Logical Forms

Prof. Sameer Singh

CS 295: STATISTICAL NLP

WINTER 2017

February 16, 2017

Based on slides from Noah Smith, Dan Klein, Tom Kwiatkowski, and everyone else they copied from.

Outline

Logical Semantics

Combinatory Categorical Grammar

Outline

Logical Semantics

Combinatory Categorical Grammar

So far….

Meaning of Words

- Word Vectors
- Parts of Speech
- Named Entities
- Word senses
- ….

Meaning of Verbs

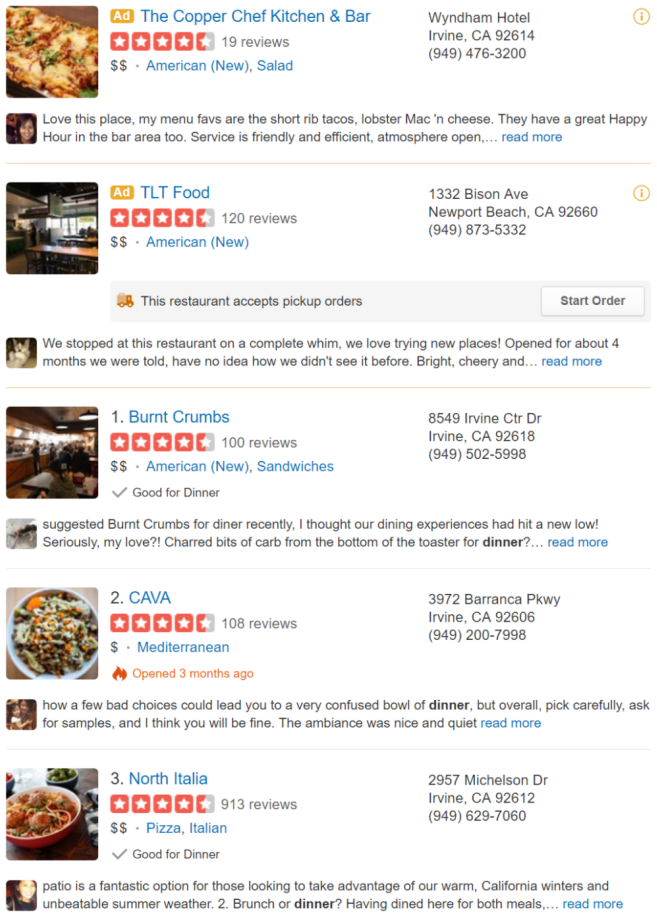
- Context-free grammars
- Thematic Roles
- Semantic Roles
- Dependency Relations
- …

Still a gap between language and actionable representations

Language "World"

What is a good Korean restaurant near UCI campus?

Meaning Representation



A (Tiny) World Model

Domain

Amir, Brook, Chen, …
Gogi Grill, Eureka, Cha Tea, UCI, …
Korean, American, Beverages, ..

a, b, c, ...
gg, er, ct, uci, ..
ko, am, be, ...

Properties

Amir, Brook, Chen, … are humans
Gogi Grill is good, Cha Tea has a long wait,
Eureka is noisy, They are restaurants..

Humans = {a, b, c, ...}
Good = {gg} Noisy={er}
Restaurant={gg,er,ct} ...

Relations

Gogi serves Korean, Eureka serves American,
Cha Tea serves Beverages, Amir likes Gogi,
Chen likes Korean, …

Serves = {(gg,ko),(er,am),
(ct,be), ...}
Likes={(a,gg),(c,ko),..} ...

Is Eureka noisy? — er in Noisy?
Does Cha Tea serve beverages? — (ct,be) in Serves?
What does Amir like? — list (a,?) in Likes

First-Order Logic

Terms

- Constant: a,b,c,gg,ct
- Variables: x,y,z

Relations

- Unary: ~~Serves(x)~~ Good(x), Noisy(x)
- Binary: Likes(x,y)
- n-ary: $R(a_1,..,a_n)$

Formula

- n-ary relation, R, and n terms $(t_1,..,t_n)$, then $R(t_1,..,t_n)$ is a formula
- If F is a formula, then so is $\neg F$
- Boolean operators: $F \vee F$, $F \wedge F$, $F \rightarrow F$
- Quantifiers: $\exists x\ F$, $\forall x\ F$

Likes(Amir,x), Noisy(er)

$\neg$ Good(er)

Likes(Amir,x) $\wedge$ Serves(x,ko)

Translating b/w FoL and NL

- Gogi is not loud $\neg Loud(gg)$
- Some humans like American $\exists x \; Human(x) \wedge Likes(x, am)$
- If a person likes Eureka, they aren't friends with Brook $\forall x \; Person(x) \wedge Likes(x, er) \Rightarrow \neg Friends(X, Brook)$
- Every restaurant has a long wait or is disliked by Amir $\forall x \; Rest(x) \Rightarrow Wait(x) \vee \neg Likes(a, x)$
- Everybody has something they don't like $\forall x \; \exists y \; \neg Likes(x, y)$
- There exists something that nobody likes $\exists y \; \forall x \; \neg Likes(x, y)$

Logical Semantics

Everybody has something they don't like.

$\forall x \exists y \neg Likes(x,y)$

The denotation of a natural language sentence is the set of conditions that must hold in the (model) world for the sentence to be true.

This is called the logical form of the sentence.

