

## Sprachorientierte KI: Syntax und Parsing

- Syntax als Untersuchungsgegenstand
- Wortartendisambiguierung
- Phrasenstrukturgrammatiken
- Parsing mit Phrasenstrukturgrammatiken
- Restringierte Phrasenstrukturgrammatiken
- Unifikationsgrammatiken
- Constraint-basierte Grammatiken
- Robustes Parsing



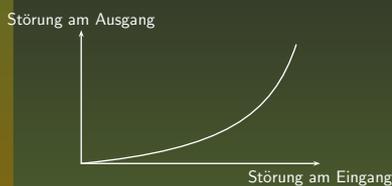
## Robustes Parsing

- Robustheit
- Robustes PSG-Parsing
- Parsing als Constraint Satisfaction
- Fehlersensitives Parsing



## Robustheit

- erwartungsgereutes Wohlverhalten bei variierenden Eingabedaten



- Adaption an veränderte Umweltbedingungen



## Robustheit

- Robustheit wogegen?
  - nichtnormgerechter Input
    - agrammatische Konstruktionen
    - kreativer Sprachgebrauch (semantische "Abweichungen")
  - Unsicherheit in den sensorischen Daten (gesprochene Sprache, Handschrift)
  - Sprechervarianz (idiolektal, dialektal or soziolektal)
  - Umgebungsgeräusche



## Robustheit

- Robustheit wogegen? (Fortsetzung)
  - mangelnde Kompetenz
    - Beschränkung auf Subsprachen
      - lexikalisch
      - grammatisch
      - semantisch/konzeptuell
  - Ressourcenbeschränkungen
    - Zeit
    - mediale und modale Restriktionen

- Analyse von fehlerhaftem Input
- Fehlertoleranz ↔ Fehlerdiagnose



## Robustheit

- drei Voraussetzungen
  - Behandlung von Inkonsistenz
  - Plausibilitätsbasierte Abwägung im Konfliktfall
  - Ausnutzen struktureller Redundanz



## Behandlung von Inkonsistenz

- formale Grammatiken basieren auf einer zweiwertigen Logik  
→ Inkonsistenz kann nicht toleriert werden
- aber: Inkonsistenz ist konstitutiv für natürliche Sprache



## Behandlung von Inkonsistenz

- Inkonsistenz ...
  - ... zwischen Äußerung und Grammatik
    - Ich habe fertig!*
    - Nicht am Montag, weil da ist schon alles voll.*
    - unvollständige Äußerungen
    - Erkennungsfehler
    - spontansprachliche Phänomene
  - ... zwischen Äußerung und Hintergrundwissen
    - Laserstrahl übersetzt!*



## Behandlung von Inkonsistenz

- Inkonsistenz ...
  - ... zwischen Äußerung und pragmatischen Erwartungen
  - ... zwischen den sprachlichen Modalitäten: MCGURK-Effekt
  - ... zwischen den Regeln der Grammatik
    - Extraposition: strukturelle vs. referentielle Distanz
    - anaphorische Referenz: Distanz vs. syntaktische Parallelen
    - lineare Anordnung: Präferenzen im deutschen Mittelfeld



## Behandlung von Inkonsistenz

- Inkonsistenz ...
  - ... zwischen verschiedenen Repräsentationsebenen der Sprache
    - Topikalisierung: *Das Zeug willst du essen?*
    - Metaphorik: *Das Auto säuft wie ein Loch!*
    - konzeptuelle Referenz: *Moses und seine Arche*
    - anaphorische Referenz: *Das Mädchen spielt mit ihrer Puppe.*
  - ... zwischen präferierter Lesart und kontextueller Einbettung  
*The horse raced past the barn fell.*



## Behandlung von Inkonsistenz

- Inkonsistenzen ...
  - ... bleiben vielfach unbemerkt
    - MCGURK-Effekt
    - inkonsistente referentielle Bezüge



## Behandlung von Inkonsistenz

- Inkonsistenzen ...
  - ... sind kommunikative Indikatoren
  - ... wecken die Aufmerksamkeit des Hörers
    - Topikalisierung
    - kontrastive Betonung
    - Metaphorik
    - alle Arten von Erwartungsverletzungen
  - ... können die verbale Interaktion steuern
    - z.B. unvollständige Äußerungen
  - ... sind grundlegend für Sprachspiele und verbalen Humor



## Behandlung von Inkonsistenz

- Lösungsansätze auf der Modellierungsebene
  - Constraint-Ignoranz
    - starkes Übergenerieren der Grammatik
    - Hoffnung: Input-Fehler liegen im hinzugekommenen Sprachbereich
  - Fehlerantizipation
    - gezieltes "Aufweichen" der Grammatik
    - z.B. Transferfehler beim L<sub>2</sub>-Erwerb, Restart-Phänomene



## Behandlung von Inkonsistenz

- Lösungsansätze auf der prozeduralen Ebene
  - Constraint-Relaxation
    - z.B. Rücknahme von Kongruenzforderungen im Fehlerfall
    - aber: Kongruenzverletzung ist kein typischer Fehlerfall
  - Modifizierte Parsing-Strategien



## Behandlung von Inkonsistenz

- Grundproblem bei allen Ansätzen:  
Suchraum wird - ohne geeignete Kompensation - extrem ausgeweitet
- Fehlerregeln werden üblicherweise nur beim Scheitern der normalen Analyse angewendet
  - kognitiv adäquat?
- Ziel:  
Ignorieren < Detektieren < Lokalisieren < Korrigieren



## Plausibilitätsbasierte Abwägung

- Plausibilitätsmaße
  - Wahrscheinlichkeit
  - Aktivierungspotenzial
  - Belohnung bzw. Strafe



## Plausibilitätsbasierte Abwägung

- Zuordnung von Plausibilitätsmaßen zu Grammatikregeln?
  1. grobkörnige Grammatik: Unabhängigkeitsannahme ist verletzt (vgl. DOP)
  2. feinkörnige Grammatik: Regelmenge wächst exponentiell



## Strukturelle Redundanz

- Syntax vs. Semantik:



## Strukturelle Redundanz

- Syntax vs. Semantik:
  - semantisch leere Äußerungen  
Der Flügelflagel gaustert  
durchs Wiruwaruwolz  
die rote Finger plaustert  
und grausig gutzt der Golz.  
*Christian Morgenstern*



## Strukturelle Redundanz

- Syntax vs. Semantik:
  - semantisch leere Äußerungen  
Der Flügelflagel gaustert  
durchs Wiruwaruwolz  
die rote Finger plaustert  
und grausig gutzt der Golz.  
*Christian Morgenstern*
  - orthografisch/syntaktisch fehlerhafte Äußerungen  
Der Ferd had fier beiner  
an jede seite einer  
und hatt er mal keiner  
umfällt.



## Strukturelle Redundanz

- Hearing lips and seeing voices:  
MCGURK AND McDONALD (1976)
  - Lippenlesen
  - visueller Stimulus kann den auditiven dominieren

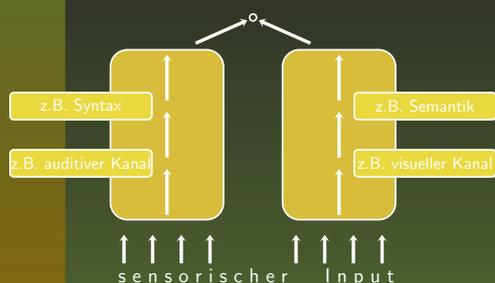


## Strukturelle Redundanz

- Beobachtungen
  - enge Kooperation verschiedener Komponenten
  - falls erforderlich is vollständige Autonomie möglich
  - optimale Arbeitsweise bei kooperativem Zusammenwirken
  - gegenseitige Kompensation von partiellen Defiziten



## Funktionale Autonomie

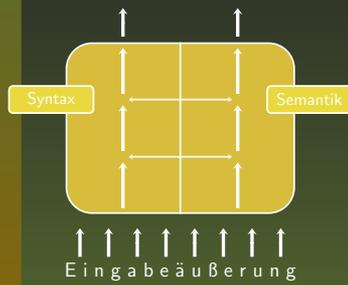


## Funktionale Autonomie

- Probleme:
  - Abwägen zwischen widersprüchlichen Resultaten:  
→ Robustheit entscheidet sich am Punkt der Informationsfusion
  - Fehlerkombinationen können zum kompletten Systemzusammenbruch führen



## Integrierte Architekturen



HPSG (POLLARD UND SAG 1994)

Construction Grammar (FILLMORE ET AL. 1988)

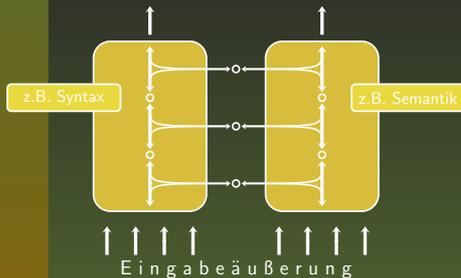
Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 23

## Integrierte Architekturen

- Probleme:
  - konjunktive Kombination von partiellen Strukturen  
→ Autonomie geht verloren

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 24

## Interaktive Architekturen



■ bidirektionale Ausbreitung von Evidenz

■ plausibilitäts-basierte Interaktion

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 25

## Interaktive Architekturen

- Robustheit gegen
  - partiellen Input
  - partiell gestörter Input
  - partielles Systemversagen

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 26

## Robustes PSG-Parsing

- Kombiniertes Chart-Parsing (MELLISH 1989, KATO 1994)
  - bottom-up-Analyse
  - im Fehlerfall: Integration der ermittelten Baum-Fragmente durch top-down-Analyse mit *dynamisch modifizierten* Grammatikregeln
  - neue Kantenintroduktionsregeln:
    - *Garbage-Rule*: eine nichtleere Teilkette im Input kann übersprungen werden
    - *Empty-Category-Rule*: eine geforderte Kategoriensequenz wird ignoriert
    - *Unknown-Word-Rule*: eine Wortform mit fehlender (oder fehlerhafter) Kategorie wird dennoch integriert

