrauchtem Adjektiv und Nomen ergibt nur manchmal *ein* Prädikat und im Satz *eine* Aussage: z.B. bei „gut \wedge Schuster“, aber nicht bei „zweifüßig \wedge Lebewesen“, wo sie in zwei implizite Prädikate zerfällt:

$$NP \text{ e } (AP_1 \wedge AP_2) \implies (NP \text{ e } AP_1) \wedge (NP \text{ e } AP_2)$$

$$NP \text{ e } (AP \wedge N) \implies (NP \text{ e } AP) \wedge (NP \text{ e } N)$$

10 Kategorien von Ausdrücken im Satz

In der Kategorienschrift unterscheidet A. zwei Aussageformen (als Anwendung von Seiendem X auf Seiendes Y verstanden):

1. X καθ' ὑποκειμένου τινός (Y) λέγεται, „ X wird von einem Zugrundeliegenden (Y) ausgesagt“:
ἄνθρωπος ζῶον ἐστίν = der Mensch ist ein Lebewesen =
($Y:N$ ist ein $X:N$) =: $X(Y)$.
2. X ἐν ὑποκειμένῳ (Y) ἐστίν, „ X kommt an einem Zugrundeliegenden Y vor“:
σῶμα λευκός = der Körper ist weiß = ($Y:N$ ist $X:AP$) =: $X[Y]$
bzw. ($Y:N$ hat $X:N$) der Körper hat die Farbe Weiß

Die semantischen Werte der so (atomar) *verbundenen* Ausdrücke sind die *Wahrheitswerte* $\{W, F\} = \{0, 1\}$.

Innerhalb des Satzes werden zehn *Kategorien* von *unverbundenen* Ausdrücken bzw. ihren semantischen Wertbereichen unterschieden:

			Bezeichnung
1.	τί ἐστί·	Was?	Substanz, Wesen (ουσία), Objekte
2.	ποσὸν	Wieviel?	Grade, Skalenwerte, Quantitäten
3.	ποιὸν	Wie?	Eigenschaften, Qualitäten
4.	πρὸς τί	bezüglich was?	Relationen
5.	ποῦ	Wo?	Orte
6.	ποτὲ	Wann?	Zeiten
7.	κεῖσθαι	Liegen	Lagen (im Raum)
8.	ἔχειν	Haben	Zustände (Medium)
9.	ποιεῖν	Tun	Tätigkeiten, Handlungen (Aktiv)
10.	πάσχειν	Erleiden	Einwirkungen (Passiv)

Wortarten bei Dionysos Thrax

Die Grammatik des Dionysos Thrax (170 - 90 v.Chr.) unterscheidet für das Griechische acht Wortarten, die weiter untergliedert werden.

Die Wortarten und ihre Formen (Parameter, Akzidenzien) sind:

1. **Nomen** (Genus, Art, Figur, Numerus, Kasus):

Genus: Maskulinum, Femininum, Neutrum

Art: primär, abgeleitet

Figur: Simplex, Kompositum, Dekompositum

Numerus: Singular, Dual, Plural

Kasus: Nominativ, Genitiv, Dativ, Akkusativ, Vokativ

An **abgeleiteten Nomen** werden 7 Unterarten genannt:

patronymische, possessivische, komparativische,
superlativische, diminutivische, denominale, deverbale.

Dann werden 24 **Unterklassen** genannt, darunter

Eigenname (Nomen Proprium), Gattungsname (Nomen Appellativum),
Adjektiv (oder Epitheon, Beiwort),
Relationsnomen, Quasirelationsnomen (z.B. Tag,Nacht),
Homonym, Synonym,
Pheronym (Benennung nach einem Ereignis, z.B. der Rächer),
Eponym (oder Dionym, Beiname: Phoibos = der Strahlende),
Ethnikon (Volksname),
Interrogativum, Indefinitum (wer auch immer),
Anaphorisches Nomen (Gleichsetzungsnomen, z.B. ein solcher),
Kollektivum, Distributivum (jeder Einzelne),
Komprehensivum (nach dem Inhalt benanntes Umfassendes, Lorbeerhain
Artnomen (was in Arten eingeteilt wird, z.B. Lebewesen),
Spezialnomen (das, was der Art nach unterschieden ist),
Ordnungsnomen (*n*-ter), Numeral (was eine Zahl bezeichnet),
Absolutes Nomen (was aus sich selbst begriffen wird, z.B. Gott),
Partizipativum (was an einem Stoff teilhat, z.B. aus Weizen)

2. **Verb** (Modus, Diathese, Art, Figur, Numerus, Person, Tempus, Konjugation):

Modus: Indikativ, Imperativ, Optativ, Konjunktiv, Infinitiv

Diathese: Aktiv, Passiv, Medium

Art: primär, abgeleitet

Figur: Simplex, Kompositum, Dekompositum

Numerus: Singular, Dual, Plural

Person: erste, zweite, dritte

Tempus: Präsens, Präteritum, Futur.

Das Präteritum hat vier Sorten: Imperfekt (sich erstreckend), Perfekt (vorliegend), Plusquamperfekt (mehr als vollendet), Aorist (nicht begrenzt).

Es gibt sechs Konjugationen der nicht endbetonten und drei Konjugationen der endbetonten Verben.

3. **Partizip** (Genus, Art, Figur, Numerus, Kasus, Diathese, Art, Figur, Tempus, Konjugation)
4. **Artikel** (Genus, Numerus, Kasus): wird den Nomen vorangestellt (gr. ho) oder nachgestellt (gr. hos)
5. **Pronomen** (Person, Genus, Numerus, Kasus, Figur, Art): ein Wort, das statt eines Nomens gebraucht wird und das bestimmte Personen bezeichnet.
6. **Präposition**: ein Wort, das allen Redeteilen vorangestellt wird, sowohl synthetisch als auch parataktisch

7. **Adverb** (Figur): ein unflektierter Redeteil, der vom Verb her ausgesagt oder dem Verb hinzugefügt wird.

Figur: Simplex, Kompositum

Hiervon unterscheidet Thrax 26 Arten nach ihrer Bedeutung oder kommunikativen Funktion, z.B.

Zeitadverbien	(nun, damals, später, heute, solange),
Zahladverbien	(<i>n</i> -mal),
Ortsadverbien,	
Wuns Chadverbien	(wenn doch),
Adverbien der Ablehnung	(nicht [gr. ouk], nirgends, keineswegs),
Adverbien des Verbotens	(nicht [gr. me]),
Adverbien des Vergleichens	(wie),
Adverbien des Vermutens	(wahrscheinlich, vielleicht),
Adverbien des Fragens	(wer, warum, wann, wie)

8. **Konjunktion** (Art): ein Wort, das Gedanken in einer bestimmten Anordnung verknüpft und das verdeutlicht, was nur implizit geäußert wird.

Art: kopulativ (und, aber, ferner),
disjunktiv (oder),
kontinuativ (wenn),
subkontinuativ (da, weil),
kausal (damit, um-zu, weil, weshalb),
dubitativ (wohl, etwa, doch-wohl-nicht),
syllogistisch (folglich, nun-aber, also),
expletiv (offenbar, ja, nämlich, durchaus, allerdings)
(ggf. adversativ: (dennoch, trotzdem))

Beachte:

- ▶ die Gliederung ist relativ fein, trennt aber nicht zwischen Unterklassen- und Formparametern (z.B. beim Nomen: Genus, Art, Figur vs. Numerus, Kasus)
- ▶ die Gliederung berücksichtigt syntaktische Eigenschaften (bei Artikel, Pronomen, Präposition, Verb) und semantische oder kommunikative Eigenschaften (bei Adverb, Konjunktion)
- ▶ die heutigen Klassifizierungen weichen z.T. in den Wortarten ab, z.B. ist das Adjektiv bei Dionysos ein Nomen.

Die Grammatik von Dionysos [2] ist –neben Euklids *Elementen*– das langlebigste Lehrbuch des Abendlands; sie wurde über 1000 Jahre lang verwendet.

Sie enthält aber keine Grammatikregeln im heutigen Sinne.

Man erkennt einige allgemeine Probleme der Klassifizierung:

- ▶ Soll man nach reinen Formeigenschaften klassifizieren?
- ▶ Wo soll man abgeleitete Wortformen einordnen?
- ▶ Form- und Bedeutungsaspekte entsprechen einander nicht eindeutig: z.B. gibt es Nomina im Singular, die eine Vielheit bedeuten, wie Demos (Volk).

Literatuur



Aristoteles.

Kategorien. Hermeneutik oder vom sprachlichen Ausdruck. Organon Band 2.

Felix Meiner, 1998.



Pierre Swiggers and Alfons Wouters.

De Tékhne Grammatikē van Dionysius Thrax: De oudste Spraakkunst in het Westen. Inleiding; Griekse tekst met Nederlandse vertaling en noten; Duitse vertaling (door Wilfred Kürschner); terminologisch apparaat en bibliografie.

Peeters, Leuven - Paris, 1998.

Strukturalismus: Bloomfield, Harris, Chomsky

Im 19. Jahrhundert untersuchten die Sprachwissenschaftler hauptsächlich den Sprachwandel (*diachrone* Sprachwissenschaft).

Im 20. Jahrhunderts herrschte die *synchrone* Sprachwissenschaft vor: die Untersuchung einer (in engem Zeitraum) gebräuchlichen Sprache als ein System von Beziehungen zwischen sprachlichen Elementen, der *Strukturalismus*.

Ansatzpunkt der Strukturalisten (de Saussure, Bloomfield, Harris, Chomsky):

- ▶ Sprachliche Elemente (Wörter, Phoneme usw.) haben einen Sinn nur innerhalb eines Systems, durch Äquivalenzen und Kontraste untereinander.
de Saussure: wie die Figuren im Schach
- ▶ Das System wird durch Kombinationen der Elemente und Beschränkungen der Kombinationsmöglichkeiten beherrscht.

Sprachliche Ausdrücke werden i.a. aus kleineren Einheiten zusammengesetzt. Wenn man solche Einheiten identifiziert hat, kann man untersuchen,

1. welche ihrer Kombinationen werden als größere Einheiten erkannt/akzeptiert,
2. welche Kombinationsoperationen unterliegen welchen Einschränkungen?

Einschränkungen an die Kombination von Sätzen zu größeren Einheiten sind relativ schwach ausgeprägt, deshalb sind diese Fragen eher für Einheiten unterhalb der Satzebene interessant, z.B.

- ▶ Wörter innerhalb von Syntagmen
- ▶ Syntagmen innerhalb von Sätzen
- ▶ Phoneme innerhalb von Wörtern

Harris:

- ▶ Mangel an expliziten Methoden in der Syntax führt zu
 - ▶ intuitivem Vorgehen,
 - ▶ ungewisser Relevanz von Beobachtungen,
 - ▶ unklaren Beziehungen zwischen Beobachtungen
- ▶ Untersuchung von Distributionsverhältnissen
 - ▶ mit expliziten Methoden möglich (Substitution)
 - ▶ nicht, um daraus Aussagen über die Bedeutungen zu erhalten,
 - ▶ weil sie unabhängig von Bedeutungen untersuchbar sind.

Endliche Folgen kleinster Einheiten

Da sprachliche Äußerungen φ in der Zeit erfolgen und i.a. nicht atomar sind, lassen sie sich in endliche *Folgen* w kleinster Einheiten a zerlegen, die nacheinander geäußert werden.

Die kleinsten Einheiten sollten wenige sein, und von einander unterscheidbar. Nehmen wir an, eine endliche Menge $\Sigma = \{a, b, \dots\}$ solcher kleinsten Einheiten sei bestimmt, z.B. die Phoneme.

Sei Σ^* die Menge *aller* endlichen Folgen von Elementen aus Σ . Den *sprachlichen* Äußerungen entspricht eine Teilmenge $L \subseteq \Sigma^*$.

- ▶ Es gibt unter den $w \in L$ gewisse Regularitäten, z.B. treten manche $a, b \in \Sigma$ nicht am Ende eines w auf, manche nicht direkt nebeneinander, manche nur gemeinsam, usw.
- ▶ Da L sehr groß oder unendlich ist, ist es schwierig, die Regularitäten zu kennen.

Distribution von $w \in \Sigma^*$ in L

Um die Regularitäten aufzufinden, versucht man, Folgen $w \in \Sigma^*$ zu klassifizieren und Regularitäten auf der Ebene der Klassen zu finden.

Definition

Die *Distribution* $D_L(w)$ von $w \in \Sigma^*$ in L ist die Menge der *Kontexte* $u_v := (u, v)$, in denen w in L vorkommt:

$$D(w) := \{(u, v) \mid u, v \in \Sigma^*, uwv \in L\}.$$

Zwei Folgen $w_1, w_2 \in \Sigma^*$ sind *distributionsäquivalent bzgl. L* , kurz: $w_1 \equiv_L w_2$, wenn sie dieselbe Distribution haben, $D(w_1) = D(w_2)$.

Zwei Folgen $w_1, w_2 \in \Sigma^*$ sind *von komplementärer Distribution*, wenn sie keine Kontexte gemeinsam haben: $D(w_1) \cap D(w_2) = \emptyset$.

Seien Σ die Wortformen und L die Sätze des Deutschen.

- ▶ *Emil* und *Willi* sind äquivalent.
- ▶ *mich* und *dich* sind nicht äquivalent: *Ich schäme dich.* $\notin L$
- ▶ *wir* und *sprecht* sind komplementär.
- ▶ $D(\text{der Hund}) \subseteq D(\text{er})$. Aber: *er ißt* $\in L$, *der Hund ißt* $\in L$?

Beachte: die Begriffe sind nur bedingt operationalisierbar

- ▶ ist L endlich, so kann man $w_1 \equiv w_2$ feststellen, da man nur die endlich vielen u_v berücksichtigen muss, die man aus den $uv \in L$ bekommt. z.B. $\Sigma =$ Phoneme, $L =$ einf.Wortformen
- ▶ wenn L unendlich ist, müßte man unendlich viele Kontexte u_v durchlaufen, was man nur im Geiste kann. z.B. $L =$ Sätze
- ▶ die Definition ist relativ dazu, was $\in L$ bedeutet: manchmal ist die *Akzeptanz* von uvw gemeint, da man L nur indirekt kennt.

Beispiel (Harris 1942)

Üblicher Begriff: Ein *Morphem* ist eine Folge von Phonemen, die eine Bedeutung hat, aber nicht aus bedeutungsvollen Teilfolgen von Phonemen zusammengesetzt ist.

Aber:

- ▶ manchmal faßt man verschiedene dieser Morpheme als Varianten *eines* Morphems auf, z.B. /knaif/ in *knife* und /naiv/ in *knives*, als dasselbe /knaif/, wo mit /f/ mal /f/ und mal /v/ gemeint ist.
- ▶ Verdopplungen werden als spezielles Affix betrachtet, obwohl sie aus unterschiedlichen Phonemfolgen bestehen: μεμένεκα (bin geblieben) und λέλυκα (habe verloren).

Besseres Verfahren zur Bestimmung der Morpheme einer Sprache:

1. Teile jeden Ausdruck in die kleinsten Phonemfolgen mit gleicher Bedeutung in *Morphemalternativen*: /knaif/ \neq /knaiv/.
2. Bilde *Morphemeinheiten*, d.h. Mengen der Morphemalternativen gleicher Bedeutung und komplementärer Distribution. Berücksichtige eine mehrelementige Einheit E nur, wenn

$$\bigcup \{D(w) \mid w \in E\} = D(e) \text{ für eine Einheit } E' = \{e\},$$

z.B. $E = \{am, are\}$ wegen $D(am) \cup D(are) = D(walk)$.

3. Unterscheiden sich die Alternativen zweier Einheiten E_1 und E_2 auf dieselbe Weise, so repräsentiere die Einheiten durch eines der Elemente und den Unterschied, z.B. $\{knife, kniv-\}$ und $\{wife, wiv-\}$ durch (/naif/, /f/ + /-z/ \mapsto /v/)

Ermittlung der Unterschiede zwischen den Alternativen:

1. Worin unterscheiden sich die Alternativen einer Einheit?
2. In welchen Kontexten kommt eine Alternative vor?
3. Welche Ähnlichkeit besteht zwischen Alternative und Kontext?
4. Welche Einheiten haben diese Unterschiede zwischen den Alternativen?

Z.B. die Kontexte B_C , in denen eine Alternative $a \in A$ vorkommt, sind selber Einheiten, die in ihre Alternativen $b \in B$, $c \in C$ zerfallen, und nur in manchen b_c tritt a auf.

Klassifizierung der Kontexte nach ähnlicher Bildung von Alternativen.

Substitutionsklassen

Die zu einem $w \in \Sigma^*$ bzgl. L distributionsäquivalenten w' bilden die *Distributions-* oder *Substitutionsklasse* von w ,

$$S(w) := \{w' \in \Sigma^* \mid w' \equiv_L w\} =: [w]_{\equiv_L}.$$

Die Elemente einer Substitutionsklasse kann man in beliebigen L -Kontexten durch einander ersetzen, ohne aus L herauszukommen:

$$\begin{aligned} w' \in S(w) &\iff w \equiv w' \iff D(w) = D(w'). \\ &\iff \forall u \in \Sigma^* \forall v \in \Sigma^* (uwv \in L \iff uw'v \in L). \end{aligned}$$

Oft ist man umgekehrt daran interessiert, welche $w, w' \in \Sigma^*$ sich bzgl. eines bestimmten Kontexts u_v gleich verhalten, d.h. an

$$w \simeq_{u_v} w' : \iff (uwv \in L \leftrightarrow uw'v \in L).$$

z.B. für den Kontext *der _ schläft nicht*.

Wortarten als Distributionsklassen?

Man kann nicht erwarten, daß die Wortarten Substitutionsklassen sind: verschiedene Formen desselben Worts können oft gerade *nicht* in denselben Kontexten vorkommen, z.B. *Baum* und *Bäumen*.

Manchmal bilden alle Wörter einer Wortart (bzw. Unterart) *in derselben Form* eine Substitutionsklasse: z.B.

$$N_{neut}^{gen,sg} = \{Houses, Kindes, \dots\}$$

Aber i.a. ist das nicht so: z.B. ist

$$N_{mask}^{nom,sg} = \{Baum, Hase, Hund, \dots\}$$

keine Substitutionsklasse (bzgl. $L =$ deutsche Sätze), da

Ich sehe den Baum $\in L$, *Ich sehe den Hase* $\notin L$

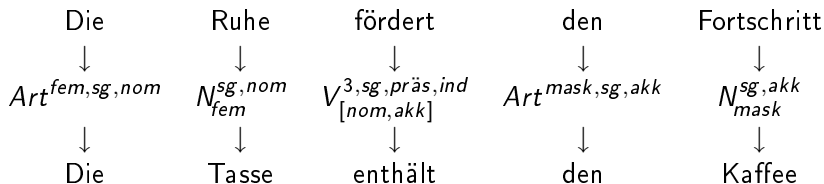
Die Distribution der konkreten Formen kann allein nicht darüber entscheiden, was ein Satz ist: es kommt u.a. auf die abstrakten Formen an (z.B. auf den an der Wortform *Baum* nicht ablesbaren Kasus).

Erstmal braucht man die abstrakten Formen, die Wörter und die Wortarten. (Vgl. Hausaufgaben)

- ▶ Erhält man *Wörter* durch Abstraktion von ihren Formen?
Die Distribution liefert etwa $D(\text{Baum}) = D(\text{Hund})$, unterscheidet aber nicht zwischen Nominativ- und Akkusativkontexten. Man muß also erstmal die Distributionen überladener Formen nach den abstrakten Formen trennen und ggf. diese geeignet zusammensetzen.
- ▶ Gibt es für abstrakte Formen, Wörter oder Wortarten *charakteristische Kontexte*? Z.B. Kontexte, in denen genau die Nomen im Nominativ, oder genau die Verben auftreten?

Satzschemata

Teilt man die Wortarten hinreichend fein in Unterarten (z.B. $V_{[nom,akk]}$: transitive Verben), so lassen sich in einem Satz Wörter derselben Unterart durch andere *derselben Form* ersetzen:



Aus jedem Satz erhält man ein *Satzschema*, eine Folge der Konkretisierungen abstrakter Formen. Aber:

Solche Schemata beschreiben die Regelmäßigkeiten im Satz zu eng: z.B. erlaubt es nicht, *Die Tasse* durch *der Becher* zu ersetzen.

Literatur



L. Bloomfield.

Language.

Holt, 1933.



Z. Harris.

A Theory of Language and Information.

Clarendon Press, 1991.



Z. S. Harris.

Papers on Syntax.

D. Reidel, 1981.



J. Lyons.

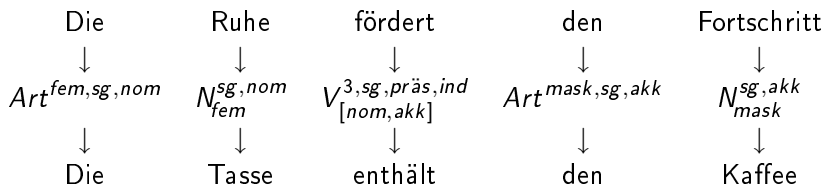
Introduction to Theoretical Linguistics.

Cambridge University Press, 1968.

Dt.: Einführung in die moderne Linguistik. C.H.Beck, 1989.

Satzschemata

Teilt man die Wortarten hinreichend fein in Unterarten (z.B. $V_{[nom,akk]}$: transitive Verben), so lassen sich in einem Satz Wörter derselben Unterart durch andere *derselben Form* ersetzen:



Aus jedem Satz erhält man ein *Satzschema*, eine Folge der Konkretisierungen abstrakter Formen. Aber:

Solche Schemata beschreiben die Regelmäßigkeiten im Satz zu eng. z.B. erlauben sie nicht, *Die Tasse* durch *der Becher* zu ersetzen.

Verschiedene konkrete Formen „eines“ Worts unterscheiden sich i.a. morphologisch höchstens durch

- ▶ durch kurze Affixe (am Anfang, im Innern oder am Ende),
- ▶ durch Lautveränderungen (Umlaute, Konsonantenwechsel),

so daß wir mit Endungstabellen, Umlautregeln u.ä. hoffentlich feststellen können, ob zwei Wortformen zum „selben“ Wort gehören.

Verschiedene Formen eines Worts sind i.a. komplementär:

Die Tasse enthält den Kaffee
* *Die Tassen enthält den Kaffee*

In diesem Fall müssen Wortformen des Kontexts „angepaßt“ werden, um wieder einen Satz zu haben: *enthält* \mapsto *enthalten*.

Bemerkung:

Wo verschiedene Formen desselben Worts nicht komplementär sind, spielen z.B. Bedeutungsänderungen mit:

<i>Die</i>	<i>Tasse</i>	<i>enthält</i>	<i>den</i>	<i>Kaffee</i>	<i>Gegenwart</i>
<i>Die</i>	<i>Tasse</i>	<i>enth<u>i</u>elt</i>	<i>den</i>	<i>Kaffee</i>	<i>Vergangenheit</i>
	<i>Wir</i>	<i>trinken</i>	<i>den</i>	<i>Kaffee</i>	<i>über Sprecher u.a.</i>
	<i><u>S</u>ie</i>	<i>trinken</i>	<i>den</i>	<i>Kaffee</i>	<i>nur über Andere</i>

Dann trenne man die Bezugssprache L in Teilsprachen $L_1 \dot{\cup} L_2$ und berücksichtige nur die Kontexte der Aussagen aus *einem* der L_i , z.B. die Distribution bzgl. $L_i = L^{präs}$ betrachten.

Wenn der Bezugsbereich L klein genug ist, sollten unterschiedliche Formen desselben Worts komplementär sein.

„Überladene“ Formen wie $Baum \in N_{mask}^{sg,nom} \cap N_{mask}^{sg,akk}$ lassen sich daran erkennen, daß *andere* Wörter in entsprechenden Kontexten verschiedene Formen verwenden:

Der	Baum	ist	alt		Der	Bär	ist	alt
Den	Baum	sehe	ich		Den	Bären	sehe	ich
Der	Bäume	gedenke	ich		Der	Bären	gedenke	ich

Wir können dann *abstrakte* Formen von Wort(unter)arten durch Kontexte aus konkreten Wortformen definieren:

Der	$N_{mask}^{sg,nom}$	ist	alt
Den	$N_{mask}^{sg,akk}$	sehe	ich
Der	$N_{mask}^{pl,gen}$	gedenke	ich

Dann bestimmt man das Paradigma einzelner Wörter wie *Baum*.

Man erhalte so zuerst nur verschiedene *Flexionsklassen*, und muß anschließend daraus Wort(unter)arten durch Vereinigung bilden:

$$N_{mask} = N_{mask, Flexionsklasse\ 1} \cup \dots \cup N_{mask, Flexionsklasse\ k} \quad (1)$$

Nehmen wir also an:

- ▶ man kann (über Kontexte konkreter Wortformen) Wortarten A mit zugehörigen abstrakten Formbereichen $Form_A$ festlegen,
- ▶ für jedes A und $F \in Form_A$ sei $A^F := \{W^F \mid W \in A\}$ die Menge der konkreten F -Formen W^F der Wörter W aus A .
- ▶ in den A^F definierenden Kontexten führen die W^F zu Sätzen.

