

# Combinatory Categorical Grammar

Vera Demberg

Universität des Saarlandes

Grammatikformalismen SS 2013

28. Mai 2013

Grammatikformalismen – Warum?

Grammatikformalismen – interessante Fälle:

Bindung:

- Peter erzählt Sabine, dass **Sandra sich** mag.
- Peter erzählt **Sabine**, dass Sandra **sie** mag.
- **Peter** erzählt Sabine, dass Sandra **ihn** mag.
- **Peter** erzählt **Torsten**, dass Sandra **ihn(?)** mag.

Lange Abhängigkeiten: (Beispiele für englische Objektrelativsätze)

- Fred likes the **man** that John likes *t*.
- Fred likes the **man** that Sally thinks that John likes *t*.
- Fred likes the **man** that Peter sees that Sally thinks that John likes *t*.

Ziele:

- Grammatikformalismus soll das erklären können
- ... aber gleichzeitig nicht übergenerieren.

Lange Abhängigkeiten: (Beispiele für englische Objektrelativsätze)

- Fred likes the **man** that John likes *t*.
- Fred likes the **man** that Sally thinks that John likes *t*.
- Fred likes the **man** that Peter sees that Sally thinks that John likes *t*.

Ziele:

- Grammatikformalismus soll das erklären können
- ... aber gleichzeitig nicht übergenerieren.

Lange Abhängigkeiten: (Beispiele für englische Objektrelativsätze)

- Fred likes the **man** that John likes *t*.
- Fred likes the **man** that Sally thinks that John likes *t*.
- Fred likes the **man** that Peter sees that Sally thinks that John likes *t*.

Ziele:

- Grammatikformalismus soll das erklären können
- ... aber gleichzeitig nicht übergenerieren.

Lange Abhängigkeiten: (Beispiele für englische Objektrelativsätze)

- Fred likes the **man** that John likes *t*.
- Fred likes the **man** that Sally thinks that John likes *t*.
- Fred likes the **man** that Peter sees that Sally thinks that John likes *t*.

Ziele:

- Grammatikformalismus soll das erklären können
- ... aber gleichzeitig nicht übergenerieren.

Koordination: Nur gleiche Kategorien können koordiniert werden.

## Beispiel

- Ich sehe einen Hund und eine Katze.
- Ich sehe und höre einen Hund.
- \*Ich sehe einen Hund und gebe.
- \*Ich schlafe und eine Katze.



Warum sind die folgenden holländischen Beispiele ein Problem für kontextfreie Grammatikformalisten?

*dat ik<sub>1</sub> Anna<sub>2</sub> Jan<sub>3</sub> zag<sub>1</sub> helpen<sub>2</sub> zwemmen<sub>3</sub>.*  
dass ich Anna Jan schwimmen helfen sah.

# Table of Contents

- 1 Combinatory Categorical Grammar (CCG)
- 2 Strict Competence Hypothesis

# Table of Contents

- 1 Combinatory Categorical Grammar (CCG)
- 2 Strict Competence Hypothesis

# Was kann CCG?

CCG = Combinatory Categorical Grammar

- Kann folgende Phänomene erklären:
  - Bindung / Kontrolle (Reflexivpronomen)
  - Lange Abhängigkeiten (Relativsätze)
  - Koordination
  - Kreuzende Abhängigkeiten (Holländisches Beispiel; → mild kontextabhängig)
- direkte Syntax-Semantik Schnittstelle

# Eigenschaften der Combinatory Categorical Grammar

## Kategorien:

- Kategorien beschreiben wie sich die Wörter im Satz verhalten.
- Lexikon: jedes Wort ist mit einer oder mehreren Kategorien assoziiert.

## Example

einfache Kategorien	komplexe Kategorien
Peter $NP$	a $NP/N$
dog $N$	sleeps $S \backslash NP$

## Operationen:

- kleine Menge aus kombinatorischen Regeln.