- Less ambiguous
- Can check truth value by querying a database
- If you know it's true, you can update database
- Questions become queries on the database
- Comprehending a document is same as chaining

λ-Calculus

Abstraction

If φ is a FOL formula

$\varphi \equiv$ Likes(Amir, v)

$\lambda v . \varphi$ $\quad v \rightarrow \{T, F\}$

def anon(v) { φ }

Application

$\lambda v . \varphi + \psi$

$[\lambda v . \varphi](\psi)$

anon(ψ)

[def anon(v){φ}](ψ)

Example of λ-applications

$[\lambda x.\ \text{Likes}(a,x)](b)$ $\quad$ $\text{Likes}(a,b)$

$[\lambda x\ \lambda y\ \text{Friends}(x,y)](a)$ $\quad$ $\lambda y\ \text{Friends}(a,y)$

$[\lambda y\ \text{Friends}(a,y)](c)$ $\quad$ $\text{Friends}(a,c)$

$[\lambda x \lambda y\ \text{Likes}(a,x) \wedge \text{Serves}(x,y)]$

(gg)

$[\lambda y\ \text{Likes}(a,gg) \wedge \text{Serves}(gg,y)]$

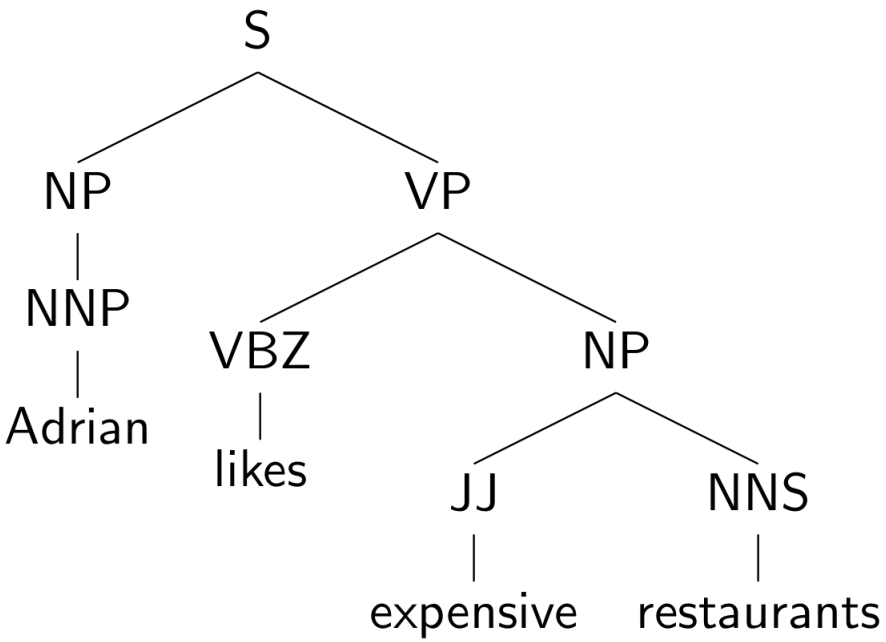
(k_o)