Notation: **Formmerkmale** werden (hier) im oberen Index angegeben, **Artmerkmale** für Angabe von Unterarten durch den unteren Index:

$$N_{mask}^{sg, dat} := \text{maskuline Nomina im Singular und Dativ} \quad (2)$$

Wortarten und Flexionsdimensionen im Deutschen

Als *Flexionsarten* sind im Deutschen zu unterscheiden:

1. die Konjugation bei Verben,
2. die Deklination bei Artikeln, Nomina, Pronomina, Adjektiven,
3. die Komparation bei Adjektiven.

Die Gesamtheit der (abstrakten) Formen einer Flexionsart stellt man in mehrdimensionalen (Form-) Merkmalen dar. Wir verwenden für das Deutsche folgende *Merkmalsdimensionen*:

- ▶ bei Nomina: Numerus, Kasus,
- ▶ bei Pronomen: Person, Genus, Numerus, Kasus
- ▶ bei Artikeln: Genus, Numerus, Kasus,
- ▶ bei Adjektiven: Genus, Numerus, Kasus, Deklinationstyp, Komparation
- ▶ bei Verben: Numerus, Person, Tempus, Modus, Infinitheit

Flexionsdimensionen im Deutschen

Jede solche Merkmalsdimension der Wortarten hat einen endlichen Bereich von Werten, die das Merkmal annehmen kann:

Dimension	Werte	Kürzel
Kasus	Nominativ, Genitiv, Dativ, Akkusativ	nom, gen, dat, akk
Numerus	Singular, Plural	sg, pl
Person	erste, zweite, dritte	1, 2, 3
Genus	Femininum, Maskulinum, Neutrum	fem, mask, neut
Komparation	Positiv, Komparativ, Superlativ	pos, komp, sup
Deklin.typ	starke, schwache, gemischte Deklin.	st, sw, ge
Tempus	Präsens, Präteritum	präs, prät
Infinitheit	Infinitiv, Partizip I, Partizip II	inf, part1, part2
Modus	Indikativ, Konjunktiv, Imperativ	ind, konj, imp

Bem. Für Ausdrücke gibt es i.a. andere Dimensionen oder Werte, z.B. bei Sätzen: Diathese als Dimension, Perfekt als Tempuswert.

Satzschemata aus Mengen konkreter Wortformen?

Geht man von einem konkreten Satz

$$s = w_1^{F_1} \dots w_n^{F_n}$$

aus und ersetzt die konkreten Wortformen $w_i^{F_i}$ durch zu ihnen „passende“ Wortarten und abstrakte Formen $A_i^{F_i} \ni w_i^{F_i}$, so erhält man *potentiell* Satz-schemata $A_1^{F_1} \dots A_n^{F_n}$:

S	$Art^{fem,sg,nom}$	$N_{fem}^{sg,nom}$	$V_{[nom,akk]}^{3,sg,präs,ind}$	$Art^{mask,sg,akk}$	$N_{mask}^{sg,akk}$
	↑	↑	↑	↑	↑
	Die	Ruhe	fördert	den	Schlaf
	↓	↓	↓	↓	↓
*S	$Art^{mask,pl,nom}$	$N_{fem}^{sg,akk}$	$V_{[nom,akk]}^{3,sg,präs,ind}$	$Art^{fem,pl,dat}$	$N_{mask}^{sg,akk}$

Das zweite sollte kein Satzschema sein, obwohl es nur Sätze ergibt!

Daß alle Einsetzungen konkrete Sätze sind, ist nur ein notwendiges Kriterium:

*S	Art ^{mask,pl,nom}	N _{fem} ^{sg,akk}	V ^{3,sg,präs,ind} [nom,akk]	Art ^{fem,pl,dat}	N _{mask} ^{sg,akk}
S	Die	Tasse	enthält	den	Kaffee

Das Schema besagt mehr, als sich in der Menge der Einsetzungen ausdrückt: eine bestimmte (Wortart,Form)-Analyse sei ein Satz.

Das Problem kommt daher, daß bei manchen Flexionsklassen die Formüberladung systematisch ist, z.B. $N_{fem}^{sg,nom} = N_{fem}^{sg,akk}$.

Also: Eine Folge $A_1^{F_1} \dots A_n^{F_n}$ von Wortarten A und abstrakten Formen $F_i \in \text{Form}_{A_i}$ ist höchstens dann ein *Satzschema* S , wenn beliebige Einsetzungen

$$w_1^{F_1} \dots w_n^{F_n}$$

der konkreten Formen $w_i^{F_i}$ von Wörtern w_i der Wortart A_i konkrete Sätze sind. Für diese Bedingung schreiben wir:

$$S \supseteq A_1^{F_1} \dots A_n^{F_n} \quad \text{oder} \quad S \rightarrow A_1^{F_1} \dots A_n^{F_n}.$$

Beachte: die A^F sind wie Mengenvariablen

- ▶ die Wahlen der $w_i \in A_i$ erfolgen unabhängig von einander
- ▶ kommt A^F mehrfach vor, so kann für jedes Vorkommen ein anderes w^F aus A^F gewählt werden.

Kontextfreie Grammatiken

Um zu definieren, was Sätze sind, braucht man natürlich viele Satz schemata, z.B. die für die Beispiele

$Art^{fem,sg,nom}$	$N_{fem}^{sg,nom}$	$V^{3,sg,präs,ind}$ [nom,akk]	$Art^{mask,sg,akk}$	$N_{mask}^{sg,akk}$
Die	Tasse	enthält	den	Kaffee
$Art^{mask,sg,nom}$	$N_{mask}^{sg,nom}$	$V^{3,sg,präs,ind}$ [nom,akk]	$Art^{mask,sg,akk}$	$N_{mask}^{sg,akk}$
Der	Becher	enthält	den	Kaffee
$Art^{mask,pl,nom}$	$N_{mask}^{pl,nom}$	$V^{3,pl,präs,ind}$ [nom,akk]	$Art^{mask,sg,akk}$	$N_{mask}^{sg,akk}$
Die	Becher	enthalten	den	Kaffee
$Pron^{3,mask,pl,nom}$		$V^{3,pl,präs,ind}$ [nom,akk]	$Art^{mask,sg,akk}$	$N_{mask}^{sg,akk}$
Sie		enthalten	den	Kaffee
$Pron^{3,mask,pl,nom}$		$V^{3,pl,präs,ind}$ [nom,akk]	$Pron^{3,mask,sg,akk}$	
Sie		enthalten	ihn	

Teilfolgen von (Wortart,Form)-Paaren A^F dieser Schemata lassen sich ihrerseits durch ihre Distribution bzgl. der Menge aller Schemata vergleichen, z.B.

- ▶ In jedem Kontext, wo $Art^{fem,sg,nom} N_{fem}^{sg,nom}$ auftritt, kann auch das Personalpronomen $Pron^{3,fem,sg,nom}$ auftreten, usw.:

$$D(Art^{fem,sg,nom} N_{fem}^{sg,nom}) \subseteq D(Pron^{3,fem,sg,nom})$$

$$D(Art^{mask,sg,nom} N_{fem}^{sg,nom}) \subseteq D(Pron^{3,mask,sg,nom})$$

- ▶ Die beiden $Art N$ -Folgen haben viele Kontexte gemein:

$$Art^{fem,sg,nom} N_{fem}^{sg,nom} \simeq Art^{mask,sg,nom} N_{mask}^{sg,nom}$$

Treten viele Folgen α in denselben Kontexten der Schemata auf, können wir sie in eine Substitutionsklasse A' zusammenfassen.

Z.B. treten

$Art^{mask,pl,nom} N_{mask}^{pl,nom}$ und $Pron^{3,mask,pl,nom}$,
 $Art^{mask,sg,akk} N_{mask}^{sg,akk}$ und $Pron^{3,mask,sg,akk}$

in denselben Kontexten der obigen Schemata auf. Ersetzen wir sie durch $NP_{mask}^{pl,nom}$ bzw. $NP_{mask}^{sg,akk}$, so lassen sich die Schemata zu

$S \rightarrow NP_{mask}^{pl,nom} V_{[nom,akk]}^{3,pl,präs,ind} NP_{mask}^{sg,akk}$

zusammenfassen, wobei (mit | für „oder“, d.h. zwei Schemata)

$NP_{mask}^{pl,nom} \rightarrow Art^{mask,pl,nom} N_{mask}^{pl,nom} | Pron^{3,mask,pl,nom}$
 $NP_{mask}^{sg,akk} \rightarrow Art^{mask,sg,akk} N_{mask}^{sg,akk} | Pron^{3,mask,sg,akk}$

Def. Kontextfreie Grammatik

Def. (Chomsky 1957)

Eine kontextfreie Grammatik $G = (\Sigma, NT, P, S)$ besteht aus

1. einer endlichen Menge Σ , dem Alphabet (Terminalen),
2. einer endlichen Menge NT , den Kategorien (Nonterminalen),
3. einer ausgezeichneten Kategorie $S \in NT$, der Satzkategorie (dem Startsymbol)
4. einer endlichen Menge $P \subseteq NT \times (\Sigma \cup NT)^*$ von Grammatikregeln

Notation: a, b, c für Elemente von Σ , u, v, w für Elemente von Σ^* , A, B, C für Elemente von NT , X, Y, Z für Elemente von $\Sigma \cup NT$, α, β, γ für Elemente von $(\Sigma \cup NT)^*$; $(A \rightarrow \alpha)$ für die Regel (A, α) .
 $A \rightarrow \alpha \mid \dots \mid \beta$ für die Regeln $(A \rightarrow \alpha), \dots, (A \rightarrow \beta)$.

Beispiel (Kontextfreie Grammatik)

$G = (\Sigma, NT, P, S)$ bestehe aus

▶ $\Sigma = \{\text{Die, den, Sie, ihn, Becher, Kaffee, enthalten}\}$

▶ $NT = \{S, NP, N, V, Art, Pron\}$

▶ $P =$

S	\rightarrow	$NP V NP$	V	\rightarrow	enthalten
NP	\rightarrow	$Art N \mid Pron$	Art	\rightarrow	Die \mid den
N	\rightarrow	Becher \mid Kaffee	$Pron$	\rightarrow	Sie \mid ihn

▶ Satzkatgorie = S

Interpretieren wir die Terminale $a \in \Sigma$ durch Einermengen $\{a\}$, \mid durch \cup , $X Y$ durch el.Verketzung und \rightarrow durch \supseteq , so ergibt sich:

$S \supseteq \{\text{Die Becher enthalten den Kaffee, \dots, Sie enthalten ihn}\}$

Man interpretiert die Nonterminale durch die *kleinsten* Mengen, die die Schemata wahr machen. (Das geht auch bei rekursiven Regeln.)

S	\supseteq	$NP V NP$	V	\supseteq	$\{\text{enthalten}\}$
NP	\supseteq	$(Art N) \cup Pron$	Art	\supseteq	$\{\text{Die, den}\}$
N	\supseteq	$\{\text{Becher, Kaffee}\}$	$Pron$	\supseteq	$\{\text{Sie, ihn}\}$

und daher z.B.

S	\supseteq	$NP V NP$
	\supseteq	$(Art N) V NP$
	\supseteq	$(\{\text{Die, den}\} \{\text{Becher}\}) V Pron$
	\supseteq	$\{\text{Die Becher, den Becher}\} V Pron$
	\supseteq	$\{\text{Die Becher, den Becher}\} \{\text{enthalten}\} \{\text{ihn}\}$
	\supseteq	$\{\text{Die Becher enthalten, den Becher enthalten}\} \{\text{ihn}\}$
	\supseteq	$\{\text{Die Becher enthalten ihn, den Becher enthalten ihn}\}$

Bem. Eine rekursive Regel wie $(NP \rightarrow NP \text{ und } NP)$ besagt, daß die Menge NP „unter Koordination abgeschlossen ist“.

Grammatikregeln als Ersetzungsregeln

Oft betont man die „Erzeugung“ von Sätzen und spricht von „generativer Grammatik“.

Def. Seien $\alpha, \beta \in (\Sigma \cup NT)^+$. Man sagt, α erzeugt β in n Schritten, kurz: $\alpha \Rightarrow^n \beta$, wenn

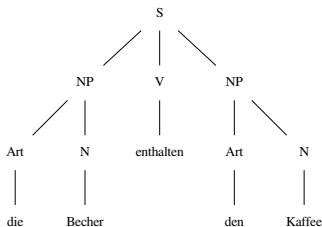
- ▶ $n = 1$ und es eine Zerlegung $\alpha = \alpha_1 A \alpha_2$ und eine Grammatikregel $A \rightarrow \gamma$ gibt, so daß $\beta = \alpha_1 \gamma \alpha_2$, oder
- ▶ $n > 1$ und es ein $\delta \in (\Sigma \cup NT)^*$ gibt, sodaß $\alpha \Rightarrow^1 \delta$ und $\delta \Rightarrow^n \beta$ gelten.

Dann stellt man die Erzeugung eines Satzes als *Ableitung* dar:

S	\Rightarrow	$NP \ V \ NP$	$(S \rightarrow NP \ V \ NP)$
	\Rightarrow	$Art \ N \ V \ NP$	$(NP \rightarrow Art \ V)$
	\Rightarrow	$Die \ N \ V \ NP$	$(Art \rightarrow Die)$
	\Rightarrow	$Die \ Becher \ V \ Art \ N$	$(N \rightarrow Becher)$
	\Rightarrow	$Die \ Becher \ V \ den \ N$	$(Art \rightarrow den)$
	\Rightarrow	$Die \ Becher \ V \ den \ Kaffee$	$(N \rightarrow Kaffee)$
	\Rightarrow	$Die \ Becher \ enthalten \ den \ Kaffee$	$(V \rightarrow enthalten)$

Syntaktische Analysen in CFGs: Syntaxbäume

Eine solche Ableitung ist eine Begründung dafür, daß der Ausdruck von der jeweiligen Kategorie ist, eine **syntaktische Analyse**. Man kann sie durch einen *Syntaxbaum* darstellen:



Da die Ableitungsschritte teilweise unabhängig voneinander gemacht werden können (zuerst *V* oder zuerst *NP* „erzeugen“?), repräsentieren gewisse Ableitungsfolgen „dieselbe Analyse“.

CFG für Programme

Programmiersprachen lassen sich durch (einfache) CFGs definieren.
Ein *Parsergenerator*, hier: ML-Yacc, erzeugt aus der Definition ein Programm zur syntaktischen Analyse:

```
%% (* Grammatik fuer arithm.Ausdruecke mit Auswertung *)
%left SUB PLUS
%left TIMES DIV
%right CARAT

%term ID of string | NUM of int
      | PLUS | TIMES | PRINT |
      SEMI | EOF | CARAT | DIV | SUB | LPAR | RPAR
%nonterm EXP of int | START of int option

%name Calc
```



```

%% (* Grammatikregeln und Werte zu den Ausdruecken *)
START : PRINT EXP      (print(Int.toString EXP);
                        print "\n"; SOME EXP)
      | EXP            (SOME EXP)
      |                (NONE)
EXP   : NUM            (NUM)
      | ID             (lookup ID)
      | EXP PLUS EXP   (EXP1+EXP2)
      | EXP TIMES EXP  (EXP1*EXP2)
      | EXP DIV EXP    (EXP1 div EXP2)
      | EXP SUB EXP    (EXP1-EXP2)
      | EXP CARAT EXP  (let fun e (m,0) = 1
                        | e (m,1) = m*e(m,1-1)
                        in e (EXP1,EXP2) end)
      | LPAR EXP RPAR  (EXP)

```

Lexer für den Parser

Spezifikation eines Lexers (für ML-Lex), der Eingabezeichen in Token (Terminale mit Werten und Eingabepositionen) umwandelt:

```
structure T = Tokens
type pos = int (* Typ der Positionen in der Eingabe *)
type svalue = T.svalue (* Semantische Werte in Token *)
type lexresult= ((T.svalue,pos) T.token list * string)
val pos = ref 0

%% (* Zum Zusammenbauen mit den Token der Grammatik: *)
%header (functor CalcLexFun(structure Tokens: Calc_TOKENS))

(* Abkuerzungen fuer die reg.Ausdruecke in den folg.Regeln *)
alpha=[A-Za-z];
digit=[0-9];
ws = [\ \t];
```

```

%% (* Lexer-Regeln: Umwandlung von Zeichen in Token *)
\n      => (pos := (!pos) + 1; lex()); (* Zeilen zaehlen *)
{ws}+   => (lex()); (* 'white space' ueberlesen *)
{digit}+ => (T.NUM(valOf(Int.fromString yytext),!pos,!pos):
"+"     => (T.PLUS(!pos,!pos)::nil,yytext);
"*"     => (T.TIMES(!pos,!pos)::nil,yytext);
";"     => (T.SEMI(!pos,!pos)::nil,yytext);
{alpha}+ => (if yytext="print"
              then (T.PRINT(!pos,!pos)::nil,yytext)
              else (T.ID(yytext,!pos,!pos)::nil,yytext));
"_"     => (T.SUB(!pos,!pos)::nil,yytext);
"^"     => (T.CARAT(!pos,!pos)::nil,yytext);
"/"     => (T.DIV(!pos,!pos)::nil,yytext);
"("     => (T.LPAR(!pos,!pos)::nil,yytext);
")"     => (T.RPAR(!pos,!pos)::nil,yytext);
"."     => (error ("ignore bad char "^yytext,!pos,!pos);lex())

```

Konstituententests

Welche Teilfolge eines Schemas bildet eine „Einheit“(Konstituente)?

Sei Σ das Lexikon aller Vollformen von Wörtern einer Sprache L und $S \subseteq \Sigma^*$ als Menge aller Sätze von L .

Der Begriff der *Konstituente* ist nur *relativ zu einer bestimmten Analyse* sinnvoll; manche Sätze haben mehrere Analysen.

Verfahren, die allein auf Grund der Wortfolge eines Satzes $w \in \Sigma^*$ „dessen“ Konstituenten bestimmen sollen, liefern daher nur *Konstituentenkandidaten*. Idee:

- ▶ als Konstituenten in den Analysen von w dürfen nur die ermittelten Kandidaten auftreten; (Vollständigkeit der Tests)
- ▶ Kandidaten können sich als untauglich herausstellen und müssen nicht in einer Analyse vorkommen;
- ▶ überlappende Kandidaten kommen nie in *einer* Analyse vor.

Grobe Formulierung

Folgende Formulierung der Tests aus Stechow/Sternefeld (1988, 107 ff.) sind zu ungenau:

1. Ersetzungsprobe: Eine Zeichenfolge, die sich durch eine andere ersetzen läßt, ohne daß sich ihre Satzhaftigkeit ändert, ist eventuell eine Konstituente.
2. Verschiebeprobe: Was sich verschieben, umstellen läßt, ist eine Konstituente. Topikalisierung: Verschieben vor das finite Verb.
3. Pronominalisierungstest: Was sich pronominalisieren läßt (d.h. worauf man sich mit einer „Proform“ beziehen kann) ist eine Konstituente.
4. Koordinationstest: Was sich koordinieren läßt, ist eine Konstituente.
5. Tilgungsprobe: In elliptischen Konstruktionen können nur Konstituenten weggelassen werden.
6. Fragetest: Wonach sich fragen läßt, das ist eine Konstituente.

- ▶ **Pronominalisierungs- und Frageprobe:** Eine Wortfolge in einem Satz, deren Ersetzung durch ein Pronomen (bzw. Fragewort) wieder zu einem Satz (bzw. einer Frage) führt, ist ein Konstituentenkandidat.

Ihr Freund kommt zu Besuch.

Er kommt zu Besuch.

Wer kommt zu Besuch?

In den Alpen ist es schön.

Dort ist es schön.

Wo ist es schön?

Das Kriterium taugt auch für andere *Pro-Formen*, die ebenfalls Ausdrücke einer bestimmten Art vertreten, z.B.:

Pro-Formen

Klasse X	Proformen $proXP$	Frageproformen $quXP$
Nomen	er, sie, es, wir, ihr, sie	wer, was
Adverbien Kausal	deswegen, deshalb, darum	weswegen, weshalb, warum wieso
Modal	so, ebenso	wie
Lokal	hier, dort, da	wo
Direktional	hierhin, dorthin, dahin	wohin
Motivational	dazu, damit	wozu (zu welchem Zweck)
Temporal	dann, solange	wann, wie lange
Instrumental	damit	womit
daß-Sätze	es	was
ob-Sätze	etwas, das	was

Bei Personalpronomen und Nominalphrasen sind die Proformen $p \in proXP$ von derselben Kategorie XP wie die Ausdrücke, die sie ersetzen; dann ist $proXP \subseteq XP$.

Im allgemeinen sind die Proformen $p \in proXP$ nicht von derselben Kategorie XP wie die ersetzten Ausdrücke: ein Pronominal*adverb* repräsentiert *Präpositional*phrasen.

Wenn die Proformen $p \in proXP$ *nur* in Kontexten stehen dürfen, in denen die repräsentierten Ausdrücke erlaubt sind, legt der Test auch den Phrasentyp der Konstituente nahe:

$$\forall u, v, w \in \Sigma^* ((uvw \in S \wedge \exists p(p \in proXP \wedge upw \in S)) \\ \Rightarrow v \text{ ist Kandidat für eine } XP\text{-Konstituente von } uvw).$$

Wegen der Überladung von *pro*-Formen, etwa bei *was*, erhält man i.a. mehrere Phrasentypkandidaten für den ersetzten Teil. Dann kommt es auf die Lesart von p an, also auf eine Analyse von upw .

- ▶ **Vertauschungsprobe:** Zwei nichtleere Wortfolgen in einem Satz, deren Vertauschung wieder einen Satz (oder eine Frage) (mit gleichem Sinn) ergibt, sind Konstituentenkandidaten.

$\forall u_1, u_2, u_3, v, w \in \Sigma^*$

$(v \neq \epsilon \wedge w \neq \epsilon \wedge u_1 v u_2 w u_3 \in S \wedge u_1 w u_2 v u_3 \in S$

$\Rightarrow v \text{ und } w \text{ sind Konstituentenkandidaten von } u_1 v u_2 w u_3).$

Die Voraussetzung, daß die vertauschten Ausdrücke nicht leer sein dürfen, schließt einfache Verschiebungen aus.

Die Einschränkung der Sinngleichheit ($u_1 v u_2 w u_3 \equiv u_1 w u_2 v u_3$) legt nahe, daß man bei diesem Test daran interessiert ist, nur die Konstituenten *einer bestimmten Lesart* zu ermitteln.

Argumente und Modifikatoren lassen sich meist vertauschen:

1. Er **erfuhr** das aus der Zeitung.
Erfuhr er das aus der Zeitung?
2. Er gibt **dem Hund den Knochen**.
Er gibt **den Knochen dem Hund**.
3. **Dein Freund aus Paris** kommt **morgen** mit dem Auto vorbei.
Morgen kommt **dein Freund aus Paris** mit dem Auto vorbei.
Dein Freund aus Paris kommt **morgen mit dem Auto** vorbei.
?Dein Freund aus Paris kommt **mit dem Auto morgen** vorbei.

Aber wo die Vertauschungen den Sinn ändern, müssen die entsprechenden Ausdrücke keine Konstituentenkandidaten sein.

1. Er gibt **dem alten** Hund **den** Knochen.
Er gibt **den** Hund **dem alten** Knochen.
2. **Dein Freund aus Paris kommt morgen** aus Rom.
Aus Paris kommt morgen dein Freund aus Rom.

Gewisse Konstituenten eines Satzes sind vertauschbar, weil die Wortstellung der Argumente einer syntaktischen Konstruktion nicht festgelegt ist, z.B. Argumente des Verbs.

Aber: Bei Koordinationen kann man die Glieder auch dann (sinngleich) vertauschen, wenn sie keine Konstituenten sind:

Er war verwirrt, weil ihn **die Frau geküßt** und **der Mann geohrfeigt** hatte.

Die Glieder der Koordination sind hier auch keine *mittelbaren* Konstituenten des Satzes.

Topikalisierungsprobe: Eine Teilfolge eines Satzes, deren Verschiebung vor das finite Verb –die Topikposition– wieder einen Satz ergibt, ist ein Konstituentenkandidat.

Hierbei zählt auch der Satzanfang bis (ausschließlich) zum finiten Verb selbst als in die Topikposition verschoben.

Auf jeden Fall komme ich zu dieser Feier.

Hast du *den* **einen feigen Hund** genannt?

Einen feigen Hund hast du *den* genannt?

Die Topikalisierungsprobe ist recht gut: die meisten Konstituenten eines finiten Satzes lassen sich in die Topikposition verschieben.

Andererseits gibt es eine Tendenz, auch Teile ins Vorfeld zu stellen, die man nicht gern als Konstituente betrachten würde, etwa

Einen feigen Hund genannt hast du *den*?

Dagegen geklagt hatten zwei Rechtsanwälte aus Ulm.

Tilgungsprobe: Entsteht aus einem Satz uvw durch Weglassen von $v \neq \epsilon$ wieder ein Satz, so ist v ein Konstituentenkandidat von uvw .

Die Probe erkennt quasi die optionalen „freien Angaben“, wie in

Das brave Kind spielt **im Sandkasten des Kindergartens**.

Sie schlägt aber auch mittelbare optionale Ergänzungen als Kandidaten vor, z.B. **des Kindergartens** oder **brave**.

