An introduction to mildly context sensitive grammar formalisms

— *Combinatory Categorial Grammar* —

Gerhard Jäger & Jens Michaelis

University of Potsdam

`{jaeger,michael}@ling.uni-potsdam.de`

Basic Categorial Grammar

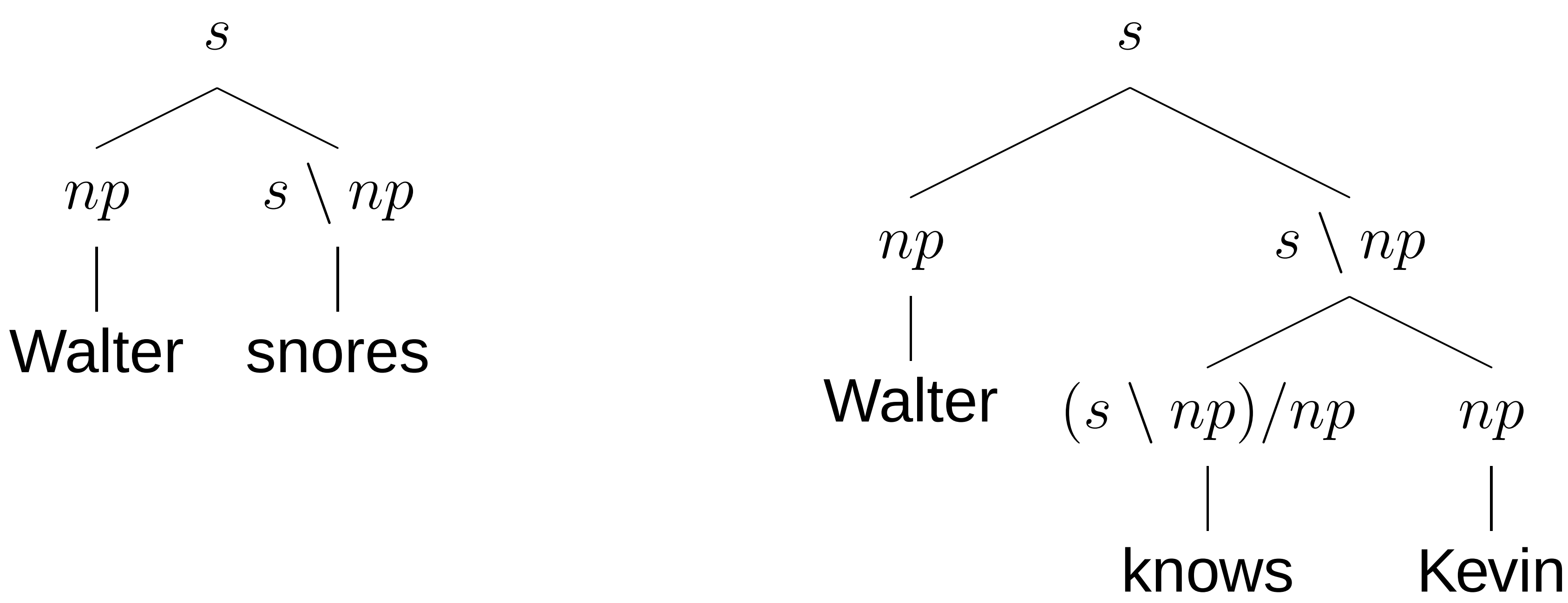
- developed by Bar-Hillel (1953)
- based on earlier work by Ajdukiewicz and others
- close correspondence between syntax and semantics
- fundamental notions: **complete** and **incomplete** expression
- also inherent in type theory and earlier versions of categorial grammar
- new contribution: **directionality of syntactic incompleteness**

A/B ... I need a B to my right to become an A
$A \setminus B$ ... I need a B to my left to become an A

Note: Type Logical CG uses different notational convention!