$\text{Likes}(a,gg) \wedge \text{Serves}(gg,k_o)$

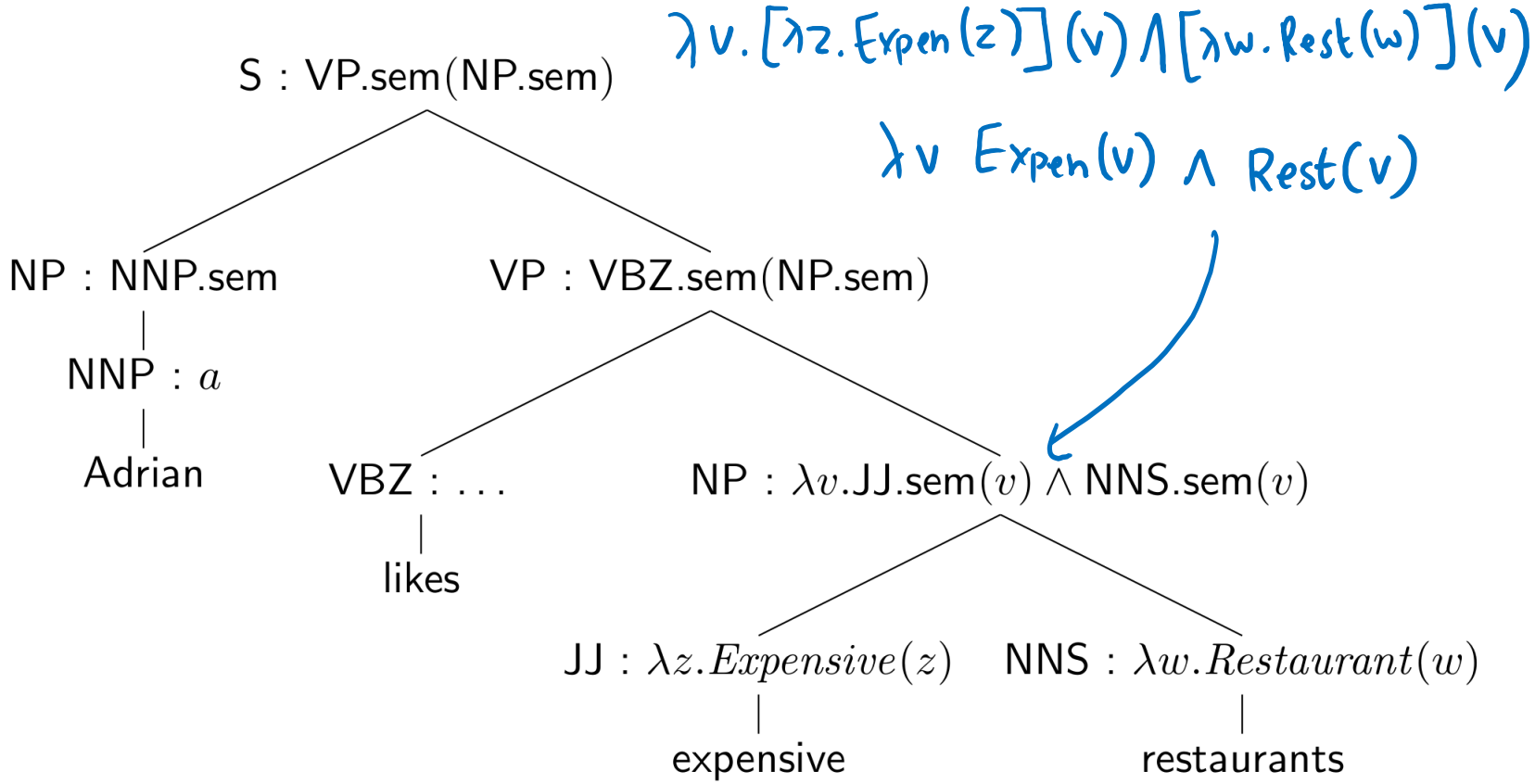
Semantic Attachments to CFG

- NNP $\rightarrow$ Adrian $\{a\}$
- VBZ $\rightarrow$ likes $\{\lambda f.\lambda y.\forall x f(x) \Rightarrow Likes(y, x)\}$
- JJ $\rightarrow$ expensive $\{\lambda x.Expensive(x)\}$
- NNS $\rightarrow$ restaurants $\{\lambda x.Restaurant(x)\}$
- NP $\rightarrow$ NNP $\{$NNP.sem$\}$
- NP $\rightarrow$ JJ NNS $\{\lambda x.$JJ.sem$(x) \wedge$ NNS.sem$(x)\}$
- VP $\rightarrow$ VBZ NP $\{$VBZ.sem(NP.sem)$\}$
- S $\rightarrow$ NP VP $\{$VP.sem(NP.sem)$\}$