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 27

## Robustes PSG-Parsing

- Kombiniertes Chart-Parsing (Fortsetzung)
  - Suchraumbeschränkung durch Beschränkung auf Hypothesen mit minimalem Strafmaß
  - Aufwand für Sätze der Länge 12

Fehler	BU-Zyk.	Lös.	TD-Zyklen		
			erste	letzte	gesamt
ohne	198	1	0	0	0
ein Wort löschen	155	7	33	315	1002
unbek. W. einfügen	156	3	132	289	1922
bek. W. einfügen	170	5	99	325	1775
unbek. W. ersetzen	150	3	42	109	1162

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 28

## Robustes PSG-Parsing

- GLR\*: Modifizierter TOMITA-Parser (LAVIE AND ROSÉ 2001)
  - Überspringen von Wortformen, die sich nicht integrieren lassen
- LCFLEX: Modifizierter Chart-Parser
  - Überspringen von Wortformen
  - einfügen von Nichtterminalen
  - "weiche" Merkmalsunifikation

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 29

## Parsing als Constraint-Satisfaction

- Initiale Belegung durch *alle* möglichen Strukturrepräsentationen
- Sukzessive Reduktion der Interpretationsvarianten durch (gewichtete) Constraints:
  - Parsing als Constraint Satisfaction
  - Parsing als Disambiguierung
- Prozedur terminiert, wenn eindeutige Interpretation vorliegt:
  - Zielorientierte Analyse

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 30

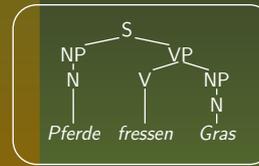
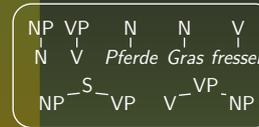
## Parsing als Constraint-Satisfaction

- Erweiterung zur Constraint-Optimierung
  - Verwendung gewichteter Constraints
    - Gewichte als Maß für die Akzeptabilität einer Constraintverletzung
  - Auswahl der Strukturbeschreibung mit minimalen Constraintverletzungen
    - Optimierungsproblem
    - plausibilitäts-basierte Entscheidung

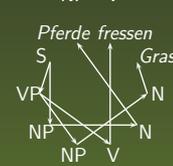


## Parsing als Constraint-Satisfaction

### Konstruktives Parsing



### Eliminatives Parsing



## Parsing als Constraint-Satisfaction

- Constraint Grammar (KARLSSON 1995)
  - Zuordnung von syntaktischen Markern zu Wortformen
  - oberflächenorientiert, funktionale Beschreibung
    - @+FMAINV finites Verb eines Satzes
    - @SUBJ grammatisches Subject
    - @OBJ direktes Object
    - @DN> Artikel modifiziert ein Nomen nach rechts
    - @NN> Nomen modifiziert eine Nomen nach rechts
  - partiell unterspezifizierte Dependenzrelationen



## Parsing als Constraint-Satisfaction

- Constraints
  - Bedingungen für die Verträglichkeit von syntaktischen Markern
  - indirekte Definition einer wohlgeformten Äußerung:
    - akzeptabel ist alles, was keine Constraints verletzt



## Parsing als Constraint-Satisfaction

- Beispiel: Subjektdetektion
  - (@w=ε! (@SUBJ) (1 VFIN) (1 ACTIVE) (NOT \*-1 @SUBJ) (NOT \*1 @SUBJ))
  - Wähle aus einer Kategorienmenge deterministisch die Kategorie @SUBJ, wenn unmittelbar rechts davon ein finites Verb im Aktiv steht und weder im linken noch im rechten Kontext des Satzes die Kategorie @SUBJ vorkommt.



## Parsing als Constraint-Satisfaction

- Disambiguierung durch Constraint-Anwendung
    - Reduzieren der Menge von Markern
    - eliminatives Verfahren
    - nicht immer ist vollständige Disambiguierung möglich
- Bill saw the little dog in the park
- @SUBJ @+FMAINV @DN> @AN> @OBJ @<NOM @DN> @<P @<ADVCL



## Parsing als Constraint-Satisfaction

- kein vollständiges Verfahren zum Constraint Satisfaction
- Standard-Ablaufschema: maximal fünffaches Abarbeiten der Constraint-Menge
- anschließend optionale Einbeziehung heuristischer Constraints zurückhaltende Anwendung
  - nur einzeln
  - anschließend erneute syntaktische Disambiguierung



## Parsing als Constraint-Satisfaction

- Einbeziehung in ein komplettes morphosyntaktisches Analysesystem
  - Vorverarbeitung
    - Umwandlung von Großbuchstaben
    - Interpunktionsanalyse
    - Behandlung fester Syntagmen
  - Lexikonaktualisierung für neue Wörter (manuell)
  - Morphologische Analyse
  - lokale morphologische Disambiguierung
  - Constraint Grammar Parsing
  - Mehrpas-Analyse



## Parsing als Constraint-Satisfaction

- Ablaufschema des Parsers
  - (1) Satzgrenzenermittlung (Interpunktion, Konjunktionen)
  - (2a) kontextabhängige Disambiguierung
  - (2b) Satzgrenzenermittlung
  - (3a) kontextabhängige Disambiguierung
  - (3b) Satzgrenzenermittlung
  - (4) morphosyntaktische Abbildung  
(morphologische → syntaktische Kategorien)
  - (5) syntaktisches Parsing



## Parsing als Constraint-Satisfaction

- Grammatikgröße (Englisch): 2000 Constraints
- Analyseaufwand
  - Entwicklungssystem (Lisp)  
3 ... 5 Wörter/s (Sparc2)
  - Produktionssystem (C)  
400 ... 500 Wörter/s (Sparc 10)  
200 Mio. Wörter (*Bank of England*) bereits getagged
- Analysequalität

	ohne Heuristik	mit Heuristik
disambiguiert (precision)	95.5%	97.4%
davon korrekt (recall)	99.7 ... 99.9%	99.6 ... 99.9%



## Parsing als Constraint-Satisfaction

- Vorteile
    - rudimentäre Robustheit
      - Abbruch, wenn vollständige Disambiguierung erreicht oder keine Constraints mehr zur Verfügung stehen
    - Parallele zwischen Aufwand und Ergebnisqualität
- "Mental effort is needed for achieving clarity, precision and maximal information. Less efforts imply (retention of) unclarity and ambiguity, i.e. information decrease. In several types of parsers, rule applications create rather than discard ambiguities: the more processing, the less unambiguous information."



## Parsing als Constraint-Satisfaction

- Vorteile (Fortsetzung)
  - Bewertung des Analysefortschritts möglich
- Nachteil: grobe oberflächensyntaktische Beschreibung

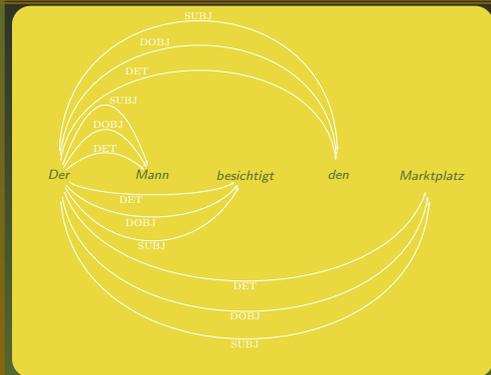


## Parsing als Constraint-Satisfaction

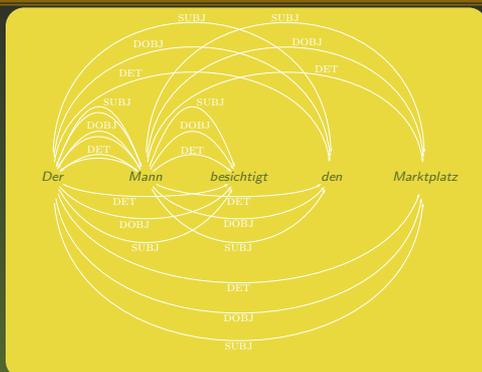
- Erweiterung: Parsing als strukturelle Disambiguierung
  - Constraint Dependency Grammar (MARUYAMA 1990)
  - vollständig spezifizierte Dependenzrelationen  
 $D \in W \times W \times L$



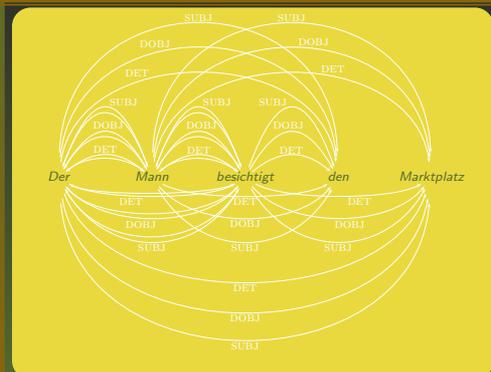
## Hypothesenraum



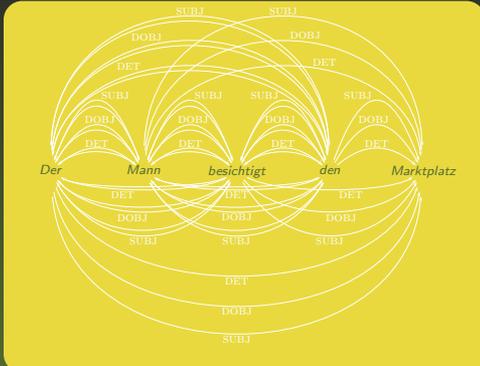
## Hypothesenraum



## Hypothesenraum

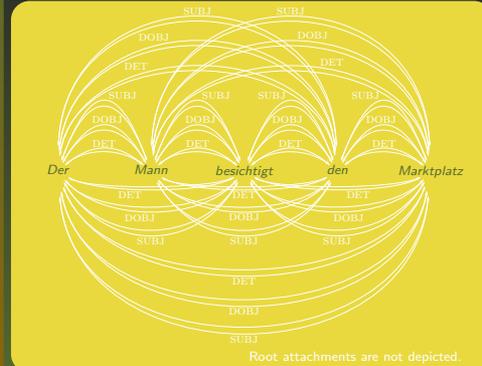


## Hypothesenraum



Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 47

## Hypothesenraum



Root attachments are not depicted.  
Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 48