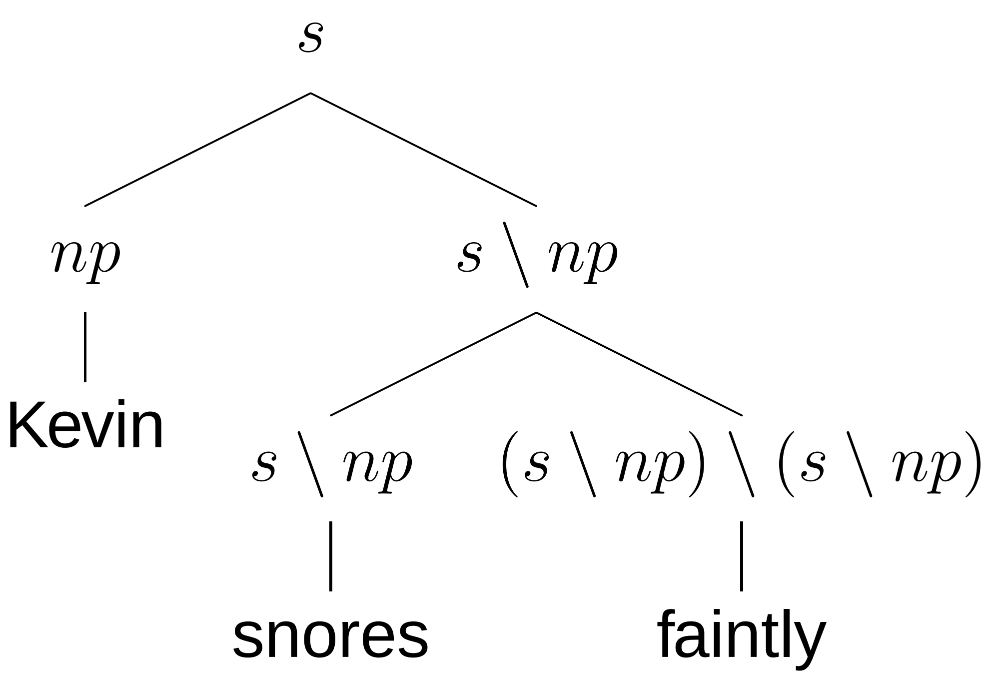
example

$$
\begin{aligned}
\textit{Walter, Kevin} &: \quad np \\
\textit{snores} &: \quad s \setminus np \\
\textit{knows} &: \quad (s \setminus np)/np
\end{aligned}
$$

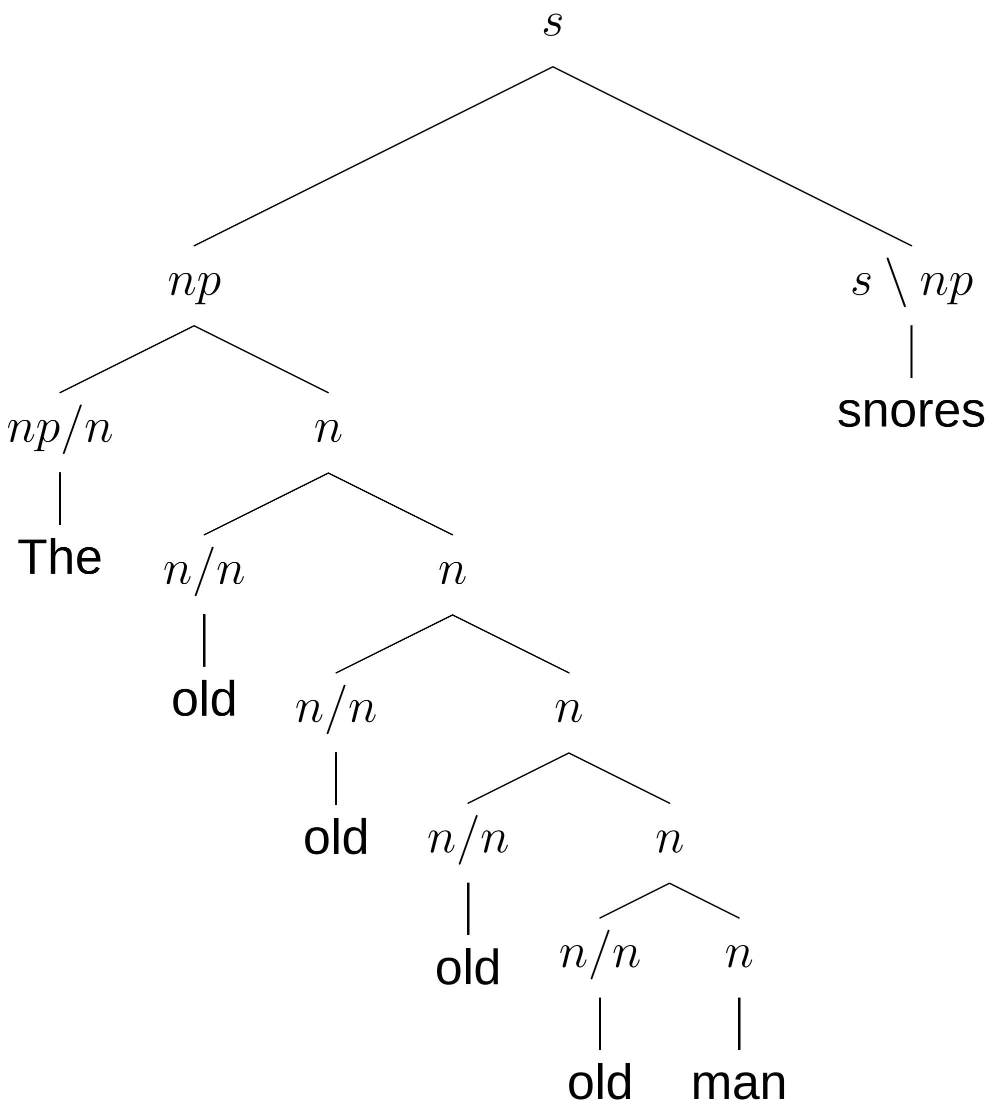


Basic Categorial Grammar

categories can be complex:

