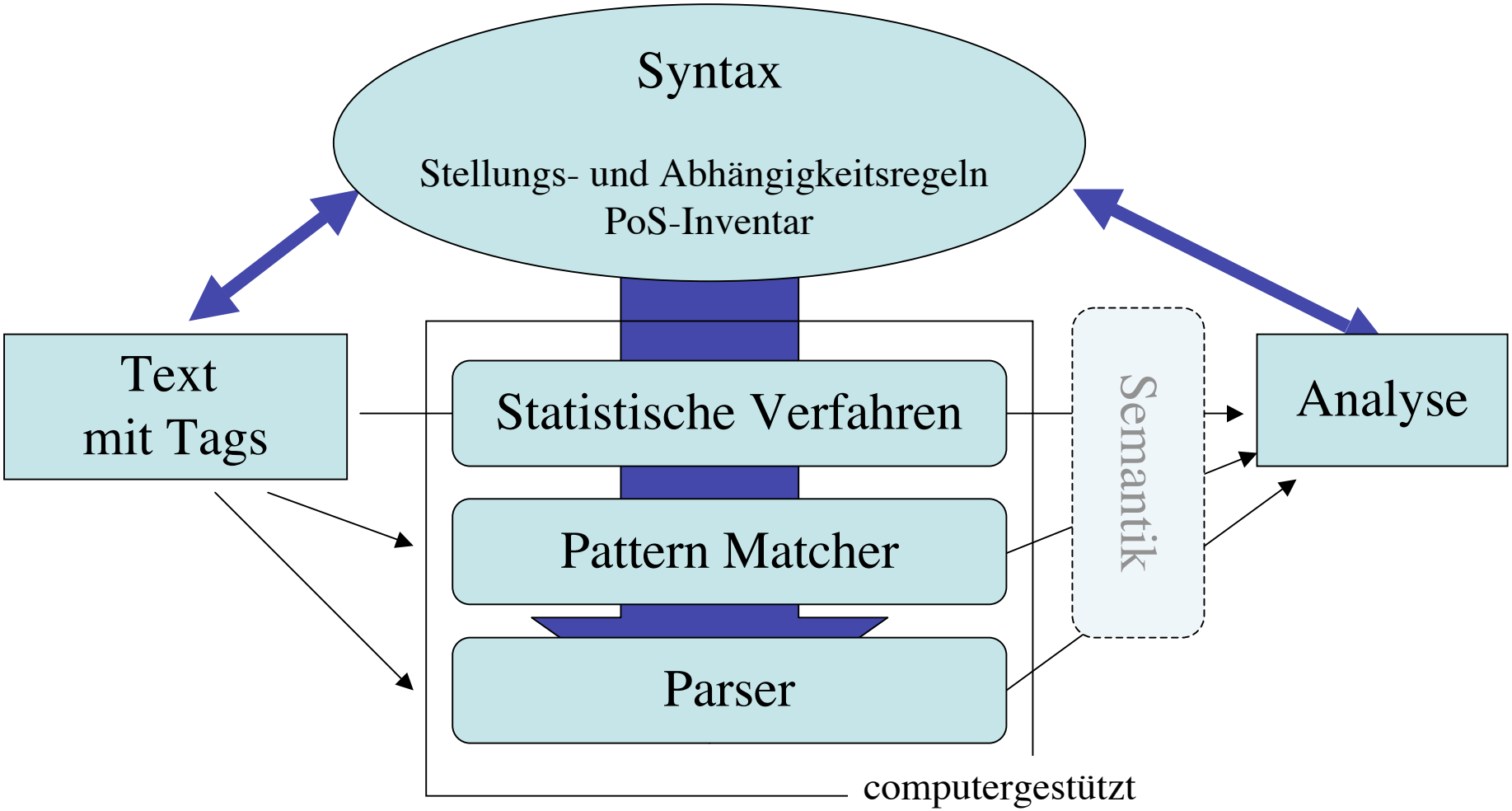
Sommersemester 2006 • Institut für Germanistik I

Vorlesung
Computerphilologie

Themenfeld
Syntaktische Repräsentation

„Wie kann man maschinell die syntaktische Struktur von Texten analysieren?“

Syntax und ihre Verwendung



Warum ist Syntax wichtig? - 1 -

- Syntaktische Strukturen sind Gegenstand literarischer wie textueller Untersuchungen (Syntaktische Eigenschaften, Historischer Stand, Syntaktische Komplexität, Lesbarkeits- und Verständlichkeitsforschung).
- Syntax definiert (mit) die Semantik natürlichsprachlicher Texte, d.h. alle semantischen Prozesse müssen syntaktisch gefiltert werden oder bleiben lexikalisch.
- Beispiele
  - das "Subjekt" könnte der Agent und das "Prädikat" die Aktion sein.
  - Eine negierte Aussage unterscheidet sich entweder gar nicht mehr von der positiven oder man weiß nicht, was negiert wird

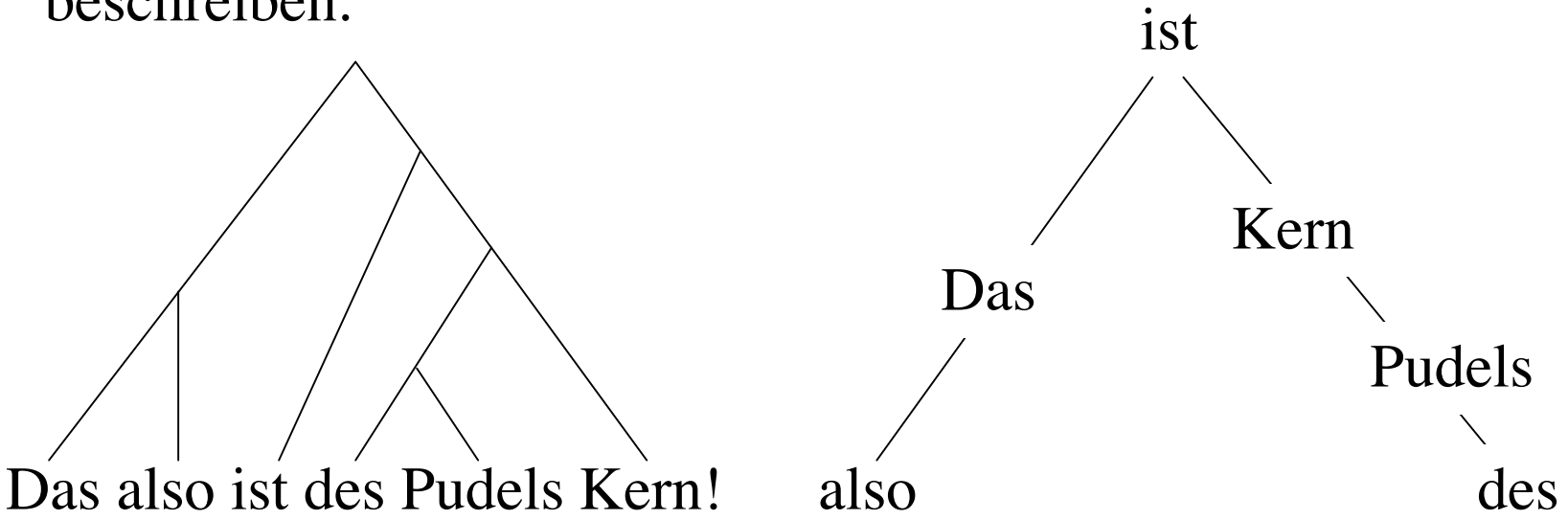
Warum ist Syntax wichtig? - 2 -

- Spezifiziert man eine spezielle Beschreibungssprache (z.B. mit XML), muß man einen Verarbeitungsprozeß formulieren, der immer mit dem Parsing beginnen wird (mindestens: Welcher Tag ist in welchem Skopus?)
- Für (partielles) syntaktisches Taggen kann man einen Parser entwerfen, der seinerseits eine Syntax liest

Wie syntaktisch annotieren?

