CS447: Natural Language Processing

http://courses.engr.illinois.edu/cs447

Lecture 15: Compositional Semantics

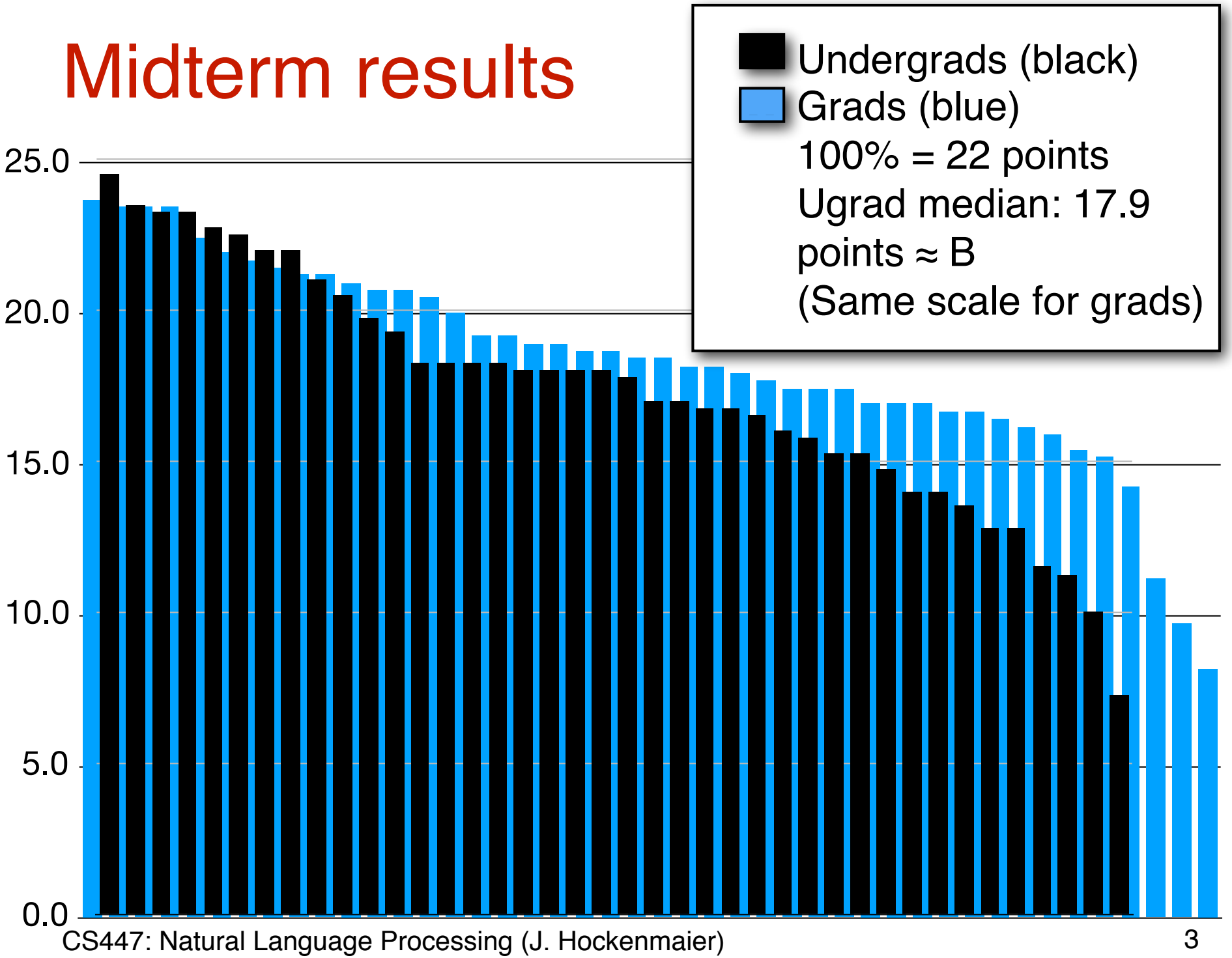
Julia Hockenmaier

juliahmr@illinois.edu

3324 Siebel Center

Admin

Midterm results



Midterm results

Converting **points to percentages:**

- 22 out of 25 points = 100% (you will soon see both in Compass)

Converting **points/percentages to letter grades:**

The final conversion will be based on the **total percentage** at the end of the semester (MPs, midterm, final, (project))
I use the **undergrads' performance** as yardstick for everybody

If I had to give letter grades for this midterm, here is a *rough* scale:

- You would need **19 points (~86%)** or more to get an A
- The **undergrad median (17.9 points = 81.4%)** would correspond to a **B** letter grade
- You would need **at least 40% (9 points)** to pass the class.