## Example

Forward Application:  $X/Y \quad Y \quad \Rightarrow_{>} \quad X$

Backward Application:  $Y \quad X \backslash Y \quad \Rightarrow_{<} \quad X$

# Eigenschaften der **Combinatory Categorical** Grammar

## Kategorien:

- Kategorien beschreiben wie sich die Wörter im Satz verhalten.
- Lexikon: jedes Wort ist mit einer oder mehreren Kategorien assoziiert.

## Example

einfache Kategorien	komplexe Kategorien
Peter $NP$	a $NP/N$
dog $N$	sleeps $S \backslash NP$

## Operationen:

- kleine Menge aus kombinatorischen Regeln.

## Example

Forward Application:  $X/Y \quad Y \quad \Rightarrow_{>} \quad X$

Backward Application:  $Y \quad X \backslash Y \quad \Rightarrow_{<} \quad X$

## Ein paar einfache Ableitungen mit CCG:

## Beispiellexicon

Wort	Kategorie	Wort	Kategorie
Peter	$NP$	a	$NP/N$
dog	$N$	sleeps	$S \backslash NP$
		saw	$(S \backslash NP) / NP$
		tired	$N/N$
		often	$(S \backslash NP) \backslash (S \backslash NP)$

## Regeln

Forward Application:  $X/Y \quad Y \quad \Rightarrow_{>} \quad X$

Backward Application:  $Y \quad X \backslash Y \quad \Rightarrow_{<} \quad X$

Wir leiten ab: *A dog sleeps.*

## Ein paar einfache Ableitungen mit CCG:

## Beispiellexicon

Wort	Kategorie	Wort	Kategorie
Peter	$NP$	a	$NP/N$
dog	$N$	sleeps	$S \backslash NP$
		saw	$(S \backslash NP) / NP$
		tired	$N / N$
		often	$(S \backslash NP) \backslash (S \backslash NP)$

## Regeln

Forward Application:  $X/Y \quad Y \quad \Rightarrow_{>} \quad X$

Backward Application:  $Y \quad X \backslash Y \quad \Rightarrow_{<} \quad X$

Wir leiten ab: *Peter saw a dog.*



## Ein paar einfache Ableitungen mit CCG:

## Beispiellexicon

Wort	Kategorie	Wort	Kategorie
Peter	$NP$	a	$NP/N$
dog	$N$	sleeps	$S \backslash NP$
		saw	$(S \backslash NP) / NP$
		tired	$N / N$
		often	$(S \backslash NP) \backslash (S \backslash NP)$

## Regeln

Forward Application:  $X/Y \quad Y \quad \Rightarrow_{>} \quad X$

Backward Application:  $Y \quad X \backslash Y \quad \Rightarrow_{<} \quad X$

Wir leiten ab: *A tired dog sleeps.*

## Ein paar einfache Ableitungen mit CCG:

## Beispiellexicon

Wort	Kategorie	Wort	Kategorie
Peter	$NP$	a	$NP/N$
dog	$N$	sleeps	$S \backslash NP$
		saw	$(S \backslash NP) / NP$
		tired	$N / N$
		often	$(S \backslash NP) \backslash (S \backslash NP)$

## Regeln

Forward Application:  $X/Y \quad Y \quad \Rightarrow_{>} \quad X$

Backward Application:  $Y \quad X \backslash Y \quad \Rightarrow_{<} \quad X$

Wir leiten ab: *Peter sleeps often.*

# CCG Combinatory Rules

CCG includes further operations in order to adequately handle English.

Coordination:  $X \quad (X \backslash X) / X \quad X \quad \Rightarrow_{\phi} \quad X$

Beispielableitung an Tafel

Peter loves and admires Sandy.

Lexikon

Wort	Kategorie	Wort	Kategorien
Peter	$NP$	and	$(X \backslash X) / X$
Sandy	$NP$	loves	$(S \backslash NP) / NP$
		admires	$(S \backslash NP) / NP$

\*Peter loves Sandy and admires.

# CCG Combinatory Rules

CCG includes further operations in order to adequately handle English.