Für eine syntaktische Annotation muß man mehr tun, als nur Wort-artentags verteilen: <w pos= 'dem'> Das </w> < w pos= 'adv'> also </w> ..... ,

sondern man muß zusätzlich die syntaktische Abhängigkeit beschreiben:



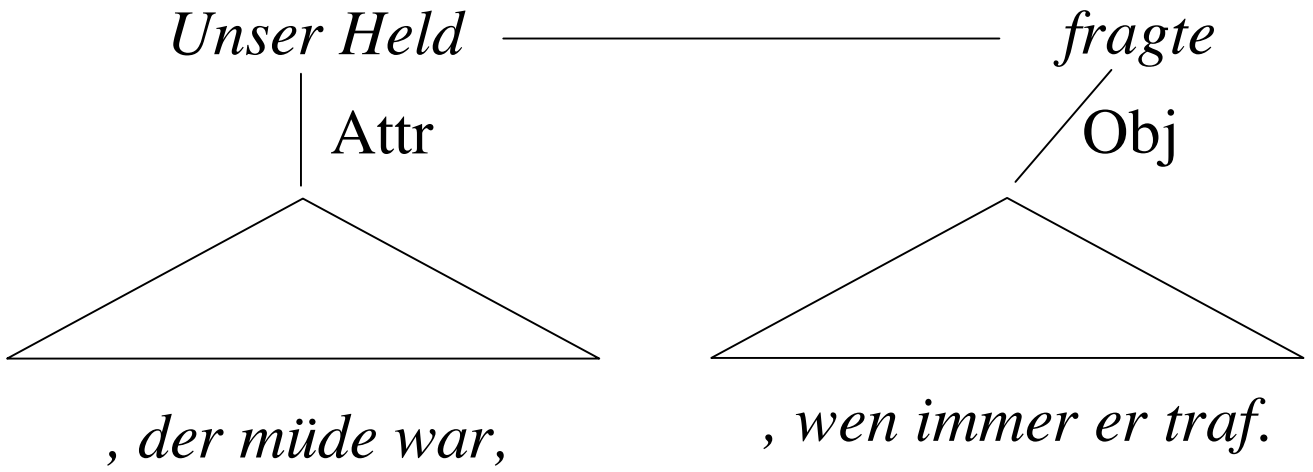
Syntaktische Annotation

Man kann sich bei syntaktischen Annotationen entscheiden zwischen

- intuitiven funktionalen Tags, wie <s func='Subjekt'> ... </s> <s func= PrepObj> ... </s> , für die man u.U. keine eindeutigen Annotationsregeln formulieren kann, oder
- man verwendet Konstituentenstruktur-/Dependenzstruktur-Tags, wie <s func= 'NP'>..., </s> , <s func= 'PP'> ... </s> , für die es Tests gibt.
- Allerdings verliert man bei den letzteren u.U. kommunikative Information.

- Darüber hinaus muß man die - prinzipiell gleichen - (Teil-)Satzrelationen beschreiben:

Funktion von Teilsätzen



Einige praktische Fragen

- Was macht man mit Satzzeichen?
  - Man sollte sie mitrepräsentieren, denn sie sind syntaktisch/semantisch bedeutungsvoll, auch wenn sie bei gesprochener Sprache nur der Intuition des Transkribenten entsprechen: <w sig='AusZ') !</w>
- Für Konjunktionen und andere Einleitungswörter muß man möglicherweise das Beziehungswort (die Beziehungs-struktur) ausdrücklich repräsentieren: <s dep='-1'> ...</s>
- Kann man eine partielle syntaktische Repräsentation einfügen?
  - Durchaus, zum Beispiel, indem man Detailstrukturen in Patterns (oder umgekehrt) einfügt.

Partielle Annotation

<m con='Anweisung'> Geht rasch ab, Gräfin tritt ein und grüßt den Herzog </m>

<w pos='pro'>Mein </w> <w pos='n'>Herr </w> </s>

Oder:

<m con='Anweisung'> <w pos=v> Geht </w> <w pos='adv'> rasch </w> <w pos='pre'> ab </w> <w sig='kom'>, </w> <w pos='n'> Gräfin </w> <w pos='v'> tritt </w> <w pos='pro'> ein ... </w> </m>

Oder:

<s Zeile='1'> Mein Herr, haben Sie Nachrichten ... </s>

Parsing

- Als Parsing bezeichnet man allgemein eine automatische formale Strukturanalyse (Beispiel: Prüfen der Konsistenz einer HTML-Annotierung, Prüfen der syntaktischen Korrektheit eines Programms)
- Speziell in der Computerlinguistik ist Parsing eine automatische formale Syntaxbeschreibung mit Ergebniszuweisung (Beispiel: Syntaxbaum zu einem Eingabesatz erzeugen). Die synt. Korrektheit der Eingabe wird dabei nicht notwendig gefordert (z.B. für Parser für gesprochene Sprache, die nur partielle Ergebnisse liefern oder Parser, die Ungrammatikalität erkennen sollen)

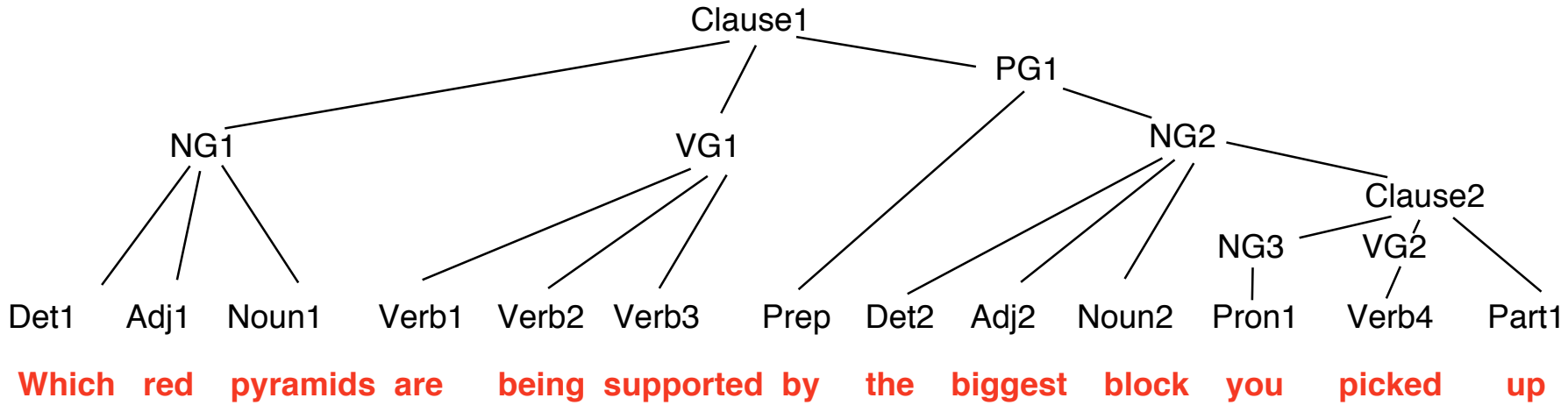
Wozu braucht man Parsing?

- Formal: Für die syntaktische Überprüfung von speziellen Tag-Sets
- Formal: Für die Auswertung von speziellen Tag-Sets
- Formal: Für automatisches Taggen
- Inhaltlich: Für die automatische Syntaxanalyse zur literarischen Interpretation
- Inhaltlich: Zur Steuerung von Segmentierern und Tokenizern

Was kann das Ergebnis einer syntaktischen Analyse sein?

- Syntaktische Strukturbeschreibung, oder
- Semantische Slot-Filler, oder
- Suchausdruck in einer Datenbank, oder
- Tabelle, oder
- Explizite logisch-semantische Repräsentation
- Oder ...

Ergebnis: Syntakt. Kategorien



Clause1 (*Which red pyramids are being supported by the biggest block you picked up?*) Clause Major Question Wh- Transitive Passive (Present in Present) Subject-Question Agent

NG1 (*Which red pyramids*) NG Determined Indefinite Question Plural Subject

- Det1 (*Which*) Determiner Question-Determiner Singular Plural
- Adj1 (*red*) Adjective
- Noun1 (*pyramids*) Noun Plural

VG1 (*are being supported*) VG Finite Passive (Present in Present)

- Verb1 (*are*) Verb Auxiliary Intensive Be Plural Present
- Verb2 (*being*) Verb Auxiliary Intensive Be Ing
  - Verb3 (*supported*) Verb Transitive Past-Participle

usw.

