TDG - ein deklarativer Grammatikformalismus für Dependenzgrammatik

Ralph Debusmann
Universität des Saarlandes
Computerlinguistik
rade@coli.uni-sb.de

Montag, 10. Dezember 2001

Kontextfreie Grammatik

- Syntax natürlicher Sprache: Formalismen bauen meist auf kontextfreier Grammatik (CFG) auf (Chomsky 57)

- Beispiele: GB (Chomsky 86), HPSG (Pollard/Sag 94), LFG (Kaplan/Bresnan 82)

- geeignet für Englisch (relativ unflexible Wortstellung), aber problematisch für die meisten anderen Sprachen (z.B. germanische Sprachen, slavische Sprachen, Hindi etc.)

Dependenzgrammatik

- alternativer Ansatz, besser geeignet für Sprachen mit freier Wortstellung
- vor allem im Ostblock vorangetrieben (Prager Schule: Sgall et al 86, Moskauer Schule: Melc'uk 1987)
- Probleme:
  - mangelnde Akzeptanz im anglo-amerikanisch geprägten linguistischen Mainstream
  - oft mangelnde Deklarativität und schlechte Algorithmisierung

Topologische Dependenzgrammatik (TDG)

- neuer Dependenz-basierter Grammatikformalismus
- beschrieben in Duchier/Debusmann 01 (ACL-Papier), Debusmann 01 (Diplomarbeit)
- weil Dependenz-basiert: elegante Behandlung von Sprachen mit freier Wortstellung möglich
- deklarativ
- effizientes Parsing als Lösen von Mengenconstraints in Mozart-Oz

Überblick

1. Dependenzbäume
2. Constraint-basiertes Dependenz-Parsing
3. Wortstellung
4. Stand der Kunst

Dependenzbäume (verglichen mit CFG-Parsbäumen)

- CFG
  - CFG-Analyse = geordneter Baum
  - Knoten mit syntaktischen Kategorien beschriftet
  - terminale und nicht-terminale Knoten
- DG
  - DG-Analyse = ungeordneter Baum
  - Ordnung durch zusätzliche Constraints
  - Kanten mit grammatischen Rollen beschriftet (mehrere Kanten mit gleicher Beschriftung möglich)
  - nur terminale Knoten (Knoten = Auftreten von Wörtern)

Constraint-basierte Beschreibung von Dependenzbäumen

- Denys Duchier (1999, 2000, 2001): neue Formalisierung von Dependenz-Parsing mithilfe von Constraints auf endlichen Mengen

- dabei: Dependenzbäume als gerichtete Graphen betrachtet

- Graphen charakterisiert durch Funktionen mothers, daughters, up, equp, down und eqdown von Knoten auf Knotenmengen und Knotenmenge $Roots$

- Constraints auf diesen Funktionen drücken Wohlgeformtheitsbedingungungen für Dependenzbäume aus

Constraints für wohlgeformte Dependenzbäume

- globale Constraints: Baumheit, Lexikon-Constraint
- lexikalisierte Constraints: Akzeptanz, Valenz

Baumheits-Constraint

- deklarative Formulierung:
  1. Jeder Knoten hat höchstens eine eingehende Kante.
  2. Ein Knoten (die Wurzel) hat keine eingehende Kante.
  3. Es gibt keine Zykel.

- Reduktion auf Mengenconstraints
  1. $$\forall w \in W : \ |\mathsf{mothers}(w)| \leq 1$$
  2. $$|Roots| = 1 \wedge W = Roots \uplus \bigcup_{w \in W} \mathsf{daughters}(w)$$
  3. $$\forall w \in W : \ w \notin \mathsf{down}(w)$$

Lexikon

- $Lexicon =$ Menge von Lexikoneinträgen e
- Record-Signatur:

$$\begin{bmatrix} \text{in} & : & 2^R \\ \text{out} & : & 2^R \end{bmatrix}$$

- Zugriff auf lexikalische Attribute in Funktionsnotation geschrieben, z.B. $\text{in}(e)$

Lexikon-Constraint

- Funktion lex bildet jedes Wort auf Menge von möglichen Lexikoneinträgen ab (lexikalische Mehrdeutigkeit):

$$\mathsf{lex}: \ W \rightarrow 2^{Lexicon}$$

- Funktion entry bildet jedes Wort auf genau einen Lexikoneintrag ab:

$$\mathsf{entry}: \ W \rightarrow Lexicon$$

- Lexicon-Constraint: jedem Knoten wird genau ein Lexikoneintrag aus der Menge der möglichen zugeordnet