Coordination:  $X \quad (X \backslash X) / X \quad X \quad \Rightarrow_{\phi} \quad X$

Beispielableitung an Tafel

Peter loves and admires Sandy.

Lexikon

Wort	Kategorie	Wort	Kategorien
Peter	$NP$	and	$(X \backslash X) / X$
Sandy	$NP$	loves	$(S \backslash NP) / NP$
		admires	$(S \backslash NP) / NP$

\*Peter loves Sandy and admires.

## CCG Combinatory Rules

Forward Composition:  $X/Y \quad Y/Z \Rightarrow_{>B} X/Z$

Backward Composition:  $Y \setminus Z \quad X \setminus Y \Rightarrow_{<B} X \setminus Z$

## Example

Peter loves and might admire Sandy.

## Lexikon

einfache	Kategorien	komplexe	Kategorien
Peter	$NP$	and	$(X \setminus X)/X$
Sandy	$NP$	loves	$(S \setminus NP)/NP$
might	$(S \setminus NP)/(S \setminus NP)$	admire	$(S \setminus NP)/NP$

\*Peter loves and might Sandy.

## CCG Combinatory Rules

Forward Composition:  $X/Y \quad Y/Z \Rightarrow_{>B} X/Z$

Backward Composition:  $Y \setminus Z \quad X \setminus Y \Rightarrow_{<B} X \setminus Z$

## Example

Peter loves and might admire Sandy.

## Lexikon

einfache	Kategorien	komplexe	Kategorien
Peter	$NP$	and	$(X \setminus X)/X$
Sandy	$NP$	loves	$(S \setminus NP)/NP$
might	$(S \setminus NP)/(S \setminus NP)$	admire	$(S \setminus NP)/NP$

\*Peter loves and might Sandy.

## CCG Combinatory Rules

Forward Type-raising:  $X \Rightarrow_T T/(T \setminus X)$

## Example

Peter loves and Paul admires Sandy.

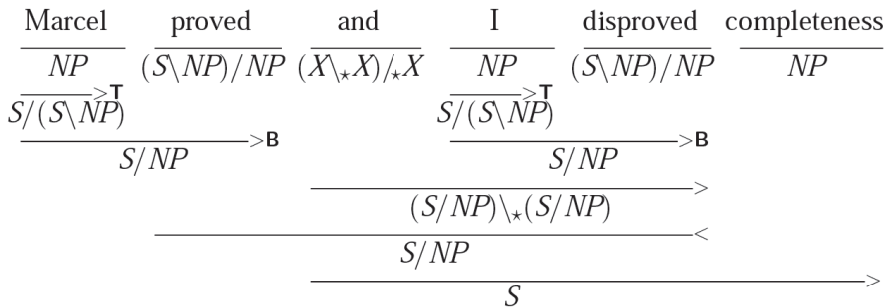
## Lexikon

einfache	Kategorien	komplexe	Kategorien
Peter	$NP$	and	$(X \setminus X)/X$
Sandy	$NP$	loves	$(S \setminus NP)/NP$
Paul	$NP$	admires	$(S \setminus NP)/NP$

## CCG Combinatory Rules

Forward Type-raising:  $X \Rightarrow_T T/(T \setminus X)$

## Example





## All rules for English

Forward Application:	$X/Y$	$Y$	$\Rightarrow >$	$X$
Backward Application:	$Y$	$X \backslash Y$	$\Rightarrow <$	$X$
Coordination:	$X \text{ conj}$	$X$	$\Rightarrow \phi$	$X$
Forward Composition:	$X/Y$	$Y/Z$	$\Rightarrow >_B$	$X/Z$
Backward Composition:	$Y \backslash Z$	$X \backslash Y$	$\Rightarrow <_B$	$X \backslash Z$
Forward Type-raising:	$X$		$\Rightarrow_T$	$T/(T \backslash X)$
Forward Generalized Composition:	$X/Y$	$(Y/Z)/\$_1$	$\Rightarrow >_{B^n}$	$(X/Z)/\$_1$
Backward Crossed Composition:	$Y/Z$	$X \backslash Y$	$\Rightarrow <_{B_x}$	$X/Z$