Aus: Winograd, T., Natural Langugae Understanding. Beispiel: SHRDLU

Ergebnis: Semantische Slot Filler

"I want to go to San Diego on May 28"

(Client Declare
(Case for *want* / e (Tense Present)
Agent = Dialog.Client.Person
Event = (Case for *go* (Tense Present)
Agent = Dialog.Client.Person
To-Place = (Case for City
Name = *San Diego*)
Date = (Case for Date
Month = *May*
Day = *28*))))

Beispiel: GUS
Nach Winograd

Ergebnis: Suchausdruck in einer Datenbank-Anfragesprache

"Liste die Namen aller Lieferanten, die mindestens alle die Teile liefern, die auch vom Lieferanten S2 geliefert werden"

```
SELECT          UNIQUE  S#
FROM        SP  SP X
WHERE       NOT EXISTS
            (SELECT *
            FROM SP SP Y
            WHERE S# = ´S2´
            AND         NOT EXISTS
                        (SELECT *
                        FROM SP
                        WHERE S# = SP X. S#
                        AND P# = SP Y. P#))
```

nach Date

Parsing-Ergebnis: Tabelle

REF	SCALE	PHASE	YIELD	TEMP
para.1	small	solid	77%	-78 to 20
TIME	ENERGY	APPARATUS	FEATURES	
	cooling		IR. N MR. MS	
REG. NO	FUNCTION	AMT.	AUTHOR ID	
78624-62-1	product	2.70 g	7a	
78624-61-0	reactant		6a	
13274-48-6	reactant	1.24 g	N-methyltriazolinedione	
	solvent	80 ml	pentane	
	solvent	40 ml	ethyl acetate	

Formular-Repräsentation des Systems SIE für eine chemische Reaktionsbeschreibung

Parsing-Ergebnis: Semantische Repräsentation

"who is the man that leads the LOKI project?"

```
[   request (referent(_5747))
    presuppose (exists (_4340))   ]
    some (_4340)
[  unique (_4407)
   single (_4407)
   instance (_4407, person)
   propval (person,_4407,sex,male)
   [  some (_4725)
      [  unique (_5033)
         single (_5033)
         instance (_5033,project)
         propval (project,_5033,name,str
      (LOKI)) ]
```

```
      instance (_4725,leading)
        propval (leading,_4725,theta,_5033)
        propval (leading,_4725,alpha,_4407)
        topic (_4407)   ]     ]
instance (_4340,identity)
propval (identity,_4340,alpha,_4407)
propval (identity,_4340,theta,_5747)
topic (_4407)
[   some (_5747)
   single (_5747)
   instance (_5747,person) ]
```

Parsingstrategien

• **Startpunkt**	top-down	vs	bottom-up
• **Regelanwendung**	deterministisch	vs	nicht-deterministisch
• **Regelanordnung**	first guess	vs	informierte Auswahl
• **Vorgehen**	Breitensuche	vs	Tiefensuche
• **Richtung**	rechts ---> links	vs	links ---> rechts
		vs	Insel-Parsing
Zielstruktur	syntaktisch	vs	semantisch
• **Informiertheit**	blind	vs	vorherschauend
• **Ambiguitäts-behandlung**	erfolgsorientiert	vs	exhaustiv

Alternativen zum vollen Parsing

- Partielle Strukturanalyse durch
  - Komplexe Stichwort-Suche und anschließend manuelle Analyse (eigentlich ein lexikalisches Verfahren),
  - Wortgruppen-Suche (Kollokationen) und manuelle Analyse,
  - Lexikalische Pattern-Analyse
  - Textsorten- oder themenabhängige Schema-Füllung
  - Einfache lexikalische Übergangsnetzwerke

DYPAR-Pattern Operationen

- Optionales Element ?
- Disjunktion |
- Wertaufnehmende Variable :=
- Wertabgebende Variable =
- Beliebiges Einzelwort $
- Beliebige Zahl $ n
- 0-n Wiederholungen *
- 1-n Wiederholungen +
- n Wiederholungen ↑
- Bis hin zu &u
- Rest der Eingabe $ r
- Alle Permutationen &c
- konsumierende Negation ~
- nicht konsumierende Negation &n
- Ersetzung von Ausdruck durch Funktionswert &i

Beispiele für Patterns im DYPAR-Format

Pattern

(&u sagen) konsumiert die Eingabe bis zum ersten "sagen" (*"Können Sie mir bitte"* ⇒ sagen)

„Bis hin zu“

(&c (Das)(ist)(so)) findet z.B. *"So ist das"* oder *"Ist das so".*

„Alle Permutationen“

(!inhaltswort):= ~ <art> bindet alles außer dem Artikel an inhaltswort

Vorteile und Nachteile von Pattern - Matchern

Vorteile:

- schnell (für zeitkritische Anwendungen geeignet)
- einfach (intuitiv formulierbar, an alle Fragestellungen anpaßbar)
- bei kleinem Umfang gut wartbar
- lokale Effekte

Nachteile:

- liefern keine Strukturbeschreibung
- ungeeignet für einige syntaktische Phänomene (offene Wortfolge, Abhängigkeiten, Fernstellung)
- umständlich für rekursive Strukturen und gleiche Strukturen an verschiedenen Stellen
- bei größerem Umfang unübersichtlich und schwer wartbar

Netzwerkgrammatiken

- Einfache Netzwerkgrammatiken (Endliche Automaten)

  Übergänge in **einem** Netz zwischen

  - Wörtern oder
  - lexikalischen Kategorien

- Rekursive Netzwerkgrammatiken (BTN / RTN)

  Übergänge in gestaffelten Netzen zwischen

  - Netzen
  - Wörtern oder
  - lexikalischen Kategorien

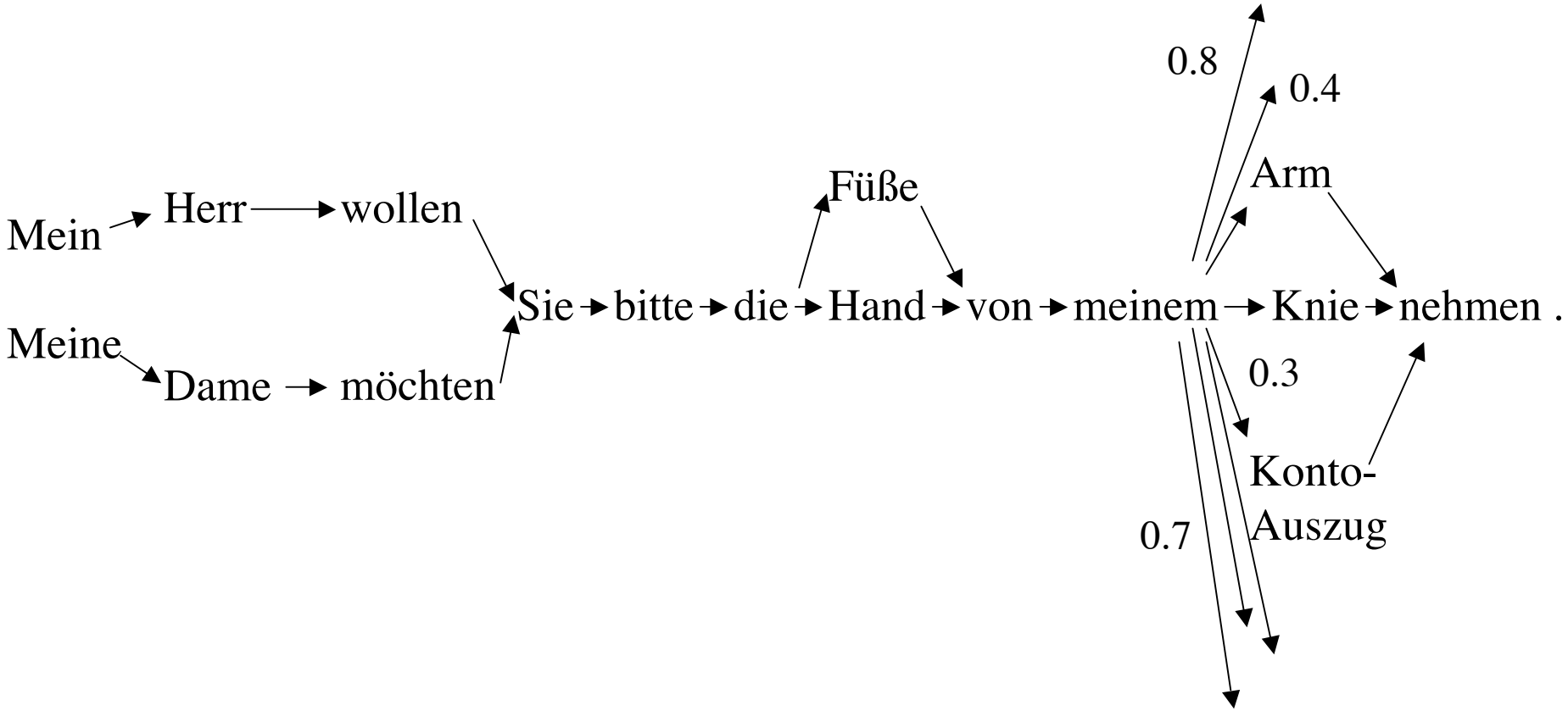
- Erweiterte Netzwerkgrammatiken (ATN)

