Emmy-Noether-Nachwuchsgruppe – Eine lexikalisierte Baumgrammatik für ein Fragment des Deutschen

Laura Kallmeyer, Timm Lichte und Wolfgang Maier

SFB 441
Universität Tübingen

9. Februar 2007

Teil I

Inhalt und Ziele des Projekts

Projektziele

Das Projekt umfasst zwei Teilziele:

- Implementierung eines Grammatikfragments für das Deutsche (Syntax und Semantik)
- Implementierung eines Parsers, der die Grammatik verarbeitet und für einen Eingabesatz eine Analyse liefert

Architekturentscheidungen

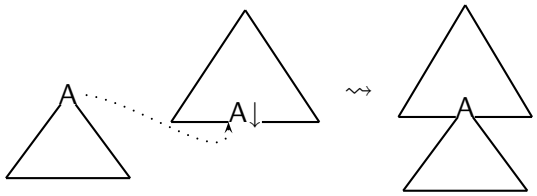
- Verwendung eines schwach kontext-sensitiven Formalismus, um gute Verarbeitbarkeit der Grammatik im allgemeinen Fall zu garantieren.
- Verwendung einer lexikalisierten Grammatik (Vorteile für Parsing).
- Verwendung eines Baumersetzungssystems. D.h., die Grammatik ist eine Menge von Bäumen, aus denen größere Bäume erzeugt werden, indem einzelne Knoten durch weitere Bäume aus der Grammatik ersetzt werden.
- Verwendung einer kompositionellen Syntax-Semantik Schnittstelle, die so definiert ist, dass die positiven Komplexitätseigenschaften des Formalismus erhalten bleiben.

$\Rightarrow$ Ausgangspunkt: Lexikalisierte Baumadjunktionsgrammatiken

LTAG: Adjunktion und Substitution

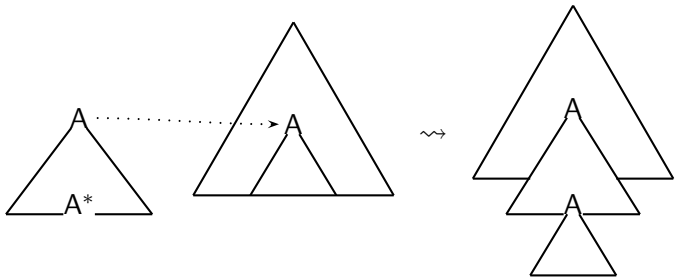
Lexicalized Tree Adjoining Grammars (LTAG): Menge von lexikalisierten elementaren Bäumen mit zwei Operationen:

Substitution:



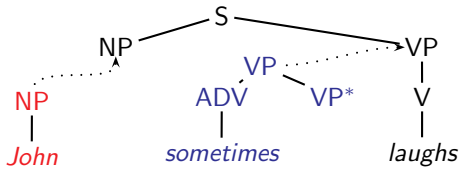
LTAG: Adjunktion und Substitution

Adjunktion:

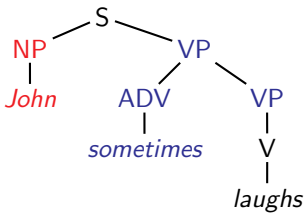


LTAG: Adjunktion und Substitution

(1) John sometimes laughs



abgeleiteter Baum:



LTAG Semantik

Kallmeyer & Romero (to appear): Entwicklung einer Syntax-Semantik Schnittstelle für LTAG.

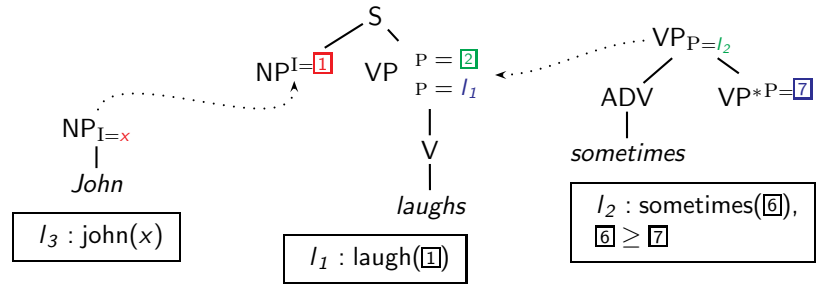
Idee: Jedem elementaren Baum ist folgendes zugeordnet:

- Eine Menge von Ty2 Ausdrücken und von Constraints, die Teilausdruckbeziehungen beschreiben, und
- eine Merkmalsstruktur, die charakterisiert, welche Argumente gefüllt werden müssen, welche Elemente als Argumente für andere Elementarbäume zur Verfügung stehen, und wie das Skopusverhalten ist.
  Die Merkmale sind Positionen im elementaren Baum zugeordnet.