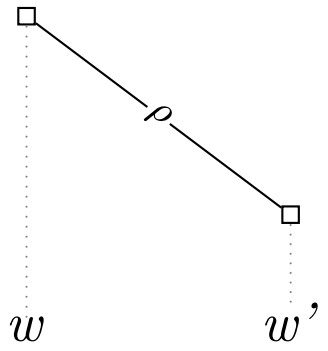
$$\forall w \in W: \ \mathsf{entry}(w) \in \mathsf{lex}(w)$$

Constraints für wohlgeformte Dependenzbäume

- bisher: globale Constraints: Baumheit, Lexikon-Constraint
- lexikalisierte Constraints: Akzeptanz, Valenz. Benutzen lexikalische Attribute in ihrer Formulierung

Akzeptanz-Constraint

- Constraint auf eingehenden Kanten:
  Jeder Knoten muss die Beschriftung seiner eingehenden Kante akzeptieren.

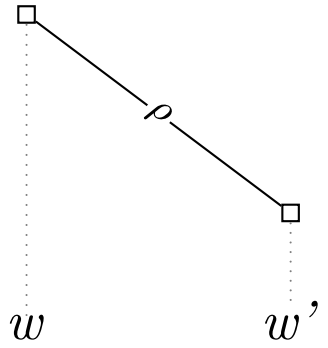


- formal:

$$\forall w, w' \in W, \rho \in Roles:$$
$$w{-}\rho{\rightarrow}w' \;\Rightarrow\; \rho \in \mathsf{in}(\mathsf{entry}(w'))$$

Valenz-Constraint

- Constraint auf ausgehenden Kanten:
  Jede ausgehende Kante eines Knotens muss durch dessen Valenz lizensiert sein.

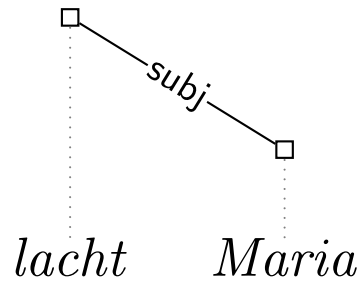


- formal:

$$\forall w, w' \in W, \rho \in Roles:$$
$$w - \rho \rightarrow w' \;\Rightarrow\; \rho \in \mathsf{out}(\mathsf{entry}(w))$$

Lexikalisierte Constraints: Beispiel

- Beispiel-Dependenzbaum:



- Lexikoneinträge-Zuordnung:

$$\text{entry}(lacht) = \begin{bmatrix} \text{in} & : & \{\} \\ \text{out} & : & \{\text{subj}\} \end{bmatrix}$$

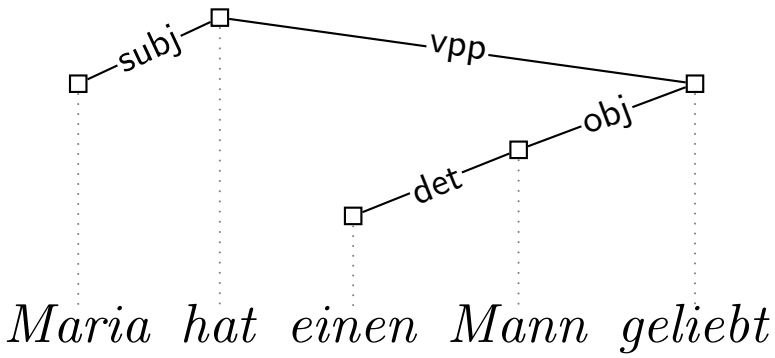
$$\text{entry}(Maria) = \begin{bmatrix} \text{in} & : & \{\text{subj}, \text{obj}\} \\ \text{out} & : & \{\} \end{bmatrix}$$

- Akzeptanz:
  $lacht$–subj→$Maria \Rightarrow \text{subj} \in \text{in}(\text{entry}(Maria))$

- Valenz:
  $lacht$–subj→$Maria \Rightarrow \text{subj} \in \text{out}(\text{entry}(lacht))$