  Übergänge in gestaffelten Netzen zwischen

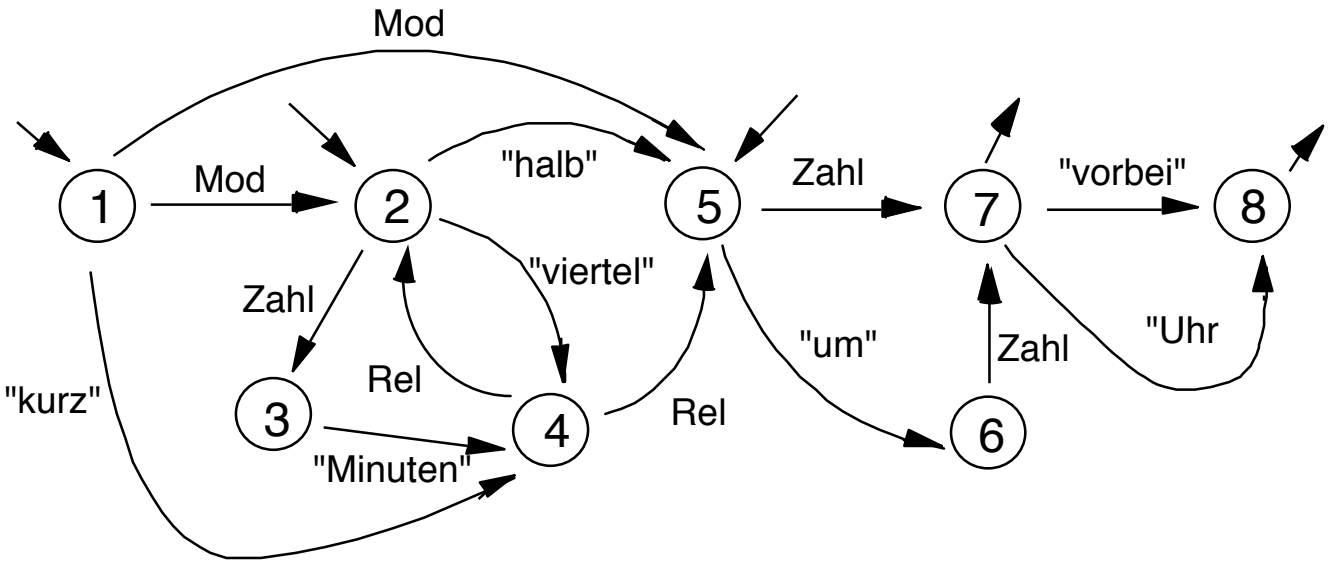
  - Netzen
  - Wörtern oder
  - lexikalischen Kategorien

  unter
  - Ausführung von Aktionen und
  - Aufbau einer Strukturbeschreibung

Lexikalischer Übergangsgraph



Endlicher kategorialer Automat für deutsche Zeitangaben



- Mod: eben, bald, gerade, gleich, genau
- Rel: vor, nach
- Zahl: 1, 2, 3, ...59, 60.

Chart-Parsing

Von Martin Kay 1973 vorgeschlagen. Von Kaplan und Kay als General Syntactic Processor implementiert.

Chart-Parsing - 1 -

Eine Chart ist ein gerichteter bewerteter Graph, der im Verlauf des Parsing-Prozesses um immer weitere Kanten erweitert wird. Der Parser liest eine Konstituentenstruktur-Grammatik und schreibt eine geklammerte Struktur.

Zunächst wird die Chart initialisiert:

- Je einen Knoten an den Satzgrenzen und zwischen je zwei Wörtern der Eingabekette einsetzen, sowie
- Kanten einsetzen, die zwei Knoten links und rechts von einem terminalen Symbol verbinden und mit der Wortklassenbezeichnung des durch sie überspannten Worts markieren. Können einem Wort mehrere alternative Kategorien zugeordnet werden, je eigene Kanten aufspannen.

Chart-Parsing - 2 -

Die Aufgabe des Parsers besteht darin, für jede im Verlauf der Analyse der Eingabe gefundene Konstituente eine mindestens zwei Knoten überspannende Kante in die Chart einzuführen.

Die Analyse gilt als abgeschlossen, wenn eine die ganze Chart überspannende Kante eingerichtet ist.

Die Chart enthält dann alle möglichen syntaktischen Zerlegungen der Eingabe.

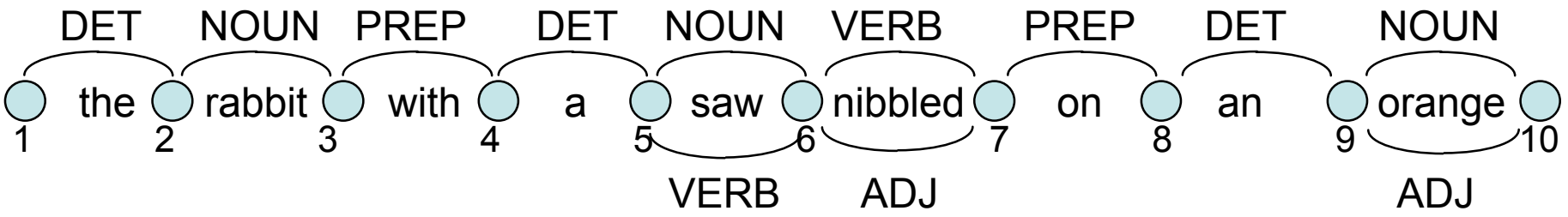
Bei unvollständigen Eingaben wird zwar keine den ganzen Satz überspannende Kante gefunden, aber die Chart enthält meist einige der Grammatik entsprechende Konstituenten als Fragmente, die dann z.B. von einer Komponente zur Rekonstruktion von Ellipsen oder sprechsprachlichen Strukturen weiterverarbeitet werden können.

Chart-Parsing - 3 -

Man unterscheidet zwei Typen von Kanten in einem Chart-Parser:

- Aktive (offene) Kanten zur Darstellung noch unvollständiger Konstituenten
- Inaktive (komplette) Kanten zur Darstellung bereits vollständiger Konstituenten

Initialisierte Chart



Aufbau der Chart

"The rabbit with a saw" nibbled on an orange"

S ($_1$NP VP)

Die Initialisierung trägt die Zahlen zwischen den Terminalen ein und schreibt die erste S-Regel in die Chart: $_1$S → $_1$NP VP
Offene Kanten: $_1$NP → $_1$DET NP2, $_1$NP →$_1$ NP2

S → NP VP
NP → DET NP2
NP → NP2
NP2 → NOUN
NP2 → ADJ NP2
NP2 → NP2 PP
PP → PREP NP
VP → VERB
VP → VERB NP
VP → VP PP

1 DET 2 NOUN 3 PREP 4 DET 5 NOUN 6

Aufbau - 2 -

In die Chart wird als 1. mögliche Fortsetzung von
S ($_1$NP VP) (= die 1. Offene Kante) eingetragen:
$_1$NP → $_1$DET NP2
das paßt am linken Rand mit der Eingabekette.
Offene Kanten: ($_1$NP →$_1$NP2) + $_1$NP → DET $_2$NP2
Komplette Kanten: $_1$DET$_2$, $_2$NOUN$_3$, usw. .....