In dem Beispiel (Grewendorf, 1988)

Dieses Lehrbuch **der Sprachwissenschaft** ist besser als jenes.

können wir sowohl *der Sprachwissenschaft* als auch *Lehrbuch der Sprachwissenschaft* weglassen, und erhalten wieder Sätze. Also sind dies Konstituentenkandidaten.

Koordinationsprobe: Ist eine Konstituente eines Satzes die Koordination zweier Ausdrücke, so sind dies (mittelbare) Konstituenten.

$$\forall o, u, v_1, v_2, w \in \Sigma^*$$

$(o \in \text{Conj} \wedge v_1 o v_2 \text{ ist Konstituente von } uv_1 o v_2 w \in S$

$\Rightarrow v_1 \text{ und } v_2 \text{ sind mb. Konstituenten von } uv_1 o v_2 w)$

Dabei sei *Conj* die Klasse der koordinierenden Konjunktionspartikel, z.B. und, aber, und oder. Wenn ein Satz $uv_1 o v_2 w \in S$ vorliegt, muß man zuerst die Koordination $v_1 o v_2$ als Konstituente erkennen.

1. Meier und Lehmann grüßen einander nicht mehr.
2. Sie tat ihre Arbeit gerne und mit größter Sorgfalt.
3. Kolumbus entdeckte Amerika und eine ganze Reihe kleinerer Inseln in der Karibik. (Aber: eine ganze Reihe kleinerer Inseln?)
4. Aber: Kommt er mit dem Auto, der U-Bahn, oder dem Rad?
5. Aber: Sie überlegt es sich heute und morgen macht sie es.

Die grobe Formulierung

Was sich koordinieren läßt, ist eine Konstituente.

ist vielleicht so gemeint:

$$\forall o, u, v_1, v_2, w \in \Sigma^* \quad (uv_iw \in S \wedge o \in \text{Conj} \wedge uv_1ov_2w \in S \\ \Rightarrow v_i \text{ ist Konstituentenkandidat von } uv_iw)$$

Aber damit wäre jedes Suffix v_1 eines Satzes $s = uv_1$ ein Kandidat: man kann den Satz zur Konjunktion $s' = sos = uv_1 \circ uv_1$ von Sätzen ergänzen, z.B.

Adam Riese lebt nicht mehr und Adam Riese lebt nicht mehr.

Adam Riese lebt nicht mehr und Adam Riese lebt nicht mehr.

Bem. Durch Weglassen eines Konjunktionsglieds erhält man i.a. keinen korrekten Satz: Peter und Maria lachen. \mapsto * Peter lachen.

Ellipsenprobe: Kann man aus einem Satz w durch Kopieren einer Wortfolge v an eine zweite Position einen neuen Satz bilden, in dem beide Kopien von v Konstituenten sind, so ist v auch eine Konstituente von w .

$\forall v, w_1, w_2, w_3 \in \Sigma^*$

$(w_1 v w_2 w_3 \in S \text{ (bzw. } w_1 w_2 v w_3 \in S)$

$\wedge w_1 v w_2 v w_3 \in S \wedge \text{ beide } v \text{ sind Konstituenten von } w_1 v w_2 v w_3$

$\Rightarrow v \text{ ist Konstituente von } w_1 v w_2 w_3 \text{ (bzw. } w_1 w_2 v w_3 \text{))}.$

Peter liebt $[\cdot]$, aber Karl verehrt **seine Freundin**.

Emil **spielt** Geige und Maria $[\cdot]$ Violine.

Manche Menschen **mögen ihren Beruf**, andere $[\cdot]$ $[\cdot]$ nicht.

Die elliptische Konstruktion eines Satzes besteht darin, aus einem anderen Satz einige, aber nicht alle Vorkommen eines Teilausdrucks wegzulassen. (Man kann nicht beliebige v in $w = w_1 w_2$ einfügen und dann w als Ellipse von $w_1 v w_2$ ansehen.)

An der Tilgungs- und Pronominalisierungsprobe sieht man:

- ▶ Sätze können *optionale* Teile enthalten, die weglafbar sind.
Das Wetter ist heute nicht schön.
- ▶ Was sich durch Pronomen ersetzen ldfbt, ist i.a. *nicht* weglafbar.
Von euch erfuh er, daB die Erde eine Kugel ist.
Von euch erfuh er es. – * Von euch erfuh er.

Aber manchmal eben doch:

Er liest das Buch. – Er liest.

Fazit Konstituententests

- ▶ Man wende die Tests nur auf *einfache*, unkoordinierte Sätze an.
- ▶ Es werden i.a. *mittelbare* Konstituentenkandidaten gefunden; sind Kandidaten für *unmittelbare* Konstituenten maximale?
- ▶ Die ermittelten Kandidaten sind problematisch, wenn sie Konjunktionen enthalten, z.B. beim Pronominalisierungstest

Ich kenne sie, aber sie kennt mich nicht.

Ich kenne sie nicht.

Unklar bleibt (z.B. bei der Topikalisierung):

- ▶ Was wird durch die Tests eigentlich ermittelt – welche *Einheiten* oder Wortgruppen?
- ▶ Betreffen die Konstituenten nur die erlaubten Wortstellungen?
- ▶ Sind die Konstituenten auch bedeutungstragende Teile?

Konstituententests unterhalb der Satzebene

Bezieht man sich nicht auf Sätze, sondern kleinere Ausdrücke, so lauten die Tests zum Teil anders.

Zur Ermittlung von Konstituenten innerhalb von Nominalphrasen:

- ▶ Frage- und Pronominalisierungstest:

	Proform <i>proAP</i>	Frageproform <i>quAP</i>
Adjektiv	solch-	was für ein
Nomen ^{gen}	dessen/sein	wessen

ein **sehr** altes Buch – ein **solches** Buch – **was für ein** Buch
eine **größere** Stadt **als** Ulm – eine **solche** Stadt
die Seiten **des** Buches – **seine** Seiten – **wessen** Seiten

Zur Ermittlung von Konstituenten innerhalb der Adjektivphrase:

- ▶ Frage nach Gradausdrücken:
ein **fast 20m** hoher Turm – ein **wie** hoher Turm

Helfen die Tests, eine innere Struktur von Ausdrücken zu finden?

Beispiel: Soll man in $NP \rightarrow Art AP N S_{rel}$ noch weiter gruppieren?

Ich habe einige grüne Männer, die vom Mars kamen, gesehen

Wenn ja, wie?

- ▶ $Art (AP (N S_{rel}))$ einige grüne Männer, die ... kamen
- ▶ $Art ((AP N) S_{rel})$ einige grüne Männer, die ... kamen
- ▶ $(Art AP) (N S_{rel})$ einige grüne Männer, die ... kamen

Weglassungen, Vertauschungen, und Topikalisierungen sind viele möglich – das besagt also für die innere Struktur nicht viel:

Ich habe einige grüne Männer gesehen, die vom Mars kamen,
grüne Männer, die vom Mars kamen, habe ich einige gesehen
grüne Männer habe ich einige, die vom Mars kamen, gesehen
Männer, die vom Mars kamen, habe ich einige grüne gesehen

Valenz: Komplemente vs. Argumente

Aristoteles bemerkte, daß die einfachsten Sätzen aus zwei Teilen bestehen: Etwas wird über etwas ausgesagt ($\tau\acute{\iota}$ κατὰ τινός). Und die Teile haben eine unterschiedliche *Funktion*: das eine *benennt* das „Zugrundeliegende“, das andere *besagt* (prädiziert) etwas.

Das Prädikat erfordert ein Argument, worüber es ausgesagt wird.

Aristoteles trennt Nomina in

- ▶ einstellige, die etwas als das benennen, was es *für sich* oder *absolut* ist, z.B. *Mensch*, und
- ▶ mehrstellige, die etwas als das benennen, was es *in Bezug auf anderes* oder *relativ* ist, z.B. *Sklave*.

Eine entsprechende Unterscheidung macht er bei den Prädikaten:

- ▶ dieses Pferd ist weiß
- ▶ dieser Berg ist hoch (im Vergleich zu anderen); höher als jener

Argumente

C.S.Peirce (Die Logik der Relative, 1897) hat die Analogie zwischen Aussagen und chemischen Verbindungen betrachtet:

John gives John to John:

Ammoniak:

John
|
John — gives — John

H
|
H — N — H

Nomen, Verben und Adjektive hat er als logische Relationen mit einer *Wertig-* oder *Stelligkeit* betrachtet: ihre *Argumentstellen* müssen von (evtl. gleichen) Argumenten besetzt sein, damit eine logische Aussage entsteht. (G. Frege: „(un)gesättigte Ausdrücke“)

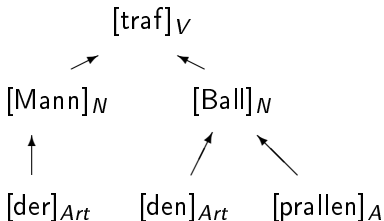
Das „logische Subjekt“ ist dann i.a. das Tupel der Argumente.

Boethius(450): **Subjectum** est id de quo dicitur id quod **praedikatur**.

Komplemente

Ebenso vergleicht L. Tesnière (Eléments de syntaxe structurale, 1959) die Verben mit Atomen, die „Aktanten an sich ziehen und in Abhängigkeit halten“. Das Verb hat eine *Wertigkeit* oder *Valenz*.

Er beschreibt Sätze durch Abhängigkeiten zwischen Wörtern, die mittelbar vom Verb abhängen (*Dependenzgrammatik*):



Die Ausdrücke, die die Wertigkeiten füllen, die *Komplemente*, müssen keine logischen Argumente sein: *Er träumt einen Traum*.

Optionale und obligatorische Komplemente

Obwohl die Komplemente *als logische Argumente* notwendig sind, müssen sie nicht immer explizit im Satz auftreten:

- ▶ Er arbeitet **an einem Roman**.
- ▶ Sie liest **ein Buch**.
- ▶ Wir sind gespannt **auf den Beweis**.
- ▶ Das Kind hat Angst **vor dem Gewitter**.

Sie können implizit „durch den Kontext“ bereitgestellt werden.

- ▶ Komplemente von Nomen und Adjektiven sind i.a. optional.

Optional: Du sollst **uns** nicht stören.

Obligatorisch: Du sollst **uns** nicht von der Arbeit abhalten.

Variable Stelligkeit

Die Stelligkeit von Verben und Adjektiven ist systematisch variabel:

1. Das Verb hat im Aktivsatz eine obligatorische Stelle **mehr** als im Passivsatz: *Er lobt sie.* vs. *Sie wird gelobt.*
2. Das Adjektiv ist im Positiv **ein-**, im Komparativ **zweistellig**:
Dieser Stein ist schwer.
Dieser Stein ist schwerer als jener Stein.

Aber: auch im Komparativ wird es einstellig verwendet:
ein älterer Herr. vs. *ein besserer Spieler als Boris*

Daher:

- ▶ Die syntaktische Stelligkeit eines Verbs oder Adjektivs (und die Form der Komplemente) variiert mit seiner Verwendung.
- ▶ Das Lexikon muß die Form der Komplemente und den Bezug zu den logischen Argumentstellen pro Verwendung angeben, z.B. lobt : $V_{[nom,akk]}^{fin}$, gelobt : $V_{[nom,akk]}^{part,aktiv}$, gelobt : $V_{[?von,nom]}^{part,pass}$

Komplemente vs. freie Angaben

Neben den optionalen Komplementen lassen sich auch andere Ausdrücke meist weglassen: die „freien Angaben“ oder Umstandsbestimmungen (Adverbiale) z.B. zu

- ▶ Ort, Zeit, Dauer, Häufigkeit bei Handlungs- u. Ereignisverben,
- ▶ Absicht, Gemütszustand des Handelnden bei Handlungsverben, *Er schrieb den Aufsatz ungerne / schlecht gelaunt.*
- ▶ Grund, Häufigkeit, Wahrscheinlichkeit bei Ereignisverben

Test(?): Komplemente sind durch Pronomen ersetzbar.

Wir putzen **den ganzen Tag** **das Haus**. * Wir putzen **ihn** **es**.

Wir putzen **es** **den ganzen Tag**. * Wir putzen **es** **ihn**.

Aber: gelegentlich sind Angaben

- ▶ *notwendig*, also Argumente: *Wir wohnen in München.*
- ▶ *nicht* erlaubt: *Heute ist 12 gerne größer als 10.*

Rektion und Kongruenz

Nicht nur Wörter, auch größere Ausdrücke haben Formen:

das große Haus ein großes Haus
die großen Häuser den großen Häusern

Kongruenz: Die „Zusammengehörigkeit“ von Wörtern ist oft durch *Formgleichheit* gekennzeichnet: die Wörter nehmen in der Gruppe die gleichen (bzw.: bestimmte) Formen an.

$$NP^{sg,nom} \rightarrow Art^{sg,nom} AP^{sg,nom} N^{sg,nom}$$

Rektion: Die Abhängigkeit einer Wortgruppe von einem Wort ist oft durch eine Formabhängigkeit gekennzeichnet. Z.B.: ein Valenzträger bestimmt oder *regiert* die Form seiner Komplemente.

$$S \rightarrow NP^{sg,nom} V_{[nom,akk]}^{sg} NP^{pl,akk}$$

Kongruenz

- ▶ Kongruenz ist eine symmetrische Relation: wenn X mit Y kongruiert (in einem Merkmal Z), dann auch Y mit X .
- ▶ Kongruenz bedeutet, daß Formen übereinstimmen müssen, damit Wörter eine Gruppe (bestimmter Art) bilden.
- ▶ Das heißt nicht, daß *alle* Wörter gleicher Form in einem Satz auch eine Einheit oder Konstituente bilden:

Wir nennen [ihn]^{sg,akk} [einen großen Denker]^{sg,akk}.

Typische Fälle von Kongruenz:

1. Artikel, Adjektiv und Nomen kongruieren im Numerus und Kasus in ($NP \rightarrow Art^{sg,dat} A^{sg,dat} N^{sg,dat}$)
2. „Subjekt“ und Verb kongruieren in Person und Numerus in ($S \rightarrow NP^{pers,sg,nom} V_{[nom]}^{pers,sg}$)
Auch bei Subjektsätzen? (*Daß $2 + 2 = 2 \cdot 2$*)^{3,sg}, *ist*^{3,sg} schön.

Rektion

- ▶ Rektion ist eine unsymmetrische Relation: wenn X Y im Merkmal m regiert, dann regiert Y X nicht im Merkmal m .
- ▶ Verschiedene Merkmale eines Worts X können von verschiedenen anderen Y, Z regiert werden: in

$$(NP \rightarrow \text{Art}_{\text{Def}}^{\text{Gen}} \text{A}^{\text{Gen, ADekl}} \text{N}_{\text{Gen}})$$

wird das Genus von Artikel und Adjektiv vom Nomen regiert, die Deklination des Adjektivs vom Artikel:

der_{def} dumme^{mask,sw} Junge_{mask}
ein_{indef} dummer^{mask,ge} Junge_{mask}

Problem: a) (bisher): Subjekt und Verb *kongruieren* im Numerus.
oder b) (O.Jespersen): Das Subjekt *regiert* den Numerus des Verbs.

$$a) (S \rightarrow NP^{\text{Num}, \text{nom}} V^{\text{Pers}, \text{Num}}) \quad b) (S \rightarrow NP^{\text{nom}}_{\text{Num}} V^{\text{Pers}, \text{Num}}).$$

Also: ist der Numerus der NP variabel: NP^{Num} , oder fest: NP_{Num} ?

Hornformelgrammatik (Definite Clause Grammar)

In der Programmiersprache Prolog kann man kontextfreie Grammatiken, erweitert um Merkmale, leicht definieren:

1. Kategorien: $\langle \text{Kat} \rangle ::= \text{Prolog-Terme}$ (außer Variable, ...)

2. Startsymbole: $\text{startsymbol}(\langle \text{Kat} \rangle) :- \langle \text{Bedingung} \rangle.$

3. Regeln:

$\langle \text{Regel} \rangle$	$::=$	$\langle \text{Kat} \rangle \text{ --> } \langle \text{RS} \rangle.$	
$\langle \text{RS} \rangle$	$::=$	$\langle \text{Kat} \rangle$	Nonterminal
		$(\langle \text{RS} \rangle , \langle \text{RS} \rangle)$	Sequenz
		$(\langle \text{RS} \rangle ; \langle \text{RS} \rangle)$	Alternative
		$[\langle \text{Atom} \rangle]$	Terminal
		$\{\langle \text{Prolog} \rangle\}$	Nebenbedingung

4. Hilfsprädikate: durch Prolog-Klauseln definierbar

Abkürzungen: X, Y, Z für $(X, (Y, Z))$ $X; Y; Z$ für $(X; (Y; Z))$

Beispiel einer DCG

```
startsymbol(s(praes)).  
startsymbol(np(Kas)) :- kasus(Kas).
```

```
kasus(nom).  
kasus(akk).
```

```
s(praes) --> np(nom), tv([nom,akk],praes), np(akk).  
np(Kas) --> art(Kas), a(Kas),  
             n([mask],Kas), {kasus(Kas)}.  
art(nom) --> ['Der'] ; [der].  
art(akk) --> ['Den'] ; [den].  
a(nom) --> [alte] ; [junge].  
a(akk) --> [alten] ; [jungen].  
n([mask],Kas) --> (['Mann'] ; ['Baum']), {kasus(Kas)}.  
tv([nom,akk],praes) --> [faellt] ; [fällt].
```

Parser Laden und Aufrufen

```
?- ['DCGparser/parser.pl', 'DCGparser/testgrammatik.pl'].  
% ... DCGparser/testgrammatik.pl compiled 0.00 sec, 4 bytes  
?- parse.
```

Beende die Eingabe mit <Punkt><Return>

```
|: der junge Mann faellt den alten Baum.
```

```
Aufruf: s(praes,_, [der, junge, 'Mann', faellt, den, alten, 'Baum']
```

Baum:

```
s(praes)  
  np(nom)  
    art(nom) der  
    a(nom) junge  
    n([mask], nom) 'Mann'  
  tv([nom, akk], praes) faellt  
  np(akk)  
  ...
```

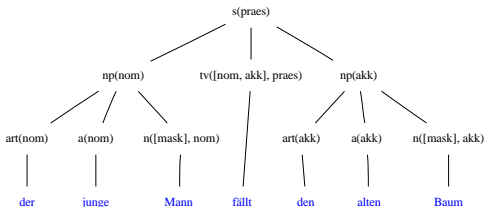

Parser Laden und Aufrufen

```
?- ['DCGparser/parser.pl', 'DCGparser/testgrammatik.pl'].  
% ... DCGparser/testgrammatik.pl compiled 0.00 sec, 4 bytes
```

```
?- parsed.
```

```
Beende die Eingabe mit <Punkt><Return>
```

```
|: der junge Mann faellt den alten Baum.
```



Beispiel: Nominalphrasengrammatik

Nominalphrasen: Ausdrücke, die als Objekte auftreten und durch Personalpronomen ersetzbar, aber keine Sätze sind

Kategorie: *NP*

Hilfskategorien: *Det, Art, Qnt, Poss, Pron, N, N', PN, MN, CN, AP*

Grammatikregeln:

- ▶ atomare *NP*: Personalpronomen, Eigennamen, artikellose Klassennomen (Plural), artikellose Massennomen (Singular)
- ▶ zusammengesetzte *NP*: $Det + AP^* + (N + NP^{gen} + \text{Objekt}) + S_{rel}$
 1. Objekte von Nomen sind optional, folgen direkt auf das Nomen
 2. Attribute wie *AP*-Folgen und S_{rel} (oder *PP*) sind optional
 3. determinierende Elemente sind Artikel, Quantoren, Possessiv- und Personalpronomen, NP^{gen} , manche Kombinationen davon
 4. *Det, AP, N* kongruieren im Numerus und Kasus
 5. *N* regiert *Det, AP* im Genus, *Det* regiert *AP* in der Deklination

Unklar:

1. Sind AP -Attribut und S_{rel} - oder PP -Attribut gleichrangig?

das (grobe Benehmen), das mich stört

↳ (das grobe Benehmen) stört mich

das grobe (Benehmen, das mich stört)

↳ (das Benehmen, das mich stört), ist grob

2. Stehen NP^{gen} -Attribute stets enger am N als Objekte, PP - und S_{rel} -Attribute?

das Auto des Chefs auf dem Hof

= das Auto, das dem Chef gehört und das auf dem Hof steht

≠ das Auto auf dem Hof des Chefs

das Auto des Mannes aus der Führungsetage

= das Auto, das dem Mann aus der Führungsetage gehört

≠ das Auto aus der Führungsetage des Mannes

NP-Grammatik ohne Merkmale

Kategorie	Konstruktion	Beispiele
<i>NP</i>	= <i>Pron</i>	er
	<i>PN</i>	Maria
	<i>CN</i>	Menschen
	<i>MN</i>	Wein
	<i>N'</i>	Kinder, die noch nicht laufen können,
	<i>Det N'</i>	kein Schwein im Stall
<i>N'</i>	= $N_{\square} NP^{gen}$	Schatten (des Körpers (des Kutschers))
	$N (S_{rel} \mid PP)$	Hund (, der) hinter dem Ofen (liegt,)
	$N_{[P]} NP^{gen} PP^P$	Angst _[vor] des Tormanns vorm Elfmeter
	$N_{[gen]} NP^{gen}$	Ermordung _[gen] des Cäsar
	$N_{[dass]}, \text{daß } S^{vl},$	These, daß die Erde eine Kugel ist,
	$N_{[dass]}, S^{vz,konj},$	Nachricht, ein Prophet komme,
	$N_{[infzu]}, S^{infzu},$	Entschluß, nicht mehr zu rauchen,
	<i>AP N'</i>	sehr große (Angst vor dem Untergang)

Kategorie	Konstruktion	Beispiele
<i>Det</i>	= <i>Art</i>	ein; der; dieser; jener
	<i>Qnt</i>	jeder; mancher; kein; viele
	<i>Poss</i>	mein; dein; sein; ihr; unser; euer; ihr
	<i>NP^{gen}</i>	des Bäckers
	<i>Pron</i>	ich
N_{\square}	= <i>PN</i>	Maria;
	<i>CN</i>	Hunde
	<i>MN</i>	Wein
	<i>NA</i>	Eitelkeit; Schönheit; Ängstlichkeit
$N_{[Arg]}$	= <i>NV_[Arg]</i>	Versuch _[infzu] ; Hoffnung _[auf] Tragen _[gen] ; Behauptung _[dass]
	<i>NA_[P]</i>	Eifersucht _[auf] ,

Das Lexikon muß klären, wie die Komplemente gemeint sind:

entschließen: $V_{[nom, refl(akk), infzu]}$ \mapsto Entschluß: $NV_{[?gen, infzu]}$

ermorden: $V_{[nom, akk]}$ \mapsto Mord: $NV_{[?gen, an]}$

NP-Grammatik mit Merkmalen

Kategorie und Merkmale: NP^{Kasus}
Person, Genus, Numerus

- ▶ Person, Numerus, Kasus für Kongruenz mit dem Verb nötig
- ▶ Nominalphrasen sind i.a. nur im Kasus variierbar (Eigennamen)
- ▶ nur (in)definite NPs, keine interrogativen oder relativierenden

Hilfskategorien und ihre Merkmale:

- ▶ $Det^{Genus, Kasus}$
Person, Numerus, Deklinationstyp
- ▶ $Art^{Genus, Kasus}$
Deklinationstyp, Numerus und $Qnt^{Genus, Kasus}$
Deklinationstyp, Numerus
- ▶ $Poss^{Person, Genus, Numerus, Kasus}$ und $Pron^{Person, Genus, Numerus, Kasus}$
- ▶ $N^{Numerus, Kasus}$
Komplementrahmen, Genus
- ▶ $N^{Deklinationstyp, Numerus, Kasus}$
Genus
- ▶ $AP^{Deklinationstyp, Komparation, Genus, Numerus, Kasus}$

(definite) Nominalphrasen

Kongruenz und Rektion werden durch Variablengleichheiten gezeigt.

Kategorie	Konstruktion
$NP_{Pers, Gen, Num}^{Kas}$	$=$
	$Pron^{Pers, Gen, Num, Kas}$ (Gen=-; Pers=3, Num=sg)
	PN_{Gen}^{Kas} (Pers=3, Num=sg)
	$N'_{Gen}^{Adekl, Num, Kas}$ (Pers=3, Adekl=st, Num=pl)
	$Det_{Pers, Num, Adekl}^{Gen, Kas}$ $N'_{Gen}^{Adekl, Num, Kas}$

```

<npgram.pl>≡
  np([Pers,Gen,Num],[Kas]) -->
    ( pron([], [Pers,Gen,Num,Kas])           % er, sie, es
    ; pn([Gen],[Kas]), { Pers=3,Num=sg } % Hans, Maria
    ; n1([Gen],[Dekl,Num,Kas]),             % alte Menschen
      { Pers=3,Num=pl,Dekl=st }
    ; det([Pers,Num,Dekl],[GenD,Kas]), % der alte Hund
      { GenD=(-) ; GenD=Gen },           % ein alter Hund
      n1([Gen],[Dekl,Num,Kas])           % ich alter Hund
      % alle Hunde, die ...
    ).

```

Determinatoren (und Adjektive) haben im Plural und in erster und zweiter Person kein echtes Genus, sondern einen Wert -, um ihr Paradigma klein zu halten.

