Vorlesung 14

# Industrielle Parser

*Dozent: Gerold Schneider*

In der letzten Vorlesung werden wir mit zwei industriellen Parsern für Englisch experimentieren. Industrielle Parser sind auf Parsinggeschwindigkeit, Implementierbarkeit und wirtschaftliche Verkaufbarkeit hin optimiert, während sie bei der linguistischen Präzision teilweise Abstriche machen. Während sie von formellen Theorien beeinflusst sind, gehen sie teilweise auch ihre eigenen Wege und verwenden proprietäre Formalismen und Notationen. Eine gute Übersicht über Parser, die man übers Internet austesten kann, industrielle wie auch solche, die sich einer bestimmten formalen Theorie verpflichtet fühlen, findet man hier in unseren Webseiten. Die zwei vorzustellenden sind auch darunter:

- Der Link Grammar Parser der Carnegie Mellon Uni, Pittsburgh : Der Link Grammar Parser verwendet eine eigene Notation und einen eigenen Formalismus, der von der Dependenzgrammatik (DG) inspiriert ist.
- Der funktionale Dependenzparser der Uni Helsinki/Conexor : Dieser Parser befolgt die Theorie der Dependenzgrammatik.

# Link Grammar

## Die grundlegende Idee

Wie DG ist Link Grammar lexikalistisch, d.h. die einzigen Grammatikregeln, die es gibt, sind Lexikoneinträge. Wie in der DG ist die Valenz das zentrale Konzept.

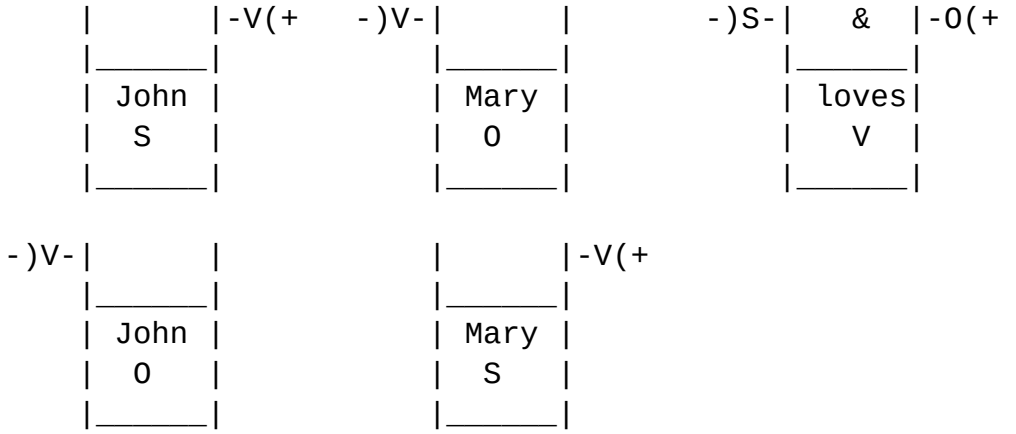
Ein Verb beispielsweise benötigt ein Subjekt und ein Objekt.Diese Valenz wird im Lexikoneintrag des Verbs ausgedrückt. Anders als DG ist Link Grammar aber knofigurational orientiert, also die Wortstellung wird berücksichtigt. Deshalb wird im Lexikoneintrag angegeben, ob die Valenzen nach links oder nach rechts erscheinen. In unmarkierter (kanonischer) Satzstellung eröffnet ein transitives Verb (z.B.- **love**) eine Subjektsvalenz (ausgedrückt durch **S**) nach links (ausgedrückt durch -) und eine Objektsvalenz ( ausgedrückt durch **O**) nach rechts ( ausgedrückt durch +):

```
loves: S- & O+ ;
```

Anders als in DG muss aber ein Dependens (Subjekt **S** und Objekt **O**) auch die vom Regens gestellte Valenzforderung annehmen, nach links (-) oder nach rechts (+):

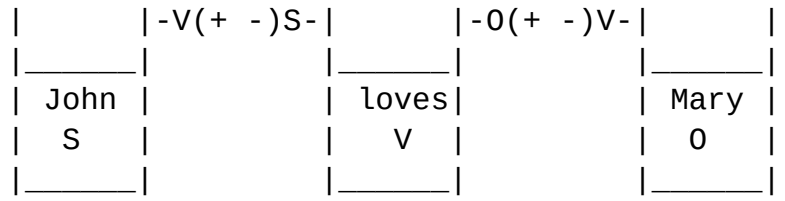
```
John: V+ or V- ;
Mary: V+ or V- ;
```