LTAG Semantik

- Keine funktionale Applikation im eigentlichen Sinne. Stattdessen werden, abhängig von den vorgenommenen Substitutionen und Adjunktionen, Unifikationen zwischen den Merkmalsstrukturen vorgenommen, die dazu führen, dass Argumentvariablen Werte zugewiesen werden.
- Das Ergebnis kann unterspezifiziert sein. Daher wird im Anschluss eine Disambiguierung vorgenommen, die die eigentlichen Lesarten berechnet.

LTAG Semantik



$\rightsquigarrow$ l_3 : john(x), l_1 : laugh(x), l_2 : sometimes(6), 6 $\geq$ l_1

Zentrale Problematik des Projekts

LTAG ist nicht ausdrucksstark genug, um Wortstellungsphänomene im Deutschen zu erfassen.

$\Rightarrow$ Zentrale Frage: Welche LTAG Erweiterung soll man wählen?

Abwägen zwischen

- benötigter Ausdrucksstärke auf der einen Seite, und
- gewünschten Komplexitätseigenschaften auf der anderen Seite.

Teil II

LTAG fürs Deutsche

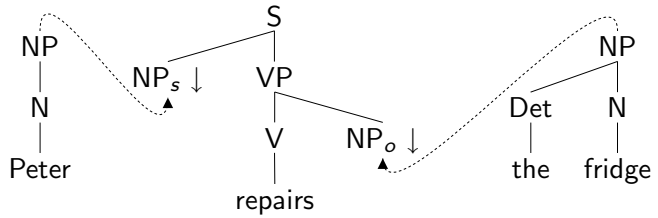
Vorgedanken

- **Wie** muss der LTAG-Formalismus angepasst werden, um dem deutschen Satzbau gerecht zu werden?
- **Wie stark?** So, dass die Grundeigenschaften der LTAG bewahrt bleiben:
  - polynomielle Parsebarkeit;
  - endliche Menge von Merkmalsstrukturen als Knotenbeschreibungen;
  - 2 Verknüpfungoperationen: Substitution und Adjunktion;
  - Abhängigkeiten in einem Lexikoneintrag (extended domain of locality).
- **Auf welche Art und Weise?** Wir werden sehen...

TAG pur

Fixierte SVO-Wortstellung im Englischen:

(1) Peter repairs the fridge. ≠ The fridge repairs Peter.



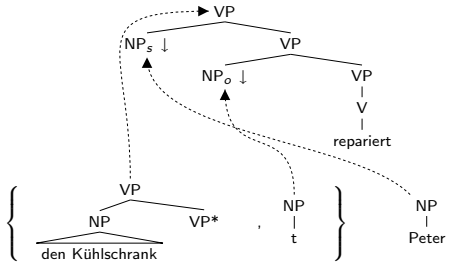
Freie Wortstellung im Deutschen

(2) Peter repariert den Kühlschrank.
Den Kühlschrank repariert Peter.

(3) dass Peter den Kühlschrank repariert.
dass den Kühlschrank Peter repariert.

Freie Wortstellung im Deutschen

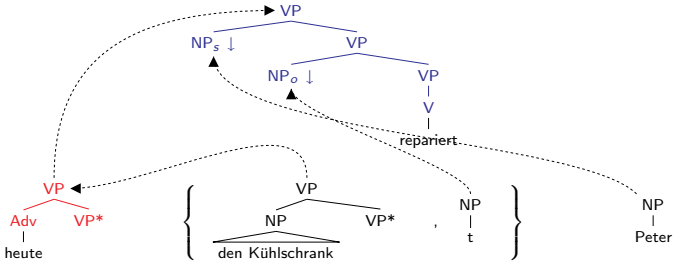
- **TAG-Analyse (1):** TAG mit Baummengen, mit Spuren
  (4) dass [den Kühlschrank]$_i$ Peter t$_i$ repariert



Freie Wortstellung im Deutschen

- **TAG-Analyse (1):** TAG mit Baummengen, mit Spuren
  Problem: Nichtlokale Anknüpfung der Baummenge

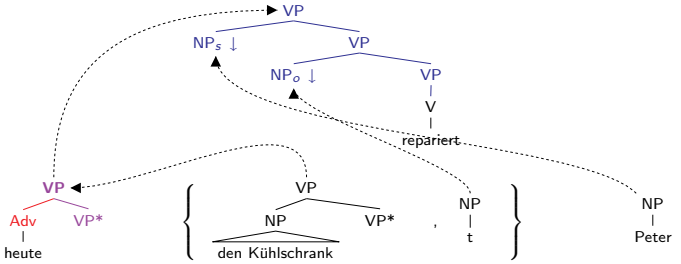
  (5) dass [den Kühlschrank]$_i$ heute Peter t$_i$ repariert