Regrade requests

We will post solutions on the class website (securely).

We will accept regrade requests until Nov 9.

How can you do better?

Come to class, and participate.

Spend time with the material after each lecture.

Read the textbook.

Use Piazza.

Come to office hours.

Let us know if you struggle.

4th Credit hour: Proposal

Upload a **one-page PDF** to Compass by this Friday

- written in **LaTeX** (not MS Word)
- with **full bibliography** of the papers you want to read or base your project on

(ideally with **links to online versions**; add url-field to your bibtex file)

- include a **motivation** of why you have chosen those papers
- for a research project: tell me whether you have the **data** you need, what **existing software** you will be using, what you will have to **implement yourself.**
- mention any **questions/concerns** that you may have
- include your names and NetId/Illinois emails
- one proposal per project is fine.

Back to the material…

Semantics

In order to understand language, we need to know its meaning.

- What is the meaning of a word?
  (**Lexical semantics**)

- What is the meaning of a sentence?
  ([Compositional] semantics)

- What is the meaning of a longer piece of text?
  (Discourse semantics)

Natural language conveys information about the world

We can compare statements about the world with the actual state of the world:

Champaign is in California. (false)

We can learn new facts about the world from natural language statements:

The earth turns around the sun.

We can answer questions about the world:

Where can I eat Korean food on campus?

We draw inferences from natural language statements

Some inferences are purely linguistic:

All blips are foos.

Blop is a blip.

Blop is a foo (whatever that is).

Some inferences require world knowledge.

Mozart was born in Salzburg.

Mozart was born in Vienna.

No, that can't be - these are different cities.

Today's lecture

Our initial question:

What is the meaning of (declarative) sentences?

Declarative sentences: *"John likes coffee"*.

(We won't deal with questions (*"Who likes coffee?"*) and imperative sentences (commands: *"Drink up!"*))

Follow-on question 1:

How can we represent the meaning of sentences?

Follow-on question 2:

How can we map a sentence to its meaning representation?

What do nouns and verbs mean?

In the simplest case, an NP is just a name: *John*
Names refer to entities in the world.

Verbs define n-ary predicates: depending on the arguments they take (and the state of the world), the result can be true or false.

What do sentences mean?

Declarative sentences (statements) can be true or false, depending on the state of the world:

John sleeps.

In the simplest case, the consist of a verb and one or more noun phrase arguments.

Principle of compositionality (Frege):
The meaning of an expression depends on the meaning of its parts and how they are put together.

First-order predicate logic (FOL) as a meaning representation language

Predicate logic expressions

Terms: refer to entities

- Variables: x, y, z
- Constants: John', Urbana'
- Functions applied to terms (fatherOf(John'))

Predicates: refer to properties of, or relations between, entities

- tall'(x), eat'(x,y), …

Formulas: can be true or false

- Atomic formulas: predicates, applied to terms: tall'(John')
- Complex formulas: constructed recursively via logical connectives and quantifiers

Formulas

Atomic formulas are predicates, applied to terms:

book(x), eat(x,y)

Complex formulas are constructed recursively by

...**negation** (¬): $\neg book(John')$

...**connectives** (∧,∨,→): $book(y) \wedge read(x,y)$

conjunction (and): $\varphi \wedge \psi$ disjunction (or): $\varphi \vee \psi$ implication (if): $\varphi \rightarrow \psi$

...**quantifiers** (∀x, ∃x)

universal (typically with implication) $\forall x[\varphi(x) \rightarrow \psi(x)]$

existential (typically with conjunction) $\exists x[\varphi(x)]$, $\exists x[\varphi(x) \wedge \psi(x)]$

Interpretation: formulas are either **true or false**.