Wortstellung

- bislang: Dependenzbäume ungeordnet
- untenstehender, ungeordneter Dependenzbaum lizensiert so $5! = 120$ Wortabfolgen:



- davon aber nur sieben erlaubt:

1. *Maria hat einen Mann geliebt.*
2. *Einen Mann hat Maria geliebt.*
3. *Geliebt hat Maria einen Mann.*
4. ?*Geliebt hat einen Mann Maria.*
5. *Einen Mann geliebt hat Maria.*
6. *Hat Maria einen Mann geliebt?*
7. ?*Hat einen Mann Maria geliebt?*

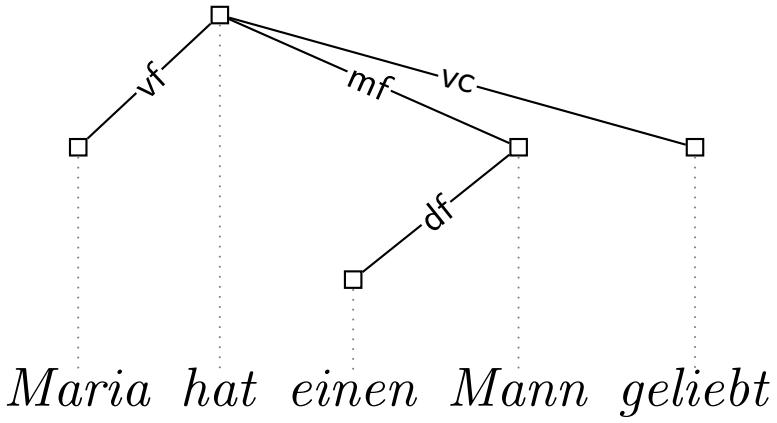
Wortstellung: die Theorie der topologischen Felder

- beschreibt deutsche Wortstellung
- teilt Satz in zusammenhängende Wortketten ein und nennt diese *topologische Felder*
- topologische Felder: Vorfeld, linke Satzklammer, Mittelfeld, rechte Satzklammer, Nachfeld.
- Beispielanalysen:

Vorfeld	(	Mittelfeld	)	Nachfeld
Maria	*hat*	*einen Mann*	*geliebt.*	
Einen Mann	*hat*	*Maria*	*geliebt.*	
Geliebt	*hat*	*Maria einen Mann.*		
Geliebt	*hat*	*einen Mann Maria.*		
Einen Mann geliebt	*hat*	*Maria.*		
	Hat	*Maria einen Mann*	*geliebt?*	
	Hat	*einen Mann Maria*	*geliebt?*	

Topologiebäume

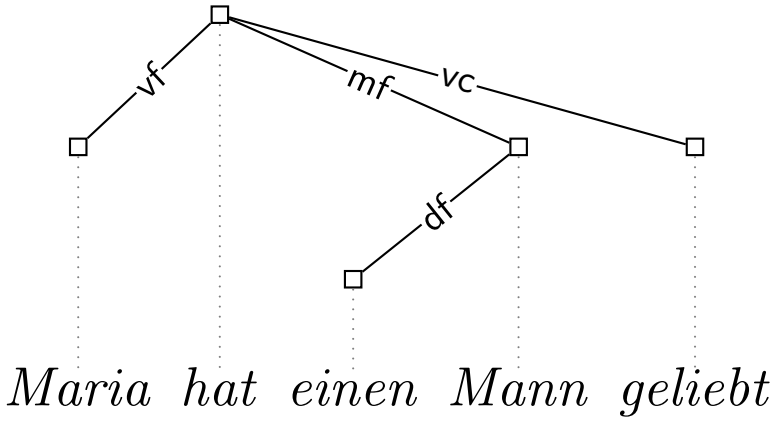
- repräsentieren topologische Struktur von Sätzen
- schließen unerlaubte Wortabfolgen aus
- Beispielanalyse:



- Felder vf, mf und vc entsprechen Vorfeld, Mittelfeld und rechter Satzklammer.