- faintly : $(s \setminus np) \setminus (s \setminus np)$
- Kevin snores faintly
- 

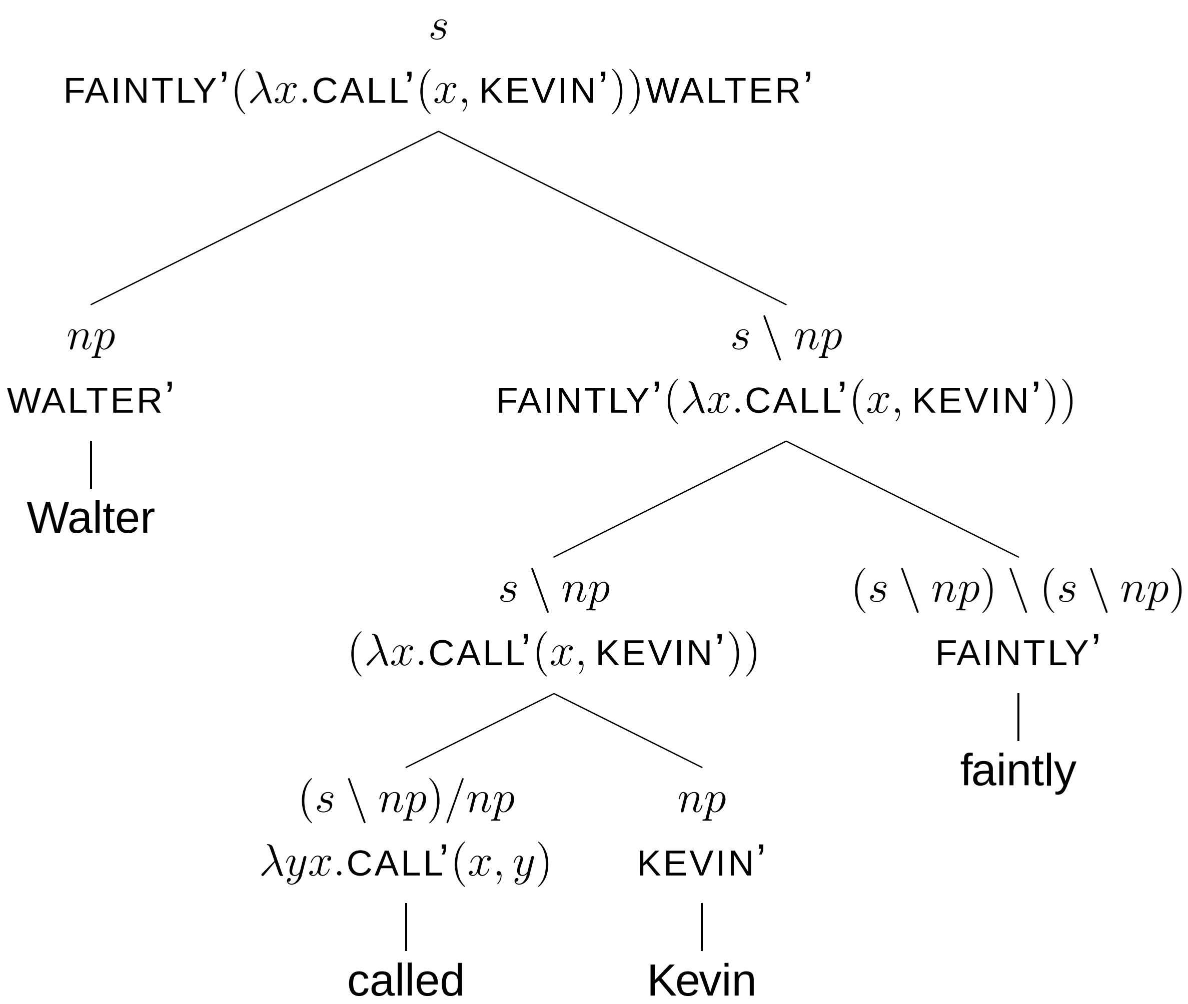
recursion


s
np
$s \setminus np$
np/n
n
snores
The
n/n
n
old
n/n
n
old
n/n
n
old
n/n
n
old
man