The syntax of FOL expressions

Term ⇒ Constant |
Variable |
Function(Term,...,Term)

Formula ⇒ Predicate(Term, ...Term) |
¬ Formula |
∀ Variable Formula |
∃ Variable Formula |
Formula ∧ Formula |
Formula ∨ Formula |
Formula → Formula

Some examples

John is a student:
$\text{student}(\text{john})$

All students take at least one class:
$\forall x\ \text{student}(x) \rightarrow \exists y(\text{class}(y) \wedge \text{takes}(x,y))$

There is a class that all students take:
$\exists y(\text{class}(y) \wedge \forall x\ (\text{student}(x) \rightarrow \text{takes}(x,y))$

FOL is sufficient for many Natural Language inferences

All blips are foos. $\forall x\ blip(x) \rightarrow foo(x)$

Blop is a blip. $blip(blop)$

Blop is a foo $foo(blop)$

Some inferences require world knowledge.

Mozart was born in Salzburg. $bornIn(Mozart, Salzburg)$

Mozart was born in Vienna. $bornIn(Mozart, Vienna)$

No, that can't be- $bornIn(Mozart, Salzburg)$

these are different cities $\wedge \neg bornIn(Mozart, Salzburg)$

Not all of natural language can be expressed in FOL:

Tense:

It was hot yesterday.

I will go to Chicago tomorrow.

Modals:

You can go to Chicago from here.

Other kinds of quantifiers:

Most students hate 8:00am lectures.

λ-Expressions

We often use **λ-expressions**
to construct complex logical formulas:

- $\lambda x.\varphi(..x...)$ is a **function** where x is a variable, and φ some FOL expression.

- **β-reduction** (called λ-reduction in textbook):
  Apply $\lambda x.\varphi(..x...)$ to some argument a:
  $(\lambda x.\varphi(..\mathrm{x}...)\ a) \Rightarrow \varphi(..\mathrm{a}...)$
  Replace all occurrences of x in $\varphi(..\mathrm{x}...)$ with a

- **n-ary functions** contain embedded λ-expressions:
  $\lambda x.\lambda y.\lambda z.give(\mathrm{x},\mathrm{y},\mathrm{z})$

CCG: the machinery

Categories:

specify subcat lists of words/constituents.

Combinatory rules:

specify how constituents can combine.

The lexicon:

specifies which categories a word can have.

Derivations:

spell out process of combining constituents.

(Combinatory) Categorial Grammar

CCG categories

Simple (atomic) categories: NP, S, PP

Complex categories (functions):

Return a **result** when combined with an **argument**

VP, intransitive verb	**S\NP**
Transitive verb	**(S\NP)/NP**
Adverb	**(S\NP)\(S\NP)**
Prepositions	**((S\NP)\(S\NP))/NP** **(NP\NP)/NP** **PP/NP**

Function application

Forward application (>):

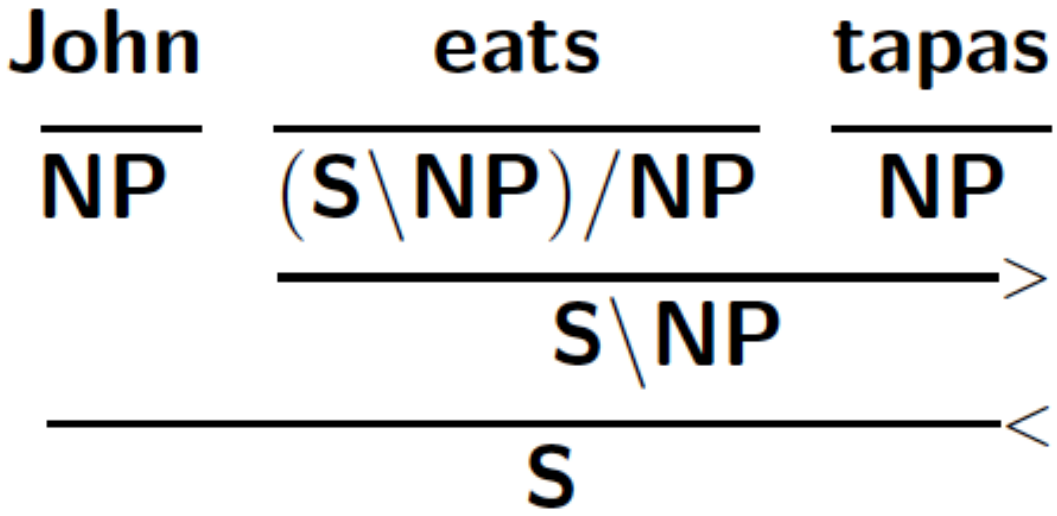
(S\NP)/NP	**NP**	$\Rightarrow_>$	**S\NP**
eats	tapas		eats tapas