S ($_1$NP VP)

$_1$NP → $_1$DET NP2

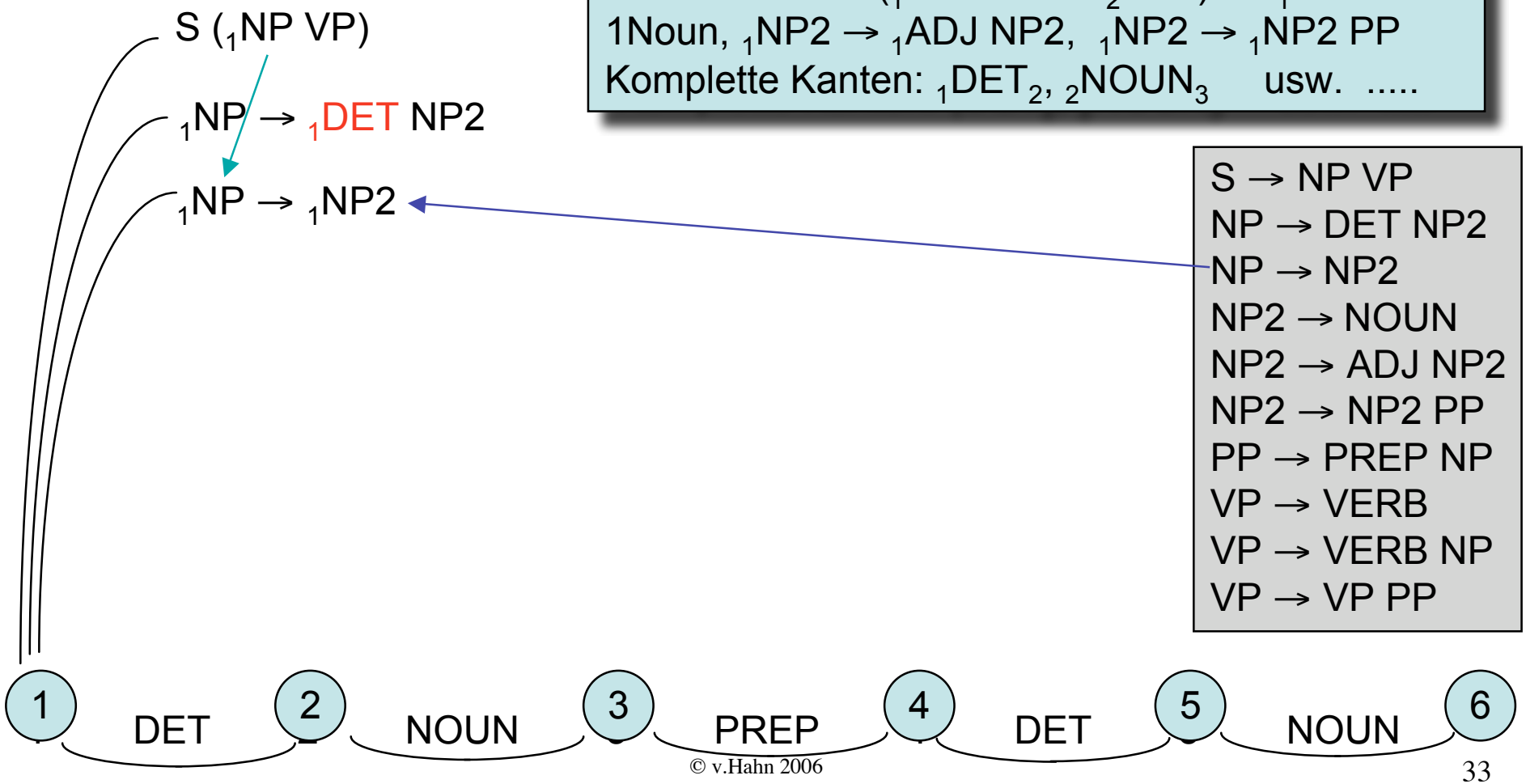
In Klammern
immer die noch
nicht bearbeiteten offen
Kanten
früherer Schritte

S → NP VP
NP → DET NP2
NP → NP2
NP2 → NOUN
NP2 → ADJ NP2
NP2 → NP2 PP
PP → PREP NP
VP → VERB
VP → VERB NP
VP → VP PP

1 DET 2 NOUN 3 PREP 4 DET 5 NOUN 6

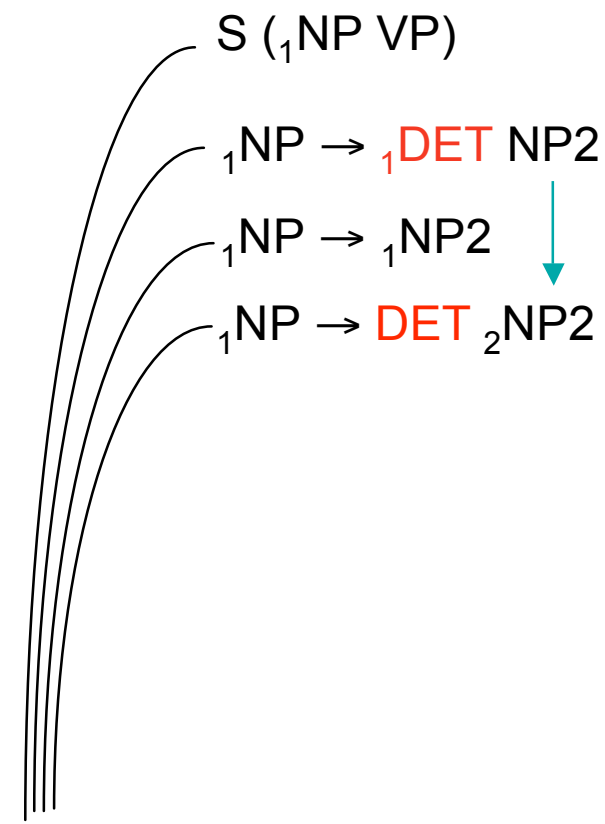
Aufbau - 3 -

In die Chart wird als 2. mögliche Fortsetzung von S ($_1$NP VP) (= älteste offene Kante) eingetragen:
$_1$NP → $_1$NP2
Offene Kanten: ($_1$NP → DET $_2$NP2) + $_1$NP2 → 1Noun, $_1$NP2 → $_1$ADJ NP2, $_1$NP2 → $_1$NP2 PP
Komplette Kanten: $_1$DET$_2$, $_2$NOUN$_3$ usw. .....



Aufbau - 4 -

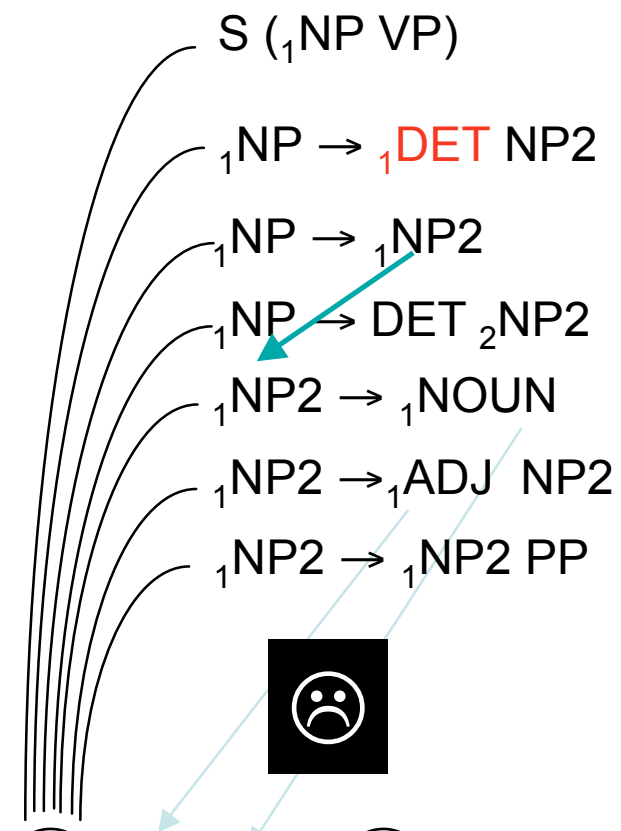
In die Chart wird als Folge von $_1$NP → $_1$DET NP2
eingetragen: $_1$NP → DET $_2$NP2
Offene Kanten:($_1$NP2 → 1Noun, $_1$NP2 → $_1$ADJ NP2,
$_1$NP2 → $_1$NP2 PP) + $_2$NP2 →$_2$ Noun,
$_2$NP2 → $_2$ADJ NP2, $_2$NP2 → $_2$NP2 PP)
Komplette Kanten: $_1$DET$_2$, $_2$NOUN$_3$ usw. .....



