# Semantic Parsing and First-Order Predicate Calculus

11-711 Advanced NLP

October 2021

(With thanks to Noah Smith)

# Key Challenge of Meaning

- We actually ***say*** very little - much more is left unsaid, because it's assumed to be widely known.

- Examples:
  - Reading newspaper stories
  - Using restaurant menus
  - Learning to use a new piece of software

# Meaning Representation Languages

- Symbolic representation that does two jobs:
  - Conveys the meaning of a **sentence**
  - Represents (some part of) the **world**
- We're assuming a very literal, context-independent, inference-free version of meaning!
  - Semantics vs. linguists' "pragmatics"
  - "Meaning representation" vs some philosophers' use of the term "semantics".
- For now we'll use **first-order logic**. Also called First-Order Predicate Calculus. Logical form.

# Representing NL meaning

- Fortunately, there has been a lot of work on this (since Aristotle, at least)
  - Panini in India too
- Especially, ***formal mathematical logic*** since 1850s (!), starting with George Boole etc.
  - Wanted to replace NL proofs with something more formal
- Deep connections to set theory

# Model-Theoretic Semantics

- Model:  a simplified representation of (some part of) the world:  **sets** of objects, properties, relations (**domain**).
- Non-logical vocabulary: like variable and function names
  - Each element **denotes** (maps to) a well-defined part of the model.  *(“Grounding”.)*
  - Such a mapping is called an **interpretation**
- Logical vocabulary: used to *compose* larger meanings
  - like reserved words in programming languages
  - or function words in grammar

# A Model

- **Domain**: Noah, Karen, Rebecca, Frederick, Green Mango, Casbah, Udipi, Thai, Mediterranean, Indian
- **Properties**: Green Mango and Udipi are crowded; Casbah is expensive
- **Relations**: Karen likes Green Mango, Frederick likes Casbah, everyone likes Udipi, Green Mango serves Thai, Casbah serves Mediterranean, and Udipi serves Indian

- n, k, r, f, g, c, u, t, m, i
- Crowded = {g, u}
- Expensive = {c}
- Likes = {(k, g), (f, c), (n, u), (k, u), (r, u), (f, u)}
- Serves = {(g, t), (c, m), (u, i)}

# Some English

- *Karen likes Green Mango and Frederick likes Casbah.*
- *Noah and Rebecca like the same restaurants.*
- *Noah likes expensive restaurants.*
- *Not everybody likes Green Mango.*

- What we want is to be able to represent these statements in a way that lets us compare them to our model.
- **Truth-conditional semantics**: need operators and their meanings, given a particular model.

# First-Order Logic

- **Terms** refer to elements of the domain: **constants**, **functions**, and **variables**
  - Noah, SpouseOf(Karen), X
- **Predicates** are used to refer to sets and relations; predicate applied to a term is a **Proposition**
  - Expensive(Casbah)
  - Serves(Casbah, Mediterranean)
- Logical connectives (**operators**):

  $\wedge$ (and), $\vee$ (or), $\neg$ (not), $\Rightarrow$ (implies), ...
- **Quantifiers** ...

# Logical operators: truth tables

| A | B | A ∧ B | A ∨ B | A ⇒ B |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

- Only really need ∧ and ¬

  “A ∨ B”  is “(¬A) ∧ (¬B)”

  “A ⇒ B” is “¬ (A ∧ ¬ B)” or “¬A ∨ B”

# Quantifiers in FOL

- Two ways to use variables:
  - refer to one anonymous object from the domain (**existential**; ∃; "there exists")
  - refer to all objects in the domain (**universal**; ∀; "for all")

- *A restaurant near CMU serves Indian food*
  $\exists x\ \text{Restaurant}(x) \wedge \text{Near}(x, \text{CMU}) \wedge \text{Serves}(x, \text{Indian})$
- *All expensive restaurants are far from campus*
  $\forall x\ \text{Restaurant}(x) \wedge \text{Expensive}(x) \Rightarrow \neg\text{Near}(x, \text{CMU})$

# Inference

- Big idea: extend the knowledge base, or check some proposition against the knowledge base.
- **Forward chaining** with modus ponens: given $\alpha$ and $\alpha \Rightarrow \beta$, we know $\beta$.
- **Backward chaining** takes a query $\beta$ and looks for propositions $\alpha$ and $\alpha \Rightarrow \beta$ that would prove $\beta$.
  - Not the same as backward reasoning (*abduction*).
  - Used by Prolog
- Both are sound, neither is complete by itself.