## Hypothesenraum

root/nil	root/nil	root/nil	root/nil	root/nil
det/2	det/1	det/1	det/1	det/1
det/3	det/3	det/2	det/2	det/2
det/4	det/4	det/4	det/3	det/3
det/5	det/5	det/5	det/5	det/4
subj/2	subj/1	subj/1	subj/1	subj/1
subj/3	subj/3	subj/2	subj/2	subj/2
subj/4	subj/4	subj/4	subj/3	subj/3
subj/5	subj/5	subj/5	subj/5	subj/4
dobj/2	dobj/1	dobj/1	dobj/1	dobj/1
dobj/3	dobj/3	dobj/2	dobj/2	dobj/2
dobj/4	dobj/4	dobj/4	dobj/3	dobj/3
dobj/5	dobj/5	dobj/5	dobj/5	dobj/4

Der	Mann	besichtigt	den	Marktplatz
1	2	3	4	5

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 49

## Parsing als Constraint-Satisfaction

### spezielle Selektorfunktionen

#### für den Relationszugriff

- $\text{dom}(X)$  modifizierter (dominierender) Knoten
- $\text{dep}(X)$  modifizierender (abhängiger) Knoten
- $\text{lab}(X)$  Kantenmarkierung

#### für den Knotenzugriff

- $\text{word}(X)$  Wortform
- $\text{cat}(X)$  (syntaktische) Kategorie
- $\text{pos}(X)$  Positionsindex
- $\text{semprop}(X)$  semantische Eigenschaften

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 50

## Parsing als Constraint-Satisfaction

### unäre Constraints

$$\text{cat}(\text{dep}(X))=D \rightarrow (\text{lab}(X)=\text{DET} \wedge \text{cat}(\text{dom}(X))=N \wedge \text{pos}(\text{dep}(X)) < \text{pos}(\text{dom}(X)))$$

Ein Artikel kann ein rechts von ihm stehendes Nomen mit der Relation DET modifizieren.

### binäre Constraints

$$(\text{dom}(X)=\text{dom}(Y) \wedge \text{cat}(\text{dom}(X))=V \wedge \text{pos}(\text{dep}(X)) < \text{pos}(\text{dom}(X)) \wedge \text{pos}(\text{dep}(Y)) < \text{pos}(\text{dom}(Y))) \rightarrow X=Y$$

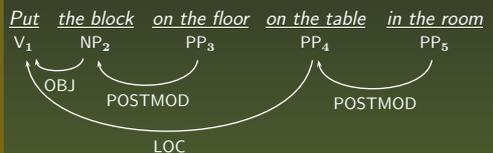
Links vom finiten Verb des Satzes kann sich nur ein einziger Modifikator befinden.

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 51

## Parsing als Constraint-Satisfaction

### vollständiges Verfahren zur Bedingungsüberprüfung (Constraint Satisfaction)

#### Beispiel: PP-Attachment



Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 52

## Parsing als Constraint-Satisfaction

### initiale Constraint-Menge

$$\text{sy1: } \text{word}(\text{dep}(X))=\text{PP} \rightarrow (\text{word}(\text{dom}(X)) \in \{\text{PP}, \text{NP}, \text{V}\} \wedge \text{pos}(\text{dom}(X)) < \text{pos}(\text{dep}(X)))$$

Eine PP modifiziert eine PP, eine NP oder ein Verb links von ihr.

$$\text{sy2: } (\text{word}(\text{dep}(X))=\text{PP} \wedge \text{word}(\text{dom}(X)) \in \{\text{PP}, \text{NP}\}) \rightarrow \text{lab}(X)=\text{POSTMOD}$$

Eine PP modifiziert eine PP oder eine NP durch eine POSTMOD-Relation.

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 53

## Parsing als Constraint-Satisfaction

### initiale Constraint-Menge (Fortsetzung)

$$\text{sy3: } (\text{word}(\text{dep}(X))=\text{PP} \wedge \text{word}(\text{dom}(X))=V \rightarrow \text{lab}(X)=\text{LOC}$$

Eine PP modifiziert ein V durch eine LOC-Relation.

$$\text{sy4: } \text{word}(\text{dep}(X))=\text{NP} \rightarrow (\text{word}(\text{dom}(X))=V \wedge \text{lab}(X)=\text{OBJ} \wedge \text{pos}(\text{dom}(X)) < \text{pos}(\text{dep}(X)))$$

Eine NP modifiziert ein V links von ihr durch eine OBJ-Relation.

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 54

## Parsing als Constraint-Satisfaction

### ■ initiale Constraint-Menge (Fortsetzung)

sy5:  $\text{word}(\text{dep}(X))=V \rightarrow (\text{mod}(X)=\text{nil} \wedge \text{lab}(X)=\text{ROOT})$

Ein Verb modifiziert nil durch eine ROOT-Relation.

Einziges binäres Constraint

sy6:  $\text{pos}(\text{dom}(X)) < \text{pos}(\text{dep}(Y)) < \text{pos}(\text{dep}(X))$   
 $\rightarrow \text{pos}(\text{dom}(X)) \leq \text{pos}(\text{dom}(Y)) \leq \text{pos}(\text{dep}(X))$

Dependenzstrukturen sind projektiv.



## Parsing als Constraint-Satisfaction



## Parsing als Constraint-Satisfaction



14 Lesarten

## Parsing als Constraint-Satisfaction

### ■ Anwendung zusätzlicher Constraints (statische Semantik)

se1:  $\text{word}(\text{dep}(X))=PP$

$\wedge \text{on-table} \in \text{semprop}(\text{dep}(X))$

$\rightarrow \neg(\text{floor} \in \text{semprop}(\text{dom}(X)))$

Der Fußboden kann sich nicht auf dem Tisch befinden.

se2:  $(\text{lab}(X)=\text{lab}(Y)=\text{LOC} \wedge \text{dom}(X)=\text{dom}(Y))$

$\wedge \text{word}(\text{dom}(X))=V \rightarrow X=Y$

Ein Verb kann nicht zwei Lokalbestimmungen haben.



## Parsing als Constraint-Satisfaction



## Parsing als Constraint-Satisfaction

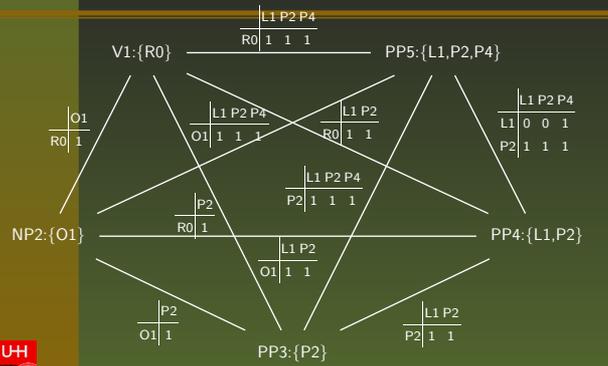


## Parsing als Constraint-Satisfaction

### ■ WALTZ-Filtering



## Parsing als Constraint-Satisfaction



4 Lesarten

## Parsing als Constraint-Satisfaction

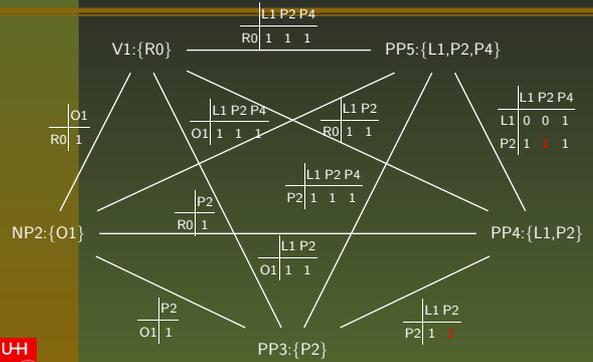
- Anwendung zusätzlicher Constraints (dynamische Semantik)

$$se3: (\text{lab}(X)=\text{lab}(Y)=\text{POSTMOD} \wedge \text{dom}(X)=\text{dom}(Y) \\ \wedge \text{block-is-on} \in \text{semprop}(\text{dep}(X)) \\ \wedge \text{block-is-on} \in \text{semprop}(\text{dep}(Y))) \\ \rightarrow X=Y$$

Kein Objekt kann sich gleichzeitig auf zwei anderen befinden.