Basic Categorial Grammar

syntactic and semantic composition

- (ideally:) syntactic and semantic incompleteness coincide
- syntactic composition concurs with semantic function application



Definition 1 (Categories)
*Let a finite set **B** of **basic categories** be given. **CAT(B)** is the smallest set such that*

1. $\mathbf{B} \subseteq \mathbf{CAT}(\mathbf{B})$
2. *If* $A, B \in \mathbf{CAT}(\mathbf{B})$*, then* $A/B \in \mathbf{CAT}(\mathbf{B})$
3. *If* $A, B \in \mathbf{CAT}(\mathbf{B})$*, then* $A \setminus B \in \mathbf{CAT}(\mathbf{B})$
4. *Nothing else is in* $\mathbf{CAT}(\mathbf{B})$

Definition 2 ((Uninterpreted) Lexicon) *Let an alphabet* Σ *and a finite set* B *of basic categories be given. A BCG-lexicon* $\mathbf{LEX}$ *is a finite relation between* Σ^+ *(the set of non-empty strings over* Σ*) and* $\mathbf{CAT}(\mathbf{B})$*.*

Rules of BCG

$$
\begin{aligned}
(x/y)\; y &\;\rightarrow\; x \\
y\; (x \setminus y) &\;\rightarrow\; x
\end{aligned}
$$

Definition 3 (BCG Grammar) *Let an alphabet Σ be given. A BCG grammar $\mathbf{G}$ is a triple $\langle\mathbf{B},\mathbf{LEX},\mathbf{S}\rangle$, where $\mathbf{B}$ is a finite set (the basic categories), $\mathbf{LEX}$ is a finite sub-relation of $\Sigma^+ \times \mathbf{CAT}(\mathbf{B})$, and $\mathbf{S}$ is a finite subset of $\mathbf{CAT}(\mathbf{B})$ (the designated categories).*

Definition 4 *Let $\mathbf{G} = \langle\mathbf{B},\mathbf{LEX},\mathbf{S}\rangle$ be a BCG grammar over the alphabet Σ. Then $\alpha \in L(\mathbf{G})$ iff there are $a_1,\ldots,a_n \in \Sigma^+$, $A_1,\ldots,A_n \in \mathbf{CAT}(\mathbf{B})$, and $S \in \mathbf{S}$ such that*

1. $\alpha = a_1 \ldots a_n$,
2. *For all i such that $1 \leq i \leq n : \langle a_i, A_i\rangle \in \mathbf{LEX}$, and*
3. $A_1,\ldots,A_n \rightarrow^* S$.

Relation to CFGs

- weakly equivalent
- embedding BCG $\rightsquigarrow$ CFG is trivial (only finitely many instances of the BCG rule schemata are needed for a given grammar; can be interpreted as CFG rules)
- embedding CFG $\rightsquigarrow$ BCG difficult to prove (proved in Bar-Hillel, Gaifman and Shamir 1960)
- embedding is straightforward though once you have the **Greibach Normal Form lemma**

Semantics

- semantic type of an expression is homomorphic image of its syntactic category

Definition 5 (Category to type correspondence)
Let τ *be a function from* $\mathbf{CAT}(\mathrm{B})$ *to* $\mathbf{TYPE}$. τ *is a* ***correspondence function*** *iff*

$$\tau(A \setminus B) = \tau(A/B) = \langle \tau(A), \tau(B) \rangle$$

Definition 6 ((Interpreted) Lexicon) *Let an alphabet* Σ, *a finite set* B *of basic categories and a correspondence function* τ *be given. An interpreted BCG-lexicon* $\mathbf{LEX}$ *is a finite sub-relation of*

$$\bigcup_{A \in \mathbf{CAT}(\mathbf{B})} (\Sigma^{+} \times \{A\} \times \mathbf{EXP}_{\tau(A)})$$

semantically annotated rules

$$\begin{aligned} (x/y) : \alpha,\ y : \beta &\quad\rightarrow\quad x : \alpha(\beta) \\ y : \beta,\ (x \setminus y) : \alpha &\quad\rightarrow\quad x : \alpha(\beta) \end{aligned}$$

coordination

- coordination is polymorphic

(1) John walked and Bill talked

(2) John walked and talked

(3) John loves and plays soccer

- general coordination scheme:

$$x \text{ and } x \rightarrow x$$

provided x is a Boolean category

- no syntax without semantics:

$$x : \alpha \text{ and } x : \beta \rightarrow x : \alpha \cap \beta$$

quantifiers

(1) John walked and John talked $\vdash$ John walked and talked

(2) Some man walked and some man talked $\not\vdash$ Some man walked and talked

quantifiers cannot have type e, i.e. category np
good hypothesis: quantifiers have category $s/(s \setminus np)$ and type $\langle\langle e, t\rangle, t\rangle$