Freie Wortstellung im Deutschen

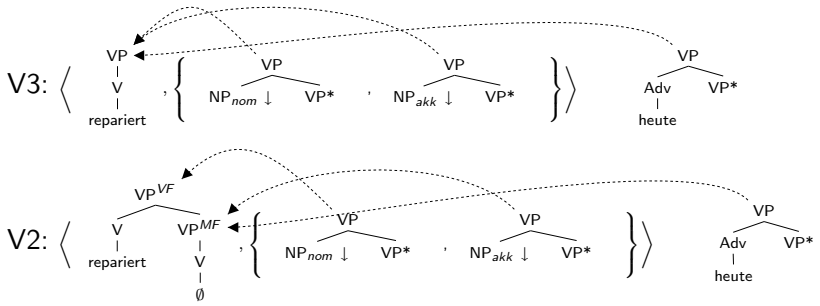
- **TAG-Analyse (1):** TAG mit Baummengen, mit Spuren
  Problem: Nichtlokale Anknüpfung der Baummenge
  Lösung: Lockerung des Lokalitätsbegriffs

Node-Sharing: Wurzelknoten von Adjunktbäumen (Hilfsbäumen) gehören sowohl dem Adjunktbaum als auch dem Zielbaum.



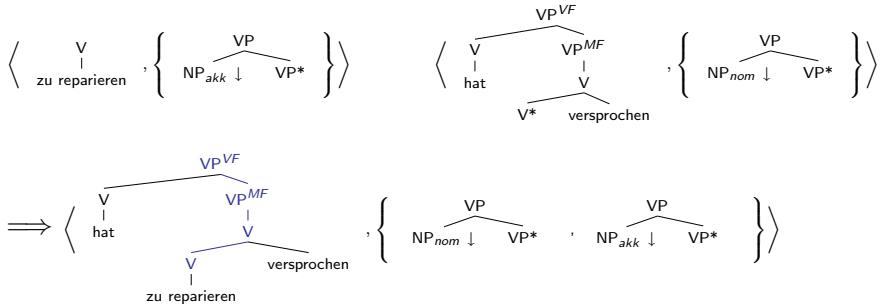
Freie Wortstellung im Deutschen

- **TAG-Analyse (1):** TAG mit Baummengen, mit Spuren
- **TAG-Analyse (2):** TAG mit Baumtupeln, ohne Spuren



Kohärente Konstruktionen

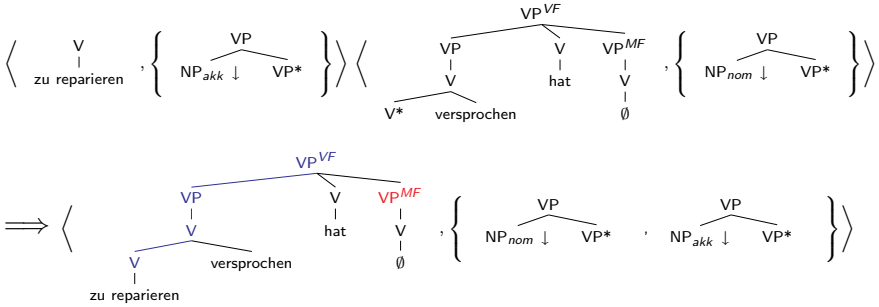
(6) [Den Kühlschrank] hat Peter *zu reparieren* versprochen.



Shared-Spine: Die Knoten zwischen dem Fuß- und Wurzelknoten des Hilfsbaums gehören sowohl zum Hilfsbaum als auch zum Zielbaum.

Voranstellungsphänomene

(7) *Zu reparieren* versprochen hat Peter [den Kühlschrank].



Shared-Tree: Alle Knoten des Hilfsbaums gehören auch zum Zielbaum.

Agenda

- Extraposition
  - Komplemente: rechtsverzweigende Komplementbäume?
  - Adjunkte: rechtsverzweigende Adjunktbäume?
  - Relativsätze: mithilfe von Ankern?
- NP-Split, NP-PP-Split
- Stellungsirregularitäten im Verbkomplex (Oberfeldphänomene, Zwischenstellung, Linksstellung)
- ...

Teil III

Parsing

Beschränkungen für den Formalismus

Ziel: polynomielles Parsen.

Daher: TAG-Erweiterung, die

- polynomiell ist, und
- deren Ableitungen einer Lokalitätsbeschränkung genügen (ein Argument verbindet sich mit dem Prädikat, von dem es abhängt).

⇒ Formalismus im schwach kontextsensitiven Bereich.

Simple RCGs

Schwach kontextsensitive Sprachen können durch einfache (simple) Range Concatenation Grammars (RCG) charakterisiert werden.

RCGs beschreiben, wie sich die Beiträge einzelner lexikalischer Elemente (einschließlich ihrer Argumente und Adjunkte) im Satz verteilen.