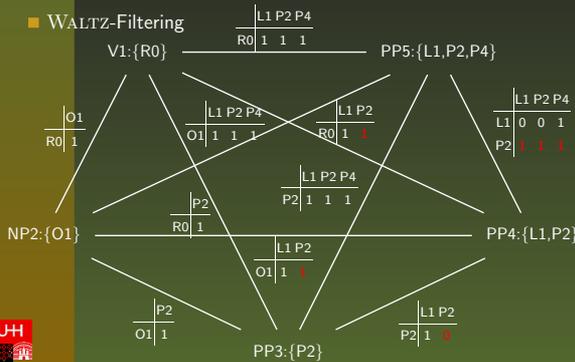


## Parsing als Constraint-Satisfaction



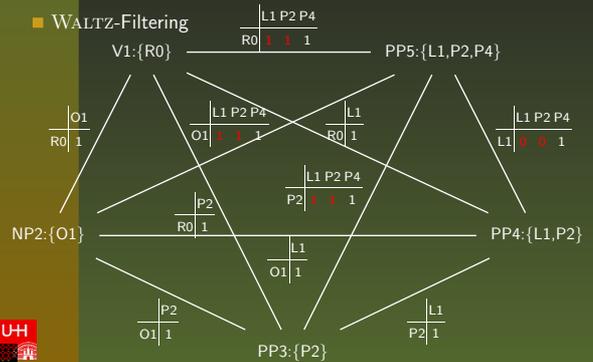
## Parsing als Constraint-Satisfaction

- WALTZ-Filtering

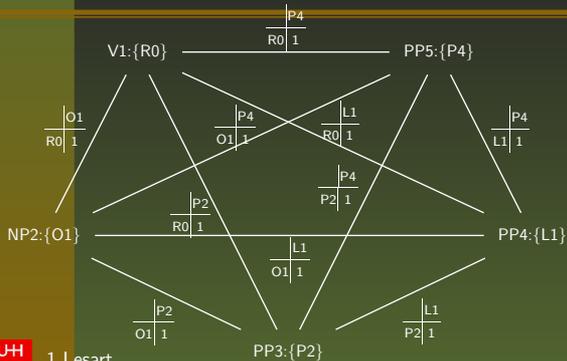


## Parsing als Constraint-Satisfaction

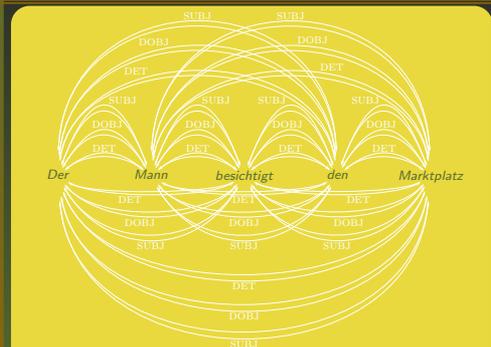
- WALTZ-Filtering



## Parsing als Constraint-Satisfaction

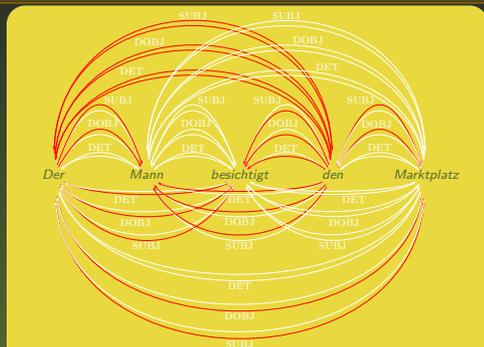


## Kantelenimation



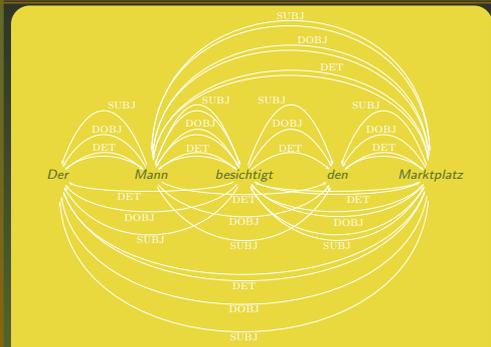
Initial state of a parsing problem with three labels (DET, SUBJ, DOBJ)

## Kantelenimation

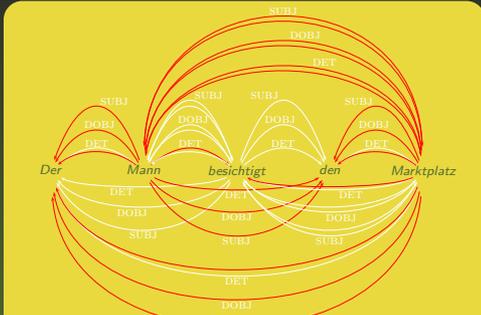


{X}: DetNom : Det : 0.0 : X|cat=det → X↑ cat=noun ∧ X.label=DET

## Kantelenimation

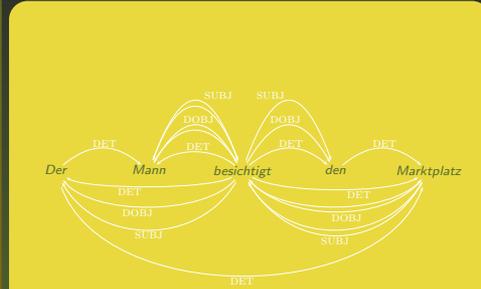


## Kantelenimation



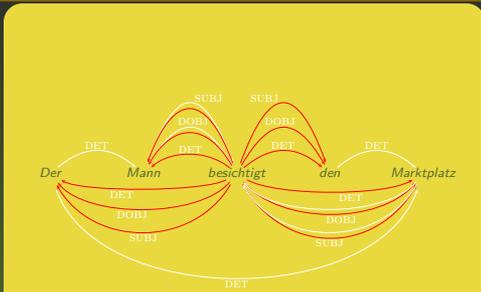
{X} : SubjObj : Verb : 0.0 : SUBJ  
 X↓cat=noun → X↑ cat=vfin ∧ X.label=SUBJ ∨ X.label=DOBJ  
 Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 71

## Kantelenimation



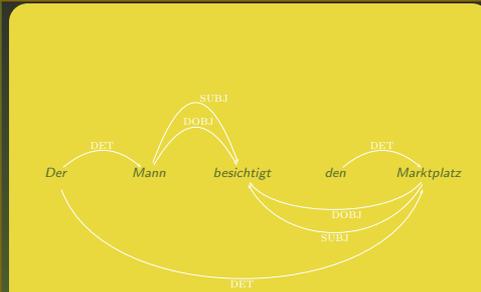
Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 72

## Kantelenimation



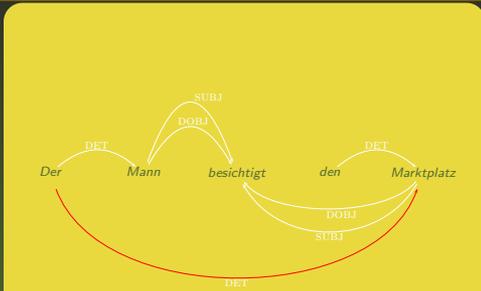
{X} : Root : Verb : 0.0 :  
 X↓cat=vfin → X↑ cat=nil  
 Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 73

## Kantelenimation



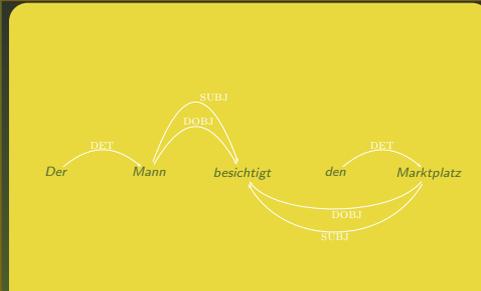
Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 74

## Kantelenimation



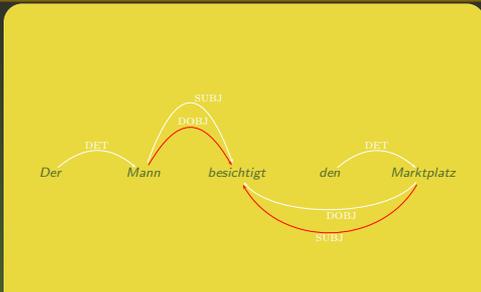
{X,Y} : Unique : General : 0.0 :  
 X↑ id=Y↑ id → X.label≠Y.label  
 Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 75

## Kantelenimation



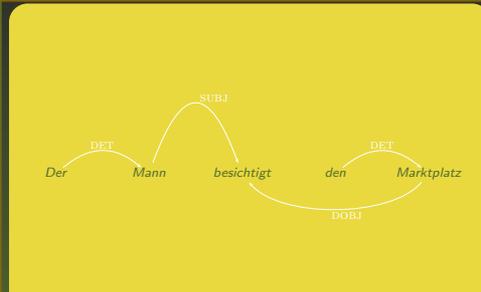
Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 76

## Kantelenimation



{X,Y} : SubjAgr : Subj : 0.0 :  
 X.label=SUBJ ∧ Y.label=DET ∧ X↓id=Y↑ id → Y↑ case=Y↓case=nom  
 Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 77

## Kantelenimation



Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 78

## Parsing als Constraint-Satisfaction

- CDG ist schwach kontextsensitiv
  - Automatische Transformation (CFG → CDG) ist möglich
- Analyseaufwand:  $\mathcal{O}(|C| \cdot n^4)$ 
  - $n$  Länge der Eingabekette
  - $C$  Constraintmenge
- Vergleich CFG-Parser (EARLEY 1970):  $\mathcal{O}(|G|^2 \cdot n^3)$ 
  - $|C| \approx |G|$  bei gleicher Sprache  $L$
  - für praktische Fälle gilt  $|G| \gg n$



## Parsing als Constraint-Satisfaction

- parallelisierte CDG-Version:  $\mathcal{O}(|C| + \log(n))$   
bei  $\mathcal{O}(n^4)$  Prozessoren (HELZEMAN UND HARPER 1992)
- sonstige Erweiterungen
  - Einbeziehung von lexikalischen Merkmalen an den Knoten
  - Worthypothesegraphen als Eingabestruktur
    - lexikalische Mehrdeutigkeit
    - Erkennungsunsicherheit



## Parsing als Constraint-Satisfaction

- keine direkte Kodierung von Existenzforderungen möglich
  - Modellierung obligatorischer Valenzen durch NEEDS-Rollen
- maximal binäre Constraints
  - Konsistenzforderungen sind auf lokale Strukturfragmente beschränkt
  - transitive Forderungen (z.B. Merkmalsprojektion) können nur approximiert werden
- gute Voraussetzungen für robuste Parsingstrategien



