Categorial Grammar

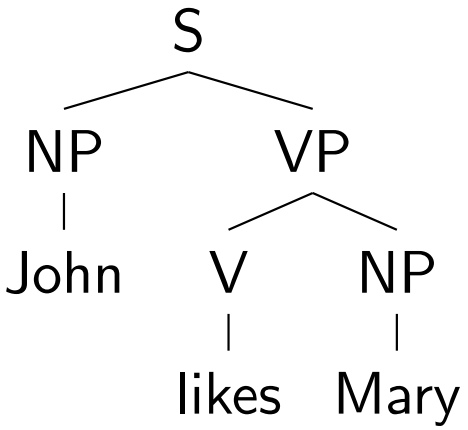
Weiwei Sun

Institute of Computer Science and Technology
Peking University

December 19, 2017

Simple vs. complex category

Phrase Structure Grammar



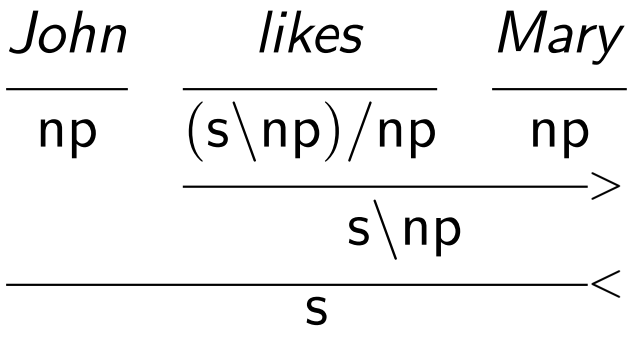
Last two lectures

- type theory
- argument structure
- λ calculus

Today's lecture: Categorial Grammar

$$\dfrac{\dfrac{\textit{John}}{\mathsf{np}} \quad \dfrac{\dfrac{\textit{likes}}{(\mathsf{s}\backslash\mathsf{np})/\mathsf{np}} \quad \dfrac{\textit{Mary}}{\mathsf{np}}}{\mathsf{s}\backslash\mathsf{np}}>}{\mathsf{s}}<$$

What is the meaning of the symbols in this analysis?



Complex category

In a CG, all constituents—and in particular the lexical elements—are associated with a very specific *category* which define their syntactic behaviour.

Simple *phrase-structure* rules

A set of universal rules defines how words and other constituents can be combined according to their categories.

Syntax vs. Semantics

Syntactic and semantic descriptions are tightly connected $\rightarrow$ CG is popular amongst logicians and semanticists.

Definition

The set of syntactic categories $\mathcal{C}$ is defined recursively:

- **Atomic categories**: the grammar for each language is assumed to define a finite set of atomic categories, usually $s, np, n, pp, ... \in \mathcal{C}$
- **Complex categories**: if X and Y $\in \mathcal{C}$, then $X/Y, X\backslash Y \in \mathcal{C}$

Complex categories X/Y or $X\backslash Y$ are functors

- X: a result
- Y: an argument
- $/$: arguments to the right of the functor
- $\backslash$: arguments to the left of the functor

Lexical categories

Complex categories encode subcategorisation information

- intransitive verb: $s\backslash np$ ▷ walked
- transitive verb: $(s\backslash np)/np$ ▷ respected
- ditransitive verb: $((s\backslash np)/np)/np$ ▷ gave

$(s\backslash np)/np$

- the verb takes a noun phrase to its right, and
- another noun phrase to its left to form a sentence.

There is no explicit difference made between phrases and words:
An intransitive verb is described in the same way as a verb phrase with an object: $s\backslash np$.

Lexical categories

Complex categories encode subcategorisation information

- intransitive verb: $s\backslash np$ ▷ walked
- transitive verb: $(s\backslash np)/np$ ▷ respected
- ditransitive verb: $((s\backslash np)/np)/np$ ▷ gave

$(s\backslash np)/np$

- the verb takes a noun phrase to its right, and
- another noun phrase to its left to form a sentence.