CFG to λ-Calculus



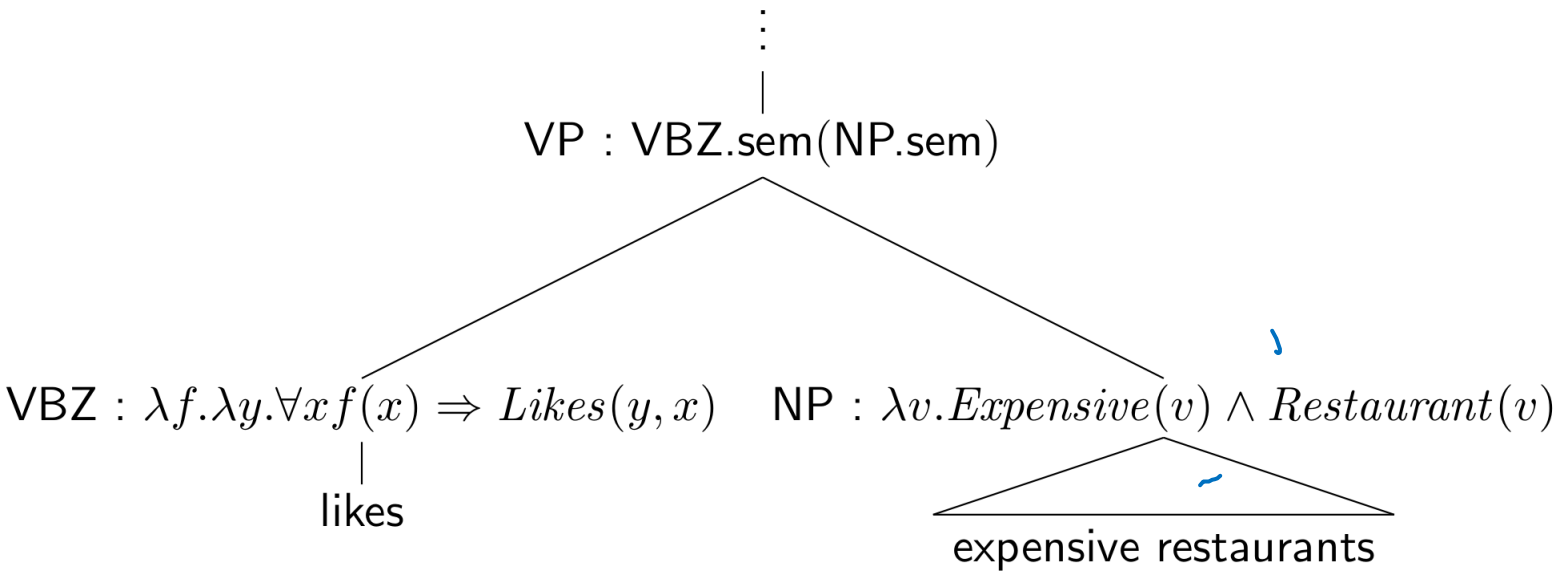
CFG to λ-Calculus



CFG to λ-Calculus

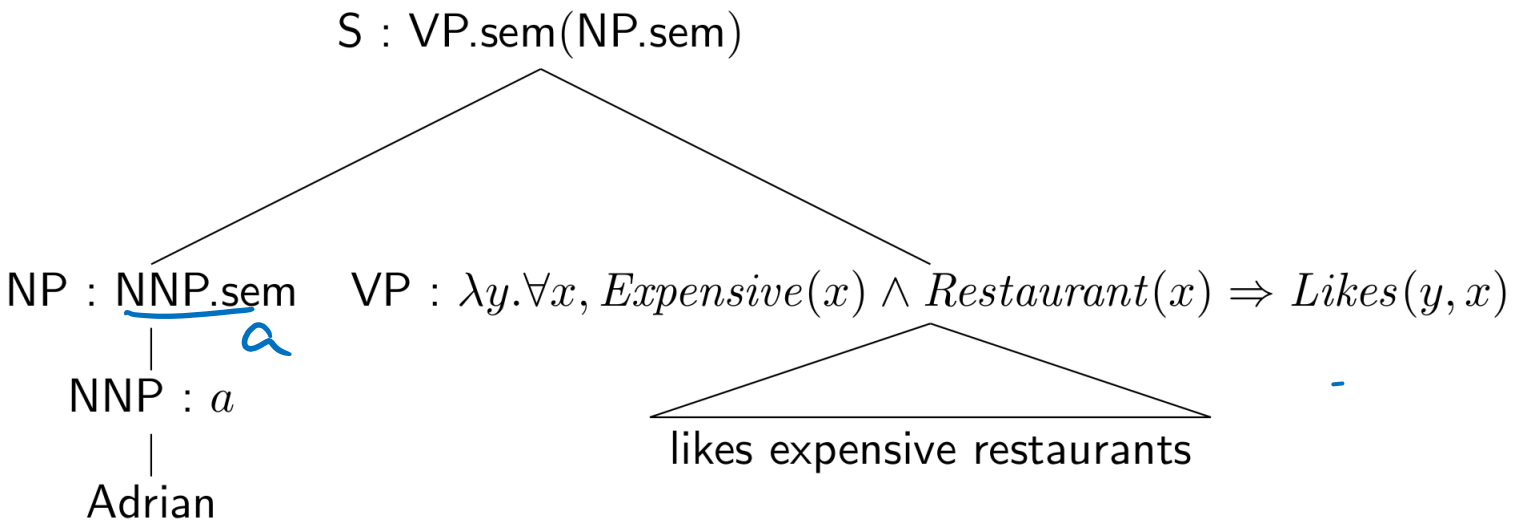
$[\lambda f.\lambda y\, \forall x\, f(x) \Rightarrow Likes(y,x)]\,(\lambda v.\, Expen(v) \wedge Rest(v))$

$\lambda y\, \forall x\quad Expen(x) \wedge Rest(x) \Rightarrow Likes(y,x)$



CFG to λ-Calculus

$\forall x \; Expen(x) \wedge Rest(x) \Rightarrow Likes(a, x)$



Tricky Cases: Transitive Verbs

I eat food. $\lambda y \lambda x \; eat(y, x)$

I eat. $\lambda y \; eat(y)$

Tricky Cases: Indefinites

Bob ate a waffle.
Amy ate a waffle.

$ate(bob, waffle)$
$ate(alice, waffle)$

$\exists x : waffle(x) \wedge ate(bob, x)$

$\exists x : waffle(x) \wedge ate(alice, x)$

Tricky Cases: Tenses and Events

Alice danced.

$danced(alice)$

$\exists e \; dance(e) \wedge agent(e, alice) \wedge start(e) < \text{"now"}$

Alice had been dancing when Bob sneezed.

$\exists e \exists e' \; dance(e) \wedge agent(e, a) \wedge sneeze(e') \wedge start(e) < start(e') \wedge agent(e', bob) \wedge end(e) = end(e') \wedge time(e') < now$

Tricky Cases: Adverbs

Bob sings terribly.

terribly (sings (bob))

$\forall e \; present(e) \wedge type(e, sing)$

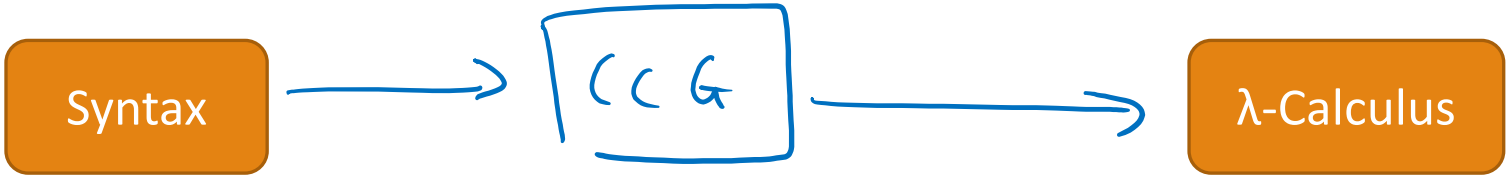
$agent(e, bob) \rightarrow manner(e, terrible$