Einfache und attribuierte Nomina mit Komplementen

Kategorie	Konstruktion
$N'_{Gen}{}^{Dekl, Num, Kas}$	$= N_{[], Gen}{}^{Num, Kas} (NP_{3, Gen', Num'}^{gen})$ $N_{[P], Gen}{}^{Num, Kas} (NP_{3, Gen', Num'}^{gen}) P_{Kas'} NP_{Pers, Gen2, Num2}^{Kas'}$ $N_{[gen], Gen}{}^{Num, Kas} NP^{gen}$ $N_{[dass], Gen}{}^{Num, Kas}$, daß S^{vl} , $N_{[dass], Gen}{}^{Num, Kas}$, $S^{vz, konj}$, $N_{[infzu], Gen}{}^{Num, Kas}$, S^{infzu} , $AP^{Dekl} N'_{Gen}{}^{Dekl, Num, Kas}$ $N_{Kompl, Gen}{}^{Num, Kas} (S_{rel} P_{Kas'} NP_{Pers', Gen', Num'}^{Kas'})$

Implementierung als DCG

$\langle npgram.pl \rangle + \equiv$

```
% N mit Komplement und Attribut: n1([Gen],[Adekl,Num,Kas])
```

```
n1([Gen],[Dekl,Num,Kas]) -->
```

```
  { deklinationstyp(Dekl) },
```

```
  n([Kompl,Gen],[Num,Kas]),
```

```
  ( np([3,-,-],[gen])
```

```
  ; []          % optionaler Gen.possessivus/subjectivus
```

```
  ),
```

```
  ( []          % Nomen ohne Komplemente: Mensch, Ruhe
```

```
    % Nomenkomplemente optional: Hoffnung
```

```
  ; { Kompl=[gen] },
```

```
    []          % gen-Objekt statt Gen.subjectivus
```

```
  ; { Kompl=[Praep], % Angst vor den Hunden
```

```
    praeposition(Praep,KasP) },
```

```
    [Praep], np([_Pers,_Gen,_Num],[KasP])
```

Implementierung als DCG

Nomina mit Objektsätzen oder Relativsätzen:

$\langle npgram.pl \rangle + \equiv$

```
; { Kompl=[dass] }, % These, dass sie rund ist
  ( ([daß] ; [dass]), s([],[v1])
    ; s([],[vz,konj]) ) % These, sie sei rund
  )
; { Kompl=[ob] }, % Untersuchung, ob sie lebt
  [ob], s([],[v1])
; { Kompl=[qu] }, % Frage, wo Maria lebt
  s([qu],[v1])
; { Kompl=[infzu] }, % Absicht, zu rauchen
  s([infzu],[ ])
; s([rel],[ ]) % Menschen, an die wir denken
).
```

praeposition(auf,akk). praeposition(in,akk).

praeposition(an,akk). praeposition(vor,dat).

Attributierung durch eine Adjektivphrase im Positiv:

$\langle np_{grm}.pl \rangle \equiv$

```
n1([Gen],[Dekl,Num,Kas]) -->
  ap([], [Kompl,Dekl,GenD,Num,Kas]), alte Hunde, die ...
  { GenD=(-) ; GenD=Gen },
  { Kompl = pos }, % TODO: Kompl=komp, sup
n1([Gen],[Dekl,Num,Kas]).
```

Determinierende Ausdrücke

Kategorie	Konstruktion
$Det_{Pers, Num, Dekl}^{Gen, Kas}$	$=$ <ul style="list-style-type: none"> <li data-bbox="546 350 913 412"> $Art_{Dekl, Num}^{Gen, Kas} \text{ (Pers=3)}$ <li data-bbox="546 433 924 495"> $\quad Qnt_{Dekl, Num}^{Gen, Kas} \text{ (Pers=3)}$ <li data-bbox="546 515 1177 578"> $\quad Poss_{Pers', Gen', Num'}^{Gen, Num, Kas} \text{ (Pers=3, Dekl=ge)}$ <li data-bbox="546 598 1105 660"> $\quad NP_{3, Gen', Num'}^{gen} \text{ (Pers=3, Dekl=ge)}$ <li data-bbox="546 681 1372 743"> $\quad Pron^{Pers, Gen, Num, Kas} \text{ (Pers} \neq 3, Kas \neq gen, Dekl=st)$

Implementierung als DCG

$\langle np_{grm}.pl \rangle + \equiv$

```
det([Pers,Num,Dekl],[Gen,Kas]) -->
  ( ( art([Dekl,Num],[Gen,Kas]),           % der, ein
    ; qnt([Dekl,Num],[Gen,Kas]),         % kein, jeder
    ; poss([_,_,_],[Gen,Num,Kas]),{Dekl=ge} % mein, unser
  ), {Pers=3}
; pron([], [Pers,Gen,Num,Kas]),          % in manchen
  { (Pers=1 ; Pers=2)                    % Formen:
    , (Kas=nom ; Kas=dat ; Kas=akk)
    , Dekl=st }                          % ich armer Student
).
```

$\langle \text{Linksrekursion, führt beim Top-Down-Parsen zur Endlosschleife} \rangle \equiv$

```
np([Pers,Gen,Num],[Kas]) -->
  np([3,_Gen,_Num],[gen]),{Dekl=ge} % des Bäckers Brote
  n1([Gen],[Dekl,Num,Kas]).
```

Nomina ohne bzw. mit Komplementen

Kategorie	Konstruktion	Beispiel
$N_{[], Gen}^{Num, Kas}$	= PN_{Gen}^{Kas} (Num=sg)	Peter, Maria
	$CN_{Gen}^{Num, Kas}$	Hund, Mensch
	MN_{Gen}^{Kas} (Num=sg)	Wasser, Ruhe
$N_{Kompl, Gen}^{Num, Kas}$	= $NA_{Kompl, Gen}^{Num, Kas}$	Besorgtheit um
	$NV_{Kompl, Gen}^{Num, Kas}$	Hoffnung auf
	$N_{[P], Gen}^{Num, Kas}$	Angst vor, Neid auf

Implementierung als DCG

Nomina mit und ohne Komplemente:

$\langle npgrm.pl \rangle + \equiv$

```
% Zusammenfassung von Nomenarten: n([Kompl,Gen],[Num,Kas])
```

```
n([],Gen,[Num,Kas]) -->
```

```
  ( pn([Gen],[Kas]), {Num=sg} % Peter, Maria
    ; cn([Gen],[Num,Kas])      % Hund, Menschen
    ; mn([Gen],[Kas]), {Num=sg} % Wasser, Ruhe
  ).
```

```
n([Kompl,Gen],[Num,Kas]) -->
```

```
  ( na([Kompl,Gen],[Num,Kas]) % Gute, Neid auf
    ; nv([Kompl,Gen],[Num,Kas]) % Hoffnung auf
  ).
```

Im Lexikon kann man auch Nomina mit Komplementen erlauben, die weder nominalisierte Adjektive noch nominalisierte Verben sind.

Adjektivphrasen

- ▶ Adjektiv mit Komplementen,
- ▶ nur im Positiv
- ▶ keine Modifikation durch Gradausdrücke/Adverbien
- ▶ nur zur Verwendung als Nomenattribut
- ▶ keine Vergleichskonstruktionen (so A er N wie NP)

Kategorie	Konstruktion	Beispiel
$AP^{Komp, Dekl, Gen, Num, Kas}$	$= (P_{Kas'} NP_{Pers', Gen', Num'}^{Kas'})$ $A_{[P]}^{Dekl, Gen, Num, Kas}$	auf sie
	$ (P_{Kas'} NP_{Pers', Gen', Num'}^{Kas'})$ $(NP_{Pers, Gen2, Num2}^{Kas2})$	neugieriges von uns
	$A_{[P, Kas2]}^{Dekl, Gen, Num, Kas}$	den Kindern empfohlenes

Implementierung als DCG

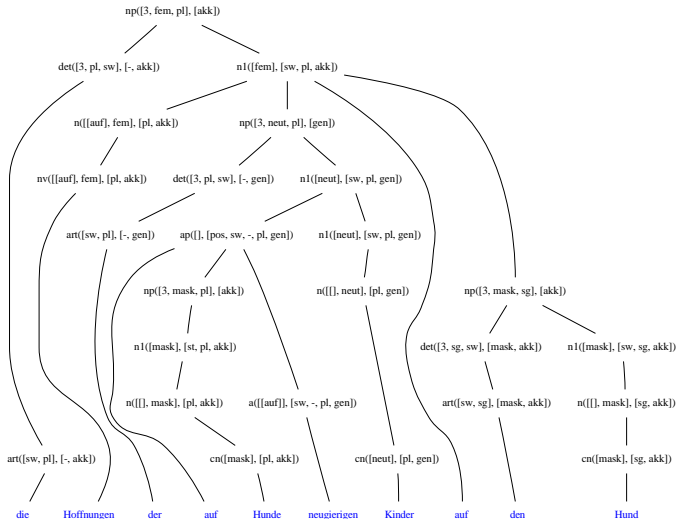
Adjektivphrasen (positiv) aus Adjektiv mit optionalem Komplement:

$\langle npgrm.pl \rangle + \equiv$

```
% Adjektivphrasen: ap([], [Komp, Dekl, Gen, Num, Kas])
```

```
ap([], [pos, Dekl, Gen, Num, Kas]) -->
  ( [] % Komplemente stets optional
    ; { Kompl=[Pr], praeposition(Pr, KasP) },
      [Pr], np([_Pers, _Gen2, _Num2], [KasP])
    ; { Kompl=[Pr, KasA0], praeposition(Pr, KasP) },
      [Pr], np([_Pers, _Gen2, _Num2], [KasP]),
      np([_PersA0, _GenA0, _NumA0], [KasA0])
    ), % Linksrekursion durch Stellung vermieden!
  a([Kompl], [Dekl, Gen, Num, Kas]).
```

Beispiel einer komplexen Analyse



Syntaktische Funktionen

Aristoteles bemerkte, daß die einfachsten Sätzen aus zwei Teilen bestehen: Etwas wird über etwas ausgesagt (τὶ κατὰ τινός).

Die Teile haben eine unterschiedliche *Funktion*: das eine *benennt* das „Zugrundeliegende“, das andere *besagt* (prädiziert) etwas.

Boethius: **Subjectum** est id de quo dicitur id quod **praedikatur**.

Man kann Äußerungen und Ausdrücke einer Sprache durch *funktionale Begriffe* beschreiben und sich interessieren für die

- ▶ kommunikativen oder pragmatischen Funktionen, d.h. welchen kommunikativen Zweck soll eine Äußerung haben?,
- ▶ für die logischen und semantischen Funktionen, d.h. bildet ein Ausdruck das logische Prädikat oder ein Argument?,
- ▶ für die syntaktischen Funktionen, d.h. welche Rolle spielt ein Ausdruck in einem größeren?

Typische funktionale Begriffe sind „Subjekt“ und „Objekt“.
Ursprünglich sind dies Begriffe der Logik, die erst spät von den Grammatikern übernommen wurden.

Von den *syntaktischen Funktionen*, also Beziehungen zwischen Teilausdruck und Gesamtausdruck, werden manchmal die *syntaktischen Relationen*, d.h. Beziehungen zwischen mehreren Teilausdrücken, unterschieden.

Beispiel: Sind *Andrew Wiles* bzw. *den Satz von Fermat* in

Andrew Wiles bewies den Satz von Fermat.

das Subjekt bzw. Objekt a) des Satzes oder b) des Verbs *bewies*?

Die Begriffe laufen nicht auf dasselbe hinaus:

*Wir sahen*_[nom,akk?|inf?] *euch* *wegfahren*_[nom].

C.Dürscheid, „Syntax. Grundlagen und Theorien“ nennt als syntaktische Funktionen (von Satzgliedern):

- ▶ Subjekt
- ▶ Prädikat(iv)
- ▶ Objekt
- ▶ Adverbial
- ▶ freier Dativ
- ▶ Attribut

Keine der Funktionen außer dem Prädikat muß im Aussagesatz durch einen Ausdruck verwirklicht sein. Zwischen dem Aufbau von Ausdrücken und den Funktionen, die sie realisieren, besteht keine direkte Korrelation.

Terminologie: **Ausdrücke in einer syntaktischen Funktion**

Subjekt XP = Satzsubjekt mit Struktur einer XP

Subjektnominalphrase = Subjekt mit Struktur einer NP

Subjektsatz = Subjekt mit Struktur eines Satzes

Objekt XP = XP -Objekt = Objekt mit Struktur einer XP

Objektnominalphrasen (nominale Objekte)

Objektsätze (satzförmige Objekte)

Objektpräpositionalphrasen (präpositionale Objekte)

Attribut XP = Attribute mit Struktur einer XP

Attributsätze (satzförmige Attribute)

XP -Prädikativ = Prädikat(iv) mit Struktur einer XP

Dagegen: **Syntaktische Funktion relativ zum Ausdruck(skopf)**

Nominalobjekt = Objekt eines Nomens, innerhalb der NP

Adjektivobjekt = Objekt eines Adjektivs, in der AP

Präpositionalobjekt = Objekt einer Präposition, in der PP

Verbalobjekt = Objekt eines Verbs = O. innerhalb des S

Syntaktische Funktionen legen die Form der Ausdrücke, die sie realisieren, nicht eindeutig fest, schränken sie aber z.T. ein.

Kann man durch formale Test erkennen, welche Funktion welcher Ausdruck in einem einfachen deutschen Satz hat?

Subjekt

1. Das Subjekt ist durch *wer|was* erfragbar; sonst durch *wem|wen*.
2. Das NP^{nom} -Subjekt kongruiert mit dem Verb in Pers und Num.
3. Infinitivprobe: Bildet man aus dem Satz den entsprechenden Verbalkomplex, indem man das Hauptverb in den Infinitiv setzt, so bleibt das Subjekt übrig.
4. Passivtransformation: Die Passivform eines Aktivsatzes enthält dessen NP^{nom} -Subjekt in der Form *von NP^{dat}*, oder gar nicht.

Nicht jeder Satz hat ein Subjekt, z.B. der folgende hat keins:

Ab jetzt wird nur noch gefaulenzt.

Formen des Subjekts (des als Subjekt fungierenden Ausdrucks):

Subjektnominalphrasen:

NP^{nom} Diese Behauptungen wundern mich. Er ist ein Genie?
 $NP^{akk|dat}$ Mich fror den ganzen Abend. Mir ist schlecht.

Subjektsätze:

daß-S Daß dies sich als falsch erwies, überraschte uns nicht.
ob-S Ob sich diese Investition lohnt, ist fraglich.
 $S_{qu(kaus)}$ Warum niemand ihm geholfen hat, ist unklar.
 $S_{qu(nom)}$ Wer andern eine Grube gräbt, ist Bauarbeiter.
 S^{infzu} Maria zu lieben ist allzeit mein Sinn.
 S^{inf} Nachdenken und dann handeln ist nicht seine Stärke.

Expletives Es:

es Es regnete sehr stark. Weitere Fragen gab es nicht.

Korrelat-Es: verdoppeltes Subjekt: Es+S bzw. Es+NP

es|das Es|Das ist nicht ausgeschlossen, daß es morgen schneit.
es Es wurden sehr viele Zwischenfragen gestellt.

Objektnominalphrasen (inklusive -subjekte)

1. *NP*-Objekte sind Nominalphrasen, die durch Pronomina, aber nicht durch kasuslose Ausdrücke ersetzt werden können.
2. *NP*-Objekte sind erfragbar durch wer, wessen, wem, wen, was.

Akkusativobjekt (direktes Objekt)

NP^{akk} Sie findet **alle Fehler**. Er arbeitet **einen Tag die Woche**.

S_{qu(akk)} Ich weiß nicht, **wen man das am besten fragen sollte**.

Dativobjekt

NP^{dat} Thomas glaubte **niemandem außer seinen Augen**.

Thomas soll **mir** keinen Unsinn reden!

S_{rel(dat)} Er vertraut nur, **wem er vertrauen muß**.

Genitivobjekt

NP^{gen} Er konnte sich **eines Tages** vieler Details erinnern.

Er konnte sich **eines Tages großer Mühen** erinnern.

*Nominativ Cassius Clay bleibt **der größte Boxer**. Ich bin **der Dumme**.

Objektpräpositionalphrasen

1. *PP*-Objekte sind *PP*-Ausdrücke, deren Präposition vom Verb bestimmt ist und daher für die Bedeutung keine Rolle spielt.
2. *PP*-Objekte können durch Pronominaladverbien (gleicher Präposition, wie in *darunter* leiden, *darauf* warten) ersetzt werden, aber durch keine anderen kasuslosen Ausdrücke.
3. *PP*-Objekte können (nur) mit entsprechendem Fragewort erfragt werden, z.B. mit *worunter* bzw. *unter wem/was*.
4. **Ausnahme:** Präpositionen, die den Genitiv fordern, z.B. *wegen*, *trotz*, leiten keine *PP*-Objekte ein.

PP^{an}[akk] Sie erinnert sich **an diesem Morgen** an den letzten Urlaub.

PP^{an}[akk] Sie erinnert sich **daran**, daß es heiß war.

PP^{vor}[dat] Sie fürchten sich **vor dem Jahresende** vor dem Finanzprüfer.

*PP*_{qu}^{an}[akk] **Woran**|An was|**Daran** erinnert sie sich?

*PP*_{qu}^{vor}[dat] **Wovor**|Vor wem|**Vor was** fürchten sich manche Leute?

Objektsätze

Manche Nebensätze und Infinitiv„sätze“ können auch die Funktion des direkten (Akkusativ-) Objekts übernehmen.

1. Objektsätze bzw. -Infinitive lassen sich durch ein Pronomen – *es, das* –, aber nicht durch kasuslose Elemente ersetzen. (Das Pronomen steht allerdings nicht an derselben Stelle im Satz.)
2. Objektsätze lassen sich mit *was* erfragen.

Objektsätze

S^{infzu} Jemand hatte vergessen, **die Tür abzusperren**.
(Jemand hatte **es** vergessen.)

ob-S Wer weiß schon, **ob das auch den Tatsachen entspricht?**
(Wer weiß **es** schon?)

daß-S **Daß er die Einzelheiten vergessen hat**, behauptet er.
(**Das (*Es)** behauptet er. Er behauptet **es**.)

S^{konj} Er behauptet, **er habe die Einzelheiten vergessen**.

Adverbiale

Adverbiale (Umstandsbestimmungen) sind Satzglieder, die den im restlichen Satz ausgedrückten Sachverhalt näher bestimmen. (Quasi: Prädikate höherer Stufe, über die Kernaussage.)

1. Adverbiale können durch Adverbien ersetzt werden.

Je nach dem, worauf sich das Adverbial bezieht, spricht man von Verbmodifikatoren

Ich durchlief diese Gegend **in großer Geschwindigkeit**.

Sie taten es **mit wenig Aufwand**.

Satzoperatoren

Nach den Mitteilungen eines Beobachters verfinsterte sich die Sonne.

In China ist der Kaiser ein Chinese. (Chr. Andersen)

Bestimmungen

der Zeit

Sie träumt **den ganzen Tag**.

Am Nachmittag passierte nichts.

des Raums

Wir arbeiten **in München**.

der Art

Er spricht **sehr schnell und undeutlich**.

und Weise

Man kommt nach oben, **indem man den Lift benutzt**.

Erhobenen Hauptes traten sie vor ihren Richter.

des Grunds

(kausal)

Wegen des Baulärms kann man hier nicht nachdenken.

(konzessiv)

Trotz des Lärms werde ich weiter arbeiten.

Obwohl das Ergebnis bekannt war, wurde es publiziert.

(konditional)

Im Zweifelsfall sollte man einen Experten fragen.

Wenn man etwas nicht versteht, sollte man fragen.

(konsekutiv)

Das Thema war uninteressant, **so daß niemand kam**.

Wir fanden es **zum Verzweifeln**.

Formen der Adverbiale:

Adverbien, undeklinierte Adjektive und Partizipien des Verbs,
Präpositionalausdrücke, selbständige Nominalausdrücke, Sätze:

Formen von Adverbialen

Adv Die Vögel singen **morgens|oft|gern**.

AdvP Hier ißt man **recht gut**, aber nicht **vorzüglich**.

Ich bin **viel zu schweigsam, um etwas Bemerkenswertes von Schwill oder Noll erzählen zu können**. (R.Wolf, S.104)

Ich erzähle diese Geschichte **hin und wieder ganz gern**.

A^{undekl} Sie freuten sich **diebisch**.

V^{part1} **Bezugnehmend auf Ihr Schreiben** teilen wir Ihnen mit, daß wir Ihren Ausführungen keineswegs folgen können.

V^{part2} **Gefürchtet von seinen Feinden** herrschte er noch Jahre.

Adverbiale Präpositionalausdrücke:

1. *PPs* mit bedeutungsvoller Präposition, die keine Objekte sind.

PP_{Ort} Er stand **vor**|**in**|**hinter** der Tür. Sie stand **daneben**.
PP_{Kaus} Der Wagen blieb **wegen eines Motorschadens** liegen.

Adverbiale Nominalausdrücke:

1. *NPs*, die nicht durch Pronomen, nur durch kasuslose Ausdrücke ersetzbar sind, also keine Objekte.

NP^{gen} Der Zufall wollte es, daß **eines Tages/damals**
ein Männerchor aus dem belgischen Mol vorbeifuhr.
NP^{akk} Er verschwand **eine Zeit/kurz** in Luzern. (R.Wolf, S.176)

Adverbialsätze:

1. Infinitivsätze und Nebensätze, die Satzglieder sind (und keine Relativsätze oder Teilsätze von Koordinierungen).

S^{infzu} Sie arbeiten sehr intensiv, ohne das Problem lösen zu können.
 $conj S$ Weil niemand sich um sie kümmerte, war sie eingerostet.
Obwohl er nichts genaues wußte, redete er die ganze Zeit.

Adverbialsätze können durch Pro-Formen, $proAdvP$, ersetzt werden:

obwohl S^{vl}	↪	trotzdem
weil S^{vl}	↪	deswegen deshalb
nachdem S^{vl}	↪	danach
während S^{vl}	↪	währenddessen
bevor S^{vl}	↪	vorher
um S^{infzu}	↪	dazu dafür

Satzadjektive

Der Duden führt adverbial gebrauchte Adjektive als eigenständige Satzglieder auf.

Damit sind Adjektivphrasen gemeint, die ein undekliniertes Adjektiv oder Partizip als Kern enthalten und sich auf die Subjekt- oder Objektnominalphrasen oder das Verb beziehen kann.

Subjektbezug	AP^{undekl}	Ich war in diesem Moment ganz ruhig.
	V^{part}	Wir waren überrascht und nicht gelangweilt.
Objektbezug	AP^{undekl}	Sie aß den Fisch niemals roh.
	V^{part}	Er trug seine Hemden ungebügelt.
Verbbezug	AP^{undekl}	Ich erwähne den Vorgang nur kurz.

Ambig: Verlassen Sie den Raum in ordentlichem Zustand

Prädikat

Die Funktion des Prädikats übernimmt meist das Verb. Es kann aus der in Person und Numerus mit dem Subjekt übereinstimmenden finiten Form eines Verbs alleine bestehen oder aus mehreren Teilen.

einteiliges Prädikat

V^{fin} Am nächsten Tag **fiel** ein Handschuh vom Himmel.

grammatischer Prädikatsteil

V^{part2} Dieser Mann hat schlecht über mich **gesprochen**.

V^{infzu} Es beginnt wieder **zu regnen**.

V^{inf} Mehr muß man vielleicht auch nicht **wissen**.

lexikalischer Prädikatsteil

N Sie fährt **Mercedes**.

A Er ärgert sich **schwarz**.

$V^{präfix}$ Sie tauchten **unter**.

Teil eines Funktionsverbs

Sie brachte die Diskussion **auf den Punkt**.

Subjektsprädikativ

1. ein Ausdruck, der bei der Infinitivprobe (s.o.) ein Teil des Verbkomplexes bleibt, fungiert als Prädikat über das Subjekt(!) des Satzes, bei wenigen Verben wie *sein*; *scheinen*; *bleiben*; *werden*; *heißen*; *genannt werden*; *gescholten werden*.

Formen:

NP^{nom} (Prädikatsnomen) Karl ist **Redakteur bei einer Tageszeitung**.

Er wurde **der zähste Marathonläufer des Landes** genannt.

NP^{gen} Alle schienen **guten Mutes**, nur Emil war **schlechter Laune**.

S^{inf} Raten heißt **den Menschen Motive zum Handeln geben, die sie nicht kennen**. (Vauvenargues)

Adv Wir bleiben **hier!**

PP Dieser Fund ist **von großer Bedeutung** für die Forschung.

V^{part2} Tim Finnigan war **betrunken**.

AP^{undekl} Die Aufgabe war **ziemlich schwierig, aber nicht unlösbar**.

Objektsprädikativ

1. ein Ausdruck, der als Prädikat über das direkte Objekt fungiert, bei wenigen Verben wie *nennen*; *heißen*; *schimpfen*; *schmähen*; *schelten*. ggf. *finden*; *halten für*

Formen:

- NP^{akk} \neq Objekt (Duden: Gleichsetzungsakkusativ)
Sie schimpften uns **einen Haufen von Faulenzern**.
Nicht nur die Historiker nennen ihn **Karl den Großen**.
- S^{inf} Dieses Benehmen nenne ich **flegeln**.
- AP^{dekl} Solche Methoden nannten sie **äußerst brutale**.