There is no explicit difference made between phrases and words:
An intransitive verb is described in the same way as a verb phrase with an object: $s\backslash np$.

Lexical categories

Complex categories encode subcategorisation information

- intransitive verb: $s\backslash np$ ▷ walked
- transitive verb: $(s\backslash np)/np$ ▷ respected
- ditransitive verb: $((s\backslash np)/np)/np$ ▷ gave

$(s\backslash np)/np$

- the verb takes a noun phrase to its right, and
- another noun phrase to its left to form a sentence.

There is no explicit difference made between phrases and words:
An intransitive verb is described in the same way as a verb phrase with an object: $s\backslash np$.

Modification

In CG, adjuncts have the following general form: X\X or X/X.

Example

- PP nominal: (np\np)/np
- PP verbal: ((s\np)\(s\np))/np

Lexicalization (1)

Lexicalization

In a lexicalized grammar, each element of the grammar contains at least one lexical item (terminal).

- G1: S$\rightarrow$ SS, S$\rightarrow$ a
- G2: S$\rightarrow$ aS, S$\rightarrow$ a

Grammar or Lexicon

In a CG,

- the lexicon specifies the categories that the words of a language can take;
- lexical entries do most of the grammatical work of mapping the strings of the language to their interpretations.

Lexicalization (2)

The Principle of Lexical Head Government

Both bounded and unbounded syntactic dependencies are specified by the lexical syntactic type of their head.

Example

(1) a. *John* $\vdash$ np ▷ *John* is a noun phrase.

b. *shares* $\vdash$ np ▷ *shares* is a noun phrase.

c. *buys* $\vdash$ (s\np)/np ▷ *buy* is a transitive verb.

d. *sleeps* $\vdash$ s\np ▷ *sleeps* is an intransitive verb.

e. *well* $\vdash$ (s\np)/(s\np) ▷ *well* can modify a s\np-like thing.

The Principle of Head Categorial Uniqueness

A single nondisjunctive lexical category for the head of a given construction specifies both the bounded dependencies that arise when its complements are in canonical position and the unbounded dependencies that arise when those complements are displaced under relativization, coordination, and the like.

admire $\vdash$ (s\np)/np

(2) a. John admires Mary.

b. the man that I believe that John admires.

c. I believe that John admires and you believe that he dislikes, the woman in the skinny skirt.

AB Categorial Grammar

Variants of categorial grammar differ in the rules they allow.

- The system defined by Ajdukiewicz (1935) and Bar-Hillel (1953) forms the basis for all variants of categorial grammar.
- In AB categorial grammar, categories can only combine through function application.

Forward application

$$X/Y \quad Y \quad \Rightarrow \quad X \qquad (>)$$

Backward application

$$Y \quad X\backslash Y \quad \Rightarrow \quad X \qquad (<)$$

Deriving a string

- A string α is grammatical if each word in the string can be assigned a category (as defined by the lexicon) so that the lexical categories of the words in α can be combined (according to the grammar rules) to form a constituent.
- The process of combining constituents in this manner is called a derivation.