(4) John and somebody walked

Names and quantifiers are conjoinable

- Montague: names also have category $s/(s \setminus np)$

Combinators

- alternative solution (Partee and Rooth 1983, among others): **Category of expressions can be changed in syntax!**
- what is needed here:

$$x \rightarrow y/(y \setminus x)$$

- called **Type Lifting** (abbreviated $\mathbf{T}_{>}$)
- usually restricted to few instances
- no syntax without semantics:

$$x : \alpha \rightarrow y/(y \setminus x) : \lambda w.w(\alpha)$$

coordination between names and quantifiers

$$
\cfrac{\cfrac{\cfrac{\cfrac{John}{\mathsf{J'} : np}\,lex}{\lambda x.x\mathsf{J'} : s/(s \setminus np)}\,\mathbf{T}_{>} \quad \cfrac{somebody}{\lambda P.\exists x Px : s/(s \setminus np)}\,lex}{\lambda P.(P\mathsf{J'}) \wedge \exists x Px : s/(s \setminus np)}\,conj \quad \cfrac{walked}{\mathsf{WALK'} : s \setminus np}\,lex}{(\mathsf{WALK'J'}) \wedge \exists x \mathsf{WALK'} x : s}\,\mathbf{A}_{>}
$$

right node raising

- coordination sometimes applies to apparent non-constituents

(5) John likes and Bill detests broccoli

- application of coordination scheme requires that **John likes** has a single Boolean category
- solution: **(forward) function composition B**$_>$

$$(x/y)\ (y/z) \rightarrow (x/z)$$

- name suggests semantics:

$$(x/y) : \alpha,\ (y/z) : \beta \rightarrow (x/z) : \lambda w.\alpha(\beta(w))$$

- combination of lifting and composition gives desired result

$$\dfrac{John}{\text{J'}}\,lex \qquad \dfrac{Bill}{\text{B'}}\,lex$$

$$\dfrac{np}{\lambda x.x\text{J'}}\,\mathbf{T}_{>} \qquad \dfrac{likes}{\text{LIKE'}}\,lex \qquad \dfrac{np}{\lambda x.x\text{B'}}\,\mathbf{T}_{>} \qquad \dfrac{detests}{\text{DETEST'}}\,lex$$

$$\dfrac{s/(s\setminus np) \qquad (s\setminus np)/np}{\lambda y.\text{LIKE'}y\text{J'}}\,\mathbf{B}_{>} \qquad \dfrac{s/(s\setminus np) \qquad (s\setminus np)/np}{\lambda y.\text{DETEST'}y\text{B'}}\,\mathbf{B}_{>}$$

$$\dfrac{s/np \qquad s/np}{\lambda z.(\text{LIKE'}z\text{J'}) \wedge (\text{DETEST'}z\text{B'})}\,Conj \qquad \dfrac{broccoli}{\text{BROCCOLI'}}\,lex$$

$$\dfrac{s/np \qquad np}{(\text{LIKE'BROCCOLI'J'}) \wedge (\text{DETEST'BROCCOLI'B'})}\,\mathbf{A}_{>}$$

$$s$$

Left node raising

- similar “non-constituent coordination” also possible in other direction

(6) John introduced Bill to Sue and Harry to Sally.

- analogous treatment requires mirror images of combinators **T**$_>$ and **B**$_>$
- **backward type lifting (T$_<$)**

$$x : \alpha \rightarrow y \setminus (y/x) : \lambda w.w(\alpha)$$

- **backward function composition (B$_<$)**

$$x \setminus y : \alpha, z \setminus x : \beta \rightarrow z \setminus y : \lambda w.\beta(\alpha(w))$$