S → NP VP
NP → DET NP2
NP → NP2
NP2 → NOUN
NP2 → ADJ NP2
NP2 → NP2 PP
PP → PREP NP
VP → VERB
VP → VERB NP
VP → VP PP

Aufbau - 5 -

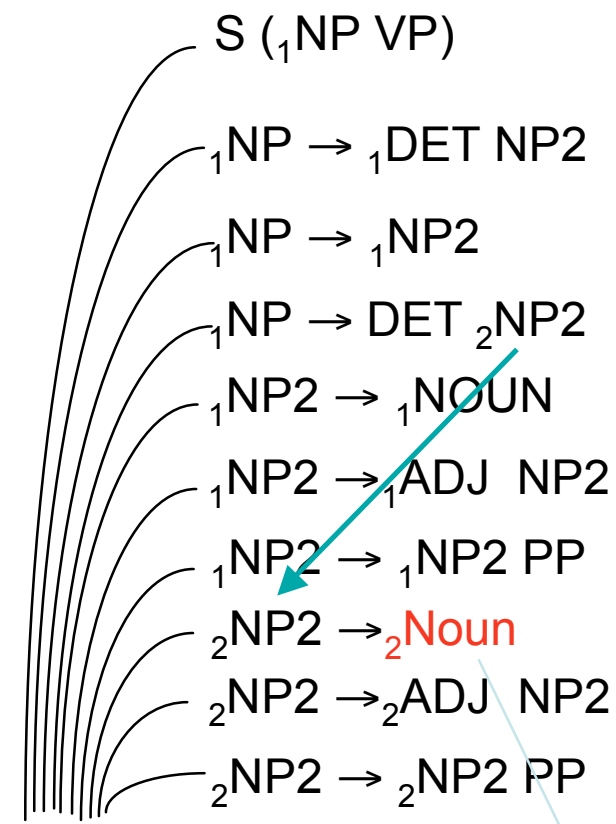
In die Chart wird als Auflösung von
$_1$NP → $_1$NP2 eingetragen: $_1$NP2 →$_1$Noun,
danach $_1$NP2 →$_1$ADJ NP2 und $_1$NP2 → $_1$NP2 PP
Neue offene Kanten: ($_2$NP2 →$_2$ Noun,
$_2$NP2 → $_2$ADJ NP2, $_2$NP2 → $_2$NP2 PP) keine neuen, weil Terminale nicht matchen und 1NP2 schon in der Chart ist.
Komplette Kanten: $_1$DET$_2$, $_2$NOUN$_3$ usw. .....



NP → NP2
NP2 → NOUN
NP2 → ADJ NP2
NP2 → NP2 PP
PP → PREP NP
VP → VERB
VP → VERB NP
VP → VP PP

Aufbau - 6 -

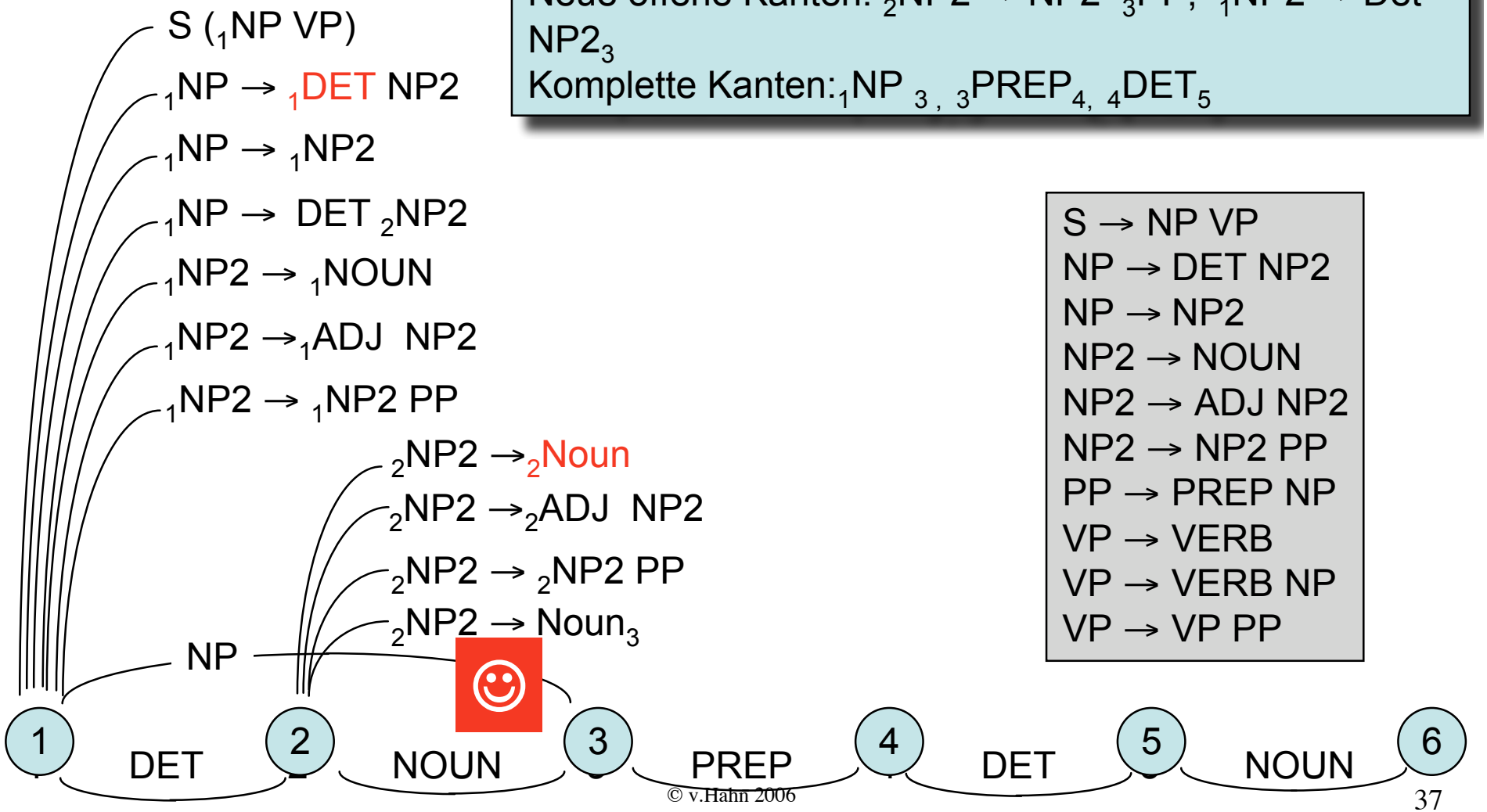
In die Chart wird als mögliche Auflösung von
$_1$NP → DET $_2$NP2 eingetragen:
$_2$NP2 →$_2$ Noun, darauffolgend $_2$NP2 → $_2$ADJ NP2,
$_2$NP2 → $_2$NP2 PP
Neue offene Kanten: $_2$NP2 → Noun$_3$
Komplette Kanten: $_1$DET$_2$, $_2$NOUN$_3$ usw. .....