Outline

Logical Semantics

Combinatory Categorical Grammar

Combinatory Categorical Grammar

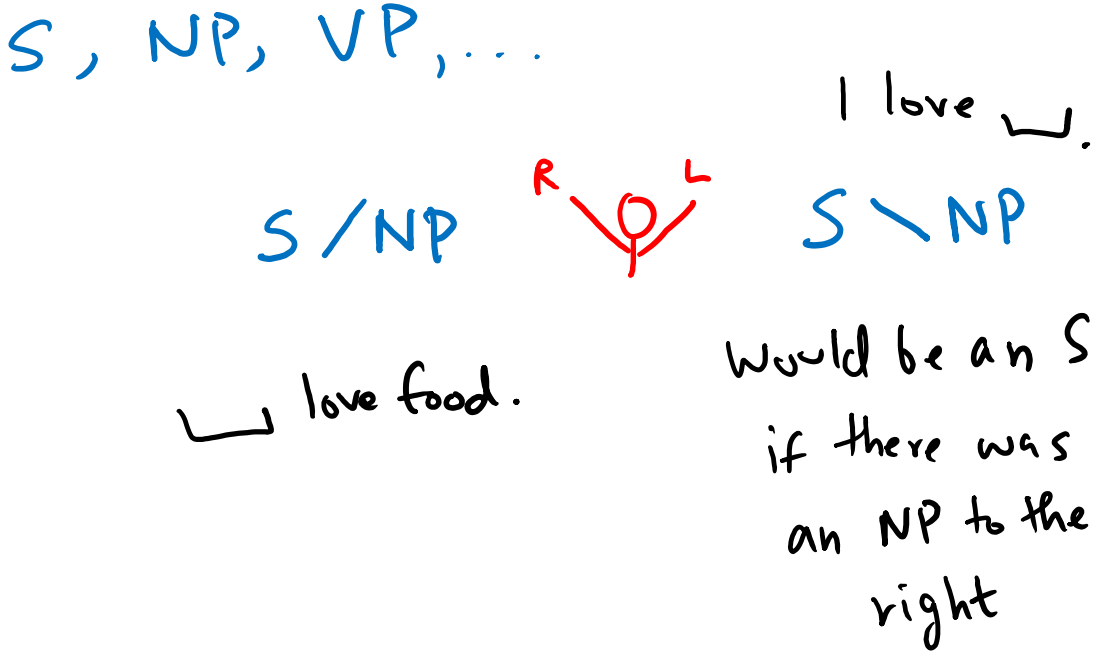


CCG Types

Instead of non-terminals, it has infinitely large set of **categories or types**

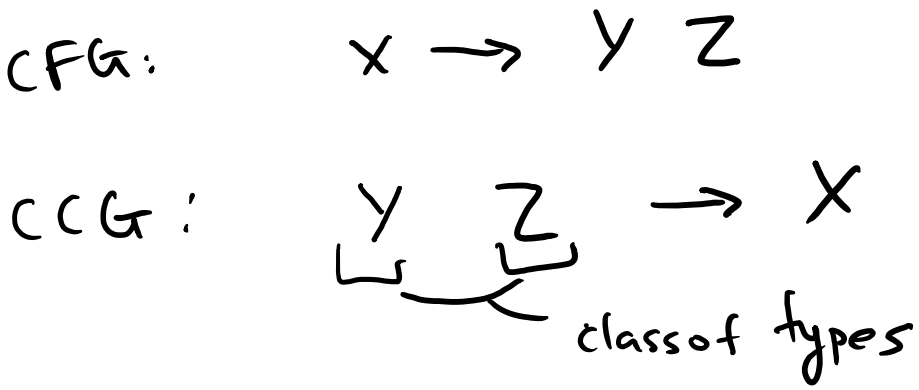
Primitive

Complex



CCG Combinators

Instead of rules, we have a small set of **generic combinators**.