Combinators

$$
\dfrac{
\dfrac{\mathit{John}}{\begin{array}{c}\textsf{J'}\\ np\end{array}}\,\mathit{lex}
\qquad
\dfrac{
\dfrac{\mathit{introduced}}{\begin{array}{c}\textsf{INTRODUCE'}\\ tvp/np\end{array}}\,\mathit{lex}
\qquad
\dfrac{
\dfrac{
\dfrac{\dfrac{\mathit{Bill}}{\begin{array}{c}\textsf{B'}\\ np\end{array}}\,\mathit{lex}}{\begin{array}{c}\lambda y.y\textsf{B'}\\ tvp \backslash (tvp/np)\end{array}}\,\mathbf{T}_<
\quad
\dfrac{\dfrac{\dfrac{\mathit{to}}{\begin{array}{c}\lambda x.x\\ pp/np\end{array}}\,\mathit{lex}\quad\dfrac{\mathit{Sue}}{\begin{array}{c}\textsf{SUE'}\\ np\end{array}}\,\mathit{lex}}{\begin{array}{c}\textsf{SUE'}\\ pp\end{array}}\,\mathbf{A}_>}{\begin{array}{c}\lambda w.w\textsf{SUE'}\\ vp \backslash (vp/pp)\end{array}}\,\mathbf{T}_<
}{\begin{array}{c}\lambda u.(u\textsf{SUE'B'})\\ vp \backslash (tvp/np)\end{array}}\,\mathbf{B}_<
\quad
\dfrac{
\dfrac{\dfrac{\mathit{Harry}}{\begin{array}{c}\textsf{H'}\\ np\end{array}}\,\mathit{lex}}{\lambda y.y\textsf{H'} : tvp \backslash (tvp/np)}\,\mathbf{T}_<
\quad
\dfrac{\dfrac{\dfrac{\mathit{to}}{\begin{array}{c}\lambda x.x\\ pp/np\end{array}}\,\mathit{lex}\quad\dfrac{\mathit{Sally}}{\begin{array}{c}\textsf{SA'}\\ np\end{array}}\,\mathit{lex}}{\textsf{SA'} : pp}\,\mathbf{A}_>}{\begin{array}{c}\lambda w.w\textsf{SA'}\\ vp \backslash (vp/pp)\end{array}}\,\mathbf{T}_<
}{\begin{array}{c}\lambda u.(u\textsf{SA'H'})\\ vp \backslash (tvp/np)\end{array}}\,\mathbf{B}_<
}{\begin{array}{c}\lambda uv.(u\textsf{SUE'B'}v) \wedge (u\textsf{SA'H'}v)\\ vp \backslash (tvp/np)\end{array}}\,\mathit{Conj}
}{\begin{array}{c}\lambda v.(\textsf{INTRODUCE'SUE'B'}v) \wedge (\textsf{INTRODUCE'SA'H'}v)\\ vp\end{array}}\,\mathbf{A}_<
}{\begin{array}{c}(\textsf{INTRODUCE'SUE'B'J'}) \wedge (\textsf{INTRODUCE'SA'H'J'})\\ s\end{array}}\,\mathbf{A}_<
$$

tvp *abbreviates* $s \backslash np/pp$

vp *abbreviates* $s \backslash np$

long distance movement

man who ate the apples
apples that the man ate

- lexical entry for relative pronoun

  who, which, that $:= n \setminus n/(s \setminus np) : \lambda QP.P(x) \wedge Q(x)$
  who(m), which, that $:= n \setminus n/(s/np) : \lambda QP.P(x) \wedge Q(x)$

Combinators

$$
\dfrac{
  \dfrac{man}{\begin{array}{c}\text{MAN'}\\ n\end{array}}\,lex
  \qquad
  \dfrac{
    \dfrac{who}{\begin{array}{c}\lambda QP.P(x)\wedge Q(x)\\ n\setminus n/(s\setminus np)\end{array}}\,lex
    \qquad
    \dfrac{
      \dfrac{ate}{\begin{array}{c}\text{EAT'}\\ s\setminus np/np\end{array}}\,lex
      \qquad
      \dfrac{
        \dfrac{the}{\begin{array}{c}\lambda P\iota xP(x)\\ np/n\end{array}}\,lex
        \qquad
        \dfrac{apples}{\begin{array}{c}\text{APPLES'}\\ n\end{array}}\,lex
      }{\begin{array}{c}\iota x\text{APPLES'}\\ np\end{array}}\,\mathbf{A}_>
    }{\begin{array}{c}\text{ATE'}\iota x.\text{APPLES'}x\\ s\setminus np\end{array}}\,\mathbf{A}_>
  }{\begin{array}{c}\lambda Px.Px\wedge\text{ATE'}(\iota y.\text{APPLES'}y)x\\ n\setminus n\end{array}}\,\mathbf{A}_>
}{\begin{array}{c}\lambda x.\text{MAN'}x\wedge\text{ATE'}(\iota y.\text{APPLES'}y)x\\ n\end{array}}\,\mathbf{A}_<
$$