Backward application (<):

NP	**S\NP**	$\Rightarrow_<$	**S**
John	eats tapas		John eats tapas

Used in all variants of categorial grammar

A (C)CG derivation



Function application

Combines a function X/Y or X\Y with its argument Y to yield the result X:

(S\NP)/NP	NP	->	S\NP
eats	tapas		eats tapas

NP	S\NP	->	S
John	eats tapas		John eats tapas

Type-raising and composition

Type-raising: X → T/(T\X)

Turns an argument into a function.

NP	→	S/(S\NP)	(subject)
NP	→	(S\NP)\((S\NP)/NP)	(object)

Harmonic composition: X/Y Y/Z → X/Z

Composes two functions (complex categories)

(S\NP)/PP	PP/NP	→	(S\NP)/NP
S/(S\NP)	(S\NP)/NP	→	S/NP

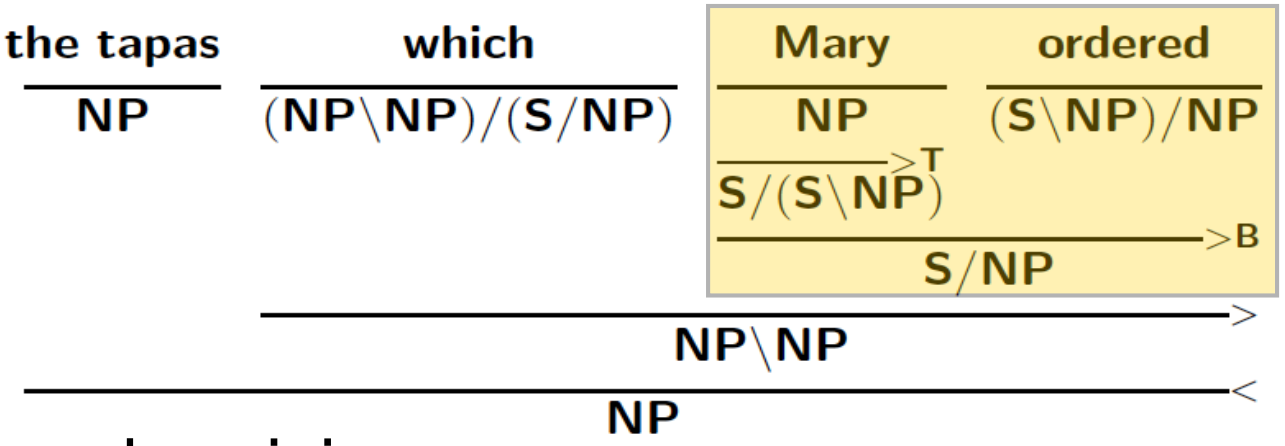
Crossing function composition: X/Y Y\Z → X\Z

Composes two functions (complex categories)