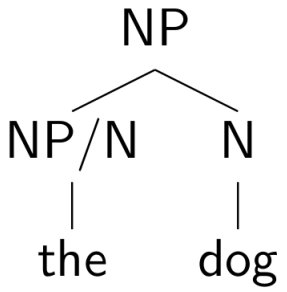
Application Combinator

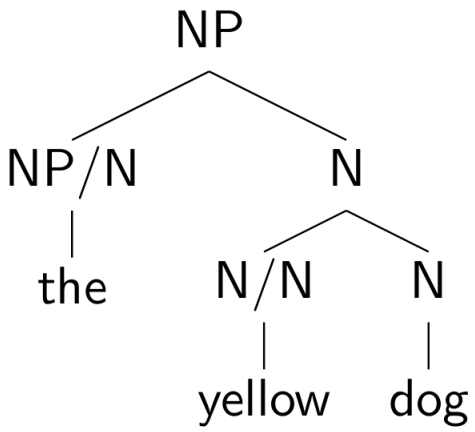
Forward

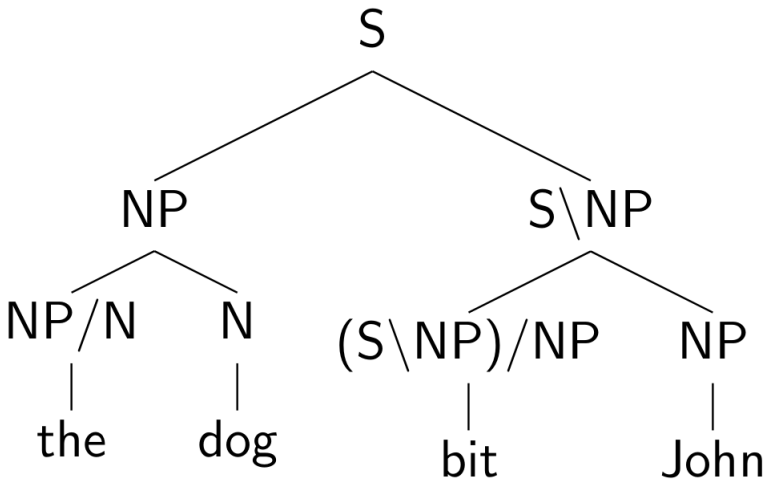
$X/Y \quad Y \Rightarrow X$

Backward

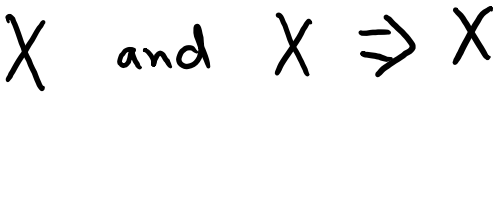
$Y \quad X\backslash Y \Rightarrow X$

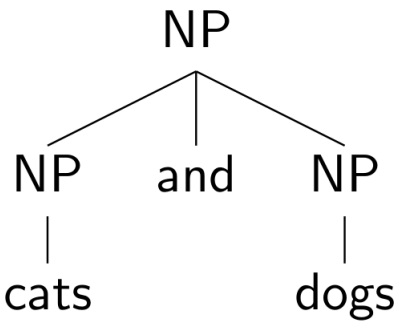