Combinators

$$\dfrac{the}{\lambda P \iota x P(x)}\; lex \qquad \dfrac{man}{\text{MAN'}}\; lex$$

$$\dfrac{\begin{matrix}\lambda P \iota x P(x) \\ np/n\end{matrix} \qquad \begin{matrix}\text{MAN'} \\ n\end{matrix}}{\begin{matrix}\iota x \text{MAN'} \\ np\end{matrix}}\; \mathbf{A}_{>}$$

$$\dfrac{np}{s/(s \setminus np)}\; \mathbf{T}_{>} \qquad \dfrac{ate}{\text{EAT'}}\; lex$$

$$\dfrac{\begin{matrix} s/(s \setminus np) \\ \lambda R.R(\iota x.\text{MAN'}x)\end{matrix} \qquad \begin{matrix}\text{EAT'} \\ s \setminus np/np\end{matrix}}{\begin{matrix} s/np \\ \lambda x.\text{EAT'}x(\iota y.\text{MAN'}y)\end{matrix}}\; \mathbf{B}_{>}$$

$$\dfrac{that}{\lambda QP.P(x) \wedge Q(x)}\; lex$$

$$\dfrac{\begin{matrix}\lambda QP.P(x) \wedge Q(x) \\ n \setminus n/(s/np)\end{matrix} \qquad \begin{matrix} s/np \\ \lambda x.\text{EAT'}x(\iota y.\text{MAN'}y)\end{matrix}}{\begin{matrix}\lambda Px.Px \wedge \text{ATE'}x(\iota y.\text{MAN'}y) \\ n \setminus n\end{matrix}}\; \mathbf{A}_{>}$$

$$\dfrac{apples}{\text{APPLES'}}\; lex$$

$$\dfrac{\begin{matrix}\text{APPLES'} \\ n\end{matrix} \qquad \begin{matrix}\lambda Px.Px \wedge \text{ATE'}x(\iota y.\text{MAN'}y) \\ n \setminus n\end{matrix}}{\begin{matrix}\lambda x.\text{APPLES'}x \wedge \text{ATE'}(\iota y.\text{MAN'}y)x \\ n\end{matrix}}\; \mathbf{A}_{<}$$

relativization

- object relativization in principle unbounded
- can be modeled via repeated forward function composition

a man [who]$_{n\backslash n/(s\backslash np)}$ [(suspects that Chapman) will eat the apples]$_{s\backslash np}$

the apples [that]$_{n\backslash n/(s/np)}$ [Keats (suspects that Chapman) will eat]$_{s\backslash np}$

ECP effects

- extraction of embedded subjects impossible

  a man who [I think that]$_{s/s}$ [Keats likes]$_{s/np}$

  *a man who [I think that]$_{s/s}$ [likes Keats]$_{s\backslash np}$

- likewise, adjuncts are extraction islands

  *a book that Peter died without knowing

- neither extraction can be derived with forward or backward composition and type lifting

non-peripheral extraction

- object gap need not be located at right periphery

packages [which I sent and which you carried]$_{n\backslash n/pp}$ to Philadelphia

people [whom I begged and whom you persuaded]$_{n\backslash n/vp}$ to take a bath

- requires more complex lexical categories for relative pronoun, like

$$n \setminus n/pp/(s/pp/np)$$

- can be schematized to

$$n \setminus n/\$/(s/\$/np)$$

for a small set of possible values of $\$$

- values of $\$$ may be sequences of arguments

Combinators

- also requires generalization of $\mathbf{B}$:

$$x/y : \alpha, y/z_1/\cdots/z_n : \beta \rightarrow x/z_1/\cdots/z_n : \lambda w_1 \cdots w_n.\alpha(\beta(w_n)\cdots(w_1)$$

$$x\backslash y_1\backslash\cdots\backslash y_n : \alpha, z\backslash x : \beta \rightarrow z\backslash y_1\cdots\backslash y_n : \lambda w_1 \cdots w_n.\beta(\alpha(w_n)\cdots(w_1))$$

Pied piping

- extracted element need not be an *wh*-phrase
- can also be a complex NP/PP containing a *wh*-phrase

a report the cover of which Keats (expects that Chapman) will design

a subject on which Keats (expects that Chapman) will speak

a report the height of the lettering on the covers of which the government prescribes

- lexical entry for relative pronoun in pied-piping construction:

$$n \setminus n/(s/np) \setminus (np/np)$$

$$\lambda f P Q x . Q x \wedge P(f x)$$

Combinators

- argument passing (via composition) inside the pied-piped phrase works as in previous examples
- therefore same island constraints for both kinds of unbounded dependencies

*a report [[a man who knows the woman that wrote]$_{np/np}$ which]$_{n\backslash n/(s/np)}$ Keats met

Heavy NP Shift and Crossed Composition

- order of post-verbal material in English rather free

John put the book on the table

John put on the table an extremely heavy book which seemed to be made of stone

- sometimes considered an extra-grammatical phenomenon
- participates in coordination though