## Parsing als Constraint-Optimierung

- Weighted Constraint Dependency Grammar (MENZEL 1995, SCHRÖDER ET AL. 2000)
- Gewichtete Constraints
- Lösungsverfahren
- Lernen von Constraint-Gewichten
- Robustheitseigenschaften
- Integration flacher Analyse



## Gewichtete Constraints

- jedes Constraint erhält ein Gewicht
- Constraints sind prinzipiell verletzbar
- Gewichte modellieren den Grad zu dem eine Constraintverletzung akzeptiert werden kann
  - reduzierte Präferenz für Strukturhypothesen, die das betreffende Constraint verletzen
- symmetrischer Zugriff zu beiden Knoten an einer Dependenzkante
  - ursprüngliche DCG erlaubt nur den Zugriff zum abhängigen Knoten



## Gewichtete Constraints

- Gewichte
- $p(c) = 0$  "harte" Constraints: müssen immer erfüllt sein, z.B. zum Lizensieren der grundlegenden Baumstrukturen
- $0 < p(c) < 1$  "schwache" Constraints: können verletzt werden, solange keine bessere Alternative verfügbar ist
- $p(c) \ll 1$  verletzbare Wohlgeformtheitsbedingungen
- $p(c) \gg 0$  Defaults, Präferenzen, usw.
- $p(c) = 1$  sinnlos, neutralisiert das Constraint



## Gewichtete Constraints

- "harte" Constraints:  
{X:SYN}: det\_noun\_modification.1 : np : 0.0 :  
 $X \uparrow \text{cat} = \text{noun} \wedge X \downarrow = \text{det} \rightarrow X.\text{label} = \text{DET}$   
*a determiner can modify a noun with label DET*
- verletzbare Constraints:  
{X:SYN}: det\_noun\_modification.2 : np : 0.1 :  
 $X.\text{label} = \text{DET} \rightarrow X \downarrow \text{pos} < X \uparrow \text{pos}$   
*a determiner has to be left of its head noun*



## Gewichtete Constraints

- verletzbare Constraints (Fortsetzung):  
{X:SYN,Y:SYN}: np\_number\_agreement : np : 0.2 :  
 $X.\text{label} = \text{DET} \wedge Y.\text{label} = \text{AMOD} \wedge X \uparrow \text{pos} = Y \uparrow \text{pos} \rightarrow X \downarrow \text{number} = Y \downarrow \text{number}$   
*determiner and adjective within a noun phrase have to agree in number*
- {X:SYN,Y:SYN}: np\_inflection\_type : np : 0.2 :  
 $X.\text{label} = \text{DET} \wedge Y.\text{label} = \text{AMOD} \wedge X \uparrow \text{pos} = Y \uparrow \text{pos} \rightarrow X \downarrow \text{itype} = Y \downarrow \text{itype}$   
*the adjective inflection (weak vs. strong) depends on the type of determiner*



## Gewichtete Constraints

- präferentielle Constraints:
  - {X:SYN,Y:SYN}: subj\_obj\_preference : clause : 0.7 :
  - X.label = SUBJ  $\wedge$  Y.label = DOBJ  $\wedge$  X↑pos=Y↑pos  $\rightarrow$  X↓pos < Y↓pos
  - the subject tends to precede the object*
- {X:SYN}: short\_attachment\_preference : clause :
  - 0.9 \* (1 - 1/abs(X↓pos-X↑pos)) :
  - X.label = DOBJ  $\rightarrow$  abs(X↓pos-X↑pos) = 1
  - shorter attachments (e.g. for the direct object) are preferred*



## Gewichtete Constraints

- Gewichtsaggregation
  - Score: numerische Bewertung für einzelne Dependenzrelationen and Paare davon
  - Grad der bisher beobachteten Constraint-Verletzungen
    - $s_0 = 1$  initiale Bewertung
    - $s_i = s_{i-1} * p(c)$  Anwendung des  $i$ -ten Constraint mit Gewicht  $p(c)$
  - Zusammenfassen aller Kantenbewertungen eines Dependenzbaums ebenfalls durch Multiplikation



## Gewichtete Constraints

- widersprüchliche Constraints
  - $\rightarrow$  das Constraint-Satisfaction-Problem wird überbeschränkt
  - $\rightarrow$  auch die beste Lösung ist im Normalfall inkonsistent
- gewichtete Constraints
  - $\rightarrow$  "probabilistisches" CS-Problem (pCSP)
  - $\rightarrow$  Bestimmen der optimalen Lösung für das pCSP
  - $\rightarrow$  Constraint-Optimierung

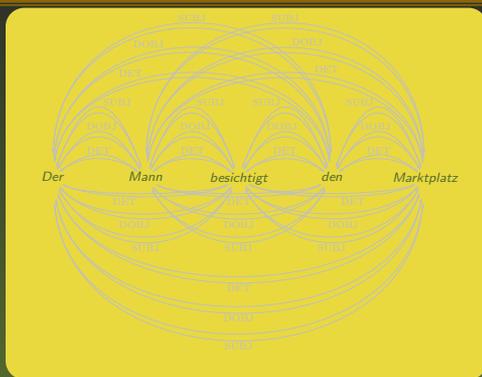


## Lösungsverfahren

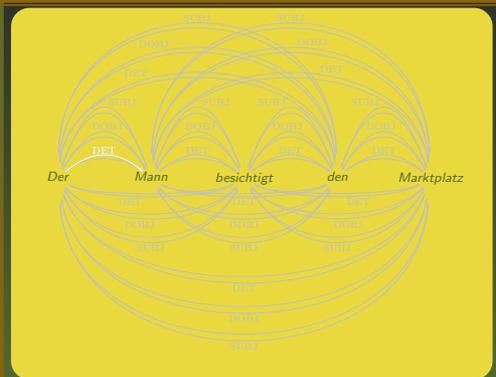
- Knoten- und Kantenkonsistenz
  - ungeeignet für gewichtete Constraints
- heuristisches Pruning
  - extrem aufwändige Konstruktion des initialen Hypothesenraums
  - keine geeigneten Heuristiken verfügbar
- Heuristische Suche
  - sukzessive Konstruktion der Lösung aus Teilstrukturen
- Transformation
  - Anwendung von lokalen Reparaturen zur Verbesserung der Gesamtbewertung



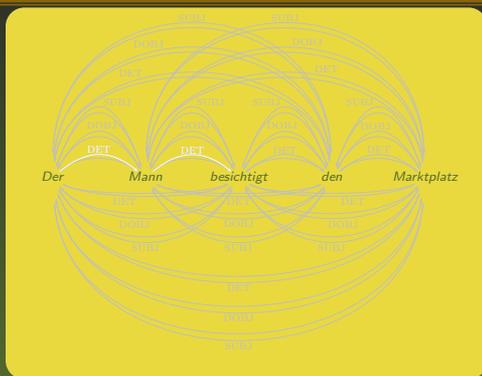
## Suche



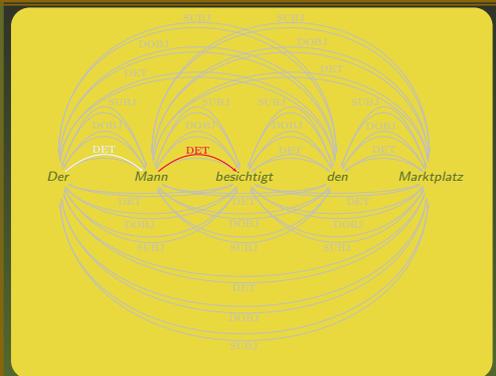
## Suche



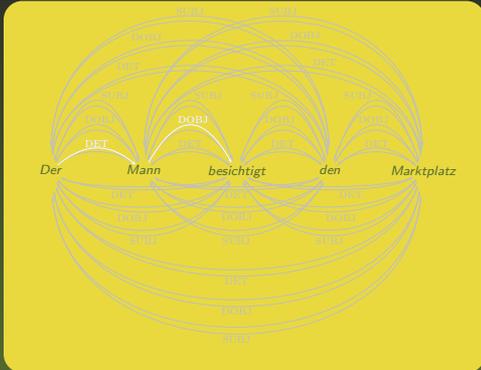
## Suche



## Suche

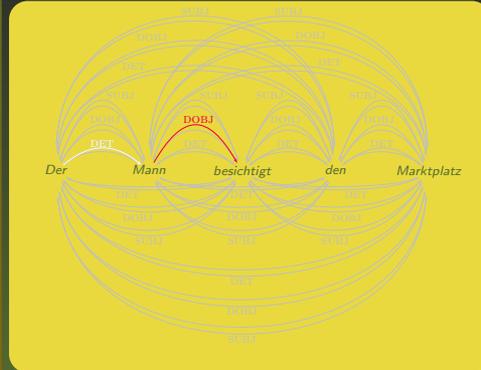


Suche



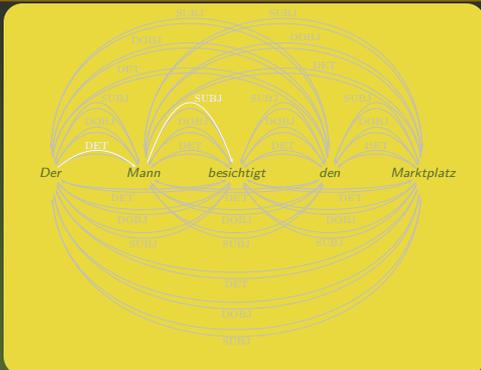
Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 95

Suche



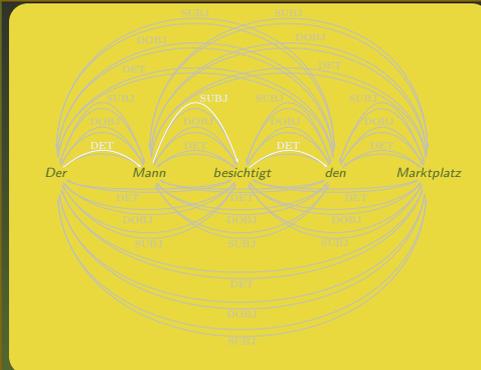
Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 96