Allgemeiner Constraint: Ordnung

- Beispielanalyse:



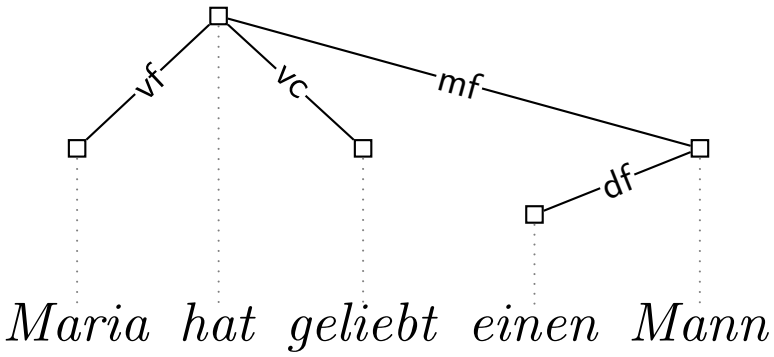
- globale Ordnung auf der Menge der Felder:

$$\text{vf} \prec \text{mf} \prec \text{vc}$$

- Ordnungs-Constraint besagt, dass Töchter im Baum analog zu dieser globalen Ordnung geordnet sein müssen:

$$Maria \prec Mann \prec geliebt$$

Verletzung des Ordnungs-Constraints



- nicht lizensiert. Globale Ordnung auf der Menge der Felder:

$$\text{vf} \prec \text{mf} \prec \text{vc}$$

- hier ist *geliebt* aber im vc vor *Mann* im mf.

Beziehung zwischen Dependenz- und Topologiebäumen

- TDG-Analyse besteht aus je einem Dependenz- und einem Topologiebaum
- beide haben die gleiche Knotenmenge aber verschiedene Kantenmengen
- allgemeiner Constraint des Kletterns setzt beide Bäume miteinander in Beziehung

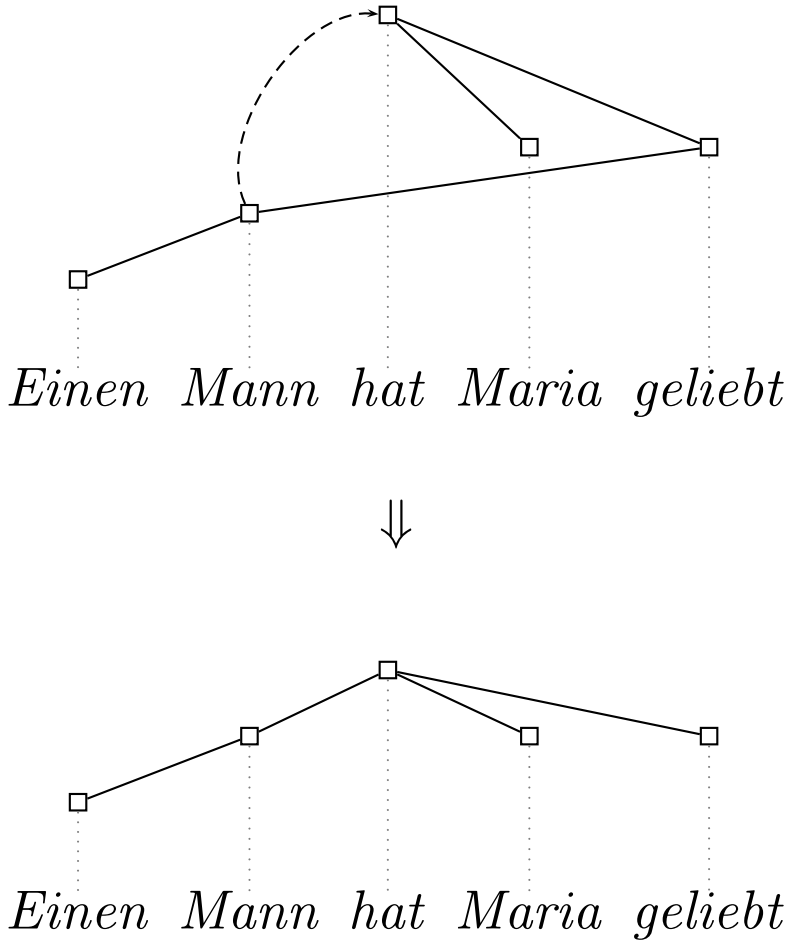
Kletter-Constraint

- Klettern: Knoten können im Topologiebaum im Vergleich zum Dependenzbaum hochklettern
- damit Topologiebaum flacher als entsprechender Dependenzbaum
- formal:

$$w \rightarrow_{\mathrm{TOP}} w' \;\Rightarrow\; w \rightarrow_{\mathrm{DEP}} w_1 \ldots w_n \rightarrow_{\mathrm{DEP}} w'$$