Die Grammatik enthält zusätzlich das Wissen, dass love ein Verb und John wie Mary Subjekt oder Objekt sein kann. Die drei Worte **John**, **Mary** und **loves**

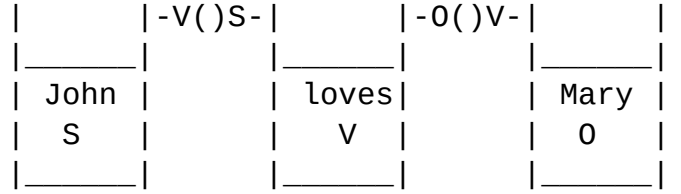



lassen sich verbinden, wie Zugwaggons oder Puzzleteile, zum Beispiel:

So passt es zusammen



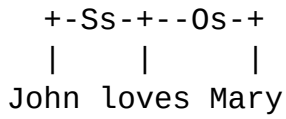

verbunden!




Man kann sich Worte als Puzzleteile vorstellen: Zum Verb-Puzzleteil passt rechts ein Subjekt-Puzzleteil und links ein Objekt-Puzzleteil.

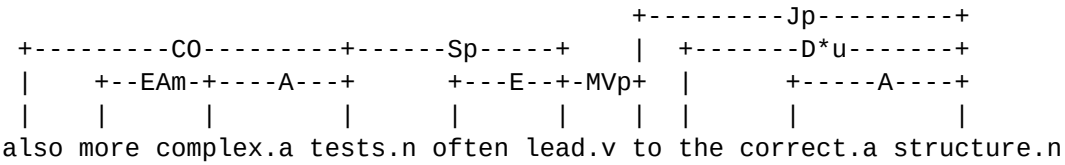
# Link Grammar Syntaxstrukturen

In Link-Grammar wird diese Struktur wie folgt dargestellt:

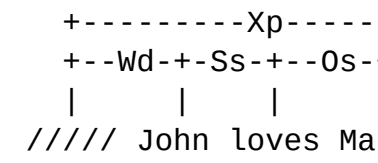



Die Verbindung (engl. Link) zwischen den beiden Worten erhält den Namen der Valenz des Kopfes (also des Verbes in diesem Fall), Kleinbuchstaben geben morphologische Zusatzinformationen an (Singular in diesem Fall)

Auch komplexere Sätze folgen diesem einfachen Muster:



Der ganze Satz mit Interpunktion wird wie folgt behandelt. Beachte, dass das Subjekt und nicht das Verb als oberste Abhängigkeit erscheint:

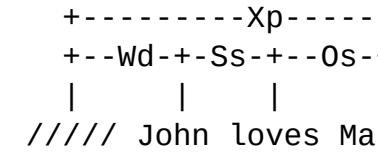



```
A dummy word called ``the wall'' is prepended to the beginning
of each sentence. This word is denoted in the parsing diagrams
as ``/////''.
W is used to attach main clauses to the wall
```

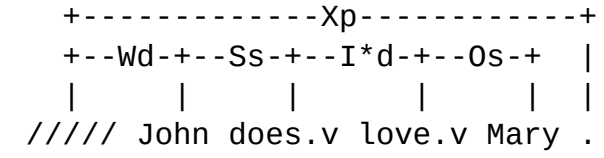
Der Link Grammar Parser ist enorm schnell und hat eine erstaunlich umfassende Grammatik, vermag aber häufig nicht dieselbe Struktur syntaktisch verwandten Sätzen zuzuordnen, wie wir sehen werden bei

- Fragen
- Passivierung
- Topikalisierung
- Spaltsätzen (Cleft)
- Dative Shift

# Fragen



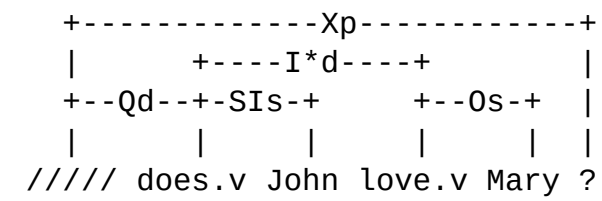

Die Behandlung von Hilfsverben:




```
I connects certain verbs with infinitives.
```

Es besteht keine offene verbale Einheit. Sowohl Subjekt als auch Objekt werden am naheliegendsten Element der Verbalkette angebunden.