Suche



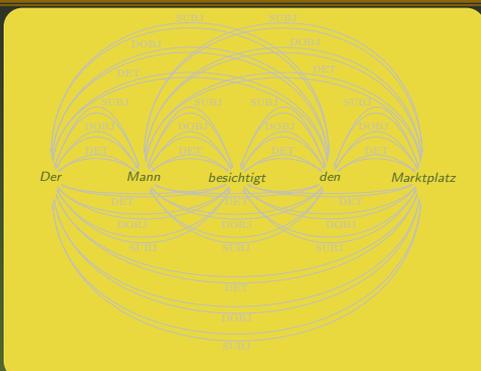
Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 97

Suche



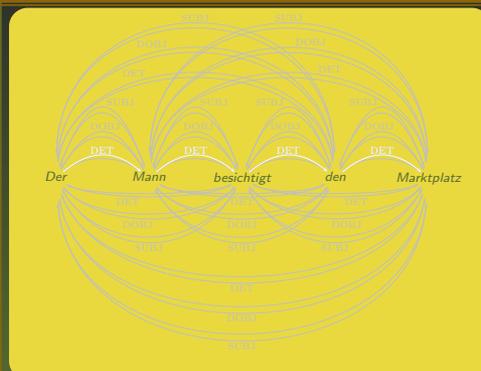
Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 98

Frobbling



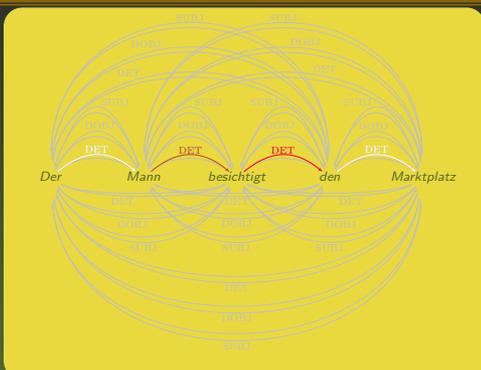
Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 99

Frobbling



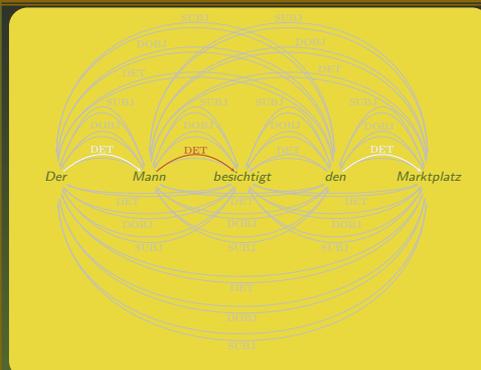
Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 100

Frobbling



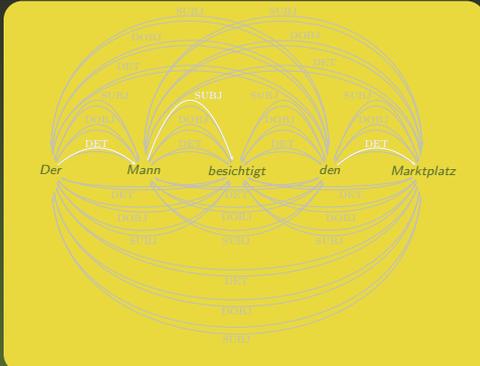
Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 101

Frobbling



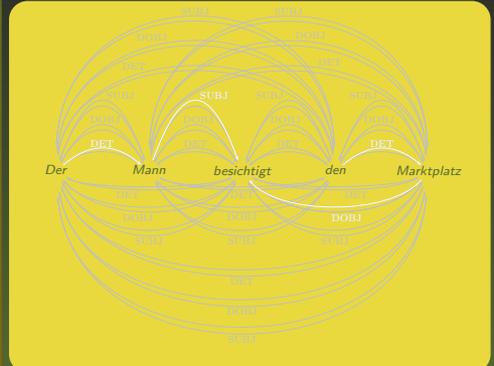
Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 102

## Frobbing



Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 103

## Frobbing



Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 104

## Frobbing

a)	Syntax	SUBJ	OBJ	...	b)	Syntax	SUBJ	OBJ	...
the <sub>1</sub>	det/2	-	-	...	the <sub>1</sub>	det/2	-	-	...
dog <sub>2</sub>	subj/3	-	-	...	dog <sub>2</sub>	subj/3	-	-	...
seems <sub>3</sub>	-	n_subj/2	-	...	seems <sub>3</sub>	-	n_subj/2	-	...
to <sub>4</sub>	to/5	-	-	...	to <sub>4</sub>	to/5	-	-	...
like <sub>5</sub>	inf/3	-	-	...	like <sub>5</sub>	inf/3	-	n_obj/7	...
the <sub>6</sub>	det/7	-	-	...	the <sub>6</sub>	det/7	-	-	...
cat <sub>7</sub>	subj/3	-	-	...	cat <sub>7</sub>	obj/5	-	-	...

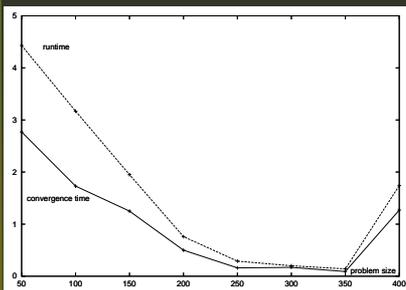
Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 105

## Frobbing

- Fehlgesteuerte Transformation mit perfektem Gedächtnis (FOTH 1998)
  - Gradientenabstiegsverfahren
  - Ausbruch aus lokalen Minima: sukzessiv komplexere Transformationsschritte → lokale Suche
  - heuristisch gesteuerte Tabu-Suche: Verwalten eines Schwellwerts für die Mindestbewertung von Teillösungen
  - schneller als best-first search für große Probleme

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 106

## Lösungsverfahren



Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 107

## GLS

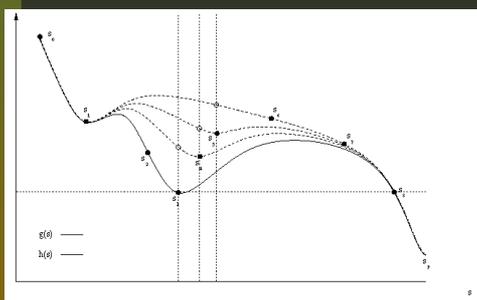
- GLS: guided local search
- Gedächtnis als Energie-Landschaft über dem Hypothesenraum (VODOURIS 1997)
- Transformation mit imperfektem Gedächtnis (SCHULZ 2000)
- augmentierte Bewertungsfunktion:

$$h(s) = g(s) + \lambda \cdot \sum_{i=1}^{|\mathcal{F}|} n_i \cdot I_i^f(s)$$

$f_i$  solution features

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 108

## GLS



Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 109

## GLS

- Nützlichkeit (utility): wo sollten die Gewichte der Bewertungsfunktion verändert werden?
- Lernstrategie:
  - hohe Kosten → hohe Nützlichkeit
  - wiederholte Reparaturversuche → geringe Nützlichkeit

$$util(s_*, f_i) = I_i^f \cdot \frac{c_i}{1 + n_i}$$

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 110

## GLS

- Experimente zur (grobkörnigen) Parallelisierung
  - hochgradig informiert: alle Prozesse kennen die beste bisher gefundene Lösung
  - zwei Strategien:
    - kooperativ: versuche immer die beste Lösung zu verbessern
    - kompetitiv: verbessere weiter die eigene Lösung, aber nutze die bessere zum Pruning des Hypothesenraums kompetitiver Modus war etwas besser



## Lösungsverfahren

- Vergleich

	Korrektheit	Vollständigkeit	Effizienz	Vorhersagbarkeit	Unterbrechbarkeit	Terminierung
Pruning	--	--	+/-	++	--	++
Suche	++	+	--	--	--	++
Transformation	+	-	-	+	++	-



## Der Parser für das Deutsche

- nur zwei Ebenen: Syntax, Referenz
- 700 handgeschriebene Constraints
- erlaubt bei Bedarf nichtprojektive Dependenzstrukturen
- weitgehend lexikalisiert:  
Valenzinformation für Verben und Präpositionen



## Beispielconstraints

- pro Satz ist nur ein Subjekt erlaubt

```
// 'Ich hatte viel Bekümmernis.'  
// *'Ich, ich, ich, ich hatte viel Bekümmernis.'  
{X!SYN/\Y!SYN} : 'doppeltes Subjekt' : uniq : 0.0 :  
  subsumes(Label, Subjekt, X.Label)  
-> "subsumes(Label, Subjekt, Y.Label);
```



## Beispiel-Constraints

- das Subject ist am wenigsten oblique

```
// 'Heute tanzt der König das Menuett.'  
// *'Heute tanzt das Menuett der König.'  
{X!SYN/\Y!SYN} : 'Subjekt-Position' : order : 0.9 :  
X.Label = SUBJ &  
subsumes(Label, Nominalobjekt, Y.Label)  
->  
X@from < Y@from |
```

```
// 'falls sich nicht _ein Investor_ findet'  
Y@cat = PRF |
```

```
(( Y@ cat = PWS | Y@ cat = PRELS | Y@ cat = PWAT |  
Y@ cat = PWAV | Y@ case = PRELAT |  
has(Y@ id, find_initial));
```