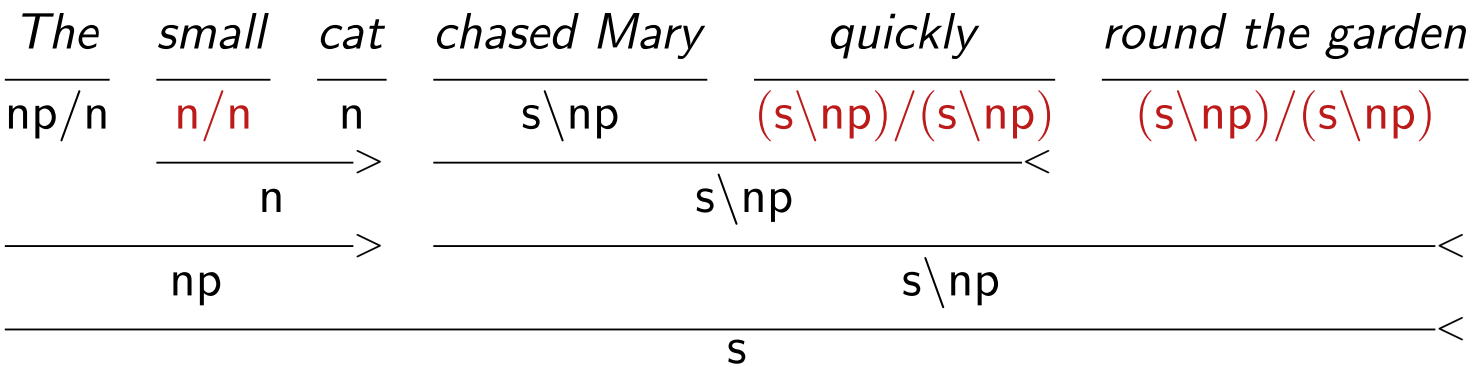
Example

$$\dfrac{\dfrac{\textit{John}}{\mathsf{np}} \quad \dfrac{\dfrac{\textit{buys}}{(\mathsf{s}\backslash\mathsf{np})/\mathsf{np}} \quad \dfrac{\textit{shares}}{\mathsf{np}}}{\mathsf{s}\backslash\mathsf{np}}>}{\mathsf{s}}<$$

Modification

In CG, adjuncts have the following general form: X\X or X/X.

Example



Not all X\X nor X/X are modifiers.

Lexicalization (4)

L. Bloomfield, *Language*

> *The lexicon is really an appendix of the grammar, a list of basic irregularities. This is all the more evident if meanings are taken into consideration, since the meaning of each morpheme belongs to it by an arbitrary tradition.*

CG's view

- If this is the case, nothing in the lexicon is predictable, hence we do not need a theory of the lexicon.
- CG argues that this dichotomy gets in the way of our understanding of how syntax can shape possible lexicons.
  - Any combinatory difference must be lexically specifiable.

Semantic interpretation (1)

Architecture

Syntactic structure (CG derivation) + Lexical interpretation

$$\Downarrow$$

Meaning representation

$$\text{buy} \vdash (\text{s}\backslash\text{np}_1)/\text{np}_2 : \text{buy}{\rightarrow_A}\text{np}_1 \wedge \text{buy}{\rightarrow_P}\text{np}_2$$

- Syntactic category: $(\text{s}\backslash\text{np})/\text{np}$
- Semantics (argument structure): the np indexed with "2" is the Patient of *buy*; the np indexed with "1" is the Agent of *buy*.

Semantic interpretation (2)

$\text{buy} \vdash (\text{s}\backslash\text{np}_1)/\text{np}_2 : \text{buy}\rightarrow_A \text{np}_1 \wedge \text{buy}\rightarrow_P \text{np}_2$

- Syntactic category: $(\text{s}\backslash\text{np})/\text{np}$
- Semantics (argument structure): the np indexed with "2" is the Patient of *buy*; the np indexed with "1" is the Agent of *buy*.

Using a Dependency Interpretation

John	*buys*	*shares*
np	$(\text{s}\backslash\text{np}_1)/\text{np}_2$	np
	$\text{s}\backslash\text{np}_1$ (>)	
s (<)		

Semantic interpretation (2)

$\text{buy} \vdash (\text{s}\backslash\text{np}_1)/\text{np}_2 : \text{buy}{\rightarrow_A}\text{np}_1 \wedge \text{buy}{\rightarrow_P}\text{np}_2$

- Syntactic category: $(\text{s}\backslash\text{np})/\text{np}$
- Semantics (argument structure): the np indexed with "2" is the Patient of *buy*; the np indexed with "1" is the Agent of *buy*.

Using a Dependency Interpretation

John	*buys*	*shares*
$\text{np} : \text{john'}$	$(\text{s}\backslash\text{np}_1)/\text{np}_2 : \text{buy'}{\rightarrow_A}\text{np}_1 \wedge \text{buy'}{\rightarrow_P}\text{np}_2$	$\text{np} : \text{shares'}$
	$\text{s}\backslash\text{np}_1$ (>)	
s (<)		

Semantic interpretation (2)

buy $\vdash (s\backslash np_1)/np_2 : buy\rightarrow_A np_1 \wedge buy\rightarrow_P np_2$

- Syntactic category: $(s\backslash np)/np$
- Semantics (argument structure): the np indexed with "2" is the Patient of *buy*; the np indexed with "1" is the Agent of *buy*.