(8) das Fahrrad hat Peter ihm noch heute vor Zeugen zu reparieren versprochen

Simple RCGs

Charakteristisch für simple RCGs:

- Verknüpfung der einzelnen lexikalischen Elemente ist kontextfrei ($\Rightarrow$ Lokalität).
- Beiträge lexikalischer Elemente dürfen nur in eine begrenzte Anzahl von nicht adjazenten Teilstrings zerlegt werden.
- Beim Auseinanderreißen/Umordnen von Teilstrings zur Erzeugung größerer Teilstringfolgen darf nicht gelöscht oder kopiert werden.

Range Concatenation Grammar für Parsing

- RCGs (also auch simple RCGs) sind polynomiell parsebar.
- Es gibt schon Parsingalgorithmen für simple RCGs, nämlich
  - traditionelle top-down Algorithmen (bottom-up, Earley auch möglich), und
  - Automatenmodelle (sogenannte *Thread Automata*), die simple RCGs parsen können.

Range Concatenation Grammar für Parsing

Idee für das Parsen des im Projekt verwendeten Grammatikformalismus:

- Beschränkung der zugelassenen Grammatiken, so dass der Beitrag eines lexikalischen Elements nur in eine begrenzte Menge von Teilstrings zerlegt werden darf.
- Umwandlung der Grammatik in eine simple RCG.
- Parsen eines Eingabesatzes mit der simple RCG.
- Ablesen der Ableitungsstruktur in der zugrundeliegenden LTAG-Variante aus dem Parsebaum des RCG-Parsens.

Teil IV

Metagrammatik

Entwicklung und Wartung

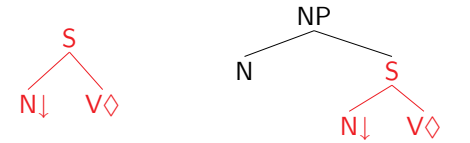
Redundanz in den Elementarbäumen und Grammatikgröße erschwert Grammatikentwicklung und -wartung.

- Kleine Änderungen an der Grammatik
  $\rightsquigarrow$ Auswirkungen auf enorm viele Elementarbäume
- Verwaltung großer Baummengen
  $\rightsquigarrow$ praktisch unmöglich (Französische TAG: >6000 Baumschemata)
- Viele Änderungen an vielen Bäumen
  $\rightsquigarrow$ Potenzielle Fehlerquelle!

Lösung: Baumschemata und Metagrammatik!

Baumgrammatiken sind groß und redundant

Jeder Elementarbaum besteht aus einem lexikalischen Element und der dazugehörigen syntaktischen Struktur.



Heinz singt Ein Lied, das **Heinz** singt

- Wörter in derselben Wortklasse verwenden dasselbe „Baumschema“
- Baumschemata ähneln sich untereinander bzw. bestehen aus identischen Baumfragmenten

Baumschemata und Metagrammatik

Baumschemata:

- Die Grammatik besteht aus unlexikalisierten Bäumen und einem Lexikon für deren Lexikalisierung
- Die Lexikalisierung wird vom Parser durchgeführt

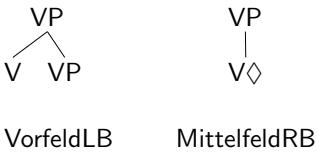
Metagrammatik:

- Baumschemata werden als Kombinationen von elementaren Baumfragmenten beschrieben
- Durch diese Abstraktion werden linguistische Generalisierungen erfasst

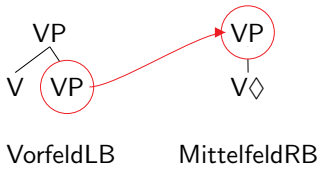
XMG: eXtensible MetaGrammar

- XMG bietet eine ausdrucksstarke Sprache, die es erlaubt, Baumfragmente zu spezifizieren und zu vollständigen Bäumen zu kombinieren.
- Der Metagrammatikcompiler interpretiert die Baumfragmentbeschreibungen und -kombinationsregeln und erzeugt vollständige Bäume (bzw. Baumschemata)
- XMG wurde für die im Projekt verwendete TAG-Variante angepasst (in Zusammenarbeit mit LORIA, Nancy)

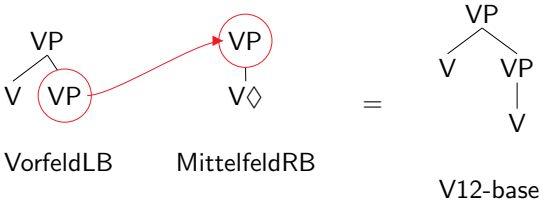
Beschreibung und Kombination von Baumfragmenten



Beschreibung und Kombination von Baumfragmenten



Beschreibung und Kombination von Baumfragmenten



Demo...

Vielen Dank!