## Beispiel-Constraints

- Numerus-Kongruenz zwischen Subjekt und Verb

```
// 'Wir gehen in den Zoo.'  
// *'Wir gehe in den Zoo.'  
{X!SYN} : 'Subjekt-Numerus' : agree : 0.1 :  
  X.Label = SUBJ  
  >  
  phrasequotes(X@id) > phrasequotes(X'id) |  
  exists(X@number) & exists(X'number) &  
  (compatible(Features, X@number, X'number) |  
  // 'eine Menge Leute sind hier'  
  X'number = pl & (exists(X@set) | X@word = Art) |  
  // 'mehr als 300,000 Abonnenten gibt es bereits'  
  X@cat=PTS & exists(X@degree) |  
  // '300 Stück wurden verkauft'  
  X'number = pl & (X@sort=measure | X@sort=number) |  
  // '1992 war ein gutes Jahr'  
  X'cat = CARD & X'number = pl & X'value < 2100 |  
  // 'Das sind ganz üble Gesellen!'  
  (X@cat = PDS & (X@word = das | X@word = dies) |  
  X@word = es | X@word = was) &  
  (X'infiniteive = sein | subsumes(Features, Modalverb, X'cat)) |  
  // 'Yahoo und Amazon haben bestätigt, daß...'  
  X'number = pl & has(X'id, KON.Label, APP.KON);
```



## Das Lexikon

- Vollformen für alle geschlossenen Wortklassen
- 6,000 Verb- und 25,000 Substantivstämme
- Kompositumszerlegung
- lexikalische Templates für unbekannte Wörter



## Integration flacher Information

- DAUM, FOTH, MENZEL 2003

- zusätzliche Information wird "weich" integriert

- kann durch die Grammatik überstimmt werden

```
{X:SYN} : tagger : POS : [ pts(X@id) ] :  
  pts(X@id) = 1.0;
```

- Trigram-Tagger (TnT, BRANTS (2000))

- im multi-tagging Modus
- regel-basierte post hoc Fehlerkorrektur
- Beschleunigung: 55.1%, labelled Recall: 50.7 % → 73.7%

- Integration eines Chunkers (TreeTagger, SCHMID (1994))

- Beschleunigung: 11.9%, labelled Recall: 73.7 % → 76.2%



## Teilbaum-Parsing

- FOTH, MENZEL (2003)
- Problemzerlegung:
  - Zerlegen des Satzes in Fragmente (z.B. aufgrund der Interpunktion)
  - Parsing der Fragmente
  - Integration der Teilanalysen durch Transformation



## Teilbaum-Parsing

- 4 Schritte
  - POS-Tagging
  - Aufbau des Hypothesenraum
  - Teilsätze analysieren
  - Integration der Baumfragmente



## Evaluation

- 1000 Sätze aus dem NEGRA-Korpus (deutscher Zeitungstext)
- gleiche Daten wie von SCHIELEN (2004) verwendet
- Obermenge der Evaluationsdaten von DUBEY AND KELLER (2003)  
Satzlänge auf 40 Wortformen beschränkt
- Dependenzstrukturen automatisch aus den Phrasenstrukturen der Baumbank extrahiert (DAUM ET AL. 2004)



## Ergebnisse

Korpus	# Sätze	Anbindung	mit Label
alle Sätze	1000	89.0%	87.0%
<60 Wortformen	998	89.1%	87.1%
<40 Wortformen	963	89.7%	87.7%
<20 Wortformen	628	92.3%	90.1%
<10 Wortformen	300	93.4%	91.0%



## Vergleich

- DUBEY AND KELLER (2003)
  - rein stochastischer Parser (modifizierter Collins-Parser)
  - Abhängigkeit der Schwesterknoten vom Kopf
  - trainiert und evaluiert auf der NRGRA Baumbank
- SCHIELEN (2004)
  - stochastischer CFG-Parser
  - starke Unterstützung durch externes Lexikon
  - angereicherte Baumbankinformation zum Training



## Vergleich

	Testset	Konstituentenstrukturen	Dependenzstrukturen
		labelled	labelled
		precision/recall/f-score	precision/recall/f-score
DUBEY & KELLER	≤ 40	70.9%/71.3%/71.09%	—/—/76.08%
SCHIELEN	alle	—/—/69.36%	—/—/81.69%
FOTH ET AL.	alle	—/—/—	87.0%/87.0%/87.0%



## Verarbeitungszeit

Zeitlimit	genutzte Zeit	unlabelled acc.	labelled acc.
pro Satz			
600 seconds	68.0s	89.0%	87.0%
400 seconds	59.3s	88.7%	86.8%
200 seconds	44.9s	88.2%	86.2%
100 seconds	31.8s	87.1%	85.0%
50 seconds	21.6s	84.6%	82.3%



## Einfluss der Textsorte

Textsorte	Sätze	mittl.	unlabelled	labelled
		Länge	accur.	accur.
Trivilliteratur	9547	14	93.1%	91.1%
Grundgesetz	1145	19	88.8%	86.7%
Verbmobil-Dialoge	1316	8	90.3%	86.3%
Heise Ticker	1894	23	89.8%	88.1%
Roman des 19. Jhd.	68	34	78.0%	75.4%



## Relativer Einfluss der Constraints

Class	Purpose	Example	Importance
agree	rection and agreement	subjects have nominative case	1.02
cat	category cooccurrence	prepositions do not modify each other	1.13
dist	locality principles	prefer the shorter of two attachments	1.01
exist	valency	finite verbs must have subjects	1.04
init	hard constraints	appositions are nominals	3.70
lexical	word-specific rules	"entweder" requires following "oder"	1.02
order	word-order	determiners precede their regents	1.11
pos	POS tagger integration	prefer the predicted category	1.77
pref	default assumptions	assume nominative case by default	1.00
proj	projectivity	disprefer nonprojective coordinations	1.09
punc	punctuation	subclauses are marked with commas	1.03
root	root subordinations	only verbs should be tree roots	1.72
sort	sortal restrictions	"sein" takes only local predicatives	1.00
uniq	label cooccurrence	there can be only one determiner	1.00
zone	crossing of marker words	conjunctions must be leftmost dependents	1.00

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 127

## Relativer Einfluss der Constraints

Class	Purpose	Example	Importance
init	hard constraints	appositions are nominals	3.70
pos	POS tagger integration	prefer the predicted category	1.77
root	root subordinations	only verbs should be tree roots	1.72
cat	category cooccurrence	prepositions do not modify each other	1.13
order	word-order	determiners precede their regents	1.11
proj	projectivity	disprefer nonprojective coordinations	1.09
exist	valency	finite verbs must have subjects	1.04
punc	punctuation	subclauses are marked with commas	1.03
agree	rection and agreement	subjects have nominative case	1.02
lexical	word-specific rules	"entweder" requires following "oder"	1.02
dist	locality principles	prefer the shorter of two attachments	1.01
pref	default assumptions	assume nominative case by default	1.00
sort	sortal restrictions	"sein" takes only local predicatives	1.00
uniq	label cooccurrence	there can be only one determiner	1.00
zone	crossing of marker words	conjunctions must be leftmost dependents	1.00

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 128

## Lernen der Constraint-Gewichte

- Lernen probabilistischer Dependenzgrammatiken  
CARROLL AND CHARNAK (1992), COLLINS (1997), EISNER (1996), GRINBERG, LAFFERTY, SLEATOR (1996)
- Problem: zu viele Parameter  
→ sparse data problem
- Constraints sind Abstraktionen über (Paare von) Dependenzrelationen  
→ weniger Parameter

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 129

## Lernen der Constraint-Gewichte

- alternative Idee: Anwendung von genetischen Algorithmen
- Verwendung der Constraint-Gewichte als Gene
- genetische Veränderungen
  - Kreuzung (crossover)
  - Mutation
- Selektion der besten Individuen aufgrund eines Fitness-Maßes

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 130

## Lernen der Constraint-Gewichte

- Kreuzung mit Genaustausch: Die Gene zweier Individuen werden "gemischt".  

$$\left. \begin{matrix} .1 & .2 & .3 & .4 \\ .8 & .6 & .3 & .2 \end{matrix} \right\} \rightarrow MC \rightarrow \left\{ \begin{matrix} .1 & .6 & .3 & .4 \\ .8 & .2 & .3 & .2 \end{matrix} \right.$$
- Kreuzung mit arithmetischer Genmittelung: Der Nachfahre erhält als Gen das gewichtete arithmetische Mittel der beiden Eltern  

$$\left. \begin{matrix} .1 & .2 & .3 & .4 \\ .8 & .6 & .3 & .2 \end{matrix} \right\} \rightarrow AC \rightarrow .45 \ .4 \ .3 \ .3$$

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 131

## Lernen der Constraint-Gewichte

- Mutation  

$$.1 \ .2 \ .3 \ .4 \rightarrow M \rightarrow .1 \ .25 \ .3 \ .2$$
  - relative Mutation: Ein Gewicht wird zufällig vergrößert bzw. verkleinert, abhängig von seinem gegenwärtigen Wert.
  - absolute Mutation: Ein Gewicht wird zufällig gesetzt.
  - Nullmutation: Das Gewicht wird auf den Wert null gesetzt.