## Binding und Kontrolle: Reflexivpronomen

The	woman	washes	herself
$NP_{NG}/N_{NG}$	$N_{sf}$	$(S \setminus NP_{sG})/NP$	$(S \setminus NP_{sf}) \setminus ((S \setminus NP_{sf})/NP)$

N: Variable für Numerus

G: Variable für Genus

s: singular

f: weiblich

m: männlich

*Peter	thinks	Sally	likes	himself
$NP_{sm}$	$S \setminus NP/S$	$NP_{sf}$	$(S \setminus NP_{sG})/NP$	$(S \setminus NP_{sm}) \setminus ((S \setminus NP_{sm})/NP)$
Peter	thinks	Sally	likes	herself
$NP_{sm}$	$S \setminus NP/S$	$NP_{sf}$	$(S \setminus NP_{sG})/NP$	$(S \setminus NP_{sf}) \setminus ((S \setminus NP_{sf})/NP)$

## Binding und Kontrolle: Reflexivpronomen

The	woman	washes	herself
$NP_{NG}/N_{NG}$	$N_{sf}$	$(S \setminus NP_{sG})/NP$	$(S \setminus NP_{sf}) \setminus ((S \setminus NP_{sf})/NP)$

N: Variable für Numerus

G: Variable für Genus

s: singular

f: weiblich

m: männlich

*Peter	thinks	Sally	likes	himself
$NP_{sm}$	$S \setminus NP/S$	$NP_{sf}$	$(S \setminus NP_{sG})/NP$	$(S \setminus NP_{sm}) \setminus ((S \setminus NP_{sm})/NP)$
Peter	thinks	Sally	likes	herself
$NP_{sm}$	$S \setminus NP/S$	$NP_{sf}$	$(S \setminus NP_{sG})/NP$	$(S \setminus NP_{sf}) \setminus ((S \setminus NP_{sf})/NP)$

## Binding und Kontrolle: Reflexivpronomen

The	woman	washes	herself
$\overline{NP_{NG}/N_{NG}}$	$\overline{N_{sf}}$	$\overline{(S \setminus NP_{sG})/NP}$	$\overline{(S \setminus NP_{sf}) \setminus ((S \setminus NP_{sf})/NP)}$

N: Variable für Numerus

G: Variable für Genus

s: singular

f: weiblich

m: männlich

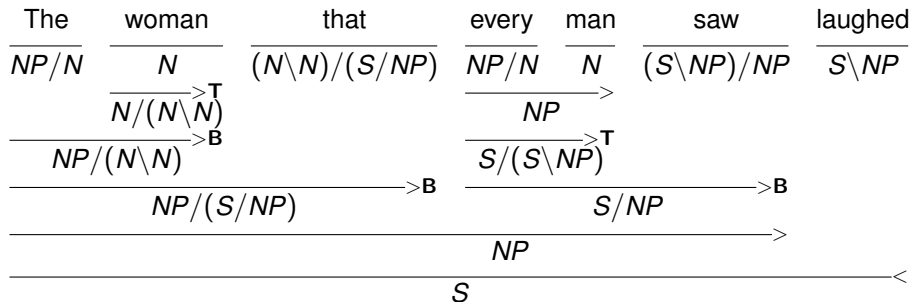
*Peter	thinks	Sally	likes	himself
$\overline{NP_{sm}}$	$\overline{S \setminus NP/S}$	$\overline{NP_{sf}}$	$\overline{(S \setminus NP_{sG})/NP}$	$\overline{(S \setminus NP_{sm}) \setminus ((S \setminus NP_{sm})/NP)}$

Peter	thinks	Sally	likes	herself
$\overline{NP_{sm}}$	$\overline{S \setminus NP/S}$	$\overline{NP_{sf}}$	$\overline{(S \setminus NP_{sG})/NP}$	$\overline{(S \setminus NP_{sf}) \setminus ((S \setminus NP_{sf})/NP)}$

## Lange Abhängigkeiten: Relativsatz

The	woman	that	every	man	saw	laughed
$\overline{NP/N}$	$\overline{N}$	$\overline{(N \setminus N)/(S/NP)}$	$\overline{NP/N}$	$\overline{N}$	$\overline{(S \setminus NP)/NP}$	$\overline{S \setminus NP}$

## Lange Abhängigkeiten: Relativsatz



# Konstituenten in CCG

CCG hat flexible Konstituenten.