Ja/Nein-Fragen:

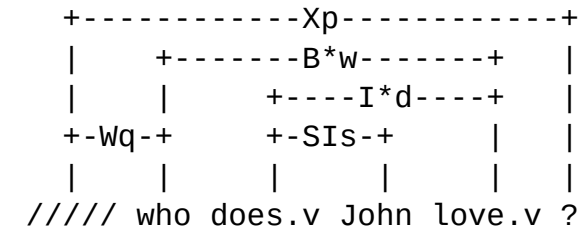



SI is used in subject-verb inversion

SI ist ein anderer Linktyp als S. Einer semantischen Komponente muss explizit mitgeteilt werden, dass der S und der SI Link miteinander verwandt sind. Die verbale Einheit wird durch das Subjekt unterbrochen. Semantisch gesehen stehen sich die Verbteile sehr nahe, so dass eine semantiknahe Syntaxstruktur hier eine überkreuzende Struktur annehmen müsste, in der die Verbteile näher beieinander liegen als der Link zum Subjekt. In GB teilt man das Verb bewusst in Hilfsverb (Kopf des IP) und Hauptverb (Kopf des VP). In HPSG und LFG leitet das Hilfsverb (einem Hebungsverb gleich) einen neuen untergeordneten subjektslosen Satz oder zumindest eine untergeordnete VP ein. Mit Argumentskomposition werden in HPSG die Verbteile zumindest soweit semantisch vereint, als dass sie die gleichen Valenzen haben.

In der obigen Struktur ist der Wall (/////) nicht wie sonst in Link Grammar mit dem Subjekt, sondern doch mit einem Verbteil, wie in DG üblich, verbunden.

Wh-Fragen:




B is used in a number of situations, involving relative clauses and questions.

B ist anderer Linktyp als O. Einer semantischen Komponente muss explizit mitgeteilt werden, dass der O und der B Link miteinander verwandt sind. Zusätzlich problematisch ist, dass B ein (zu) genereller Linktyp ist, der auch mit vielen anderen Links verwandt ist.

# Passivierung

```
  +-Ss-+--Os-+
  |    |     |
John loves Mary

  +-Ss-+-Pv-+-MVp+-Js+
  |    |    |    |   |
Mary is.v loved by John
```

```
Pv is used to connect forms of "be" to passive participles.
MV connects verbs (and adjectives) to modifying phrases like
adverbs, prepositional phrases ...
```

Syntaktisch ist diese Analyse korrekt, sie drückt aber nichts über die Verwandtschaft des Passivverbs mit dem Aktivverb aus. Der MV-Link wird für allerlei Verbkomplemente und -adjunkte verwendet, die thematische Funktion bleibt also verborgen.

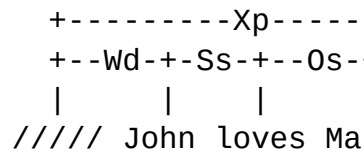
# Topikalisierung

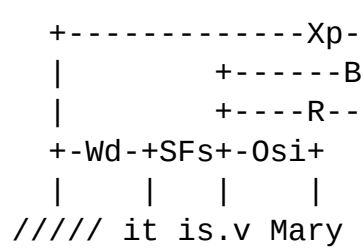
Der Parser findet keine Antwort auf Sätze wie"Mary, John loves", was verzeihlich ist, da diese Form grammatisch selten ist:

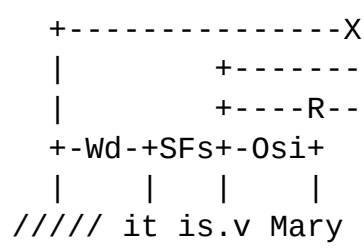
```
No complete linkages found.
```

```
[Mary] [,] [John] [loves] [.]
```