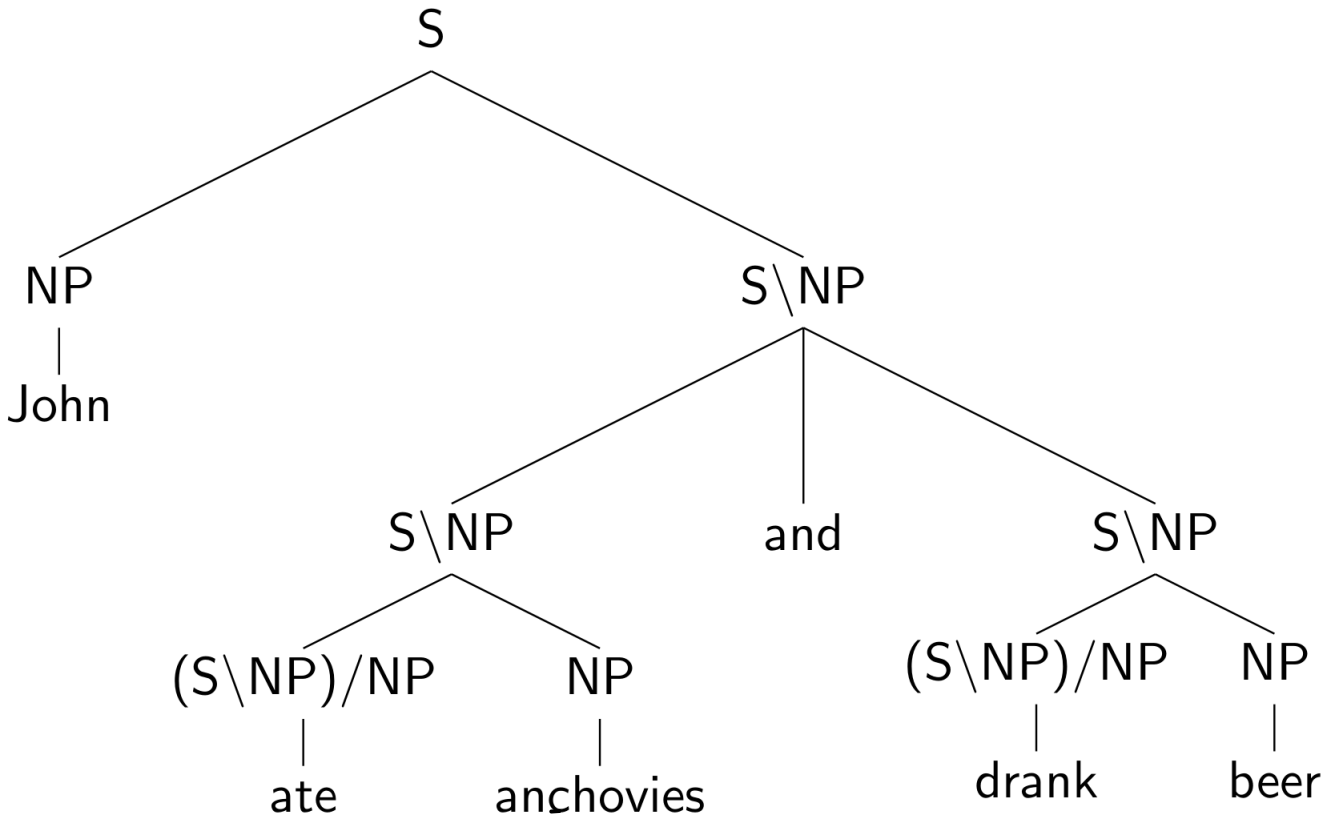




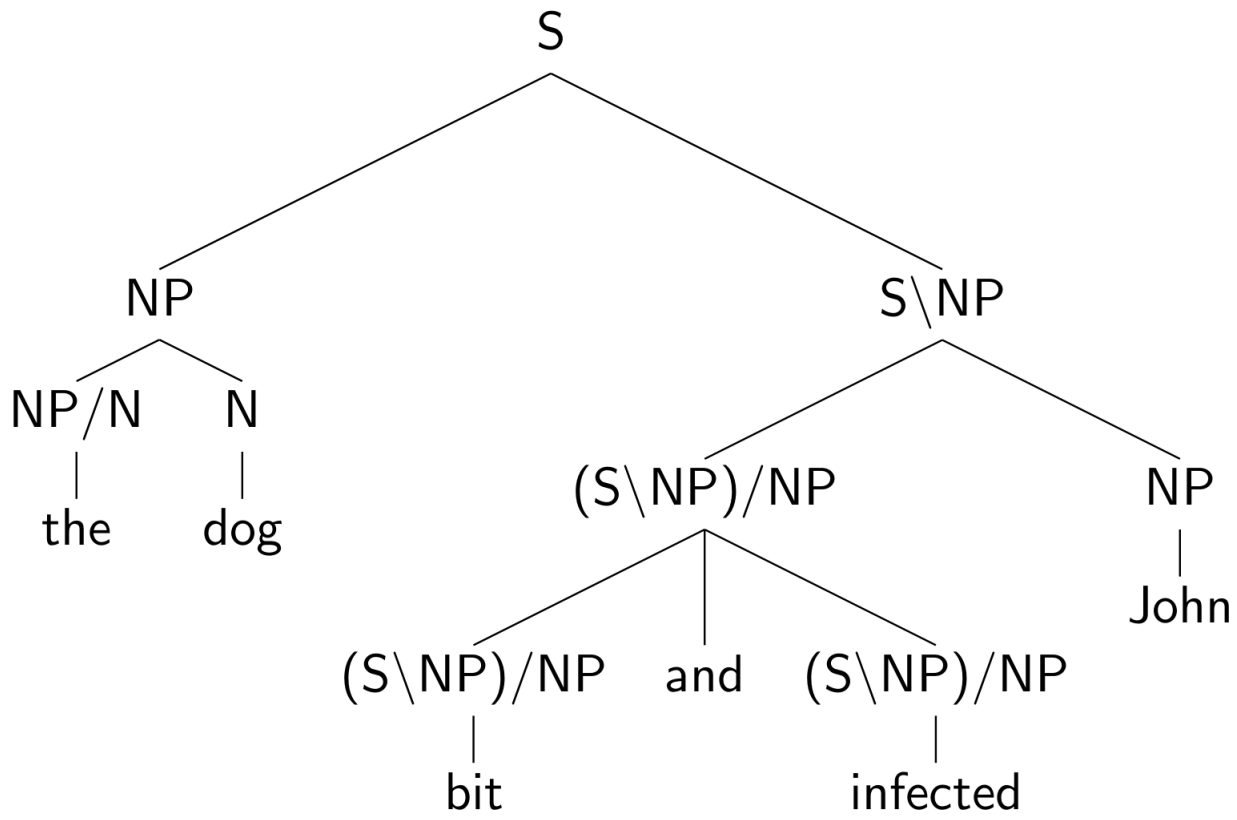
Conjunction Combinator







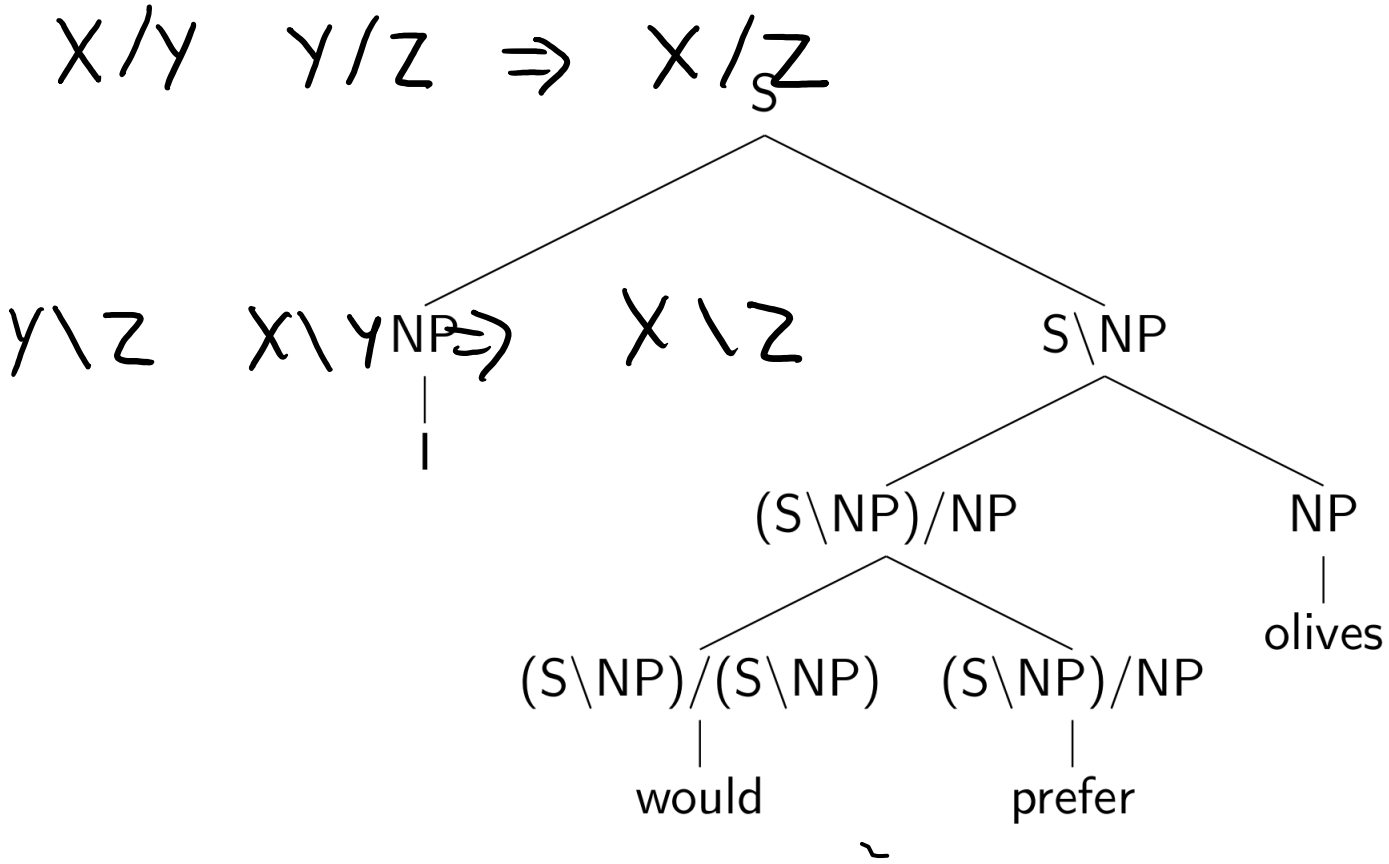
Conjunction Combinator



Composition Combinator

Forward

Backward



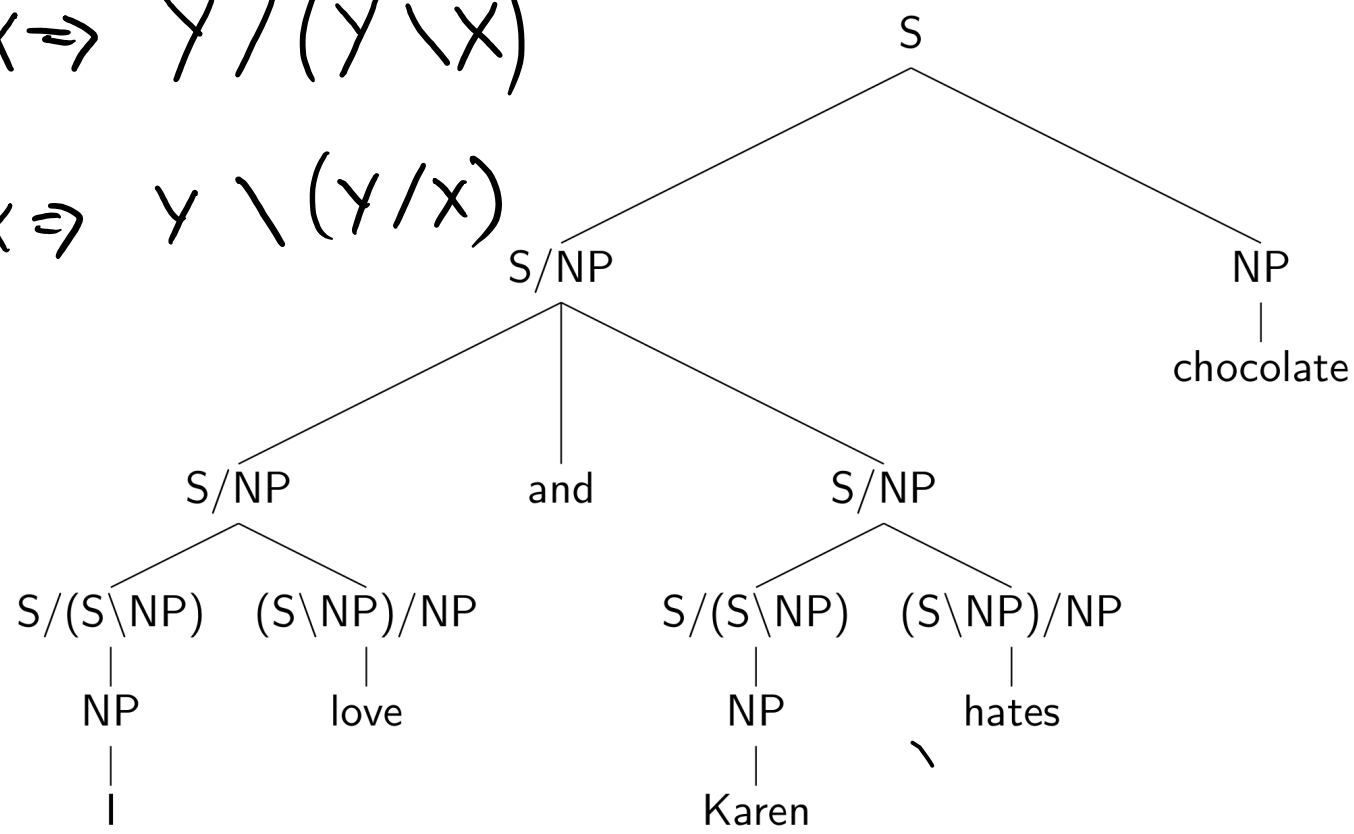
Type-Raising Combinator

Forward

$X \Rightarrow Y/(Y\backslash X)$

Backward

$X \Rightarrow Y\backslash(Y/X)$



Upcoming...

Homework

- Homework 3 is due on February 27
- Write-up and data has been released.

Project

- Status report due in 2 weeks: March 2, 2017
- Instructions coming soon
- Only 5 pages

Summaries

- Paper summaries: **February 17**, February 28, March 14
- Only 1 page each