Objekte bei N,A,P

Auch Nomina und Adjektive können nominale, präpositionale und satzförmige Objekte haben, wie in

Er war ein Liebhaber **des Halbschattens**. (A. Andersch)

Das Hauptargument **gegen seine frühere Meinung** ist seine jetzige.

Er konnte seine Behauptung, **daß $e^2 \cdot \pi > 13$ sei**, nicht beweisen.

Er konnte seine Behauptung, **$e^2 \cdot \pi$ sei größer als 13**, nicht beweisen.

Er war **des Predigens** überdrüssig und **aller Sorgen** ledig.

Sie war empört **über die Beschuldigung**.

Wir waren gespannt, **ob es klappt**.

Wir waren begierig, **das Ergebnis zu erfahren**.

Wird das Nomen oder Adjektiv prädikativ verwendet, so nennt man seine Objekte manchmal auch Objekte zweiten Grades (des Verbs).

Attribute zu N,A

Attribute sind syntaktische Funktionen innerhalb der Satzglieder.

Neben den oben genannten Objekten innerhalb von Nominal- und Adjektivphrasen sind die Attribute meist Ausdrücke, die eine untergeordnete Aussage darstellen.

Attribute in der NP:

AP^{dekl} eine **sehr beliebte** Dozentin.

S_{rel} eine Dozentin, **die sehr beliebt war**,

PP die Hackordnung **im Hühnerstall**

Attribute in der AP: (wie Adverbiale in prädikativen APs)

PP eine (**am Nordpol** sehr seltene) Tierart

NP eine (**20 Grad Celsius** warme) Strömung

GradP eine (**nicht sehr** warme) Strömung

AdvP eine (**sehr oft** von ihr zitierte) Bemerkung

Kategorialgrammatiken

K.Ajdukiewicz (Die syntaktische Konnexität, 1935) hat funktionale Begriffe in die syntaktischen Kategorien eingebaut: bei der Verkettung wird der eine Ausdruck als Funktion, der andere als sein Argument behandelt.

$$\frac{u : A/B, \quad v : B}{u \cdot v : A} (/) \qquad \frac{u : B, \quad v : B \setminus A}{u \cdot v : A} (\backslash)$$

Interpretiert man Kategorien durch formale Sprachen (Mengen von Zeichenreihen), so entspricht dies den „Kürzungsregeln“

$$A/B \cdot B \subseteq A \qquad B \cdot B \setminus A \subseteq A$$

oder den kontextfreien Regeln

$$A \rightarrow A/B \cdot B \qquad A \rightarrow B \cdot B \setminus A$$

Funktionale Eigenschaften von Ausdrücken werden direkt durch ihre Kategorien ausgedrückt, nicht wie in einer CFG durch die Regeln zu diesen Kategorien:

Beispiel

1. attributive Verwendung des Adjektivs:

CFG: Kategorie A Regel $N \rightarrow A \cdot N$

KG: Kategorie $A := N/N$

2. transitives Verb im einfachen Satz:

CFG: Kategorie TV

Regeln

$VP \rightarrow TV \cdot NP$

$S \rightarrow NP \cdot VP$

KG: $VP := NP \setminus S$

$TV := VP/NP = (NP \setminus S)/NP$

Es folgt: $NP \cdot (TV \cdot NP) \subseteq NP \cdot (NP \setminus S) \subseteq S$

Kategorialgrammatiken (Bar-Hillel 1960)

Die Menge Cat der **Kategorien**ausdrücke wird aus einer nicht-leeren Menge von *Basiskategorien* durch

$$A, B \in Cat \Rightarrow A/B \in Cat, B \setminus A \in Cat$$

aufgebaut. Eine **Kategorialgrammatik** $G = (\Sigma, T, S)$ besteht aus

- (i) einem endlichen Alphabet bzw. Lexikon Σ ,
- (ii) einer Kategorisierung $T : \Sigma \rightarrow \mathcal{P}_{fin}(Cat)$, und
- (iii) einer Satzkategorie $S \in Cat$.

Die von G definierte Sprache ist $L(G) := \{w \in \Sigma^* \mid T \vdash w : S\}$, wobei $T \vdash v : A$ besagt, daß $v : A$ durch die Axiome

$$u : B, \quad \text{für } B \in T(u),$$

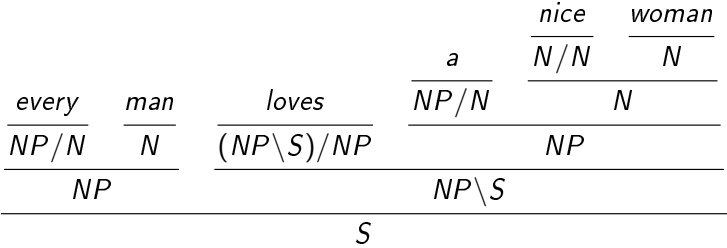
mit den Schlußregeln (/) und (\) beweisbar ist.

Beispiel als CFG

CFG:

$S \rightarrow NP VP$ $TV \rightarrow loves$
 $VP \rightarrow TV NP$ $Det \rightarrow a \mid every$
 $NP \rightarrow Det N$ $N \rightarrow man \mid woman$
 $N \rightarrow A N$ $A \rightarrow nice$

Kurzform: Wörter als Blätter, ihre Kategorien als Präterminale



Wie man sieht, ist das ein „umgedrehter“ Syntaxbaum.

Da die Kürzungsregeln als kontextfreie Grammatikregeln aufgefaßt werden können (s.o.), gilt offenbar:

Prop.: Zu jeder Kategorialgrammatik $G = (\Sigma, T, S)$ gibt es eine kontextfreie Grammatik $G' = (\Sigma, N, P, S)$ mit $L(G) = L(G')$.

Es gilt aber auch die Umkehrung:

Satz (H.Gaifman 1957) Zu jeder kontextfreien Grammatik $G = (\Sigma, N, P, S)$ gibt es eine Kategorialgrammatik $G' = (\Sigma, T, S)$ mit $L(G) = L(G')$.

Der Beweis benutzt, daß man kontextfreie Grammatiken in eine Normalform (Greibach-NF) bringen kann, und konstruiert daraus eine Kategorialgrammatik.

Bemerkung: Bloomfield hatte *Attribute* durch die Distribution definiert: in einer Zusammensetzung $A B$ ist A ein Attribut, wenn in jedem Kontext $A B$ durch B ersetzt werden kann, d.h. wenn $D_L(A B) \subseteq D_L(B)$ ist.

In der Kategorialgrammatik ist ein (vorangestelltes) B -Attribut ein Ausdruck der Kategorie B/B :

$$\frac{v : B/B \quad w : B}{v \cdot w : B}$$

Analog: nachgestellte B -Modifikatoren haben die Kategorie $B \setminus B$.

Ob ein Ausdruck w ein Satz ist, wird bei Ajdukiewicz und Bar-Hillel darauf zurückgeführt, daß $w : A$ und $A \subseteq S$ für eine Kategorie A . Die Inklusion muß durch Anwenden der „Kürzungsregeln“ gezeigt werden. Man braucht i.a. aber weitere Möglichkeiten.

Lambek-Kategorialgrammatik

Zunächst sind Ausdrücke der Kategorie A/B Funktionen, die Ausdrücke der Kategorie B zu Ausdrücken der Kategorie A erweitern, z.B. $\text{every} : NP/N$, $\text{to} : PP/NP$.

Man kann mit A/B aber auch Ausdrücke beschreiben, die „von der Kategorie A wären, wenn ein Ausdruck der Kategorie B folgte“, d.h. als A -Ausdrücke mit einer B -Lücke am Ende:

Beispiel: Relativsätze sind nachgestellte Nomenmodifikatoren, also $S_{rel} := N \setminus N$. Da das Relativpronomen ein vorgezogenes Objekt ist, folgt darauf „ein Satz mit einer NP-Lücke am Ende“:

$$\begin{array}{c} \text{woman} : N \\ \hline \text{whom} : (N \setminus N) / (S / NP) \quad \frac{\quad ?}{\text{he talks to} : S / NP} \\ \hline \text{whom he talked to} : N \setminus N \\ \hline \text{woman whom he talked to} : N \end{array}$$

Man würde gerne aus den Kategorisierungen der Pronomen, Verben und Präpositionen

$$he : NP, \quad talks : (NP \setminus S) / PP, \quad to : PP / NP$$

schließen, daß, weil für beliebige $x : NP$ gilt:

$$\frac{\frac{he : NP}{\quad} \quad \frac{talks : (NP \setminus S) / PP \quad \frac{to : PP / NP \quad x : NP}{to \ x : PP}}{talks \ to \ x : NP \setminus S}}{he \ talks \ to \ x : S},$$

man ohne ein solches $x : NP$ noch hat: $he \ talks \ to : S / NP$.

Man will mit **temporäreren Annahmen** argumentieren können, hier:

$$\frac{he : NP, \ talks : (NP \setminus S) / PP, \ to : PP / NP, \ x : NP \vdash he \ talks \ to \ x : S}{he : NP, \ talks : (NP \setminus S) / PP, \ to : PP / NP \vdash he \ talks \ to : S / NP}$$

Lambek-Kalkül

J.Lambek (The mathematics of sentence structure, 1958) hat einen Beweiskalkül vorgeschlagen, in dem man hypothetische Annahmen verwenden kann, um Inklusionen zwischen Kategorien zu beweisen.

1. **Kategorien** werden aus einer Menge $\{B_i \mid i \in \mathbb{N}\}$ von atomaren Kategorien durch (A/B) , $(A \setminus B)$ und $(A \cdot B)$ erzeugt.
2. **Sequenzen** $C_1, \dots, C_{n+1} \triangleright C_0$ sind Folgen von Kategorien C_i .
3. **Axiome** sind alle Sequenzen $A \triangleright A$, für atomare Kategorien A .

3. **Schlußregeln** zur Herleitung von Sequenzen sind folgende, wobei Γ , Δ und Θ für endliche Folgen von Kategorien stehen, und A , B für beliebige Kategorien :

$$\frac{\Theta \triangleright A \quad \Gamma, B, \Delta \triangleright C}{\Gamma, \Theta, A \setminus B, \Delta \triangleright C} (\setminus L)$$

$$\frac{A, \Gamma \triangleright B}{\Gamma \triangleright A \setminus B} (\setminus R)$$

$$\frac{\Theta \triangleright A \quad \Gamma, B, \Delta \triangleright C}{\Gamma, B/A, \Theta, \Delta \triangleright C} (/ L)$$

$$\frac{\Gamma, A \triangleright B}{\Gamma \triangleright B/A} (/ R)$$

$$\frac{\Gamma, A, B, \Delta \triangleright C}{\Gamma, A \cdot B, \Delta \triangleright C} (\cdot L)$$

$$\frac{\Gamma \triangleright A \quad \Delta \triangleright B}{\Gamma, \Delta \triangleright A \cdot B} (\cdot R)$$

$$\frac{\Theta \triangleright B \quad \Gamma, B, \Delta \triangleright A}{\Gamma, \Theta, \Delta \triangleright A} (\textit{Schnitt})$$

Wir schreiben $\Gamma \vdash A$, wenn $\Gamma \triangleright A$ mit diesen Regeln in endlich vielen Schritten aus den Axiomen herleitbar ist.

Eine **Lambek'sche Kategorialgrammatik** $G = (\Sigma, T, S)$ ist eine Kategorialgrammatik, mit einer Kategorisierung $T : \Sigma \rightarrow \mathcal{P}_{fin}(Cat)$ der aus Basiskategorien mit \backslash , $/$ und \cdot aufgebauten Kategorien Cat , die den Kategorien A die Wortmengen

$$A^G := \{ a_1 \cdots a_n \mid n \in \mathbb{N}, a_1, \dots, a_n \in \Sigma, \\ A_1 \in T(a_1), \dots, A_n \in T(a_n), \\ A_1, \dots, A_n \vdash A \}$$

zuordnet. Die **von G definierte Sprache** ist $L(G) := S^G$.

Mit den Regeln $(\backslash R)$ und $(/ R)$ kann man zum Nachweis der Inklusion $C_1 \cdots C_n \subseteq C_0$ auch Hilfsannahmen verwenden. Die Regeln $(\backslash L)$ und $(/ L)$ entsprechen den Kürzungsregeln.

Interpretation der Kategorien

Sei $\mathcal{M} = (M, \cdot, 1)$ ein Monoid, d.h. eine Menge M mit einer Operation $\cdot : M \times M \rightarrow M$ und einem Element $1 \in M$, so daß

$$(x \cdot y) \cdot z = x \cdot (y \cdot z) \text{ und } x \cdot 1 = x = 1 \cdot x$$

für alle $x, y, z \in M$. Man denke an die Folgen über Σ und die Verkettung \cdot von Folgen, d.h. $(M, \cdot, 1) = (\Sigma^*, \cdot, \epsilon)$.

Den Basiskategorien C seien Teilmengen $C^{\mathcal{M}} \subseteq M$ zugeordnet. Dann interpretiert man die Produktkategorien $(A \cdot B)$ durch das elementweise Produkt der Mengen $A^{\mathcal{M}}$ und $B^{\mathcal{M}}$:

$$(A \cdot B)^{\mathcal{M}} := A^{\mathcal{M}} \cdot B^{\mathcal{M}} := \{u \cdot v \mid u \in A^{\mathcal{M}}, v \in B^{\mathcal{M}}\}$$

Für die Divisionskategorien geht es nicht entsprechend, weil auf M i.a. keine Divisionsoperation definiert ist. Stattdessen nimmt man

$$(A/B)^{\mathcal{M}} := \{u \in M \mid \{u\} \cdot B^{\mathcal{M}} \subseteq A^{\mathcal{M}}\},$$

$$(B \setminus A)^{\mathcal{M}} := \{u \in M \mid B^{\mathcal{M}} \cdot \{u\} \subseteq A^{\mathcal{M}}\}.$$

Eine Folge $\Gamma = C_1, \dots, C_n$ von Kategorien werde interpretiert durch

$$\Gamma^{\mathcal{M}} := (C_1 \cdot C_2 \cdots C_n)^{\mathcal{M}}.$$

Proposition: Die Axiome und Schlußregeln sind bei der Interpretation in Monoiden \mathcal{M} korrekt: Ist $\Gamma \vdash C$, so ist $\Gamma^{\mathcal{M}} \subseteq A^{\mathcal{M}}$.

Beweis: Für die Axiome $A \vdash A$ ist $A^{\mathcal{M}} \subseteq A^{\mathcal{M}}$ klar. Angenommen, der Beweis endet mit einer Anwendung von

$$\frac{\Theta \triangleright A \quad \Gamma, B, \Delta \triangleright C}{\Gamma, \Theta, A \setminus B, \Delta \triangleright C} (\setminus L)$$

Nach Induktion ist $\Theta^{\mathcal{M}} \subseteq A^{\mathcal{M}}$ und $\Gamma^{\mathcal{M}} \cdot B^{\mathcal{M}} \cdot \Delta^{\mathcal{M}} \subseteq C^{\mathcal{M}}$, also

$$\begin{aligned} \Gamma^{\mathcal{M}} \cdot \Theta^{\mathcal{M}} \cdot (A \setminus B)^{\mathcal{M}} \cdot \Delta^{\mathcal{M}} &\subseteq \Gamma^{\mathcal{M}} \cdot A^{\mathcal{M}} \cdot (A \setminus B)^{\mathcal{M}} \cdot \Delta^{\mathcal{M}} \\ &\subseteq \Gamma^{\mathcal{M}} \cdot B^{\mathcal{M}} \cdot \Delta^{\mathcal{M}} \subseteq C^{\mathcal{M}}. \end{aligned}$$

Analog für die anderen Regeln, mit der Definition von $(A/B)^{\mathcal{M}}$. \square

Beispiele für Herleitungen von Sequenzen

Im Lambek-Kalkül sind die Kürzungsregeln herleitbar, etwa (/):

$$\frac{\frac{A \triangleright A \quad B \triangleright B}{A/B, B \triangleright A} (/ L)}{A/B \cdot B \triangleright A} (\cdot L)$$

Aber auch andere Beziehungen wie $((A \setminus B)/C)^M \subseteq (A \setminus (B/C))^M$:

$$\frac{C \triangleright C \quad \frac{A \triangleright A \quad B \triangleright B}{A, A \setminus B \triangleright B}}{A, (A \setminus B)/C, C \triangleright B}}{A, (A \setminus B)/C \triangleright B/C}}{(A \setminus B)/C \triangleright A \setminus (B/C)}$$

Weitere Beispiele

Folgende Beziehungen gelten bei der Interpretation in \mathcal{M} :

1. $(A \setminus B) / C = A \setminus (B / C)$ (Argumentreihenfolge)
2. $A \subseteq (B / A) \setminus B, A \subseteq B / (A \setminus B)$ (Anhebung)
3. $A / (B \cdot C) = (A / C) / B$ (Currying)
4. $(A / B) \cdot (B / C) \subseteq A / C$ (Funktions-Komposition)
5. $A / B \subseteq (A / C) / (B / C), A / B \subseteq (C / A) \setminus (C / B)$ (Geach-Regeln)

Dazu beweise man die entsprechenden Sequenzen im Lambek-Kalkül, z.B. $A / B \triangleright (A / C) / (B / C)$ für 5.

Die wesentliche Beziehung zwischen Produkt und Divisionen ist:

$$C \subseteq B \setminus A \iff B \cdot C \subseteq A \iff B \subseteq A / C$$

Das besagt u.a.: A / C ist das größte B mit $B \cdot C \subseteq A$.

Eigenschaften des Lambek-Kalküls

Lemma (Lambek 1958)

- (i) Jede herleitbare Sequenz kann auch ohne Verwendung der Schnittregel hergeleitet werden.
- (ii) Es gibt ein Verfahren, mit dem man für jede Sequenz $\Gamma \triangleright A$ feststellen kann, ob sie herleitbar ist.

Proposition Die durch $G = (\Sigma, T, S)$ definierte Sprachfamilie $\{A^G \mid A \in \text{Cat}\}$ ist die Familie der kleinsten $\llbracket A \rrbracket \subseteq \Sigma^+$ mit

1. $A^T := \{a \in \Sigma \mid A \in T(a)\} \subseteq \llbracket A \rrbracket$, und
2. $\llbracket A_1 \rrbracket \cdots \llbracket A_n \rrbracket \subseteq \llbracket A \rrbracket$, falls $A_1, \dots, A_n \vdash A$.

Im Unterschied zu kontextfreien Grammatiken werden hier die Bedeutungen *unendlich vieler* Kategorien simultan definiert. Wie Chomsky schon den 60er Jahren vermutete, werden so aber keine zusätzlichen Sprachen definiert:

Satz (Pentus 1992) Zu jeder Lambek-Grammatik G gibt es eine kontextfreie Grammatik G' mit $L(G) = L(G')$.

Linguistische Anwendungen des Lambek-Kalküls

Typanhebung: durch $typ(C)$ werde einer syntaktischen Kategorie ein semantischer Typ zugeordnet, insbesondere

- ▶ $typ(S) := t$, die Wahrheitswerte,
- ▶ $typ(PN) := e$, die Individuen,
- ▶ $typ(N) = typ(VP) := (e \rightarrow t)$, Eigenschaften von Individuen.
- ▶ $typ(X \setminus Y) = typ(Y / X) := (typ(X) \rightarrow typ(Y))$, Funktionen.

$\frac{Maria}{PN \simeq e}$	$\frac{arbeitet}{VP \simeq (e \rightarrow t)}$	$\frac{Jeder}{Det \simeq ?}$	$\frac{Student}{N \simeq (e \rightarrow t)}$	$\frac{arbeitet}{VP \simeq (e \rightarrow t)}$
$S \simeq t$		$NP \simeq ?$		$S \simeq t$

Links ist nötig: $VP \subseteq PN \setminus S$, rechts $NP \subseteq S / VP$, $Det \subseteq NP / N$,
und damit

$$typ(NP) = typ(VP) \rightarrow typ(S) = (e \rightarrow t) \rightarrow t,$$

$$typ(Det) = typ(N) \rightarrow typ(NP) = (e \rightarrow t) \rightarrow ((e \rightarrow t) \rightarrow t)$$

Montague bringt beides auf den zweiten Fall, indem er die kontextfreie Regel $NP \rightarrow PN$ als Typanhebung interpretiert:

Um PN wie eine NP auf die VP anwenden zu können, brauchen wir

$$PN \subseteq S/VP,$$

und setzt man hierin $VP \subseteq PN \setminus S$ ein, so heißt das:

$$PN \subseteq S/(PN \setminus S).$$

Diese Typanhebung ist im Lambek-Kalkül beweisbar:

$$\frac{\frac{PN \triangleright PN \quad S \triangleright S}{PN, PN \setminus S \triangleright S}}{PN \triangleright S/(PN \setminus S)}.$$

Koordination von Konstituentenfolgen:

Dowty[1] benutzt die Typanhebung und Funktionskomposition,

$$\frac{X}{(Y/X)\backslash Y}(T) \quad \text{und} \quad \frac{X\backslash Y \quad Y\backslash Z}{X\backslash Z}(C)$$

(als Regeln geschrieben), zur Koordinierung von nicht-Konstituenten, z.B. von *NP*-Paaren. Sei $TV := VP/NP$, $DTV := TV/NP$ und $X := DTV\backslash VP$ in der Kategorie des Koordinators.

$$\frac{\frac{\frac{\frac{give}{DTV}}{DTV\backslash TV} T \quad \frac{\frac{\frac{Bill}{NP} \quad \frac{a \text{ book}}{NP}}{TV\backslash VP} T}{DTV\backslash VP} C}{DTV\backslash VP} C \quad \frac{and}{X\backslash X/X} \quad \frac{\frac{\frac{\frac{Max}{NP} \quad \frac{a \text{ record}}{NP}}{TV\backslash VP} T}{DTV\backslash VP} C}{DTV\backslash VP} C}{DTV\backslash VP} C}{VP}$$

Beispiel Die Lambek-Grammatik mit den Kategorisierungen

$der, ein : NP/N,$ $Buch, Student : N,$
 $der : (N \setminus N)/(S/NP)$ $liest : (NP \setminus S)/NP$

erlaubt eine Analyse von *der Student, der ein Buch liest* : NP. Man führt dazu

$NP/N, N, (N \setminus N)/(S/NP), NP/N, N, (NP \setminus S)/NP \triangleright NP$

mit der Schnittregel und (*) $NP \setminus (S/NP) \triangleright (NP \setminus S)/NP$, auf

$NP/N, N, (N \setminus N)/(S/NP), NP/N, N, NP \setminus (S/NP) \triangleright NP$

zurück; diese Sequenz kann allein mit (/ R) und (\ R), also in KG, hergeleitet werden, aber für (*) braucht man die Linksregeln!

Vorteile der Kategorialgrammatik:

- ▶ Linguistische Information im Lexikon $T : \Sigma \rightarrow \text{Cat}$ konzentriert
- ▶ Flexible Kategorisierung, z.B. bei Koordination, quasi kein Konstituentenbegriff mehr (ggf. Nachteil)
- ▶ Enger Zusammenhang zwischen syntaktischen Kategorien und semantischen Typen

Nachteile:

- ▶ Mit X/Y und $Y \setminus X$ wird nicht zwischen Komplement und Kontext unterschieden
- ▶ Tendiert zur Übergenerierung z.B. durch Typanhebungsregeln

Verfeinerungen durch strukturierte Kategorienannahmen Γ und Beschränkung der Regelanwendungen auf Teilannahmen möglich.



David Dowty.

Type raising, functional composition, and non-constituent conjunction.

In R.T.Oehrle et al. (eds.), editor, *Categorial Grammars and Natural Language Structures*, pages 153–197. D.Reidel, 1988.



J. Lambek.

The mathematics of sentence structure.

American Math. Monthly 65, 154–170, 1958.



J. Lambek.

On the calculus of syntactic types.

In R. Jacobson, editor, *Structure of Language and its Mathematical Aspects*, pages 166–178. Providence, 1961.

Montague-Grammatik (ohne Semantik)

R. Montague hat um 1970 eine Alternative zur Generativen Grammatik von N. Chomsky ausgearbeitet, die eine –durch eine Übersetzung der natürlichsprachlichen Ausdrücke in logische Formeln angegebene– modelltheoretische Semantik hat.

Die *syntaktischen Kategorien* sind:

A, B	$:=$	e	(Entitätenbezeichner)
		t	(Aussagen)
		A/B	(A -Ausdrücke, denen ein B -Ausdruck fehlt)
		$A//B$	(A -Ausdrücke, denen ein B -Ausdruck fehlt)

Jeder syntaktischen Kategorie wird ein Typ der Logiksprache L_{Typ} zugeordnet. Ausdrücke der Kategorien A/B und $A//B$ verhalten sich syntaktisch verschieden, aber semantisch gleich, und erhalten deshalb denselben Typ.