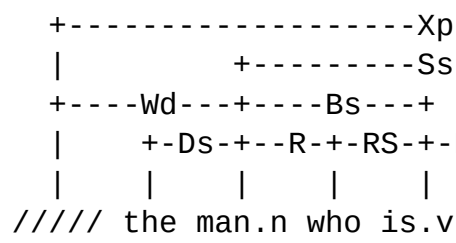
# Spaltsätze (Cleft)



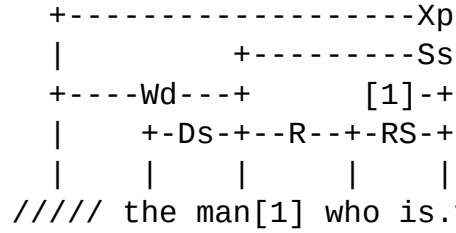



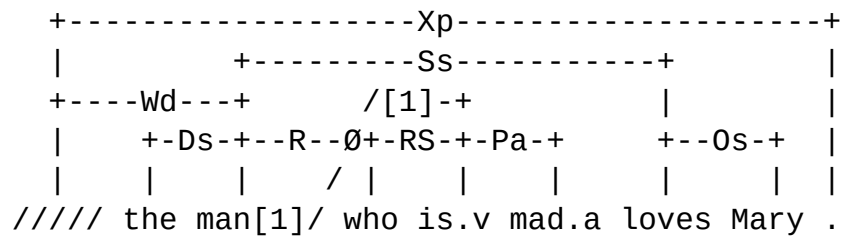

Für die unbeschränkte Abhängigkeit (Objekt von love) wird erneut - wie bei den Fragen - der B-Link eingesetzt. Die Links B-R-O bilden einen Zyklus, was in Dependenzgrammatik nicht möglich wäre. Der R-Link wird auch verwendet zur Einleitung von Relativsätzen:

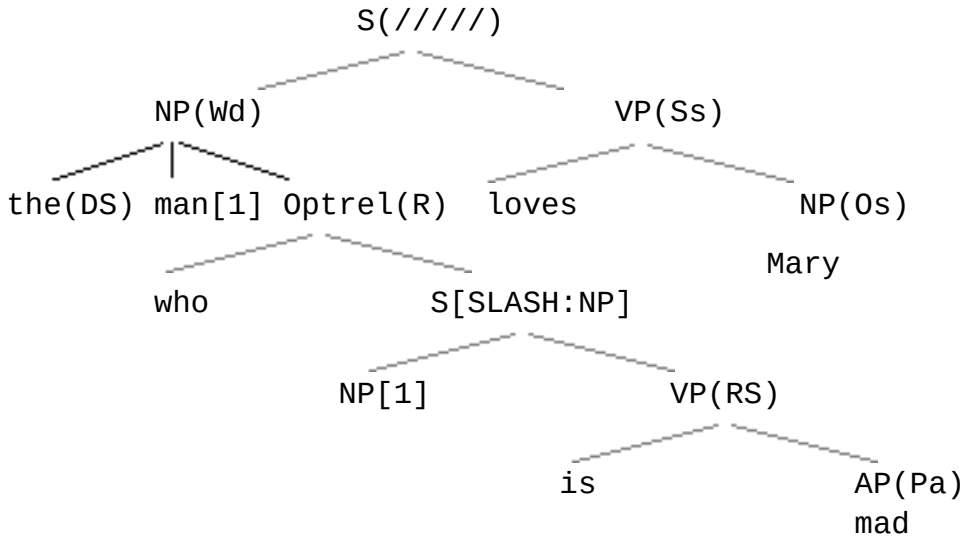



Der B-Link drückt die Valenzforderung des Verbes (is) im Relativsatz aus. Man ist sowohl Subjekt (kliengeschriebenes **s** in **Bs**) des Verbes *is* als auch des Verbes *loves.* In HPSG erhielten das Subjekt der beiden Verben eine Koreferenz:

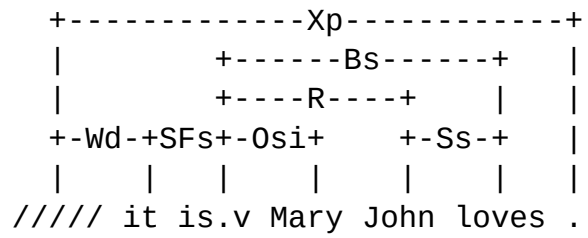


Diese Abhängigkeit ist eigentlich überkreuzend (Ø = Überkreuzung):