“Spurious ambiguity”= Überflüssige Ambiguität

Es gibt 24 verschiedene (korrekte) Ableitungen für:

*Peter caught a big cat.*

aber alle haben die gleiche semantische Interpretation.

# Table of Contents

- 1 Combinatory Categorical Grammar (CCG)
- 2 Strict Competence Hypothesis**



# The Strict Competence Hypothesis

## Strong Competence Hypothesis (Bresnan and Kaplan, 1982)

The Strong Competence Hypothesis asserts that there exists a direct correspondence between the rules of a grammar and the operations performed by the human language processor.

### Competence

Competence is the **'ideal' language system** that makes it possible for speakers to produce and understand an infinite number of sentences in their language, and to distinguish grammatical sentences from ungrammatical sentences.

### Performance

Linguistic performance is governed by **principles of cognitive structure** such as memory limitations, distractions, shifts of attention and interest, and (random or characteristic) errors.

# The Strict Competence Hypothesis

## Strong Competence Hypothesis (Bresnan and Kaplan, 1982)

The Strong Competence Hypothesis asserts that there exists a direct correspondence between the rules of a grammar and the operations performed by the human language processor.

## Competence

Competence is the '**ideal**' **language system** that makes it possible for speakers to produce and understand an infinite number of sentences in their language, and to distinguish grammatical sentences from ungrammatical sentences.

## Performance

Linguistic performance is governed by **principles of cognitive structure** such as memory limitations, distractions, shifts of attention and interest, and (random or characteristic) errors.

# The Strict Competence Hypothesis

## Strong Competence Hypothesis (Bresnan and Kaplan, 1982)

The Strong Competence Hypothesis asserts that there exists a direct correspondence between the rules of a grammar and the operations performed by the human language processor.

+

## Rule-to-Rule Assumption (Bach, 1976)

Each syntactic rule corresponds to a rule of semantic interpretation.  
( $\Rightarrow$  entities combined by syntactic rules must be semantically interpretable)

=

## Strict Competence Hypothesis (Steedman, 1992)

Structures manipulated by the processor are isomorphic to the constituents listed in the grammar.

# The Strict Competence Hypothesis

## Strong Competence Hypothesis (Bresnan and Kaplan, 1982)

The Strong Competence Hypothesis asserts that there exists a direct correspondence between the rules of a grammar and the operations performed by the human language processor.

+

## Rule-to-Rule Assumption (Bach, 1976)

Each syntactic rule corresponds to a rule of semantic interpretation.  
( $\Rightarrow$  entities combined by syntactic rules must be semantically interpretable)

=

## Strict Competence Hypothesis (Steedman, 1992)

Structures manipulated by the processor are isomorphic to the constituents listed in the grammar.

# The Strict Competence Hypothesis

## Strong Competence Hypothesis (Bresnan and Kaplan, 1982)

The Strong Competence Hypothesis asserts that there exists a direct correspondence between the rules of a grammar and the operations performed by the human language processor.

+

## Rule-to-Rule Assumption (Bach, 1976)

Each syntactic rule corresponds to a rule of semantic interpretation.  
( $\Rightarrow$  entities combined by syntactic rules must be semantically interpretable)

=

## Strict Competence Hypothesis (Steedman, 1992)

Structures manipulated by the processor are isomorphic to the constituents listed in the grammar.

# Ausblick nächste Sitzungen:

Nächste Sitzungen:

4. Juni CCG

- CCG: Wiederholung / Fragen
- Semantikkonstruktion für CCG

10. Juni Tree-adjoining grammar

Übungsblatt für 5. Juni auf Kurshomepage.