Die syntaktischen Kategorien der PTQ-Grammatik

Montague, *The Proper Treatment of Quantification in Ordinary English* [1] verwendet folgende Kategorien:

Kategorie	Informelle Bezeichnung	Basisausdruck
e	Eigennamen	John, he _n
t	Satz	–
CN := t//e	Gemeinname	man
IV := t/e	Verbalphrase	walks
T := t/IV	Nominalphrase (3.sg), Term	–
TV := IV/T	Transitives Verb	find, love
IAV := IV/IV	VP-Adverb	slowly
t/t	Satzadverb	necessarily
IAV/T	Präposition	in
IV/t	Verb mit Satzkomplement	believe
IV//IV	Verb mit Infinitivkomplement	try
DET := T/CN	Determinator	every, a, the

Die Logiksprache L_{Typ} der getypten λ -Terme

Die Menge Typ der einfachen Typen σ, τ ist:

$$\begin{array}{ll} \sigma, \tau & := e & \text{(Individuen)} \\ & | t & \text{(Wahrheitswerte)} \\ & | (\sigma \rightarrow \tau) & \text{(Funktionen)} \end{array}$$

Die Sprache L_{Typ} besteht aus den Termen t vom Typ τ , kurz: $t : \tau$, für jedes $\tau \in Typ$, die wie folgt aufgebaut sind:

1. Konstante $c : \tau$ und Variable x^τ sind Terme vom Typ τ .
2. Ist s ein Term vom Typ σ und t ein Term vom Typ $(\sigma \rightarrow \tau)$, so ist $t(s)$ (oder: $(t \cdot s)$) ein Term vom Typ τ .
3. Ist x^σ eine Variable und t ein Term vom Typ τ , so ist $\lambda x^\sigma t$ ein Term vom Typ $(\sigma \rightarrow \tau)$.

Terme vom Typ t heißen *Formeln* und werden mit φ, ψ bezeichnet, solche vom Typ e *Individuenterme*.

Wir nehmen an, daß es folgende Konstanten $c : \tau$ gibt:

$$\begin{array}{ll} \neg : t \rightarrow t & \\ \vee : t \rightarrow (t \rightarrow t) & \exists^\tau : (\tau \rightarrow t) \rightarrow t \\ \wedge : t \rightarrow (t \rightarrow t) & \forall^\tau : (\tau \rightarrow t) \rightarrow t \end{array}$$

Für diese Konstanten wird die übliche Schreibweise benutzt, z.B.

$$\begin{array}{ll} (\varphi \wedge \psi) & \text{statt } \wedge(\varphi)(\psi), \\ \forall x^\sigma \varphi & \text{statt } \forall^\sigma(\lambda x^\sigma \varphi), \end{array}$$

Logische Formeln sind also in den Termen von $L_{\mathcal{T}yp}$ enthalten.

Man möchte die Anwendung der durch $\lambda x^\sigma . r$ definierten Funktion auf das mir s bezeichnete Argument mit den Termen ausrechnen können. Daher sollte die Gleichung (sog. β -Reduktion)

$$(\lambda x^\sigma . r)(s) = r[x/s]$$

wobei $r[x/s]$ die Ersetzung von (frei vorkommenden) x in r durch s meint, bei jeder Interpretation

$$\mathcal{D} = (\langle D_\sigma \mid \sigma \in \text{Typ} \rangle, \langle c^\tau \mid c^\tau \in \text{Const} \rangle)$$

gelten; wobei $D_{(\sigma \rightarrow \tau)} \subseteq \{f \mid f : D_\sigma \rightarrow D_\tau\}$ und $c^\tau \in D_\tau$.

Die Interpretation von $t(s)$ und $\lambda x^\sigma . r$ ist so zu definieren, daß für jede Belegung g und Terme passender Typen gilt:

$$\llbracket (\lambda x^\sigma . r)(s) \rrbracket_g^{\mathcal{D}} = \llbracket r \rrbracket_{g[x/\llbracket s \rrbracket_g^{\mathcal{D}}]}^{\mathcal{D}} = \llbracket r[x/s] \rrbracket_g^{\mathcal{D}}.$$

Übersetzung der syntaktischen Kategorien

Syntaktischen Kategorien A werden Typen A' von L_{Typ} zugeordnet; Ausdrücke der Kategorie A werden später in λ -Terme vom Typ A' übersetzt.

Jeder syntaktischen Kategorie A wird durch

$$\begin{array}{ll} e' & := e & (A/B)' & := (B' \rightarrow A') \\ t' & := t & (A//B)' & := (B' \rightarrow A'). \end{array}$$

ein Typ A' von L_{Typ} zugeordnet.

Übersetzung der syntaktischen Kategorien

Ignorieren wir die Verben mit Satzkomplement, so erhalten wir folgende Typen und entsprechende Objekte:

Kategorie A	Typ A' von L_{Typ}	Objekt vom Typ A'
e	e	Individuum
t	t	Wahrheitswert
CN := t//e	$e \rightarrow t$	Individueneigenschaft
IV := t/e	$e \rightarrow t$	Individueneigenschaft
T := t/IV	$(e \rightarrow t) \rightarrow t$	Eigenschaft von Individueneigenschaften
TV := IV/T		
IAV := IV/IV	$(e \rightarrow t) \rightarrow (e \rightarrow t)$	Modifikator von Individueneigenschaften
t/t	$t \rightarrow t$	Eigenschaft von Wahrheitswerten
IAV/T		
DET := T/CN	$(e \rightarrow t) \rightarrow ((e \rightarrow t) \rightarrow t)$	

Basisausdrücke und ihre Übersetzung

Zuerst werden den atomaren Ausdrücken α der Kategorie A (evtl. zusammengesetzte) Terme α' des Typs A' zugeordnet, meist Konstante. Wir nehmen (für die extensionale Teilsprache) an:

Die syntaktischen Kategorien enthalten folgende **Basisausdrücke**:

1. Kategorie CN (Gemeinnamen): *man, woman, book*
2. Kategorie e (Individuennamen): *John, Mary, he_n ($n \in \mathbb{N}$),*
3. Kategorie IV (intransitive Verben): *walk, talk,*
4. Kategorie TV (transitive Verben): *find, see, read*
5. Kategorie DET: *the, a, some, every.*

Wir schreiben oft $w : A$ oder w^A statt „ w hat die Kategorie A “.

Bem: Eigennamen und Pronomen haben bei Montague die Kategorie T, hier die Kategorie e. Das macht die Übersetzung uniformer und wird durch die Syntaxregel (S 1) kompensiert, mit der sie in Ausdrücke der Kategorie T umgewandelt werden können.

Jedem Wort w der Kategorie A wird eine Konstante w' vom Typ A' zugeordnet, außer bei

- ▶ den Pronomina $he_n : e$, denen die Variablen $x_n : e$ zugeordnet werden, und
- ▶ den Quantoren, die wie folgt übersetzt werden:

$$(every^{DET})' := \lambda P^{e \rightarrow t} \lambda Q^{e \rightarrow t} \forall x^e (P(x) \rightarrow Q(x))$$

$$(a^{DET})' := \lambda P^{e \rightarrow t} \lambda Q^{e \rightarrow t} \exists x^e (P(x) \wedge Q(x))$$

$$(the^{DET})' := \lambda P^{e \rightarrow t} \lambda Q^{e \rightarrow t} \exists x^e ((\forall y^e (P(y) \leftrightarrow x \doteq y) \wedge Q(x)).$$

Die Übersetzung von α^{DET} hat den richtigen Typ
 $DET' = (CN' \rightarrow T') = (e \rightarrow t) \rightarrow ((e \rightarrow t) \rightarrow t)$.

Für Basisausdrücke anderer Kategorien erhalten wir z.B. folgende Übersetzungen:

$$(man^{CN})' := man' : CN' = man' : (e \rightarrow t),$$

$$(walk^{IV})' := walk' : IV' = walk' : (e \rightarrow t),$$

$$(John^e)' := John' : e.$$

$$(he_n^e)' := x_n : e.$$

Bem.: Die Übersetzung des bestimmten Artikels funktioniert nicht besonders gut, und der unbestimmte Artikel sollte nicht überall wie ein Existenzquantor behandelt werden.

Bildung und Übersetzung zusammengesetzter Ausdrücke

Die Bildung komplexer Ausdrücke geben wir durch numerierte Regeln der folgenden Form an:

$$(S \text{ Nr.}) \frac{\alpha_1 : A_1, \dots, \alpha_k : A_k}{\alpha : A}$$

Diese Regel besagt: ist α_1 ein Ausdruck der Kategorie A_1 usw., so ist der (aus $\alpha_1, \dots, \alpha_k$ gebildete) Ausdruck α von der Kategorie A .

Die Übersetzung eines zusammengesetzten Ausdrucks wird durch entsprechende Regeln

$$(T \text{ Nr.}) \frac{\alpha'_1 : A'_1, \dots, \alpha'_k : A'_k}{\alpha' : A'}$$

angegeben, die angeben, wie die Übersetzung von α aus den Übersetzungen der Teilausdrücke gebildet wird.

Bildung und Übersetzung komplexer Ausdrücke

$$(S\ 1) \frac{\alpha : e}{\alpha : T}$$

$$(T\ 1) \frac{\alpha' : e}{\lambda P^{e \rightarrow t}[P(\alpha')] : (e \rightarrow t) \rightarrow t}$$

$$(S\ 2) \frac{\delta : DET, \quad \xi : CN}{\delta \xi : T}$$

$$(T\ 2) \frac{\delta' : (CN' \rightarrow T'), \quad \xi' : CN'}{\delta'(\xi') : T'}$$

$$(S\ 3_n) \frac{\xi : CN, \quad \varphi : t}{\xi \text{ such that } \varphi[he_n/he] : CN}$$

$$(T\ 3_n) \frac{\xi' : CN', \quad \varphi' : t}{\lambda x_n^e(\xi'(x_n) \wedge \varphi') : CN'}$$

$$(S\ 4) \frac{\alpha : T, \quad \delta : IV}{\alpha \delta^{3.sg} : t}$$

$$(T\ 4) \frac{\alpha' : T', \quad \delta' : IV'}{\alpha'(\delta') : t}$$

$$(S\ 5) \frac{\delta : TV, \quad \beta : T}{\delta \beta^{acc} : IV}$$

$$(T\ 5) \frac{\delta' : (T' \rightarrow IV'), \quad \beta' : T'}{\delta'(\beta') : IV'}$$

$$(S \ 11a) \frac{\varphi : t, \ \psi : t}{\varphi \text{ and } \psi : t}$$

$$(S \ 11b) \frac{\varphi : t, \ \psi : t}{\varphi \text{ or } \psi : t}$$

$$(S \ 12a) \frac{\delta : IV, \ \gamma : IV}{\delta \text{ and } \gamma : IV}$$

$$(S \ 12b) \frac{\delta : IV, \ \gamma : IV}{\delta \text{ or } \gamma : IV}$$

$$(S \ 13) \frac{\alpha : T, \ \beta : T}{\alpha \text{ or } \beta : T}$$

$$(T \ 11a) \frac{\varphi' : t, \ \psi' : t}{(\varphi' \wedge \psi') : t}$$

$$(T \ 11b) \frac{\varphi' : t, \ \psi' : t}{(\varphi' \vee \psi') : t}$$

$$(T \ 12a) \frac{\delta' : e \rightarrow t, \ \gamma' : e \rightarrow t}{\lambda x^e (\delta'(x) \wedge \gamma'(x)) : e \rightarrow t}$$

$$(T \ 12b) \frac{\delta' : e \rightarrow t, \ \gamma' : e \rightarrow t}{\lambda x^e (\delta'(x) \vee \gamma'(x)) : e \rightarrow t}$$

$$(T \ 13) \frac{\alpha' : T', \ \beta' : T'}{\lambda P^{e \rightarrow t} (\alpha'(P) \vee \beta'(P)) : T'}$$

$$(S\ 14_n) \frac{\alpha : T, \quad \varphi : t}{\varphi[he_n/\alpha] : t}$$

$$(T\ 14_n) \frac{\alpha' : T', \quad \varphi' : t}{\alpha'(\lambda x_n^e.\varphi') : t}$$

Darin werden zwei Varianten der Ersetzung $\alpha[x/\beta]$ benutzt:

1. Die Ersetzung $\varphi[he_n/\alpha]$ in (S 14_n) ist so definiert: falls $\alpha \equiv he_k$ ist, werden alle he_n in φ durch he_k (im jeweils gleichen Kasus) ersetzt; andernfalls wird das erste Vorkommen von he_n durch α und die übrigen durch he (bzw. *she*, *it*) ersetzt, wobei der Kasus vom jeweiligen Vorkommen von he_n und das Genus vom ersten Basisausdruck der Kategorie CN oder T in α übernommen wird.
2. Die Ersetzung $\varphi[he_n/he]$ in (S 3) ist etwas anders: alle Vorkommen von he_n in φ sollen durch he bzw. *she*, *it* ersetzt werden, wobei das Genus sich nach dem Genus des ersten Basisausdrucks der Kategorie CN in ξ richtet.

Übersetzung von Personalpronomen und Eigennamen

Als Terme aufgefaßte Pronomen und Eigennamen werden mit (T 1) übersetzt:

$$\begin{aligned} ((he_n^e)^T)' &= \lambda P^{e \rightarrow t} [P((he_n^e)')] : (e \rightarrow t) \rightarrow t \\ &= \lambda P^{e \rightarrow t} [P(x_n^e)] : (e \rightarrow t) \rightarrow t. \end{aligned}$$

$$\begin{aligned} ((John^e)^T)' &= \lambda P^{e \rightarrow t} [P((John^e)')] : (e \rightarrow t) \rightarrow t \\ &= \lambda P^{e \rightarrow t} [P(John'^e)] : (e \rightarrow t) \rightarrow t. \end{aligned}$$

In einem Modell \mathcal{D} ist der Wert $\llbracket \lambda P[P(John')]' \rrbracket^{\mathcal{D}} \in D_{(e \rightarrow t) \rightarrow t}$ dieses Terms nicht das Individuum $\llbracket John' \rrbracket^{\mathcal{D}} \in D_e$, sondern *die charakteristische Funktion der Menge aller seiner Eigenschaften*.

Das wird so kompliziert gemacht, damit die Übersetzung nach demselben Schema verläuft wie bei koordinierten Termen wie *John or Mary* und quantifizierten Termen wie *every man* (s.u.).

Übersetzung atomarer Aussagen mit intransitivem Verb

Einfache Aussagen werden wie folgt übersetzt.

$$\begin{aligned}(((John^e)^T walk^{IV_s})^t)' &= {}_4 ((John^e)^T)'((walk^{IV})') : t \\ &= {}_1 \lambda P^{e \rightarrow t}[P(John'^e)](walk'^{e \rightarrow t}) : t\end{aligned}$$

Mit einer β -Reduktion kann der rechte Ausdruck vereinfacht werden zu

$$walk'^{e \rightarrow t}(John'^e) : t,$$

ohne daß sich sein Wert bei einer Belegung g über einer Struktur \mathcal{D} ändert.

Obwohl die Kategorie der Terme nur Nominalphrasen der dritten Person im Singular enthält, enthalten die Ausdrücke der Kategorie IV nicht die Vollformen, sondern die Infinitive (bzw. Stammformen) der darin vorkommenden Verben. Deshalb hat die Morphologie der Vollform des Verbs keinen Einfluß auf das Übersetzungsergebnis.

Aussagen mit koordinierten Eigennamen werden mit (T 4), (T 13) und den obigen Übersetzungen von Termen übersetzt wie bei

$$\begin{aligned}
 & (((John\ or\ Mary)^T\ walk^{IV_s})^t)' \\
 & =_4\ ((John^T\ or\ Mary^T)^T)'((walk^{IV})') \\
 & =_{13}\ \lambda P[(John^T)'(P) \vee (Mary^T)'(P)]((walk^{IV})') : t \\
 & =\ \lambda P[\lambda Q[Q(John'^e)](P) \vee \lambda Q[Q(Mary'^e)](P)](walk'^{e \rightarrow t}) : t.
 \end{aligned}$$

Durch Anwendungen von β -Reduktionen kann dieser Ausdruck weiter vereinfacht werden zu

$$\begin{aligned}
 & \lambda P[\lambda Q[Q(John'^e)](P) \vee \lambda Q[Q(Mary'^e)](P)](walk'^{e \rightarrow t}) : t \\
 & \rightarrow_{\beta}\ \lambda P[P(John') \vee P(Mary')](walk'^{e \rightarrow t}) : t \\
 & \rightarrow_{\beta}\ [walk'^{e \rightarrow t}(John'^e) \vee walk'^{e \rightarrow t}(Mary'^e)] : t,
 \end{aligned}$$

wieder ohne daß sich der Wert bei einer Belegung über einer Typstruktur \mathcal{D} ändert. Die entstandene Formel ist die, die man als Ergebnis erwartet haben sollte.

Die Regel (S 14) ersetzt ein Pseudopronomen he_n durch einen Term, z.B.

$$\frac{\frac{\text{every} : DET \quad \text{man} : CN}{\text{every man} : T} (S 2) \quad \frac{\frac{\vdots}{he_n \text{ walks} : t} \quad \frac{\vdots}{he_n \text{ talks} : t}}{he_n \text{ walks and } he_n \text{ talks} : t} (S 11)}{\text{every man walks and he talks} : t} (S 14)$$

Beachte, daß in den Syntaxregeln (S 3), (S 4) und (S 14) mehr geschieht als nur die Verkettung der Teilausdrücke; sie entsprechen also keinen kontextfreien Regeln.

Was heißt „abstrakte Syntax“ ?

In der Montague-Grammatik entspricht die Syntaxregel

$$(S\ 4) \frac{\alpha : T, \quad \delta : IV}{\alpha \delta^{3.sg} : t} = \frac{\alpha : T, \quad \delta : IV}{s_4(\alpha, \delta) : t}$$

der durch $s_4(\alpha, \delta) = \alpha \delta^{3.sg}$ definierten Funktion $s_4 : T \times IV \rightarrow t$.

Der Syntaxregel ist eine Übersetzungsregel zugeordnet,

$$(T\ 4) \frac{\alpha' : T', \quad \delta' : IV'}{\alpha' \cdot \delta' : t'} = \frac{\alpha' : T', \quad \delta' : IV'}{s'_4(\alpha', \delta') : t'}$$

die einer Bedeutungsfunktion $t_4 := s'_4 : T' \times IV' \rightarrow t'$ entspricht.

Also hat man in der Montague-Grammatik:

- ▶ Syntaxregel = Konstruktionsfunktion $s : A_1 \times \dots \times A_n \rightarrow A$
- ▶ Syntaktische Struktur = „Analysebaum“ = aus den Namen der Aufbaufunktionen s gebildeter „Konstruktions“-Term

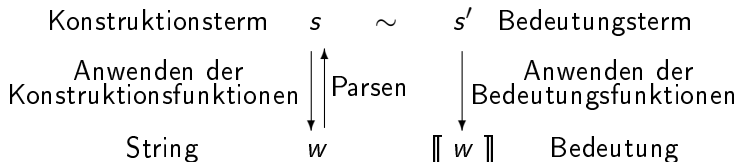
Zu jedem Konstruktor s gibt es eine Bedeutungsfunktion s' :



$$\frac{t_1 : A_1, \dots, t_n : A_n}{s(t_1, \dots, t_n) : A} (S \ k) \qquad \frac{t'_1 : A'_1, \dots, t'_n : A'_n}{s'(t'_1, \dots, t'_n) : A'} (T \ k)$$

Abstrakte Syntax:

- ▶ Kategorie = Typausdruck
- ▶ syntaktische Struktur = Konstruktionsterm,
- ▶ „Syntaxregel“ = getypter Funktionsname („Konstruktor“)

Die Bedeutung von Ausdrücken kann dann induktiv über den Aufbau der Konstruktionsterme definiert werden.



-  R.Thomason (ed.).
Formal Philosophy. Selected Papers of Richard Montague. 1976
-  David R.Dowty, Robert E.Wall, Stanley Peters.
Introduction to Montague Semantics. 1989
Springer, 1990.

Felderstruktur

P. Diderichsen (1943)[2] und J. Fourquet (1938)[3] haben die Struktur der einfachen deutschen Sätze untersucht und die Position des Verbs als „absolut“ und die der übrigen Teile als „relativ“ dazu betrachtet. (G.Bech 1955[1] untersuchte infinite Konstruktionen.)

In den europäischen Sprachen besteht das Prädikat im einfachen Satz allein aus einem finiten Vollverb oder aus einem finiten Hilfsverb und einem infiniten Vollverb.

Anders als im Englischen oder Französischen stehen Hilfs- und Vollverb im Deutschen nicht nebeneinander:

*Wir **sollten** diese Theorie **kennen**.*

*Maria **wird** das Buch nicht **lesen**, das Fritz ihr schenkte.*

*Wenn die Sonne aufgeht, **habe** ich schon **gegessen**.*

Dadurch zerfällt der einfache Satz in drei „Felder“:

Vorfeld** | **Prädikatsteil** | **Mittelfeld** | **r. Prädikatsteil** | **Nachfeld

Verberstsätze: Fragen, Befehle, Nebensätze ohne Konj.

Satzform		Felder			
Typ	Tempus	Vorfeld	(Mittelfeld)
<i>ve</i>	<i>präs</i> <i>prät</i>	–	$V_{\text{voll}}^{\text{präs}} \text{prät}$	Kompl.	$(V_{\text{voll}}^{\text{präf}})$
		–	$V_{\text{mod}}^{\text{präs}} \text{prät}$	Kompl.	$V_{\text{voll}}^{\text{inf}}$
	<i>perf</i> <i>plusq</i>	–	$V_{\text{aux}}^{\text{präs}} \text{prät}$	Kompl.	$V_{\text{voll}}^{\text{part2}} (V_{\text{aux}}^{\text{part2}})$
		–	$V_{\text{aux}}^{\text{präs}} \text{prät}$	Kompl.	$V_{\text{voll}}^{\text{inf}} V_{\text{mod}}^{\text{inf}}$
	<i>fut1</i>	–	$V_{\text{werden}}^{\text{präs}}$	Kompl.	$V_{\text{voll}}^{\text{inf}}$
		–	$V_{\text{werden}}^{\text{präs}}$	Kompl.	$V_{\text{voll}}^{\text{inf}} V_{\text{mod}}^{\text{inf}}$
	<i>fut2</i>	–	$V_{\text{werden}}^{\text{präs}}$	Kompl.	$V_{\text{voll}}^{\text{part2}} V_{\text{aux}}^{\text{inf}}$
		–	$V_{\text{werden}}^{\text{präs}}$	Kompl.	$V_{\text{aux}}^{\text{inf}} V_{\text{voll}}^{\text{inf}} V_{\text{mod}}^{\text{inf}}$

► Entscheidungsfragen

Vorfeld	(Mittelfeld)	Nachfeld
–	Gibt	er das Rauchen	auf?	
–	Will	Peter das Buch	lesen	, das ...?
–	Hatte	jeder das Buch	gelesen (gehabt)?	
–	Hat	sie das Buch	lesen wollen?	
–	Wird	jeder den Test	bestehen?	
–	Wird	er den Text	haben lesen müssen?	

► Befehle und Ausrufe

Vorfeld	(Mittelfeld)	Nachfeld
–	Hoffe	nicht darauf,	–	daß ...!
–	Möge	(es) uns	erspart bleiben,	daß ...!

► Nebensätze ohne einleitende Konjunktion

Vorfeld	(Mittelfeld)	Nachfeld
–	Stellt	Willi eine Frage		, dann

Verbzweitsätze: Aussagesätze

Satzform		Felder			
Typ	Tempus	Vorfeld	(Mittelfeld)
<i>vz</i>	<i>präs</i> <i>prät</i>	Kompl.	$V_{\text{voll}}^{\text{präs}} \text{prät}$	Kompl.	$(V_{\text{voll}}^{\text{präf}})$
		Kompl.	$V_{\text{mod}}^{\text{präs}} \text{prät}$	Kompl.	$V_{\text{voll}}^{\text{inf}}$
	<i>perf</i> <i>plusq</i>	Kompl.	$V_{\text{aux}}^{\text{präs}} \text{prät}$	Kompl.	$V_{\text{voll}}^{\text{part2}}$
		Kompl.	$V_{\text{aux}}^{\text{präs}} \text{prät}$	Kompl.	$V_{\text{voll}}^{\text{inf}} V_{\text{mod}}^{\text{inf}}$
	<i>fut1</i>	Kompl.	$V_{\text{werden}}^{\text{präs}}$	Kompl.	$V_{\text{voll}}^{\text{inf}}$
		Kompl.	$V_{\text{werden}}^{\text{präs}}$	Kompl.	$V_{\text{voll}}^{\text{inf}} V_{\text{mod}}^{\text{inf}}$
	<i>fut2</i>	Kompl.	$V_{\text{werden}}^{\text{präs}}$	Kompl.	$V_{\text{voll}}^{\text{part2}} V_{\text{aux}}^{\text{inf}}$
		Kompl.	$V_{\text{werden}}^{\text{präs}}$	Kompl.	$V_{\text{aux}}^{\text{inf}} V_{\text{voll}}^{\text{inf}} V_{\text{mod}}^{\text{inf}}$

► Aussagesätze:

Vorfeld	(Mittelfeld)	Nachfeld
Sie	wird muß	das Buch	lesen,	das ...
Niemand	hatte	das Buch	gelesen gehabt.	
Er	gab	das Studieren	auf.	

► Ergänzungsfragen:

Vorfeld	(Mittelfeld)	Nachfeld
Wer	hat	die Lösung	gefunden,	die ...?
Womit	wäre	ihnen	geholfen gewesen?	
Weshalb	gab	er das Studieren	auf?	