Kletter-Constraint: Beispiel

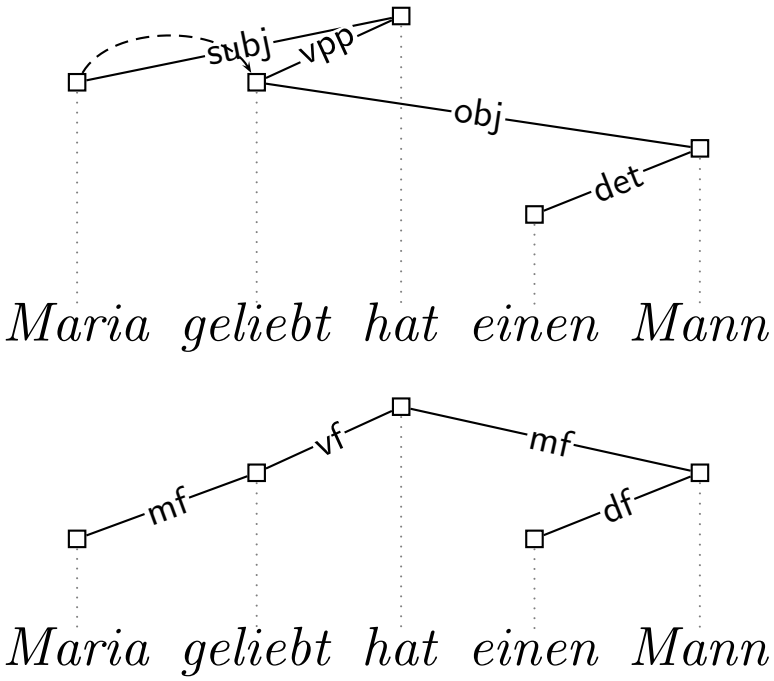
- Beispiel:



- Instanziierung des Kletter-Constraints:

$$hat \rightarrow_{\mathrm{TOP}} Mann \;\Rightarrow\; hat \rightarrow_{\mathrm{DEP}} geliebt \rightarrow_{\mathrm{DEP}} Mann$$

Verletzung des Kletter-Constraints



- nicht lizensiert: *Maria* nicht geklettert, sondern zur Schwester *geliebt* bewegt

Behandelte Phänomene

- Fragen, Aussagesätze, Nebensätze

  Gibt Maria dem Mann einen Korb?
  Maria gibt dem Mann einen Korb.
  Ich glaube, dass Maria dem Mann einen Korb gibt.

- Relativsätze (mit Extraposition, Rattenfänger-Konstruktionen)

  Maria hat einen Mann, der lacht, gefunden.
  Maria hat einen Mann gefunden, der lacht.
  Maria hat einen Mann gefunden, mit dem sie lacht.

- Verbalkomplex-Phänomene:

  (dass) Maria den Mann lachen gesehen hat.
  (dass) Maria den Mann lachen hat sehen.

- Scrambling, Intraposition, Fronting etc.

Was geht alles?

- Grammatik fürs Deutsche, behandelt viele sehr schwierige Phänomene
- kleines Grammatikfragment fürs Holländische
- TDG-Parser in Mozart-Oz, trotz fehlender Optimierung effizient: 20-Wort-Satz in 700ms, 50-Wort-Satz in 5s auf 700MHz-Maschine
- TDG-Entwicklungsumgebung in Mozart-Oz, mit statisch getypter Grammatik-Eingabesprache und graphischer Benutzeroberfläche
- TDG-Parser schon in anderen Projekten verwendet: NEGRA (automatische Grammatik-Generierung aus Korpus), Softwareprojekt "Computerspiel" (Englische Grammatik)

Was fehlt noch?

- Verbesserung der TDG-Entwicklungsumgebung
- Verbesserung des Grammatikformalismus: z.B. elegantere Spezifikation des Relativsatz-Constraints
- Verbesserung der Abdeckung: Phänomene und Breite
- Entwicklung einer Morphologie-Schnittstelle
- Entwicklung einer Syntax-Semantik-Schnittstelle (CHORUS)
- Ellipsen und Koordination (CHORUS)
- Einbindung von Präferenzen (CHORUS)