S → NP VP
NP → DET NP2
NP → NP2
NP2 → NOUN
NP2 → ADJ NP2
NP2 → NP2 PP
PP → PREP NP
VP → VERB
VP → VERB NP
VP → VP PP

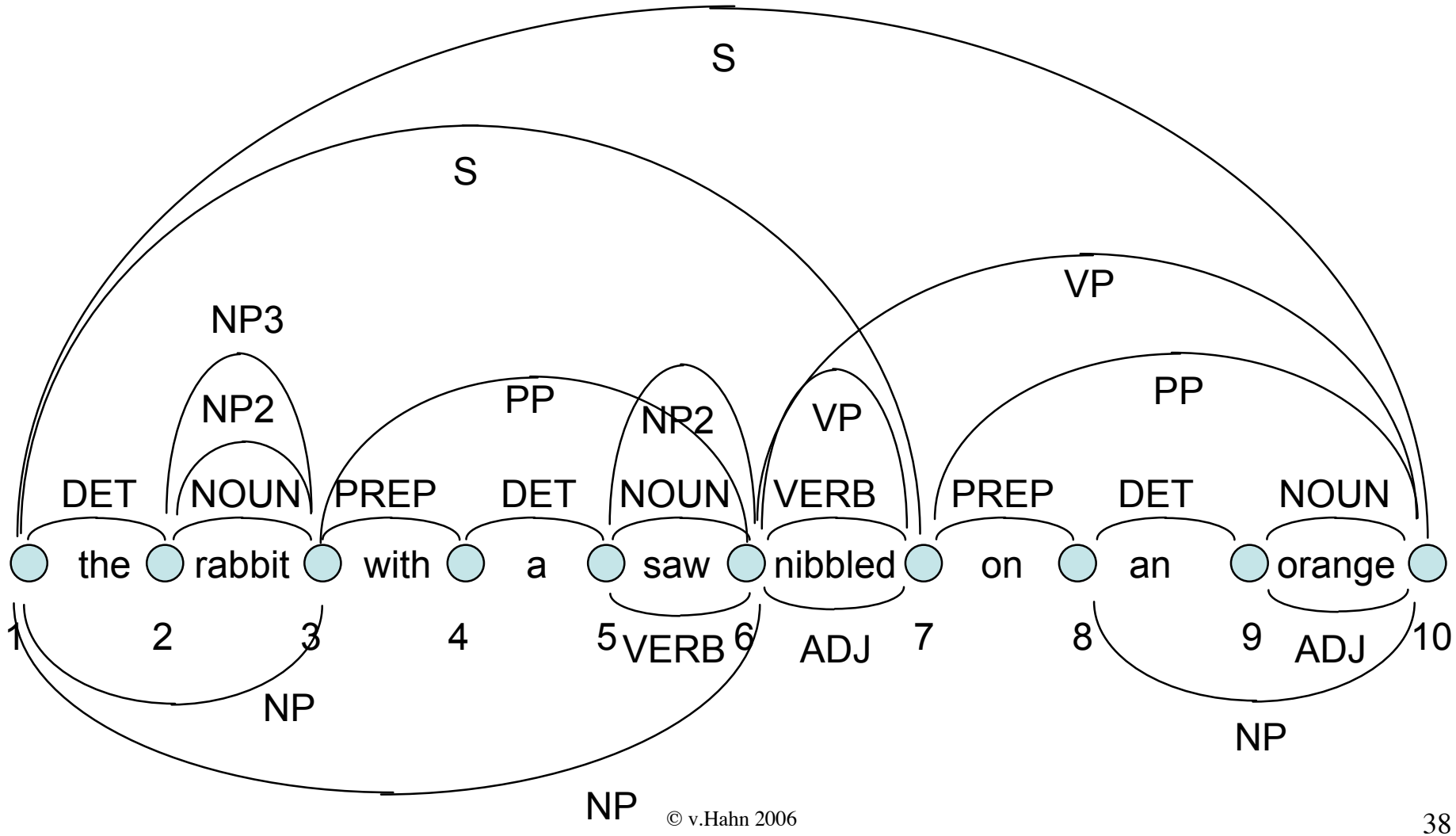
Aufbau - 7 -

In die Chart wird eingetragen: $_2$NP2 → Noun$_3$
Neue offene Kanten: $_2$NP2 → NP2 $_3$PP, $_1$NP2 → Det NP2$_3$
Komplette Kanten: $_1$NP $_3$, $_3$PREP$_4$, $_4$DET$_5$



S → NP VP
NP → DET NP2
NP → NP2
NP2 → NOUN
NP2 → ADJ NP2
NP2 → NP2 PP
PP → PREP NP
VP → VERB
VP → VERB NP
VP → VP PP

Komplette Chart



Nachteil

- Diese Verfahren muß man selbst implementieren oder wenigstens mit guten Programmierkenntnissen anpassen

Implementieren =
Planen, programmieren
und zum Einsatz bringen

Online Parser für deutsche Sprache

http://www.ifi.unizh.ch/CL/InteractiveTools.html

- Der UIS-Parser (Universität Zürich)
- Ein Parser für Text-Analyse (Universität Köln).
- Das Babel-System. Ein HPSG-basierte Parser für Deutsch (DFKI Saarbrücken)

Der UIS-Parser

http://www.ifi.unizh.ch/CL/UIS/parser.html

- Erzeugt eine Konstituentenstruktur und eine funktionale Struktur.
- Benutzt keine semantischen oder statistischen Informationen
- ausgelegt auf grammatisch korrekte Eingaben
- ein Bottom-up Chart-Parser, der mit Phrasenstruktur-, ID- und LP-Regeln arbeitet.

UIS–Parser

Der UIS–Parser ist ein Modul des Projektes zur Erstellung eines Universitäts–Informations–Systems. Er erzeugt eine Konstituentenstruktur und eine funktionale Struktur. Hier eine kurze Leistungsbeschreibung.

Bitte hier deutschen Satz (oder Teilsatz) eingeben. Ein vollständiger Satz muss mit Punkt, Fragezeichen oder Ausrufezeichen abgeschlossen werden.

Das Buch, das du mir gegeben hast, war sehr interessant.

- Umlaute bitte direkt eingeben (ä, ö, ü) oder mit Anführungszeichen umschreiben ("a, "o, "u; z.B. `sch"one Gr"usse`).
- Anstatt 'ß' bitte immer 'ss' eingeben. (Schweizer–Deutsche Rechtschreibung)
- *Hinweis: Wenn das Eingabefeld leer ist, wird der gewählte Beispielsatz verarbeitet.*

Oder wählen Sie einen der Beispielsätze:

Alle neuen Studenten brauchen ein Vorlesungsverzeichnis.

Nur Parser–Ausgabe
Lexikon–Ausgabe und Parser–Ausgabe

Sende Satz an Parser | Eingabefeld leeren

UIS-Oberfläche

Martin Volk < volk@ifi.unizh.ch>
Source: ***http://www.ifi.unizh.ch/CL/UIS/parser.html***
Date of last modification: ***March 16, 2000***

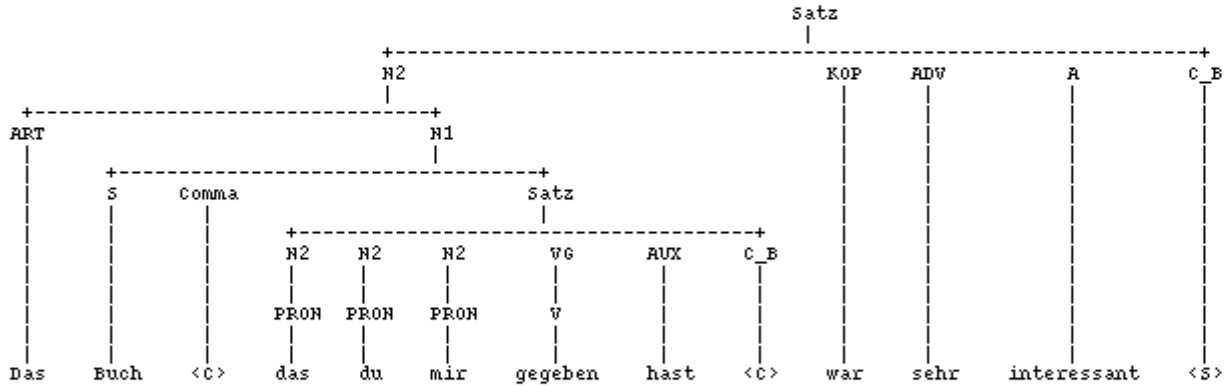
UIS Ergebnis

Parsing–Ergebnis

Eingabe: Das Buch, das du mir gegeben hast, war sehr interessant.

```
[Das,Buch,<C>,das,du,mir,gegeben,hast,<C>,war,sehr,interessant,<S>]

 "Das" xxxxxxxxx
 "Buch" xxxxxxxxx
 "<C>" xxx
 "das" xxxxxxxxx
 "du" xxx
 "mir" xxxxx
 "gegeben" xxx
 "hast" xxx
 "<C>" xxxxxxxxx
 "war" xxxxx
 "sehr" xx
 "interessant" xxxx
 "<S>" xxx
```




```
 Satz(typ:V2nd)
--- N2(typ:DEF..person:3..numerus:SG..kasus:NOM..genus:NEUTR)
------ ART(lemma:das..typ:DEF..numerus:SG..kasus:NOM..gross:(+)..genus:NEUTR..decl:SCHW)
--------- Das
------ N1(lemma:Buch..numerus:SG..kasus:NOM..genus:NEUTR..decl:SCHW)
--------- S(lemma:Buch..numerus:SG..kasus:NOM..genus:NEUTR..decl:SCHW)
------------ Buch
--------- Comma(lemma:<C>)
------------ <C>
--------- Satz(typ:RELAT)
------------ N2(typ:RELAT..person:_1197..numerus:SG..kasus:AKK)
--------------- PRON(lemma:das..typ:RELAT..person:_1197..numerus:SG..kasus:AKK..genus:NEUTR)
------------------ das
------------ N2(typ:PERS..person:2..numerus:SG..kasus:NOM..genus:_1245)
--------------- PRON(lemma:du..typ:PERS..person:2..numerus:SG..kasus:NOM..genus:_1245)
------------------ du
------------ N2(typ:PERS..person:1..numerus:SG..kasus:DAT..genus:_1141)
--------------- PRON(lemma:ich..typ:PERS..person:1..numerus:SG..kasus:DAT..genus:_1141)
```