Using a Dependency Interpretation

$$\dfrac{\dfrac{\dfrac{\textit{John}}{np : john'} \quad \dfrac{\textit{buys}}{(s\backslash np_1)/np_2 : buy'\rightarrow_A np_1 \wedge buy'\rightarrow_P np_2} \quad \dfrac{\textit{shares}}{np : shares'}}{s\backslash np_1 : buy'\rightarrow_A np_1 \wedge buy' \rightarrow_P shares'}>}{s}<$$

Semantic interpretation (2)

$\text{buy} \vdash (\text{s}\backslash\text{np}_1)/\text{np}_2 : \text{buy}{\rightarrow_A}\text{np}_1 \wedge \text{buy}{\rightarrow_P}\text{np}_2$

- Syntactic category: $(\text{s}\backslash\text{np})/\text{np}$
- Semantics (argument structure): the np indexed with "2" is the Patient of *buy*; the np indexed with "1" is the Agent of *buy*.

Using a Dependency Interpretation

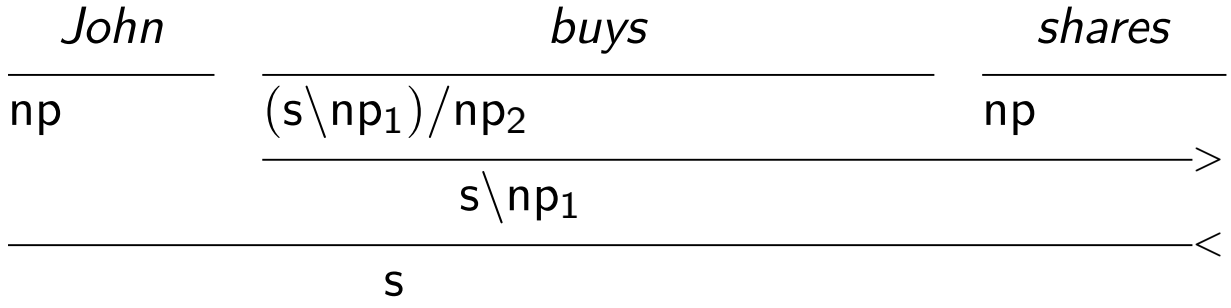
$$\dfrac{\dfrac{\textit{John}}{\text{np} : \text{john'}} \quad \dfrac{\dfrac{\textit{buys}}{(\text{s}\backslash\text{np}_1)/\text{np}_2 : \text{buy'}{\rightarrow_A}\text{np}_1 \wedge \text{buy'}{\rightarrow_P}\text{np}_2} \quad \dfrac{\textit{shares}}{\text{np} : \text{shares'}}}{\text{s}\backslash\text{np}_1 : \text{buy'}{\rightarrow_A}\text{np}_1 \wedge \text{buy'} \rightarrow_P \text{shares'}}>}{\text{s} : \text{buy'} \rightarrow_A \text{john}' \wedge \text{buy'}{\rightarrow_P}\text{shares'}}<$$

Semantic interpretation (3)

buy $\vdash$ $(\mathsf{s}\backslash\mathsf{np}_1)/\mathsf{np}_2 : \lambda x.\lambda y.\mathsf{buy'}(y, x)$

- Syntactic category: $(\mathsf{s}\backslash\mathsf{np})/\mathsf{np}$
- Semantics (λ expression): the np indexed with "2" is the second argument of *buy*; the np indexed with "1" is the first argument of *buy*.

Using λ calculus



Semantic interpretation (3)

buy $\vdash$ $(\mathsf{s}\backslash\mathsf{np}_1)/\mathsf{np}_2 : \lambda x.\lambda y.\mathsf{buy'}(y, x)$

- Syntactic category: $(\mathsf{s}\backslash\mathsf{np})/\mathsf{np}$
- Semantics (λ expression): the np indexed with "2" is the second argument of *buy*; the np indexed with "1" is the first argument of *buy*.