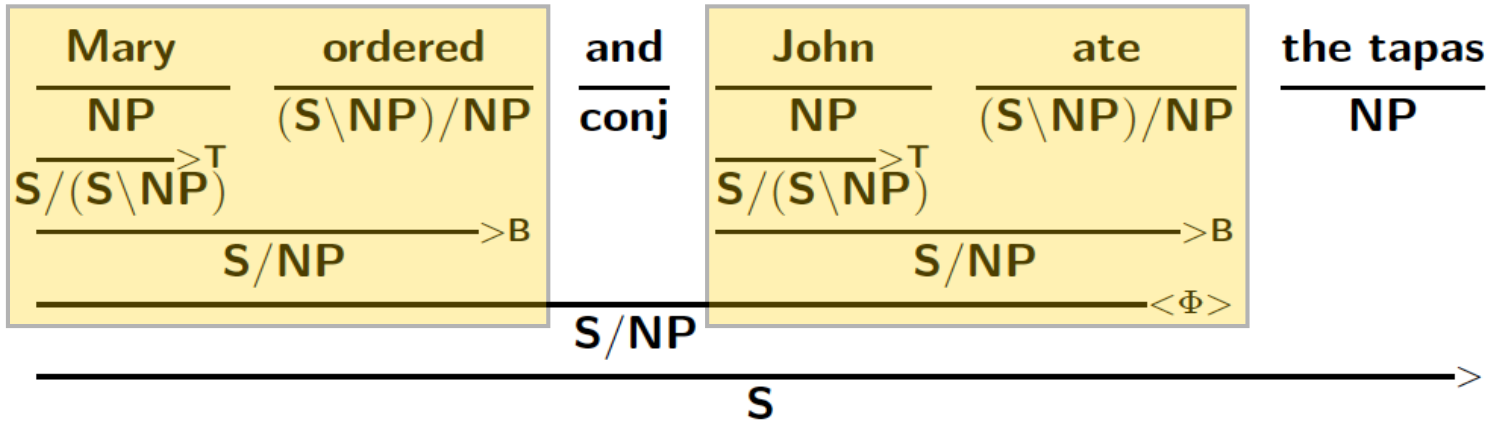
(S\NP)/S	S\NP	→	(S\NP)\NP

Type-raising and composition

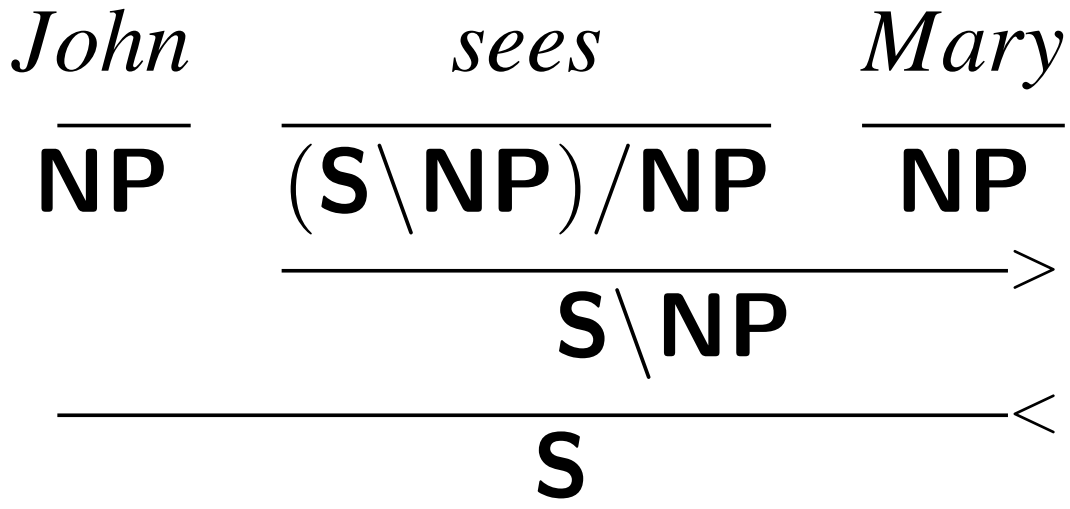
Wh-movement (relative clause):



Right-node raising:



An example



Using Combinatory Categorial Grammar (CCG) to map sentences to predicate logic

CCG semantics

Every syntactic constituent has a semantic interpretation:

Every **lexical entry** maps a word to a syntactic category and a corresponding semantic type:

John=(**NP**, john') Mary= (**NP**, mary')
loves: (**(S\NP)/NP** λx.λy.loves(x,y))