Ein Parser für Text-Analyse

(http://nebsy.nt.fh-koeln.de:8080/dk_analyse.html)

- Ein WWW-System zur datenbankengestützten Segmentierung von Satzteilen und zur Analyse präpositionaler Phrasen von Texten der deutsche Sprache.
- Das Analysesystem bietet die Möglichkeit, Texte auf das Auftreten von modalen, lokalen, instrumentalen, kausalen und temporalen Relationen zu untersuchen.
- Analyse:
  - Eingabe des Textes für die Textanalyse
  - Eingabe von unbekannten Substantiven
  - Ausgabe des Ergebnisses der Textanalyse als Ergebnistabelle
  - Ausgabe des Ergebnissen der Textanalyse als Graph
  - Umketten der Relationen in Graph

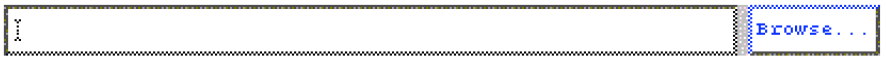
Text-Parser-Eingabe

Sie haben 2 Möglichkeiten für die Texteingabe:

1. **Sie können den Text als eine Datei von Ihrem Computer aus an den Server übertragen. (1. Möglichkeit)**
2. **Sie können den Text direkt in das Formularfeld eingeben. (2. Möglichkeit)**

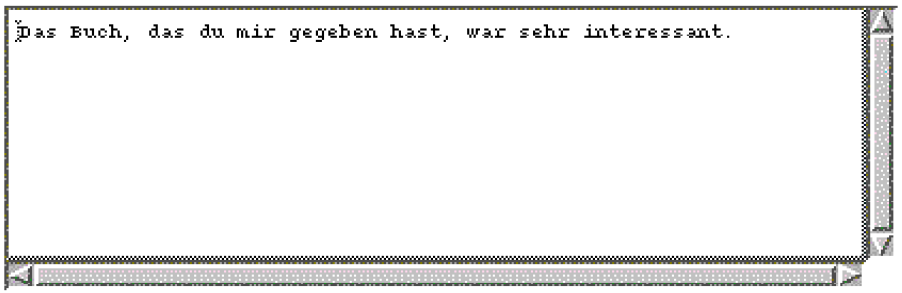
- **1. Möglichkeit:**
  Geben Sie den Dateinamen in das Formularfeld ein und starten Sie anschließend die Übertragung an den WWW-Server.

  Bitte geben Sie den Dateinamen ein:



- **2. Möglichkeit:**
  Bitte geben Sie den zu analysierenden Text ein und starten Sie das Textanalyseprogramm.

  Bitte hier den Text eingeben :



Ergebnis der Analyse :

Es wurden insgesamt 4 Beziehungen gefunden.

Ergebnistabelle

Beziehungen	Anzahl	Substantive
Lokale	0	keine
Temporale	0	keine
Kausale	0	keine
Modale	1	das Buch < [0] Adv: sehr
Instrumentale	0	keine
Verb - Infinitive	3	das Buch < [sein] { }; das Buch < [sein] Adj: interessant; du < [geben] { }

Sie können jetzt eine Grafik, mit denen von Ihnen ausgewählten Beziehungen, auf dem Bildschirm darstellen lassen.

Von Ihnen vorgegebener Text für die Analyse :

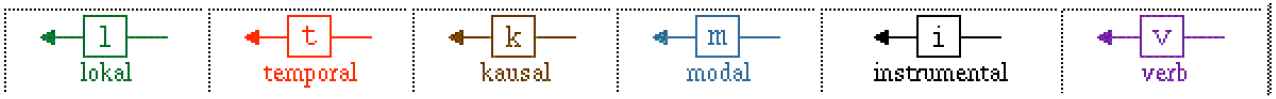
Das Buch, das du mir gegeben hast, war sehr interessant.

Zurück zu der Ergebnistabelle.

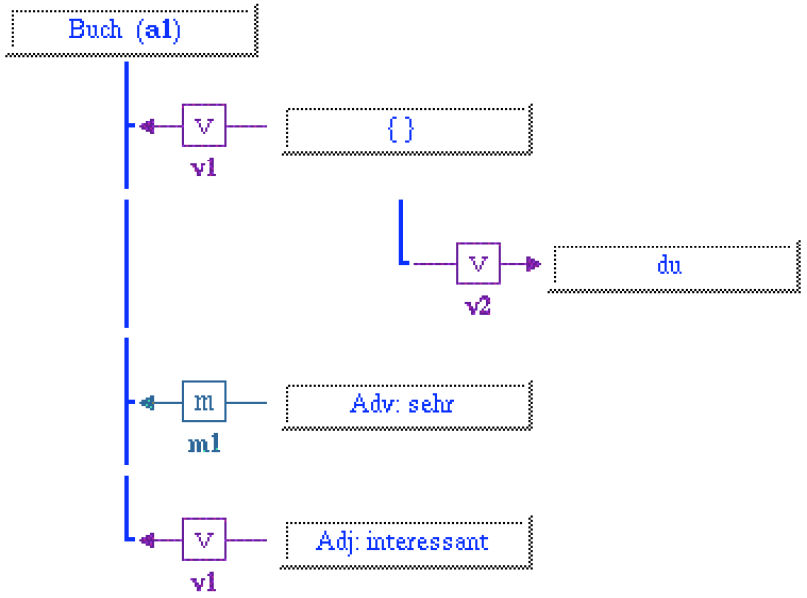
Zur Eingabe für die Grafikdarstellung.

Text-Parser-Ergebnis

Ergebnis der Textanalyse als Grafik:



Es werden 4 Beziehungen als Grafik dargestellt.



Text-Parser-Graphik

Benutzte Abkürzungen für die Vollphrasen der Substantive :

1. **a1** Substantiv : "Buch"
   Vollphrase : "**das** Buch"

Erläuterung zur der Vollphrasendarstellung :

- **Artikel** werden **fett** dargestellt
- *Adjektive werden kursiv dargestellt*

Das Babel-System

`http://www.dfki.de/~stefan/Babel/Interaktiv/`

- Verwendet eine HPSG-Grammatik.
- Eingabesätzen werden syntaktische und semantische Strukturen zugeordnet.
- Es gibt ein Public-Domain-Lexikon, das online erweitert werden kann.

Babel-System

Das Babel–System: HPSG–Interaktiv

Die Java–Version

Von dieser Seite aus können Sie das Babel–System interaktiv testen. (Beispielanfrage)

Geben Sie einen deutschen Satz ein, der dem System bekannte Wörter enthält:

Das Buch, das du mir gegeben hast, war sehr interessant << Reset

Zwischen Groß– und Kleinschreibung wird nicht unterschieden.
Für 'ä' ist 'ae' und für 'ß' 'ss' einzugeben.
Kommata werden (leider) ignoriert.
Sätze sollen mit '.' oder '?' enden. Bei Eingaben ohne Satzzeichen werden alle Phrasen zurückgegeben.

Ausgabeflags:

Ausgabe von Syntaxbäumen:	◉ja ○nein
Ausgabe von Merkmalstrukturen:	◉ja ○nein
Ausgabe mittels fegramed:	◉ja
Ausgabe von Töchtern in MS:	◉ja ○nein
Ausgabe von logischen Formen:	◉ja ○nein
Ausgabe der Anzahl der konstruierten Zeichen:	◉ja ○nein

St. Mü. (Stefan.Mueller@dfki.de)

Literatur

- Winograd, Terry, Language as a Cognitive Process. Addison-Wesley, 1993
- Allen, James, Natural Language Understanding. Benjamin/Cummings, 1995
- Carstensen, K.U., Ebert, C., Endriss, Jekat, S., Klabunde, R., Langer, H. (Hrsg.), Computerlinguistik und Sprachtechnologie - Eine Einführung-. Spektrum 2001