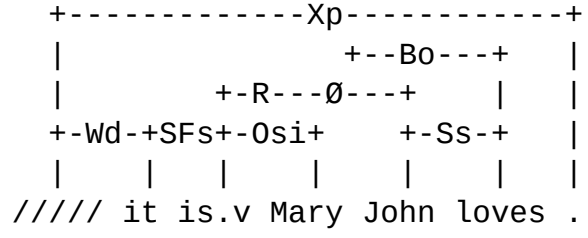



Im obigen Spaltsatz tritt das noch deutlicher zu Tage:




Hier verbindet sich der Bs-Link mit dem Verb *is* ! Erst via einen weitern Link (Osi) wird das Objekt erreicht. Erst das Wissen, über diese Mehrfachverbindung, in der Subjektslink Bs (kleines s) zudem in einen Objektslink (Osi) verwandelt wird, erlaubt eine sinnvolle Interpreation dieses Satzes. Die adäquate Struktur dieses Satzes wäre (Ø zeigt Überkreuzung an):



# Dative Shift

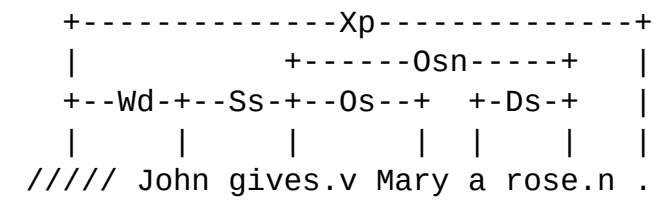
Als Dative Shift bezeichnet man in TG /GB die Transformation, die den Satz

`John gives Mary a rose.`

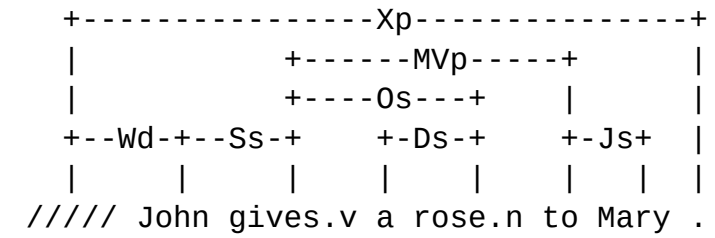
in

`John gives a rose to Mary.`

transformiert.

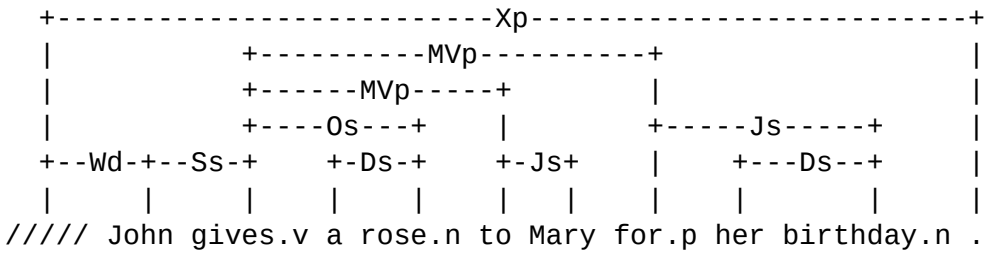



Diese Analyse unterscheidet korrekt zwischen dem direkten Objekt **Os** und dem indirekten Objekt **Osn.**




MV connects verbs (and adjectives) to modifying phrases like adverbs, prepositional phrases

Diese Analyse behandelt die **PP** *to Mary* nicht wie ein Komplement, sondern gleich wie ein Adjunkt.



## Schlussfolgerungen

Der Parser ist u.a. so schnell, weil die Grammatik kontextfrei ist und somit keine überkreuzenden Verbindungen (=unbeschränkte Abhängigkeiten) erlaubt. Deshalb erscheinen viele der Strukturen nicht wirklich linguistisch motivierte, sondern ad hoc Lösungen zu sein.

Trotz einiger linguistischer Mängel lässt sich Link Grammar sinnvoll einsetzen, z.B. in unserem ExtrAns Projekt, und die Grösse der Grammatik wie auch die Geschwindigkeit des Parsers sind beindruckend.