John [put on the table]$_{s\backslash np/np}$ and [opened]$_{s\backslash np/np}$ an extremely heavy book which seemed to be made of stone

Combinators

- can be handled with **crossed backward function composition** $\mathbf{B}_{<}^{\times}$

$$x/y,\ z \setminus x \rightarrow z/y$$

- semantics as in harmonic (=non-crossed) function composition

Combinators

$$
\dfrac{\dfrac{John}{np}\,lex \qquad \dfrac{\dfrac{\dfrac{put}{s \backslash np/pp/np}\,lex \qquad \dfrac{\dfrac{\dfrac{on}{pp/np}\,lex \quad \dfrac{\dfrac{the}{np/n}\,lex \quad \dfrac{table}{n}\,lex}{np}\,\mathbf{A}_{>}}{pp}\,\mathbf{A}_{>}}{s \backslash np \backslash (s \backslash np/pp)}\,\mathbf{T}_{<}}{s \backslash np/np}\,\mathbf{B}^{\times}_{<} \qquad \dfrac{Principia}{np}\,lex}{s \backslash np}\,\mathbf{A}_{>}}{s}\,\mathbf{A}_{<}
$$

Combinators

- effect of $\mathbf{B}^{\times}_{<}$:
  - "forward looking gaps" ($/np$) can originate from any linear position
  - "backward looking gaps" ($\backslash np$) must be left peripheral
- seems to cover subject/object asymmetry in English correctly
- **crossed forward composition** would have mirror-image like effect

$$x/y,\ y \setminus z \rightarrow x \setminus z$$

Combinators

$$\dfrac{\dfrac{man}{n}\,lex \qquad \dfrac{\dfrac{who}{n \backslash n/(s \backslash np)}\,lex \qquad \dfrac{\dfrac{I\ think\ that}{s/s}\ \ldots \qquad \dfrac{likes\ Keats}{s \backslash np}\ \ldots}{s \backslash np}\,\mathbf{B}^{\times}_{>}}{n \backslash n}\,\mathbf{A}_{>}}{n}\,\mathbf{A}_{<}$$

Dutch

- recall the Dutch/Swiss German cross-serial dependencies

dat Jan Marie Pieter Arabisch laat zien schrijven
THAT JAN MARIE PIETER ARABIC LET SEE WRITE
'that Jan let Marie see Pieter write Arabic'

- can be dealt with using a generalized version of $\mathbf{B}^{\times}_{>}$

$$x/y,\ y \setminus z/w \rightarrow x \setminus z/w$$

Combinators

- Lexion
  - *Jan, Marie, Pieter, Arabisch* $:= np$
  - *laat* $:= s \setminus np \setminus np/vpi$
  - *zien* $:= vpi \setminus np/vpi$
  - *schrijven* $:= vpi \setminus np$

Combinators

$$
\dfrac{\dfrac{Jan}{np}\,lex \quad \dfrac{\dfrac{Marie}{np}\,lex \quad \dfrac{\dfrac{Pieter}{np}\,lex \quad \dfrac{\dfrac{Arabisch}{np}\,lex \quad \dfrac{\dfrac{\dfrac{laat}{s \backslash np \backslash np / vpi}\,lex \quad \dfrac{zien}{vpi \backslash np / vpi}\,lex}{s \backslash np \backslash np \backslash np / vpi}\,\mathbf{B}^{\times}_{>} \quad \dfrac{schrijven}{vpi \backslash np}\,lex}{s \backslash np \backslash np \backslash np \backslash np}\,\mathbf{B}^{\times}_{>}}{s \backslash np \backslash np \backslash np}\,\mathbf{A}_{<}}{s \backslash np \backslash np}\,\mathbf{A}_{<}}{s \backslash np}\,\mathbf{A}_{<}}{s}\,\mathbf{A}_{<}
$$

Combinators

- how do we prevent derivation of the (ungrammatical)

  dat Jan Marie laat Pieter zien Arabisch schrijven
- solution:
  - availability of combinatorial rules is cross-linguistically parameterized
  - English has $\mathbf{B}^{\times}_{<}$ while Dutch has $\mathbf{B}^{\times}_{>}$ etc.
  - furthermore, **instances** of combinatorial rules may be restricted for a particular language
  - Dutch: forward application

    $$(x/y),\ y \to x$$

    is only licit if the first atom in $y \neq vpi$

Conclusion

- main features of CCG
  - strong connection between syntax and semantics
  - strictly compositional
  - mono-stratal
  - (almost) lexicalized
- differences to other versions of Categorial Grammar
  - language specific parametrization of combinatory rules
  - language specific parametrization of rule instances