Using λ calculus

$$\dfrac{\dfrac{\dfrac{\textit{John}}{\mathsf{np} : \mathsf{john'}} \quad \dfrac{\textit{buys}}{(\mathsf{s}\backslash\mathsf{np}_1)/\mathsf{np}_2 : \lambda x.\lambda y.\mathsf{buy'}(y, x)} \quad \dfrac{\textit{shares}}{\mathsf{np} : \mathsf{shares'}}}{\mathsf{s}\backslash\mathsf{np}_1}\,{>}}{\mathsf{s}}\,{<}$$

Semantic interpretation (3)

buy $\vdash$ $(\mathrm{s}\backslash\mathrm{np}_1)/\mathrm{np}_2 : \lambda x.\lambda y.\mathrm{buy'}(y, x)$

- Syntactic category: $(\mathrm{s}\backslash\mathrm{np})/\mathrm{np}$
- Semantics (λ expression): the np indexed with "2" is the second argument of *buy*; the np indexed with "1" is the first argument of *buy*.

Using λ calculus

$$\dfrac{\dfrac{\dfrac{\textit{John}}{\mathrm{np} : \mathrm{john'}} \quad \dfrac{\dfrac{\textit{buys}}{(\mathrm{s}\backslash\mathrm{np}_1)/\mathrm{np}_2 : \lambda x.\lambda y.\mathrm{buy'}(y, x)} \quad \dfrac{\textit{shares}}{\mathrm{np} : \mathrm{shares'}}}{\mathrm{s}\backslash\mathrm{np}_1 : \lambda y.\mathrm{buy'}(y, \mathrm{shares'})}>}{\mathrm{s}}<$$

Semantic interpretation (3)

buy $\vdash$ $(\mathrm{s}\backslash\mathrm{np}_1)/\mathrm{np}_2 : \lambda x.\lambda y.\mathrm{buy'}(y, x)$

- Syntactic category: $(\mathrm{s}\backslash\mathrm{np})/\mathrm{np}$
- Semantics (λ expression): the np indexed with "2" is the second argument of *buy*; the np indexed with "1" is the first argument of *buy*.

Using λ calculus

$$\dfrac{\dfrac{\dfrac{\textit{John}}{\mathrm{np} : \mathrm{john'}} \quad \dfrac{\textit{buys}}{(\mathrm{s}\backslash\mathrm{np}_1)/\mathrm{np}_2 : \lambda x.\lambda y.\mathrm{buy'}(y, x)} \quad \dfrac{\textit{shares}}{\mathrm{np} : \mathrm{shares'}}}{\mathrm{s}\backslash\mathrm{np}_1 : \lambda y.\mathrm{buy'}(y, \mathrm{shares'})}>}{\mathrm{s} : \mathrm{buy'(john',shares')}}<$$

Semantic interpretation (4)

buy $\vdash$ $(\mathsf{s}\backslash\mathsf{np}_1)/\mathsf{np}_2 : \lambda x.\lambda y.\mathsf{buy'}(y, x)$

- Syntactic category: $(\mathsf{s}\backslash\mathsf{np})/\mathsf{np}$
- Semantic type (type theory): "buy" $\Rightarrow \langle e, \langle e, t\rangle\rangle$
  Type theory could easily admit and handle n-place functors; such a functor can take the n arguments one by one

Using types

$$\dfrac{\dfrac{\dfrac{\textit{John}}{\mathsf{np}} \quad \dfrac{\textit{buys}}{(\mathsf{s}\backslash\mathsf{np}_1)/\mathsf{np}_2} \quad \dfrac{\textit{shares}}{\mathsf{np}}}{\mathsf{s}\backslash\mathsf{np}_1}\,{>}}{\mathsf{s}}\,{<}$$