► Wunschsätze:

Vorfeld	(Mittelfeld)	Nachfeld
Uns	möge	es	erspart bleiben	, daß ...
Man	nehme	zwei Tabletten	ein.	

Verbletztsätze: Nebensätze mit Konj.

Satzform		Felder			
Typ	Tempus	Vorfeld	(Mittelfeld)
<i>vl</i>	<i>präs prät</i>	daß/ob	–	Kompl.	$V_{\text{voll}}^{\text{präs prät}}$
		daß/ob	–	Kompl.	$V_{\text{voll}}^{\text{inf}} V_{\text{mod}}^{\text{präs prät}}$
	<i>perf plusq</i>	daß/ob	–	Kompl.	$V_{\text{voll}}^{\text{part2}} V_{\text{aux}}^{\text{präs prät}}$
		daß/ob	–	Kompl.	$V_{\text{aux}}^{\text{präs prät}} V_{\text{voll}}^{\text{inf}} V_{\text{mod}}^{\text{inf}}$
	<i>fut1</i>	daß/ob	–	Kompl.	$V_{\text{voll}}^{\text{inf}} V_{\text{werden}}^{\text{präs}}$
		daß/ob	–	Kompl.	$V_{\text{voll}}^{\text{inf}} V_{\text{mod}}^{\text{inf}} V_{\text{werden}}^{\text{präs}}$
		daß/ob	–	Kompl.	$V_{\text{werden}}^{\text{präs}} V_{\text{voll}}^{\text{inf}} V_{\text{mod}}^{\text{inf}}$
	<i>fut2</i>	daß/ob	–	Kompl.	$V_{\text{voll}}^{\text{part2}} V_{\text{aux}}^{\text{inf}} V_{\text{werden}}^{\text{präs}}$
		daß/ob	–	Kompl.	$V_{\text{werden}}^{\text{präs}} V_{\text{voll}}^{\text{part2}} V_{\text{aux}}^{\text{inf}} V_{\text{mod}}^{\text{inf}}$

- ▶ Min Konjunktion eingeleitete Nebensätze [ggf.: Konj. in (]

Vorfeld	(Mittelfeld)	Nachfeld
wenn	–	er das Problem	lösen kann	, das ...
daß	–	niemand das Buch	gelesen hat	, das ...
ob	–	unsere Kraft dazu	ausreicht	, es zu tun

- ▶ Relativsätze

Vorfeld	(Mittelfeld)	Nachfeld
–	–	der das Problem	gelöst hatte	, das ...
–	–	mit dessen Mitteln es	getan wird	

- ▶ Mit eine Interrogativphrase eingeleitete Nebensätze

Vorfeld	(Mittelfeld)	Nachfeld
wessen Rat	–	man	folgen soll	
wie	–	du das Problem	lösen willst	, das ...
wer	–	das	behauptet hatte	

Wortstellung im Mittelfeld

Die Wortstellung im Mittelfeld ist relativ frei wählbar; es gibt aber auch hier einige Einschränkungen an die Ordnung < der Wörter:

- ▶ *Abfolgeregel für Pronomina:*

$proNP^{nom} < proNP^{akk} < proNP^{dat}$.

Beispiel: . . . , hat sie es ihm gegeben.

- ▶ *Abfolgeregel für Adverbiale:* $AdvP_{kausal} < AdvP_{modal}$.

Beispiel: Weil er wegen des Tadels sorgfältig schreibt, . . .

Für mit Konjunktionen eingeleitete Nebensätze, also Sätze mit Verbletzstellung, gelten anscheinend folgende Einschränkungen;

$proNP^{akk} < NP^{dat} < (NP^{akk} - proNP^{akk})$

$proNP^{dat}, proNP^{akk} < (NP^{nom} - proNP^{nom})$

a) Relative Wortstellung von Nominalargumenten:

<i>conj</i>	<i>NP^{nom}</i>	<i>NP^{dat}</i>	<i>NP^{akk} - pro</i>	<i>V^{finit}</i>
weil	die Frau	dem Kind	das Spielzeug	schenkt
	die Frau	dem Kind	* es	
	die Frau	ihm	das Spielzeug	
	die Frau	ihm	* es	
	sie	dem Kind	das Spielzeug	
	sie	dem Kind	* es	
	sie	ihm	das Spielzeug	
	sie	ihm	* es	
weil	sie / die Frau	sich / ihm / dem Kind	* sie / die Ruhe	gönnt

Literatur



G. Bech.

Studien über das deutsche Verbum infinitum. 2., unveränderte Auflage mit einem Vorwort von Catherine Fabricius-Hansen.
Niemeyer, Tübingen, 1983.



P. Diderichsen.

Ganzheit und Struktur. Ausgewählte sprachwissenschaftliche Abhandlungen.
Wilhelm Fink, München, 1976.



J. Fourquet.

L'ordre des éléments de la phrase en germanique ancien.
Les belles lettres, Paris, 1938.

Erzeugung von Grammatikregeln

Statt viele einzelne Grammatikregeln von Hand zu schreiben, kann man sie, soweit sie sich einem Schema fügen, automatisch *erzeugen*.

Dazu definieren wir ein Prädikat (+Cat \dashrightarrow -Cats) durch

⟨Metaregeln zur Konstruktion von Grammatikregeln⟩ ≡
(Kategorie \dashrightarrow Konstituenten) :- ...

wo in ... gesagt wird, wie und welche Konstituentenkategorien erzeugt werden sollen. Dieses Prädikat wird wegkompiliert:

⟨Behandlung von Metaregeln⟩ ≡
:- op(1100,xfx,' \dashrightarrow '). % neues Infix-Relationssymbol
ersetze_metaregeln :- tell('satzregeln.pl'),
 (A \dashrightarrow B), % Instanz von (A \dashrightarrow B) beweisen
 portray_clause((A \dashrightarrow B)), % und als DCG-Regel
 nl, fail. % in 'satzregeln.pl' schreiben
ersetze_metaregeln :- told. % Datei schliessen.

Erzeugung von Grammatikregeln für einfache Sätze

1. Für jede Verbstellung, Diathese und jedes Satztempus sagen die Schemata zur Felderstruktur (s.o.), aus welchen (Hilfs-)Verbformen der linke und rechte Prädikatsteil bestehen.
2. Aus dem Aktiv-Komplementrahmen (im Lexikon) läßt sich der für das Passiv berechnen, z.B. [nom, akk] \mapsto [?von, nom].
3. Für jede Diathese lassen sich aus dem Komplementrahmen und dem Prädikatsteil V^{fin} Listen von Kategorien mit passend belegten Merkmalen erzeugen, und (erweitert um Kategorien für Adverbiale) in Vorfeld, Mittelfeld, Nachfeld aufteilen,
4. Durch Einfügen der Prädikatsteile zwischen die Felder ergibt sich eine Folge von Konstituentenkategorien.
5. Für die DCG-Regel ($S_{Art}^{Form} \rightarrow$ Konstituenten) sind ggf. weitere Merkmale festzulegen, z.B. Reflexiv- und Interrogativmerkmale.

Beispiel Aus Komplementrahmen von Verben sollen Schemata einfacher Sätze mit Stellungsvarianten erzeugt werden, aber mit gewissen Einschränkungen:

- ▶ Finitheitsmerkmale: nur 3. sg,
- ▶ Diathese: nur Aktiv,
- ▶ Modus nur Indikativ, Tempus beschränkt,
- ▶ keine Präfixverben, keine Modalverben,
- ▶ Komplementabkürzungen: nur nom, akk, refl (akk), daß, ob,
- ▶ expandierte Akürzungen u.U. nicht in der NP-Grammatik,
- ▶ keine Ergänzung um Adverbiale.

Die Erzeugung der Satz schemata aus den Komplementrahmen der Verben ist quasi eine Implementierung des „X-bar-Schemas“:

$$\bar{X} \rightarrow \dots (\text{Komplemente}) \dots X \dots (\text{Komplemente}) \dots$$

<Grammatik mit Metaregel> ≡

```
(s([Def],[Temp,Vst,aktiv]) --> Konstituenten) :-  
    ( Vst=ve, Def=qu ; (Vst=vz;Vst=vl), Def=def ),  
    member(Vst,[vz,vl]), % Auswahl für Teilgrammatik!  
    member(Rahmen,[nom,refl(akk)],[nom,daß]),  
    praedikatsteile(aktiv,Vst,  
                    VFIN,VINF, % = Praedikatsteile  
                    Rahmen,Pers,Num,Temp,Mod),  
    expandiereSubjCat([Pers,Num],Rahmen,SubjRahmen),  
    anordnung(SubjRahmen,SubjRahmenPerm),  
    nachfeld(SubjRahmenPerm,SubjRahmenPerm2,Nachfeld),  
    verbstellung(SubjRahmenPerm2,Vst,RahmenPraed),  
    append(RahmenPraed,Nachfeld,RahmenPraed2),  
    expandiere([Pers,Num],RahmenPraed2,VFIN,VINF,Cats),  
    listToTuple(Cats,Konstituenten).
```

Für Aktivsätze im Präsens, haben-Perfekt, Futur1, Indikativ, 3.Sg:

$\langle \text{praedikatsteile}/9, \text{praedikatsteile}/8 \rangle \equiv$

praedikatsteile(aktiv, Vst, VFIN, [VINFIN],
Rahmen, Pers, Num, Temp, ind) :-
praedikatsteile(VFIN, VINFIN,
Vst, Rahmen, Pers, Num, Temp, aktiv),
(Vst = ve ; Vst = vz ; Vst = vl).

praedikatsteile(v([Rahmen], [Pers, Num, praes]), [],
Vst, Rahmen, Pers, Num, praes, aktiv).

praedikatsteile([hat], v([Rahmen], [part2]),
Vst, Rahmen, 3, sg, perf, aktiv).

praedikatsteile([wird], v([Rahmen], [inf]),
Vst, Rahmen, 3, sg, fut1, aktiv).

Bem. i.A. sind die Prädikatsteile von der Verbstellung abhängig.

⟨Expansion der Abkürzungen zu Konstituentenkategorien⟩≡

expandiereSubjCat([Pers,Num],[Obj|Objs],[ObjE|Objs]) :-
expandiereCat(Obj,ObjE),

kongruenz_sfin(ObjE,[Pers,Num,nom]).

expandiereCat(nom,np([Pers,Gen,Num],nom)) :- !.

expandiereCat(daß,s(daß)) :- !.

expandiereCat(akk,np([Pers,Gen,Num],[akk])) :- !.

expandiereCat(daß,s([daß],[_,v1,aktiv])) :- !.

expandiereCat(ob,s([ob],[_,v1,aktiv])) :- !.

expandiereCat(X,X) :- !. % (Vereinfachung)

kongruenz_sfin(np([Pers,_Gen,Num],[Kas]),[Pers,Num,Kas]).

kongruenz_sfin(s([daß],[_Temp,v1,aktiv]),[3,sg,nom]).

⟨Besetzung der Satzklammern durch Verbteile⟩≡

anordnung([A,B],[A,B]). anordnung([A,B],[B,A]).

% daß-Sätze ins Nachfeld, ergänze daß-Korrelat:

nachfeld([daß|ArgsPraed],[?(kor(daß))|ArgsPraed],[daß]).

nachfeld([A|ArgsPraed],[A|ArgsPraed2],Nachfeld) :-

nachfeld(ArgsPraed,ArgsPraed2,Nachfeld).

nachfeld([],[],[]).

verbstellung(Args,ve,ArgsPraed) :-

append([vfin|Args],[vinf],ArgsPraed).

verbstellung([Arg|Args],vz,ArgsPraed) :-

append([Arg,vfin|Args],[vinf],ArgsPraed).

verbstellung(Args,vl,ArgsPraed) :-

append(Args,[vinf,vfin],ArgsPraed).

$\langle \text{Expandiere relativ zu Finitheitsmerkmalen und Verbteilen} \rangle \equiv$
 expandiere(Fin, [vfin|Args], Vfin, Vinf, [Vfin|Kats]) :-
 !, expandiere(Fin, Args, Vfin, Vinf, Kats).
 expandiere(Fin, [vinf|Args], Vfin, Vinf, VinfKats) :-
 !, expandiere(Fin, Args, Vfin, Vinf, Kats),
 append(Vinf, Kats, VinfKats).
 expandiere(Fin, [Arg|Args], Vfin, Vinf, [Kat|Kats]) :-
 !, expandiereCat(Fin, Arg, Kat),
 expandiere(Fin, Args, Vfin, Vinf, Kats).
 expandiere(Fin, [], Vfin, Vinf, []).

 expandiereCat([Pers, Num], refl(akk),
 pron([refl], [Pers, _Gen, Num, akk])) :- !.
 expandiereCat([Pers, Num], X, C) :- expandiereCat(X, C).

Für daß- und ob-Sätze erlauben wir dieselben Konstituenten wie in definiten Verbletztsätzen, ergänzt um die einleitende Konjunktion:

$\langle \text{Untergeordnete Sätze mit einleitender Konjunktion} \rangle \equiv$

$(s([\text{Def}], [\text{Temp}, \text{vl}, \text{aktiv}]) \text{ -->}$

$[\text{Conj}], \text{Konstituenten}) :-$

$(s([\text{def}], [\text{Temp}, \text{vl}, \text{aktiv}]) \text{ --> Konstituenten}),$

$((\text{Def}, \text{Conj}) = (\text{daß}, \text{dass}) ; (\text{Def}, \text{Conj}) = (\text{ob}, \text{ob})).$

Folgende expliziten Regeln sollen auch in die Datei `satzregeln.pl` geschrieben werden, deshalb stehen sie hier als Metaregeln:

$\langle \text{Lexikalische Regeln} \rangle \equiv$

$(\text{kor}(\text{daß}) \text{ --> } [\text{es}]).$

$(\text{pron}([\text{refl}], [3, \text{sg}, \text{akk}]) \text{ --> } [\text{sich}]).$

Benutzung: (siehe Dateien `metaregeln.pl`, `np.pl` der Kursseite)

?- `[metaregeln]`.

?- `ersetze_metaregeln, [np]`.

Das zweite erzeugt `satzregeln.pl` und lädt es mit `np.pl`.

Einige der erzeugten Regeln

⟨Regeln mit reflexivem Objekt⟩≡

```
s([qu],[praes, ve, aktiv]) -->
    v([[nom, refl(akk)]], [A, B, praes]),
    np([A, _, B], [nom]),
    pron([refl], [A, _, B, akk]),
    [].
```

% Beispiel: ärgert ihr euch

```
s([qu],[praes, ve, aktiv]) -->
    v([[nom, refl(akk)]], [A, B, praes]),
    pron([refl], [A, _, B, akk]),
    np([A, _, B], [nom]),
    [].
```

% Beispiel: ärgert sich Herr Meier

Problem: Einschränkung pron < pron(refl) nötig

Regeln im Perfekt und Futur-1 ≡

```
s([qu],[perf, ve, aktiv]) -->
    [hat],
    pron([refl], [3, -, sg, akk]),
    np([3, -, sg], [nom]),
    v([[nom, refl(akk)]], [part2]).
% Beispiel: hat sich Frau Meier geärgert
```

```
s([def],[fut1, vl, aktiv]) -->
    np([3, -, sg], [nom]),
    pron([refl], [3, -, sg, akk]),
    v([[nom, refl(akk)]], [inf]),
    [wird].
% Beispiel: (weil) Herr Meier sich wundern wird
```

<Regeln mit daß-Objektsätzen>≡

```
s([qu], [praes, ve, aktiv]) -->
    v([[nom, daß]], [A, B, praes]),
    np([A, _, B], [nom]),
    kor(daß),
    [],
    s([daß], _).
% Beispiel: wissen die Leute es, daß ...
```

```
s([qu], [praes, ve, aktiv]) -->
    v([[nom, daß]], [A, B, praes]),
    kor(daß),
    np([A, _, B], [nom]),
    [],
    s(daß).
% Beispiel: wissen es die Leute, daß ...
```

⟨Regeln mit Objektsätzen im Perfekt und Futur⟩≡

```
s([qu], [perf, ve, aktiv]) -->  
  [hat],  
  kor(daß),  
  np([3, -, sg], [nom]),  
  v([[nom, daß]], [part2]),  
  s([daß], _).
```

% Beispiel: hat es Herr Meier gewußt, daß ...

```
s([qu], [fut1, ve, aktiv]) -->  
  [wird],  
  np([3, -, sg], [nom]),  
  kor(daß),  
  v([[nom, daß]], [inf]),  
  s(daß).
```

% Beispiel: wird Herr Meier es wissen, daß ...

Beispiele

Das Lexikon `lexikon_v.pl` sei erweitert um:

$\langle \text{Lexikalische Regeln} \rangle_{+ \equiv}$

`v([[nom,daß]],[3,sg,praes]) --> [glaubt].`

`v([[nom,refl(akk)]]],[3,sg,praes]) --> [aergert].`

$\langle \text{Analysen mit den erzeugten Satzregeln} \rangle_{\equiv}$

?- parse. Maria aergert sich.

Baum:

`s([def], [praes, vz, aktiv])`

`np([3, fem, sg], [nom])`

`pn([fem], [nom]) 'Maria'`

`v([[nom, refl(akk)]]), [3, sg, praes]) aergert`

`pron([refl], [3, sg, akk]) sich`

⟨*Beispiel mit Objektsatz*⟩≡

parse. Maria glaubt es dass der Hund Maria beisst.

Baum:

```
s([def], [praes, vz, aktiv])
  np([3, fem, sg], [nom])
    pn([fem], [nom]) 'Maria'
  v([[nom, daß]], [3, sg, praes]) glaubt
  kor(daß) es
  s([daß], [praes, vl, aktiv])
    dass
    np([3, mask, sg], [nom])
      det([3, sg, sw], [mask, nom])
        art([sw, sg], [mask, nom]) der
      n1([mask], [sw, sg, nom])
        n([], [mask], [sg, nom])
          cn([mask], [sg, nom]) 'Hund'
    np([3, fem, sg], [akk])
      pn([fem], [akk]) 'Maria'
  v([[nom, akk]], [3, sg, praes]) beisst
```

Grammatical Framework

- ▶ Programmiersprache zur Grammatikentwicklung
 1. endliche Datentypen für Parameterbereiche
 2. funktionaler Programmierstil
 3. Funktionsdefinition durch Argumentmuster
 4. reguläre Ausdrücke in der Morphologie
 5. Typprüfung
 6. Module und Funktoren zum Aufbau großer Grammatiken
- ▶ Grammatikformat
 1. Trennung von abstrakter und konkreter Syntax
 2. abstrakte Konstruktionen als getypte Funktionsdeklarationen
 $f : A_1 \times \dots \times A_n \rightarrow A_0$, Kategorien A_i als abstrakte Typen
 3. konkrete Konstruktionen als Funktionen $f^\circ : A_1^\circ \times \dots \times A_n^\circ \rightarrow A_0^\circ$
mit definierten Typen A_i° als konkreten Kategorien
 4. äquivalent zu „multiple context-free grammar“ (MCFG)
 5. Behandlung unzusammenhängender Konstituenten
 6. Mehrsprachigkeit: Übersetzung via gleicher abstrakter Syntax

Beispiel

Man kann in GF Sprachen definieren, die nicht kontextfrei sind, z.B. die Sprache $S = \{a^n b^m c^n d^m \mid n, m \geq 1\}$ über $\Sigma = \{a, b, c, d\}$.

Die abstrakte Syntax deklariert neben S zwei Kategorien A, B , Konstruktionen zum Aufbau von A - und B -Ausdrücken, und deren Verbindung zu S -Ausdrücken:

```
 $\langle NonCF.gf \rangle \equiv$   
  abstract NonCF = {  
  
    cat S ; A ; B ;  
  
    fun a0 : A ;   addA : A -> A ;   -- (an,cn)'s  
      b0 : B ;   addB : B -> B ;   -- (bm,dm)'s  
      glue : A -> B -> S ;         -- an bm cn dm 's  
  }
```


Die konkrete Grammatik implementiert die Typen A, B, S und die Konstruktion der entsprechenden Ausdrücke als Paare von oder einfache Zeichenreihen:

$\langle \text{NonCFConcr.gf} \rangle \equiv$

concrete NonCFConcr of NonCF = {

lincat S = {s : Str} ;

A = {s1 : Str; s2 : Str}; B = {s1 : Str; s2 : Str};

lin a0 = {s1= "a"; s2= "c"} ; b0 = {s1= "b"; s2= "d"} ;

addA a = {s1 = a.s1 ++ "a" ; s2 = a.s2 ++ "c"} ;

addB b = {s1 = b.s1 ++ "b" ; s2 = b.s2 ++ "d"} ;

glue a b = {s = a.s1 ++ b.s1 ++ a.s2 ++ b.s2} ;

} ;

$M.s$ holt das s -Feld aus einem Verbund M , $++$ verkettet Token.

Nach dem Laden der konkreten Grammatik kann man aus der konkreten in die Abstrakte Syntax Parsen, umgekehrt Linearisieren:

```
<shell: gf>≡
```

```
> i NonCFConcr.gf
+ reading NonCF.gfo
- compiling NonCFConcr.gf... wrote file NonCFConcr.gfo
linking ... OK
Languages: NonCFConcr
```

```
NonCF> p "a a b b b c c d d d" -- Parsen
glue (addA a0) (addB (addB b0)) -- Analyse
```

```
NonCF> l glue (addA a0) (addB (addB b0)) -- Linearisiere
a a b b b c c d d d -- Konkrete Form
```

Man sieht: Teilausdrücke müssen keine Teilzeichenreihen sein!

Abstrakte versus konkrete Grammatik in GF

- ▶ Abstrakte Syntax
 - ▶ kann und soll sprachunabhängig sein,
 - ▶ benennt Wort- und Ausdruckskategorien, aber läßt Morphologie, Kongruenz, Formdimensionen und -Werte frei
 - ▶ benennt die grammatischen Konstruktionen, aber läßt Wortstellung, Wortwahl u.a. frei
- ▶ Konkrete Syntax
 - ▶ ist spezifisch für eine Einzelsprache,
 - ▶ legt die Morphologie und das Lexikon der Sprache fest,
 - ▶ legt Details der grammatischen Konstruktionen der Sprache fest, z.B. Wortstellung, Kongruenz, Rektion usw.,
- ▶ Zwischenebene (partielle Konkretisierung)
 - ▶ ist spezifisch für eine Sprachfamilie, z.B. romanische Sprachen,
 - ▶ legt der Familie gemeinsame Details fest

Format der abstrakten Grammatiken

Eine abstrakte Grammatik besteht aus:

- ▶ dem Namen der Grammatik als Deklaration eines Moduls,
- ▶ der Definition von Flags (Compilerhinweisen),
- ▶ der Einführung von Kategorien als Typdeklarationen: $C : Type$,
- ▶ der Einführung abstrakter Grammatikregeln als Deklaration getypter Funktionen $f : C_1 \rightarrow C_2 \rightarrow C$ (Termkonstruktoren).

Die mit den Termkonstruktoren gebildeten (getypten) Terme $t : C$ sind die syntaktischen Analysen (Konstruktionsbeschreibungen) der konkreten Ausdrücke der Kategorie C .

Die Reihenfolge der Argumente der f hat nichts mit der Wortstellung der konkreten Syntax zu tun.

Erlaubte Typen der abstrakten Syntax

Grundtypen sind

- ▶ abstrakte Kategorien C (durch `cat C` eingeführt)
- ▶ vordefinierte Typen: `Int`, `Float`, `String`

Komplexe Typen sind

- ▶ Funktionstypen: $A \rightarrow B$
- ▶ (von Individuen x) abhängige Typen: $(x : A) \rightarrow B$

Beispiel: abstrakte Hello-Grammatik

Hier ist eine „abstrakte Grammatik“ die zwei Konstruktionen für Anredeformen und drei Adressaten vorsieht:

$\langle \text{Examples/Hello.gf} \rangle \equiv$

```
abstract Hello = {                                -- Moduldeklaration
  flags                                           -- Compilerhinweise
  startcat = Greeting ;
  cat                                             -- Kategorien
  Greeting ; Recipient ;
  fun                                           -- Konstruktoren
  World, Mum, Friends : Recipient ;
  Hello : Recipient -> Greeting ; -- höflich
  Hi : Recipient -> Greeting ;   -- familiär
}
```

Es ist nur *ein* Modul pro Datei erlaubt. Der Modulname Hello muß mit dem Dateinamen Hello.gf übereinstimmen!

Format der konkreten Grammatiken in GF

Konkrete Grammatiken „implementieren“ die abstrakten, indem sie

- ▶ die Kategorien C durch „Linearisierungs“-Typen“ C° definieren,
- ▶ die Konstruktoren $f : C_1 \rightarrow \dots \rightarrow C_n \rightarrow C$ durch „Linearisierungs“-Funktionen $f^\circ : C_1^\circ \rightarrow \dots C_n^\circ \rightarrow C^\circ$ definieren

Die Linearisierungstypen sind Verbundtypen `{lab : type; ... }`. Sie haben normalerweise ein Feld namens `s`, das

- ▶ vom Typ `Str` (String) ist und den konkreten Ausdruck enthält:
`{ s : Str ; p : Param ; ... }`
- ▶ von einem Tabellentyp ist und ein Paradigma enthält:
`{ s : Number => Case => Str ; p : Gender ; ... }`

Konkrete Grammatiken werden oft mit Hilfsmodulen für Morphologie, Lexikon usw. implementiert.