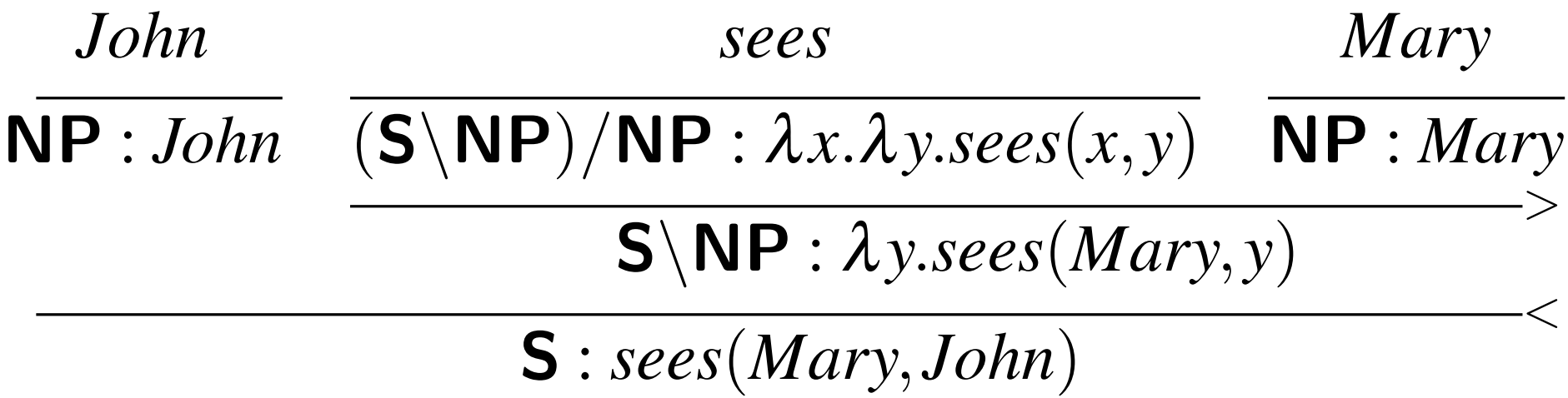
Every **combinatory rule** has a syntactic and a semantic part:

Function application: $\mathbf{X/Y}{:}\lambda x.f(x)\ \ \mathbf{Y}{:}a \rightarrow \mathbf{X}{:}f(a)$

Function composition: $\mathbf{X/Y}{:}\lambda x.f(x)\ \ \mathbf{Y/Z}{:}\lambda y.g(y) \rightarrow \mathbf{X/Z}{:}\lambda z.f(\lambda y.g(y).z)$

Type raising: $\mathbf{X}{:}a \rightarrow \mathbf{T/(T\backslash X)}\ \lambda f.f(a)$

An example with semantics



Supplementary material: quantifier scope ambiguities in CCG

Quantifier scope ambiguity

"Every chef cooks a meal"

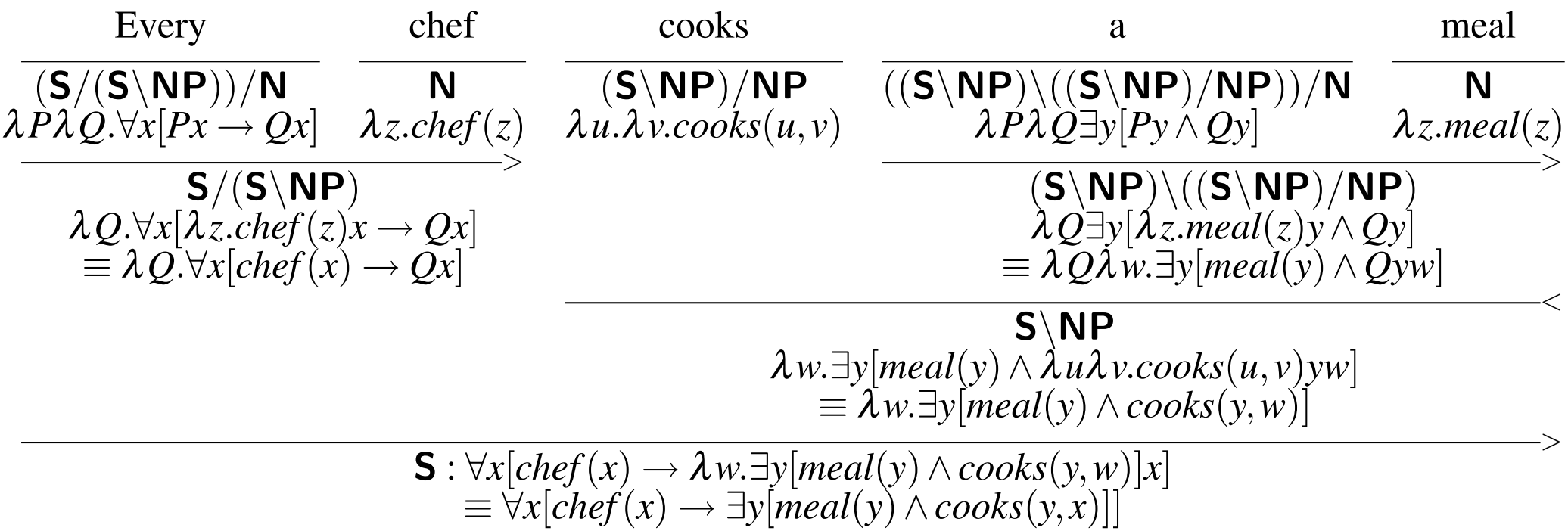
- **Interpretation A:**
  For every chef, there is a meal which he cooks.