Semantic interpretation (4)

buy $\vdash$ $(\mathsf{s}\backslash\mathsf{np}_1)/\mathsf{np}_2 : \lambda x.\lambda y.\mathsf{buy'}(y, x)$

- Syntactic category: $(\mathsf{s}\backslash\mathsf{np})/\mathsf{np}$
- Semantic type (type theory): "buy" $\Rightarrow \langle e, \langle e, t\rangle\rangle$
  Type theory could easily admit and handle n-place functors; such a functor can take the n arguments one by one

Using types

$$\dfrac{\dfrac{\textit{John}}{\mathsf{np} : \mathsf{john'}@e} \quad \dfrac{\dfrac{\textit{buys}}{(\mathsf{s}\backslash\mathsf{np}_1)/\mathsf{np}_2 : \mathsf{buy'}@\langle e, \langle e, t\rangle\rangle} \quad \dfrac{\textit{shares}}{\mathsf{np} : \mathsf{shares'}@e}\,>}{\mathsf{s}\backslash\mathsf{np}_1}}{\mathsf{s}}\,<$$

Semantic interpretation (4)

buy $\vdash$ $(\mathsf{s}\backslash\mathsf{np}_1)/\mathsf{np}_2 : \lambda x.\lambda y.\mathsf{buy'}(y, x)$

- Syntactic category: $(\mathsf{s}\backslash\mathsf{np})/\mathsf{np}$
- Semantic type (type theory): "buy" $\Rightarrow \langle e, \langle e, t\rangle\rangle$
  Type theory could easily admit and handle n-place functors; such a functor can take the n arguments one by one

Using types

$$\dfrac{\dfrac{\textit{John}}{\mathsf{np} : \mathsf{john'}@e} \quad \dfrac{\dfrac{\textit{buys}}{(\mathsf{s}\backslash\mathsf{np}_1)/\mathsf{np}_2 : \mathsf{buy'}@\langle e, \langle e, t\rangle\rangle} \quad \dfrac{\textit{shares}}{\mathsf{np} : \mathsf{shares'}@e}}{\mathsf{s}\backslash\mathsf{np}_1 : \mathsf{buy'}(\mathsf{shares'})@\langle e, t\rangle}>}{\mathsf{s}}<$$

Semantic interpretation (4)

buy $\vdash$ $(\mathsf{s}\backslash\mathsf{np}_1)/\mathsf{np}_2 : \lambda x.\lambda y.\mathsf{buy'}(y, x)$

- Syntactic category: $(\mathsf{s}\backslash\mathsf{np})/\mathsf{np}$
- Semantic type (type theory): "buy" $\Rightarrow \langle e, \langle e, t\rangle\rangle$
  Type theory could easily admit and handle n-place functors; such a functor can take the n arguments one by one

Using types

$$\dfrac{\dfrac{\textit{John}}{\mathsf{np} : \mathsf{john'}@e} \quad \dfrac{\dfrac{\textit{buys}}{(\mathsf{s}\backslash\mathsf{np}_1)/\mathsf{np}_2 : \mathsf{buy'}@\langle e, \langle e, t\rangle\rangle} \quad \dfrac{\textit{shares}}{\mathsf{np} : \mathsf{shares'}@e}}{\mathsf{s}\backslash\mathsf{np}_1 : \mathsf{buy'}(\mathsf{shares'})@\langle e, t\rangle}>}{\mathsf{s} : \mathsf{buy'}(\mathsf{john'})(\mathsf{shares'})@t}<$$

The principle of type transparency

The principle of Categorial Type Transparency

For a given language, the semantic type of the interpretation together with a number of language-specific directional parameter settings uniquely determines the syntactic category of a category.

The inverse of Type Transparency

For any category, the semantic type is a function of the syntactic type.

- CGs put into the lexicon most of the information that is captured in CFG rules.
- In CGs, all constituents and lexical elements are associated with a syntactic "category."
- In CGs, syntactic information is tightly related to semantic information.

Example

S $\rightarrow$ NP VP

VP $\rightarrow$ TV NP

TV $\rightarrow$ married|finds|...

married := (s\np)/np.