Erlaubte Linearisierungstypen C' in $\text{lincat } C = C'$

Grundtypen:

- ▶ Tokenliste: Str
- ▶ Parameterbereich: P , falls $\text{param } P = \dots$ (endlich)
- ▶ Zahlabschnitt: $\text{Ints } n$, für $\{0, 1, \dots, n-1\}$

Komplexe Typen:

- ▶ Tabellentyp: $P \Rightarrow T$, mit Parametertyp P (endliche Funktion)
- ▶ Verbundtyp: $\{\text{label}_1 : T_1 ; \dots ; \text{label}_k : T_k \}$
- ▶ Tupeltyp: $T_1 * \dots * T_k$, statt $\{p_1 : T_1 ; \dots ; p_k : T_k\}$

Dazu: Hilfstypen nur für Hilfs„operationen“ ($\text{oper } f : T$)

- ▶ Tokentyp: Tok
- ▶ Funktionstyp: $A \rightarrow B$
- ▶ Parametertyp: $P\text{Type}$
- ▶ beliebiger Typ: Type

Beispiel: konkrete Hello-Grammatik (Deutsch)

$\langle \text{Examples/HelloGer.gf} \rangle \equiv$

```
concrete HelloGer of Hello = {
  lincat          -- Implementierung der cat's:
    Greeting = {s : Str} ; -- Verbundtyp mit Feld
    Recipient = {s : Str} ; -- s vom Typ String
  lin            -- Implementierung der fun's
    World      = {s = "Welt"} ;
    Mum        = {s = "Mutti"} ;
    Friends    = {s = "Freunde"} ;

    Hello rec  = {s = "Hallo, " ++ rec.s ++ "!"} ;
    Hi rec     = {s = "Na, " ++ rec.s ++ "?" } ;
}
```

Durch `lin f x = ... x ...` ; wird `f` implementiert, passend zur Deklaration `f : T` der abstrakten Grammatik.

Beispiel: konkrete Hello-Grammatik (Russisch)

$\langle \text{Examples/HelloRus.gf} \rangle \equiv$

```
concrete HelloRus of Hello = {
  flags
    coding=utf8 ;
  lincat
    Greeting, Recipient = {s : Str} ;
  lin
    Mum      = {s = ""} ;           -- Mama
    Friends  = {s = ""} ;           -- Druschba
    World    = {s = ""} ;           -- Mir
    Hello r  = {s = "" ++ r.s ++ "!" } ; -- Sdrasdvuitje
    Hi rec   = {s = "" ++ rec.s ++ "!" } ; -- Privet r !
}
```

Problem: LaTeX kennt trotz `\usepackage[T2A]{fontenc}` und `\usepackage[utf8x]{inputenc}` kyrillische Buchstaben nicht

Modularten von GF

Es gibt mehrere Arten von Modulen:

- ▶ abstrakte Module: `abstract AM = ... ;`
- ▶ konkrete Module: `concrete CM of AM = ... ;`
- ▶ Hilfsmodule: `resource R = ... ;`
- ▶ Schnittstellen und partielle Instanzen davon

In den Hilfsmodulen werden Hilfsdatentypen und Operationen deklariert und definiert, nach folgendem Format:

<Schema eines Hilfsmoduls> \equiv

```
resource R = {  
  flags f1 = v1 ; f2 = v2 ;  
  oper E : Type = T ;  
  oper h : C -> E -> E = \c,e -> t  
  param P = C0 | C1 P1 | C2 P1 P2  
}
```

Öffnen von Modulen

Hilfsmodule können ihre Typen und Operationen an mehrere andere (Hilfs- oder konkrete) Module vererben:

⟨Vererbung von R and C und C'⟩≡

concrete C of A = open R in { ... } ;

concrete C' of A' = open R in { ... } ;

Linguistisches Beispiel: miniNP

```
<Examples/miniNP.gf>≡  
  abstract miniNP = {  
    flags  
      startcat = NP ;  
  cat  
    Det ; CN ; NP ;  
  fun  
    DetCN : Det -> CN -> NP ;  
    Tree : CN ;  
    the, these : Det ;  
  }
```

Um eine konkreten Grammatik dazu anzugeben, schreibt man zuerst einen Hilfsmodul für die Nomendeklination:

Hilfsmodul für Nomendeklination im Deutschen

$\langle \text{Examples/NDeclGer.gf} \rangle \equiv$

```
resource NDeclGer = {
  param          -- Merkmalbereiche und ihre Werte
    Number = Sg | Pl ;
    Case    = Nom | Gen | Dat | Acc ;
    Gender  = Masc | Fem | Neut ;

  oper          -- Formmerkmale ; Artmerkmale
    N : Type = {s : Number => Case => Str; g : Gender} ;

  regN : Str -> Str -> Gender -> N = \x,y,genus ->
    { s = table{
      Sg => table{ Gen => x+"s"+"es"; _ => x } ;
      Pl => table{ Dat => y+"n"+"en"; _ => y } }
    g = genus } ;
}
```

Wir betrachten den Numerus bei Det und NP als inhärentes Merkmal und lesen seinen Wert in der NP am Determinator ab:

<Examples/miniNPGer.gf>≡

```
concrete miniNPGer of miniNP = open NDeclGer in {  
  flags coding=utf8 ;
```

```
  lincat
```

```
    Det = { s : Gender => Case => Str ; n : Number } ;
```

```
    CN  = { s : Number => Case => Str ; g : Gender } ;
```

```
    NP  = { s : Case => Str ; g : Gender ; n : Number } ;
```

```
  lin
```

```
    DetCN det cn =
```

```
      { s = table{ c => det.s ! cn.g ! c ++  
                                     cn.s ! det.n ! c } ;
```

```
      g = cn.g ;
```

```
      n = det.n } ;
```

Dann werden die beiden Determinatoren verschieden:

$\langle \text{Examples/miniNPGer.gf} \rangle + \equiv$

```
the = { n = Sg ; s = table {
  Masc => table Case ["der"; "des"; "dem"; "den"] ;
  Fem  => table Case ["die"; "der"; "der"; "die"] ;
  Neut => table Case ["das"; "des"; "dem"; "das"] }
} ;
these = { n = Pl ; s = table {
  g => table Case ["die"; "der"; "den"; "die"] } } ;
Tree = regN "Baum" "Bäume" Masc ;
}
```

Bem: Da Determinatoren im Plural keine Formen nach dem Genus unterscheiden, könnte man alternativ nehmen:

$\langle \text{Weniger Formen für Det:} \rangle \equiv$

```
lincat Det = { s : DForm => Str }
param DForm = DSg Gender Case
           | DPl Case
```


Unzusammenhängende Konstituenten

Beispiel: Zusammenhängende vs. unzusammenhängende AP

- ▶ die Birke ist *kleiner als die Ulme*
- ▶ ein *kleinerer* Baum *als die Ulme*

In einer kontextfreien Grammatik ist das nur mühsam darzustellen.

Als GF-Grammatik: abstrakt eine AP für attributive und prädikative Verwendung, konkret aus zwei Zeichenreihen bestehend. In der abstrakten Grammatik braucht man:

1. Formen
 - ▶ deklinierte und undeklinierte A und AP
 - ▶ positive und komparierte A
2. AP aus Adjektiv mit Vergleichsobjekt
3. Nominalphrasen mit attributiver AP (dekliniert)
4. VP aus Kopula und prädikativer AP (undekliniert)

Modulare Grammatikerweiterung

Man kann eine Grammatik modular erweitern durch

- ▶ Erweitern der abstrakten Grammatik:

```
abstract GAE = GA ** {  
  cat C ; ... ;  
  fun f = ... }
```

- ▶ Erweitern der Hilfsmodule:

```
resource ResE = Res ** {  
  param = ... ;  
  oper T : Type = ... ;  
  oper f = ... }
```

- ▶ Erweitern der konkreten Grammatik:

```
concrete GAEC of GAE = GAC ** open ResE in {  
  lincat C = ... ;  
  lin f = ... }
```

Implementierung der APs in GF

Wir erweitern miniNPGer.gf.

$\langle \text{Examples/ANP.gf} \rangle \equiv$

```
abstract ANP = miniNP ** {  
  cat A ; AP ;  
  fun UseA : A -> AP ;           -- A | Aer  
    Compare : A -> NP -> AP ;    -- so A wie NP |  
                                -- Aer als NP  
  AttrPos : AP -> CN -> CN ;    -- so Ae CN wie NP  
  AttrCmp : AP -> CN -> CN ;    -- Aere CN als NP  
  small : A ;                    -- klein  
  Bush : CN ;                   -- Strauch  
  Tree1 : CN ;                  -- Ahorn  
}
```

Das könnte man dann um Konstruktionen für Sätze erweitern.

Die konkrete Grammatik ANPGer.gf muß Adjektive komparieren und deklinieren. Wir definieren den Typ A und die Adjektivformen:

(Examples/ADeclGer.gf)≡

```
resource ADeclGer = NDeclGer ** {
  param AForm = ASg Gender Case | AP1 | InDecl ;
        Degree = Pos | Cmp ;
  oper A : Type = { s : Degree => AForm => Str } ;
  mkA : Str -> A = \klein ->
    let mkTab : Str -> AForm => Str = \adj ->
      table{ InDecl          => adj ;
             ASg Masc Nom   => adj + "e" ;
             ASg (Fem|Neut) (Nom|Acc)
                 => adj + "e" ;
             -                => adj + "en"
            }
    in { s = table { Pos => mkTab klein ;
                   Cmp => mkTab (klein + "er") } } ;
}
```

<Examples/ANPGer.gf>≡

concrete ANPGer of ANP =

miniNPGer ** open ADeclGer in

{ flags coding=utf8 ;

lincat A = { s : Degree => AForm => Str } ;

AP = { s : Degree => AForm => Str ; -- Adjektiv

s2 : Degree => Case => Str } ; -- Vergl.glt

lin small = mkA "klein" ;

Bush = regN "Strauch" "Sträucher" Masc ;

Tree1 = regN "Ahorn" "Ahorne" Masc ;

UseA a = { s = a.s ; s2 = \\d,c => [] } ;

Compare a np =

{ s = table{ Pos => \\af => "so" ++ a.s!Pos!af ;

Cmp => \\af => a.s!Cmp!af };

s2 = table{ Pos => \\c => "wie" ++ np.s!c ;

Cmp => \\c => "als" ++ np.s!c }

} ;

$\langle \text{Examples/ANPGer.gf} \rangle_{+ \equiv}$

```
lin AttrPos ap cn =
  { s = \\n,c => ap.s!Pos!(case n of {
                                Sg => ASg cn.g c ;
                                Pl => AP1 })
    ++ cn.s!n!c ++ ap.s2!Pos!c ;

    g = cn.g
  } ;
AttrCmp ap cn =
  { s = \\n,c => ap.s!Cmp!(case n of {
                                Sg => ASg cn.g c ;
                                Pl => AP1 })
    ++ cn.s!n!c ++ ap.s2!Cmp!c ;

    g = cn.g
  } ;
}
```

NPs variieren in Numerus und Kasus; deren Werte muß man bei der AP in AFormen umrechnen.

Zum Beispiel erhält man damit:

Beispiele ≡

```
i Examples/ANPGer.gf
```

```
ANP> p "der kleine Baum"
```

```
DetCN the (AttrPos (UseA small) Tree)
```

```
ANP> p "den so kleinen Baum wie den Ahorn"
```

```
DetCN the (AttrPos (Compare small (DetCN the Tree1))  
           Tree)
```

```
ANP> p "der kleinere Baum als die Ahorne"
```

```
DetCN the (AttrCmp (Compare small (DetCN these Tree1))  
           Tree)
```

Die Grammatik erlaubt sinnvolle und unsinnige iterierte Attribute:

Beispiele ₊≡

```
ANP> p "die kleineren kleinen Bäume"
```

```
DetCN these (AttrCmp (UseA small)  
             (AttrPos (UseA small) Tree))
```

```
ANP> p "die kleineren so kleinen Bäume wie die Sträucher"
```

Beispiel: Franz Beckenbauer, SZ-Magazin 11.Juni 2010:

Was man wissen muß: Pelé ist ein mindestens ebenso großartiger Sambatänzer wie Fußballer gewesen.

(Eine andere Konstruktion, vom Typ A -> CN -> CN -> CN ???)

Um eine AP als Prädikat benutzen zu können, erweitern wir die Grammatik um VPs und einfache Sätze:

$\langle \text{Examples/S.gf} \rangle \equiv$

```
abstract S = ANP ** {
  flags startcat = S ;
  cat S ; VP ;
  fun PredPos : AP -> VP ;
    PredCmp : AP -> VP ;
    Pred : NP -> VP -> S ;
}
```


Um die VP zu bilden, braucht man von der AP die Komponenten im gewünschten Grad und die undeklinierte Form/den Nominativ:

$\langle \text{Examples/SGer.gf} \rangle \equiv$

```
concrete SGer of S = ANPger ** open ADeclGer in {
  lincat
    S = { s : Str } ;
    VP = { s : Number => Str } ;
  lin
    PredPos ap = { s = table{
      Sg => "ist" ++ ap.s!Pos!InDecl ++ ap.s2!Pos!Nom ;
      Pl => "sind" ++ ap.s!Pos!InDecl ++ ap.s2!Pos!Nom } };
    PredCmp ap = { s = table{
      Sg => "ist" ++ ap.s!Cmp!InDecl ++ ap.s2!Cmp!Nom ;
      Pl => "sind" ++ ap.s!Cmp!InDecl ++ ap.s2!Cmp!Nom } };
    Pred np vp = { s = np.s!Nom ++ vp.s!np.n } ;
  }
```

⟨Beispiel mit gf⟩≡

> linearize

Pred (DetCN the Bush)

(PredCmp (Compare small (DetCN the Tree)))

der Strauch ist kleiner als der Baum

> parse -cat=S

"die kleineren Bäume als der Ahorn sind klein"

Pred (DetCN these (AttrCmp (Compare small

(DetCN the Tree1))

Tree))

(PredPos (UseA small))

Die abstrakte Syntax erlaubt es also, Ausdrücke AP vorzusehen, die

- ▶ sowohl prädikativ wie attributiv gebraucht werden können,
- ▶ absolute (einstellige) Prädikate oder relative (zweistellige) Prädikate darstellen können,
- ▶ sowohl im Positiv als auch im Komparativ verwendbar sind.

Natürlich kann man den adverbialen Gebrauch und den Superlativ als Komparationsstufe ergänzen.

Die konkrete Syntax erlaubt es,

- ▶ die AP als zusammenhängende oder als unzusammenhängende Konstituente in andere Ausdrücke einzubauen,

indem man die beiden Felder AP.s und AP.s2, auf passende Parameter angewendet, zu einer Zeichenreihe verkettet oder nicht.

GF-Resource-Grammars: Abstrakte Grammatik (Module)

Modul „Grammar“:

```
abstract Grammar =
```

```
  Noun,
```

```
  Verb,
```

```
  Adjective,
```

```
  Adverb,
```

```
  Numeral,
```

```
  Sentence,
```

```
  Question,
```

```
  Relative,
```

```
  Conjunction,
```

```
  Phrase,
```

```
  Text,
```

```
  Structural,
```

```
  Idiom ;
```

Abstrakte Kategorien

Die Teilgrammatiken Adjective, Adverb, Noun, Verb etc. kommunizieren über den Modul Cat; eine Funktion in Verb, die NPs braucht, muß nur den Typ NP aus Cat kennen, nicht die NP-Konstruktionen aus Noun.

```
abstract Cat = Common ** {
  cat
  -- Sentences
  S ;      -- declarative sentence e.g. "she lived here"
  QS ;    -- question             e.g. "where did she live"
  RS ;    -- relative             e.g. "in which she lived"
  Cl ;    -- declarative clause, with all tenses
  ClSlash;-- clause missing NP (S/NP in GPSG)
  SSlash ;-- sentence missing NP e.g. "she has looked at"
  Imp ;   -- imperative           e.g. "look at this"
```

-- Interrogatives
QCl ; -- question clause, with all tenses
IP ; -- interrogative pronoun
IComp ; -- interrogative complement of copula
IDet ; -- interrogative determiner
IQuant; -- interrogative quantifier
-- Relative clauses and pronouns
RCl ; -- relative clause, with all tenses
RP ; -- relative pronoun
-- Verb phrases
VP ; -- verb phrase
Comp ; -- complement of copula, such as AP
VPSlash ; -- verb phrase missing complement
-- Adjectival phrases
AP ; -- adjectival phrase

-- Nouns and noun phrases
CN ; -- common noun (without determiner)
NP ; -- noun phrase (subject or object)
Pron ; -- personal pronoun
Det ; -- determiner phrase
Predet ; -- predeterminer (prefixed Quant)
Quant ; -- quantifier ('nucleus' of Det)
Art ; -- article
Num ; -- number determining element
Card ; -- cardinal number
Ord ; -- ordinal number (used in Det)
-- Numerals
Numeral ; -- cardinal or ordinal in words
Digits ; -- cardinal or ordinal in digits
-- Structural words
Conj ; -- conjunction
Subj ; -- subjunction
Prep ; -- preposition, or just case

```

-- Words of open classes
V ;      -- one-place verb      "sleep"
V2 ;     -- two-place verb      "love"
V3 ;     -- three-place verb    "show"
VV ;     -- verb-phrase-complement verb "want"
VS ;     -- sentence-complement verb "claim"
VQ ;     -- question-complement verb "wonder"
VA ;     -- adjective-complement verb "look"
V2V ;    -- verb with NP and V complement "cause"
V2S ;    -- verb with NP and S complement "tell"
V2Q ;    -- verb with NP and Q complement "ask"
V2A ;    -- verb with NP and AP complement "paint"
A ;      -- one-place adjective  "warm"
A2 ;     -- two-place adjective  "divisible"
N ;      -- common noun          "house"
N2 ;     -- relational noun      "son"
N3 ;     -- three-place relational noun "connection"
PN ;     -- proper name          "Paris"

```

```

}
```


Abstrakte NP-Grammatik

```
abstract Noun = Cat ** {           -- extend module Cat
  fun
    DetCN      : Det -> CN -> NP ;   -- the man
    UsePN      : PN -> NP ;          -- John
    UsePron    : Pron -> NP ;        -- he

    PredetNP   : Predet -> NP -> NP; -- only the man
    PPartNP    : NP -> V2  -> NP ;   -- the man seen
    AdvNP      : NP -> Adv -> NP ;   -- Paris today
    RelNP      : NP -> RS  -> NP ;   -- Paris, which is here
    DetNP      : Det -> NP ;         -- these five

    DetQuantOrd : Quant -> Num -> Ord -> Det ; -- these five
                                                    best
    DetQuant    : Quant -> Num          -> Det ; -- these five
```

```
NumSg      : Num ;
NumPl      : Num ;
...
IndefArt   : Art ;
DefArt     : Art ;

DetArtOrd  : Art -> Num -> Ord -> Det ; -- the (five)
                                                best
DetArtCard : Art -> Card -> Det ; -- the five
DetArtSg   : Art -> CN -> NP ; -- the man
DetArtPl   : Art -> CN -> NP ; -- the men

MassNP     : CN -> NP ; -- (beer)
PossPron   : Pron -> Quant ; -- my (house)
```


Konkrete Grammatik (German)

Eine konkrete Grammatik (zu einer abstrakten) gibt an:

1. für jede abstrakt deklarierte Kategorie A eine konkrete Implementierung durch einen (Record-)Typ A° ,
2. (endliche) Hilfstypen und Werte für synt.morph. Merkmale,
3. für jede abstrakte Konstruktion $f : A_1 \times \dots \times A_n \rightarrow A_0$ eine Implementierung $f^\circ : A_1^\circ \times \dots \times A_n^\circ \rightarrow A_0^\circ$,
4. morphologische Operationen zur Bildung von Wortparadigmen,
5. ein Lexikon

Die abstrakte Grammatik liefert (aus abstrakten Konstruktoren aufgebaute) *Terme* $t : A$ als syntaktische Strukturen, die durch *Linearisierung* in Objekte $t^\circ : A^\circ$ abgebildet werden, die meist eine ausdrückbare Komponente $t^\circ.s : \text{String}$ haben.

Konkrete Grammatik (German)

```
concrete CatGer of Cat =
  CommonX - [Tense,TPres,TPast,TFut,TCond] **
  open ResGer, Prelude in {

lincat
  -- Tensed/Untensed
  S  = {s : Order => Str} ;
  QS = {s : QForm => Str} ;
  RS = {s : GenNum => Str ; c : Case} ;
  SSlash = {s : Order => Str} ** {c2 : Preposition} ;
  -- Sentence
  Cl = {s : Mood => ResGer.Tense => Anteriority
        => Polarity => Order => Str} ;
  ...
  Imp = {s : Polarity => ImpForm => Str} ;
  -- Question, Relative
  ...
```

```

-- Noun
CN = {s : Adjf => Number => Case => Str ; g : Gender} ;
NP = {s : Case => Str ; a : Agr} ;
Pron = {s : NPForm => Str ; a : Agr} ;
Det = {s : Gender => Case => Str ; n : Number ; a : Adjf} ;
Quant = {s : Number => Gender => Case => Str ;
          a : Adjf} ;
Art = {s : Bool => Number => Gender => Case => Str ;
        a : Adjf} ;
Predet = {s : Number => Gender => Case => Str} ;
Num = {s : Gender => Case => Str ; n : Number ;
        isNum : Bool} ;
Card = {s : Gender => Case => Str ; n : Number} ;
Ord = {s : AForm => Str} ;

```

```

-- Preposition, Adjective
Prep = {s : Str ; c : Case} ;
AP = {s : AForm => Str ; isPre : Bool} ;

-- Open lexical classes, e.g. Lexicon
A = {s : Degree => AForm => Str} ;
A2 = {s : Degree => AForm => Str ; c2 : Preposition} ;

N = {s : Number => Case => Str ; g : Gender} ;
N2 = {s : Number => Case => Str ; g : Gender
      ** {c2 : Preposition} ;
N3 = {s : Number => Case => Str ; g : Gender
      ** {c2,c3 : Preposition} ;
PN = {s : Case => Str} ;

...
}

```

Konkrete Konstruktionen (Grammatikregeln)

Die Implementierung der z.B. durch

```
fun f : A -> B -> C
```

deklarierten Konstruktionen erfolgt durch Angabe einer Funktion f° in der Form

```
lin f a b = term
```

unter Verwendung von

- ▶ Funktionsanwendung $(\backslash x \rightarrow \text{value}) x' = \text{value}[x/x']$
- ▶ Feldselektion $\{\text{label}=\text{value}; \dots\}.\text{label}' = \text{value}'$
- ▶ Tabellenanwendung $\text{table}\{\text{lab} \Rightarrow \text{val}; \dots\}!\text{lab}' = \text{val}'$
- ▶ Tokenverkettung $[a,b,c]++[d,e] = [a,b,c,d,e]$
- ▶ Stringverkettung $"abc"+"de" = "abcde"$

NP-Konstruktionen (NounGer)

```
concrete NounGer of Noun =
  CatGer ** open ResGer, Prelude in {

lin
  DetCN det cn = {
    s = \\c => det.s ! cn.g ! c ++
              cn.s ! adjfCase det.a c ! det.n ! c ;
    a = agrP3 det.n ;
    isPron = False
  } ;

  DetNP det = {
    s = \\c => det.s ! Neutr ! c ; ---- genders
    a = agrP3 det.n ;
    isPron = False
  } ;
```

```
UsePN pn = pn ** {a = agrP3 Sg} ;
```

```
UsePron pron = {  
  s = \\c => pron.s ! NPCase c ;  
  a = pron.a  
} ;
```

```
DetQuant quant num =  
  let  
    n = num.n ;  
    a = quant.a  
  in {  
    s = \\g,c => quant.s ! n ! g ! c ++ num.s!g!c ;  
    n = n ;  
    a = a  
  } ;
```

```
...  
}
```

Beispiel: Mehrsprachige Grammatik

Wir laden drei Sprachen der „Resource-Library“ und übersetzen einen Satz aus dem Englischen ins Deutsche und Italienische: p dient zum Parsen der Eingabe, l zum Linearisieren der Analyse:

```
cd gf/lib/resource
```

```
gf
```

```
> i german/LangGer.gf
```

```
> i english/LangEng.gf
```

```
> i italian/LangIta.gf
```

```
...
```

```
Languages: LangGer LangEng LangIta
```

```
Lang> p -lang=LangEng
```

```
    "the new book is better than the old paper"
```

```
    | l -lang=LangGer,LangIta
```

```
das neue Buch ist besser als das alte Papier
```

```
il nuovo libro è migliore che la vecchia carta
```

Der Syntaxbaum ist ein mit den Konstruktoren gebildeter Term:

```
> p "the new book is better than the old paper"
```

```
PhrUtt NoPConj (UttS (UseCl TPres ASimul PPos (PredVP  
(DetArtSg DefArt (AdjCN (PositA new_A) (UseN book_N)))  
(UseComp (CompAP (ComparA good_A (DetArtSg DefArt (AdjCN  
(PositA old_A) (UseN paper_N)))))))))) NoVoc
```

Literatur:

A.Ranta: <http://www.cs.chalmers.se/~aarne/GF/>

Janna Khagai: Language Engineering in Grammatical Framework
VDM Verlag Dr. Müller, 2008