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 132

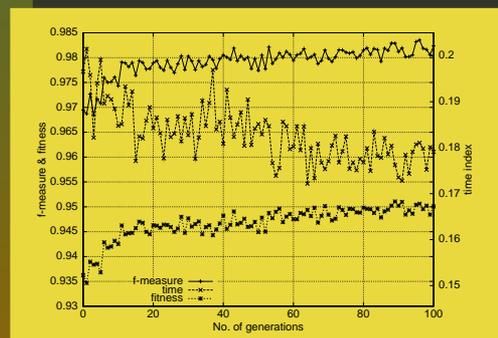
## Lernen der Constraint-Gewichte

- konstante Populationsgröße (30)
- gemischtes Fitnessmaß: 0.8 Akkuratheit, 0.2 Verarbeitungszeit
- Training: 220 Sätze, Test: 90 Sätze
- erstes Experiment: Kann eine manuell optimierte Grammar weiter verbessert werden?  

$$p_{ac} = 0.25, p_{mc} = 0.25, p_r = 0.02, p_a = 0.002 \text{ and } p_z = 0.002.$$

Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 133

## Lernen der Constraint-Gewichte



Wolfgang Menzel: Sprachorientierte KI: Syntax und Parsing – p. 134

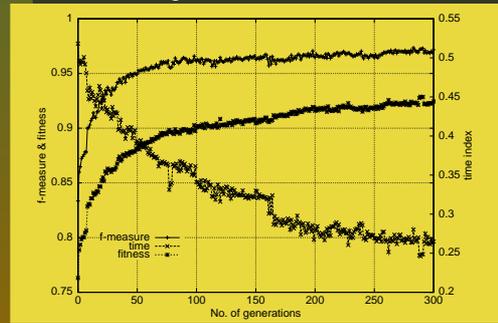
## Lernen der Constraint-Gewichte

- zweites Experiment: Können die Gewichte ausgehend von einer zufälligen Initialisierung gelernt werden?  
 $p_{ac} = 0.25$ ,  $p_{mc} = 0.25$ ,  $p_r = 0.02$ ,  $p_a = 0.0$  and  $p_s = 0.0$ .
- Kleine Parametermodifikation und zufällige Störungen um lokale Minima zu überwinden



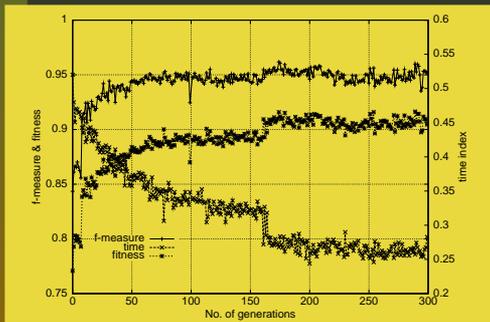
## Lernen der Constraint-Gewichte

Resultate auf den Trainingsdaten



## Lernen der Constraint-Gewichte

Resultate auf den Testdaten



## Robustheitseigenschaften

- Robustheit gegen ...
  - ... fehlerhaften Input
  - ... Zeitdruck
  - ... unvollständige Eingaben



## Robustheit gegen fehlerhaften Input

- strikte Trennung zwischen dem Raum möglicher Strukturbeschreibungen und der Grammatik
  - Constraints sind vollständig unabhängig
  - praktisch beliebigem Input wird eine Strukturbeschreibung zugeordnet



## Robustheit gegen fehlerhaften Input

- FOTH, MENZEL, SCHRÖDER 2004
- künstlich gestörte Sätze
  - geänderte Flexion (50%)
  - Löschen des Wortes (30%)
  - Austausch benachbarter Schwesterkonstituenten (20%)
- pro Durchlauf im Mittel jedes 10. Wort verändert

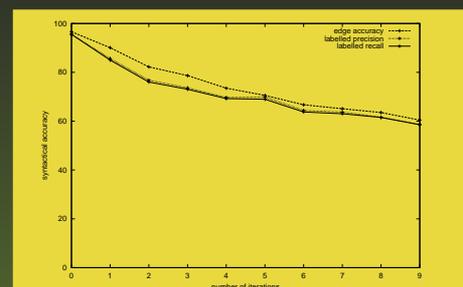


## Robustheit gegen fehlerhaften Input

- 10 Durchläufe:
  - Dann lassen Sie uns doch noch einen Termin ausmachen
  - Dann lassen Sie uns  $\emptyset$  noch einen Termin ausmachen
  - Dann lasse Sie uns noch einen Termin ausmachen
  - Dann lasse Sie noch einen Termin ausmachen **uns**
- Auswirkung der Fehlertypen
  - Störung der Konstituentenreihenfolge (88.1% Akkuratheit)
  - geänderte Flexion (83.1% Akkuratheit)
  - Löschen einer Wortform (78.7% Akkuratheit)



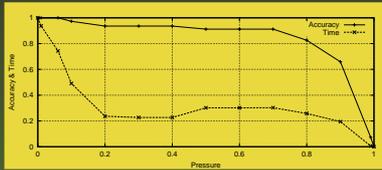
## Robustheit gegen fehlerhaften Input



## Robustheit gegen Zeitdruck

- Suche: Beschleunigung durch Constraintverschärfung

$$p'(c) = \begin{cases} 0 & p(c) < \vartheta \\ p(c) & \text{else} \end{cases}$$



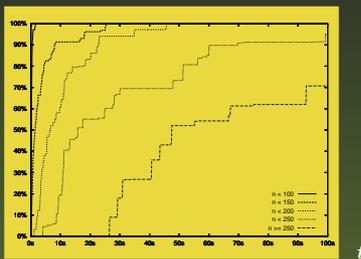
## Robustheit gegen Zeitdruck

- Transformation: Schwellwert für die minimale Qualität einer Lösung
  - Reduktion der mittleren Laufzeit um den Faktor 2...3
- Transformation: Qualität einer Lösung wächst mit der Zeit
  - starke Anytime-Eigenschaft: Berechnung kann zu (fast) beliebigen Zeitpunkten unterbrochen werden



## Robustheit gegen Zeitdruck

- Transformation: Qualität einer Lösung wächst mit der Zeit



starke Anytime-Eigenschaft: Unterbrechung zu (fast) beliebigen Zeitpunkten möglich



## Robustheit gegen unvollständigen Input

- Voraussetzung für inkrementelle Analyse
- Coling 2000

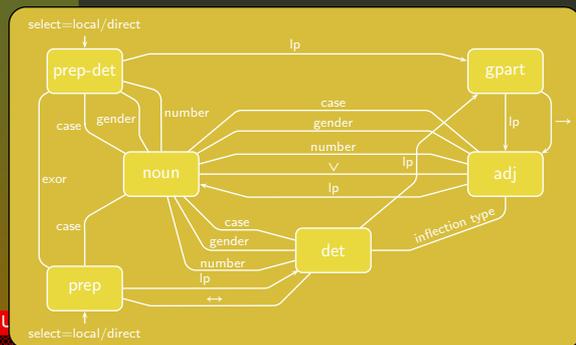


## Fehlersensitives Parsing

- MENZEL 1992, MENZEL/SCHRÖDER 1998
- modell-basierte Diagnose
  - Simulation von Constraint-Verletzungen
  - hohe Präzision auch bei transitiven Forderungen (Kongruenz, Projektion)
  - verschiedene Fehlerperspektiven (Regeln vs. Fakten)
  - aber: Struktur muss bekannt sein



## Fehlersensitives Parsing



## Fehlersensitives Parsing

- WCDG:
  - Struktur kann ermittelt werden
  - Constraintverletzungen können als Diagnosen interpretiert werden
  - aber: transitive Forderungen werden nicht korrekt behandelt



## Fehlersensitives Parsing

- Kopplung der modellbasierten Diagnose mit dem WCDG-Parser
  - WCDG-Parser
    - ermittelt Strukturbeschreibung
    - diagnostiziert einige Fehlertypen (fehlendes oder überflüssiges syntaktisches Material, Anordnungsfehler)
  - Konstruktion eines Bedingungsnetzes für die Modellbasierte Diagnose aus dem Syntaxbaum
  - modellbasierte Diagnose
    - vollständige Berücksichtigung transitiver Bedingungen
    - Berücksichtigung verschiedener Fehlerperspektiven

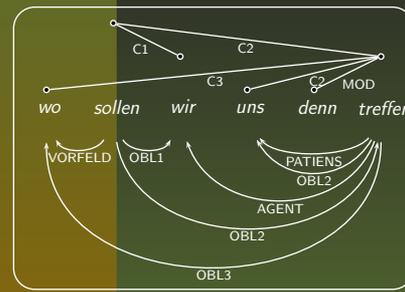


## Fehlersensitives Parsing

- höhere Robustheit durch Mehrebenenanalyse
- Syntax
  - Valenzmöglichkeit und -notwendigkeit, Rektion, Anordnung, Kongruenz, ...
- Semantik
  - Thematische Rollen, selektionale Restriktionen, ...
- Mapping
  - syntaktische Funktionen ↔ thematische Rollen
- TRUESWELL, TANENHAUS, GARNSEY (1994):
  - modular representation, but integrated processing



## Fehlersensitives Parsing



Repräsentationsebenen:  
 syntaktische Struktur  
 Funktor-Argument Struktur  
 Informationsstruktur

weitere zu ergänzende Ebenen:  
 Prosodie  
 Weltwissen  
 referentielle Beziehungen

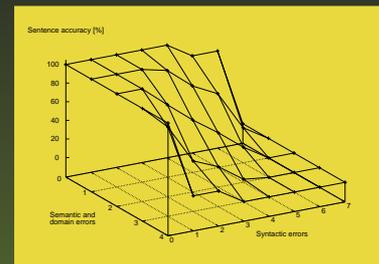


## Fehlersensitives Parsing

- Abgleich von Konflikten zwischen der syntaktischen Form und semantischen Erwartungen
  - semantische Constraints unterstützen die syntaktische Disambiguierung
    - plausibilitätsgesteuerte Verarbeitung von fehlerhaften Äußerungen
    - dynamische Integration von Kontextwissen
  - syntaktische Evidenz kann die semantischen Erwartungen überschreiben
    - Kommunikation von neuem Inhalt
    - Erwartungsverletzungen erzeugen Aufmerksamkeit



## Fehlersensitives Parsing



## Fehlersensitives Parsing

- Verarbeitung spiegelt den kognitiven Aufwand wider:  
 Ignorieren < Detektieren < Lokalisieren < Diagnostizieren < Korrigieren
- Unterstützung der Analyse durch Weltwissen möglich
  - propositional repräsentierte graphische Information
  - propositionaler Inhalt einer einfachen Geschichte