# Funktionaler Dependenzparser

Der funktionale Dependenzparser lehnt sich an die Dependenzgrammatiktheorie an. Eine Einführung findet sich hier. Die Java-basierten graphischen Strukturen sind alleine einen Versuch wert. Die Grammatik dieses Parsers ist weniger umfassend, vermag aber einige Verwandtschaften auszudrücken:

- Fragen
- Passivierung
- Topikalisierung
- Spaltsätzen (Cleft)
- Dative Shift

## Fragen

```
John loves Mary.
       loves
   subj/  \obj
    John  Mary

John does love Mary.
```

Dies ist eine typische Dependenzstrukur mit Kopf *love* und Dependentien *John* und *Mary*.

```
          love
verb-chain/  \obj
       does  Mary
   subj/
    John
```

Alle Verbteile sind durch die Verb-chain verbunden.

```
Does John love Mary?
          love
verb-chain/  \obj
       Does  Mary
   subj/
    John
```

Die Frage erhält dieselbe Analyse wie der Aussagesatz. Das System erkennt trotz der unterbrochenen verbalen Einheit zwischen Hilfsverb und Hauptverb (bewusste Aufteilung des Verbes in Hilfs- und Hauptverb / unbeschränkte Abhängigkeit) die dem Aussagesatz engst verwandte Struktur.

```
Who does John love?

  John        love
              vc|         %vc=verb-chain
              does
           subj/
             Who
```

Hier versagt die Grammatik ganz kläglich aufgrund des kleinen Vokabulars, das *love* offenbar nicht korrekt enthält. Richtig und elegant, da wie der entsprechende Aussagesatz, wird der folgende Satz analysiert:

```
What does John eat?

        eat
   obj/ vc|          %vc=verb-chain
   What  does
      subj/
       John
```

Der Parser wird mit unbeschränkten Abhängigkeiten fertig, da er teilweise kontextsensitiv arbeitet und also Überkreuzungen behandeln kann. Die gelieferte Depenzstruktur lässt die überkreuzende Abhängigkeit gar nicht mehr erkennen.

# Passivierung

```
John loves Mary.

       loves
   subj/  \obj
    John  Mary

Mary is loved by John.

       loved
     vc/  \agent
     is   by
 subj/     \pcomp
   Mary    John
```

Diese Analyse ist nicht nur korrekt, sondern erkennt im Passivsatz die thematische Rolle (agent) des by-PP.

# Topikalisierung

```
Mary, John loves.
```

```
       loves
   subj/  \obj
    John  Mary
```

Korrekt, genau dieselbe Analyse wie der kanonische Satz. (Überkreuzung in PSG !)

# Spaltsätze (Cleft)

```
It is Mary John loves.
```

liefert keine zusammenhängende Analyse, mit einem anderen Verb (z.B. It is apples John eats) eine falsche:

```
      is
 subj/  \comp
    It  eats
         |subj
        John
         |attr
        apples
```

```
It is Mary that John loves.
```

Der Parser liefert eine etwas ungewöhnliche Analyse (auch für ein anderes Verb als love), die die semantischen Verhältnisse höchstens auf grossen Umwegen ausdrückt:

```
        is
   subj/  \comp
      It  Mary
   mod|
     loves
  obj/  \subj
  that  John
```

## Dative Shift

```
John gives Mary a flower.

        gives
  subj/ |dat \obj    %dat=dative
  John  Mary flower
              |det
              a

John gives a flower to Mary.

        gives
  subj/ |dat \obj    %dat=dative
  John  to  flower
   pcomp/     |det
      Mary    a
```

Diese beiden Analysen sind eng verwandt und erkennen insbesondere den Dativ in beiden Varainten korrekt, anders als Link Grammar.

## Schlussfolgerungen

Die Grammatik dieses Parsers ist teilweise kontext-sensitiv und erlaubt somit echt überkreuzende Verbindungen (=unbeschränkte Abhängigkeiten). Die Grammatik ist allerdings nicht ganz so umfassend und umfangreich wie diejenige der Link Grammar.

---

*Gerold Schneider <gschneid@ifi.unizh.ch>*

*Date of last modification: 31 January 2000, 3:40 am. Moin, Moin!*