$$\forall x[chef(x) \rightarrow \exists y[meal(y) \wedge cooks(y,x)]]$$

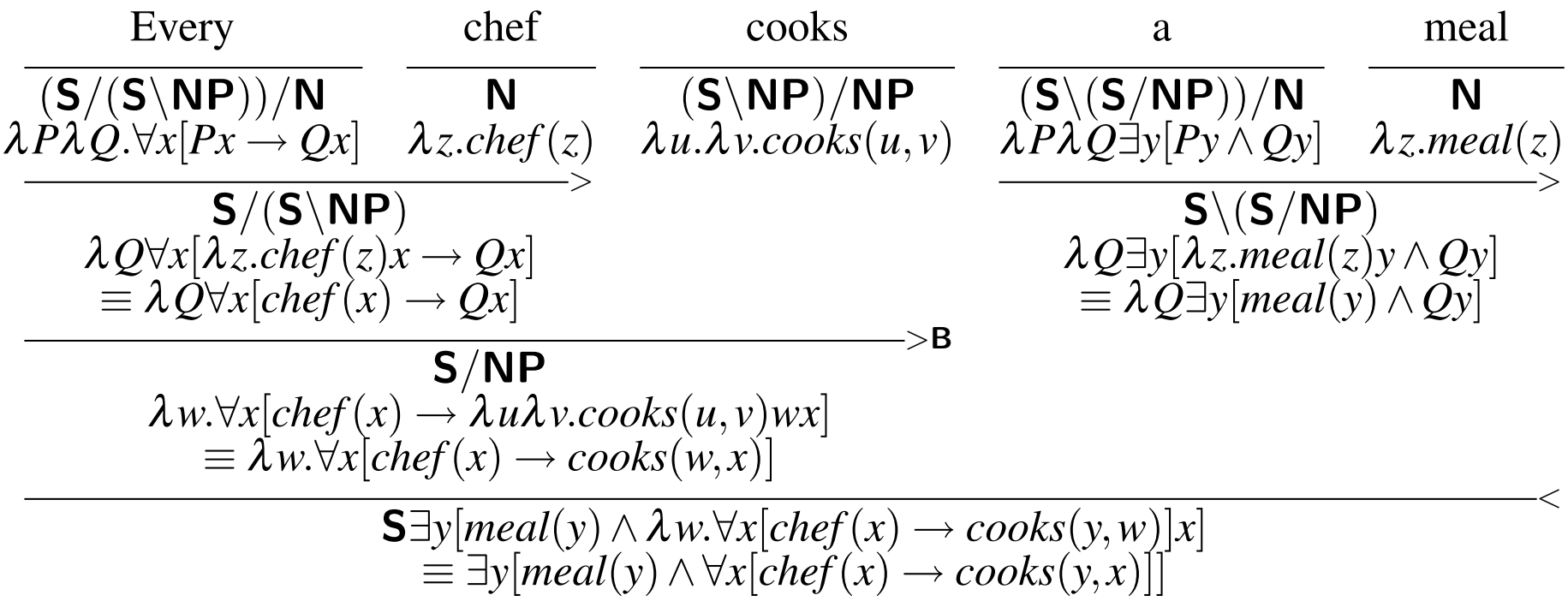
- **Interpretation B:**
  There is some meal which every chef cooks.

$$\exists y[meal(y) \wedge \forall x[chef(x) \rightarrow cooks(y,x)]]$$

Interpretation A



Interpretation B



Additional topics

Representing events and temporal relations:

- Add event variables *e* to represent the events described by verbs, and temporal variables *t* to represent the time at which an event happens.

Other quantifiers:

- What about "*most* / *at least two* / … *chefs*"?

Underspecified representations:

- Which interpretation of *"Every chef cooks a meal"* is correct? This might depend on context. Let the parser generate an underspecified representation from which both readings can be computed.

Going beyond single sentences:

- How do we combine the interpretations of single sentences?