# Inference example

- Starting with these facts:

  Restaurant(Udipi)

  $\forall x$ Restaurant(x) $\Rightarrow$ Likes(Noah, x)

- We can "turn a crank" and get this *new* fact:

  Likes(Noah, Udipi)

# FOL: Meta-theory

- Well-defined set-theoretic semantics
- **Sound:** can't prove false things
- **Complete:** can prove everything that logically follows from a set of axioms (e.g., with "resolution theorem prover")

- Well-behaved, well-understood
- Mission accomplished?

# FOL: But there are also “Issues”

- “Meanings” of sentences are ***truth values***.
- *Extensional* semantics (vs. *Intensional*); Closed World issue
- Only ***first-order*** (no quantifying over *predicates* [which the book does without comment!]).
- Not very good for ***“fluents”*** (time-varying things, real-valued quantities, etc.). Heard of Zeno?
- Brittle: ***anything*** follows from *any* contradiction(!)
- ***Goedel incompleteness***: “This statement has no proof”!

# FOL: But there are also "Issues"

- "Meanings" of sentences are ***truth values***.
- *Extensional* semantics (vs. *Intensional*); Closed World issue
- Only ***first-order*** (no quantifying over *predicates* [which the book does without comment]).
- Not very good for ***"fluents"*** (time-varying things, real-valued quantities, etc.)
- Brittle: ***anything*** follows from *any* contradiction(!)
- ***Goedel incompleteness***: "This statement has no proof"!
  - (Finite axiom sets are incomplete w.r.t. the real world.)

- **So:** Most systems use the FOL **descriptive** apparatus (with extensions) but not its **inference** mechanisms.

# Lots More To Say About MRLs!

- See chapter 17 for more about:
  - Representing events and states in FOL
  - Dealing with optional arguments (e.g., “eat”)
  - Representing time
  - Non-FOL approaches to meaning
- Interest in this topic (in NLP) waned during the 1990s and early 2000s.
  - It has come back, with the rise of semi-structured databases like Wikipedia.

# Connecting Syntax and Semantics

# Semantic Parsing

- Goal:  transform a NL statement into MRL (for now, FOL).
- Sometimes called “semantic analysis.”
- As described earlier, this is the literal, context-independent, inference-free meaning of the statement

# “Literal, context-independent, inference-free” semantics

- Example: *The ball is red*
- Assigning a specific, grounded meaning involves deciding *which* ball is meant
- Would have to resolve *indexical terms* including pronouns, normal NPs, etc.
- Logical form allows compact representation of such indexical terms (vs. listing all members of the set)
- To retrieve a specific meaning, we combine LF with a particular context or situation (set of objects and relations)
- So LF is a function that maps an initial discourse situation into a new discourse situation (from *situation semantics*)

# Compositionality

- The meaning of an NL phrase is determined by combining the meaning of its sub-parts.
- There are obvious exceptions ("*hot dog*," "*straw man*," "*New York*," etc.).
- Note: J&M II book uses an event-based FOL representation, but I'm using a simpler one without events.

- **Big idea:** start with parse tree, build semantics on top using FOL with λ-expressions.

# Extension: Lambda Notation

- A way of making anonymous functions.
- λx. (*some expression mentioning x*)
  - Example: λx.Near(x, CMU)
  - Trickier example: λx.λy.Serves(y, x)
- Lambda reduction: substitute for the variable.
  - (λx.Near(x, CMU))(LulusNoodles)
    becomes
    Near(LulusNoodles, CMU)

# Lambda reduction: order matters!

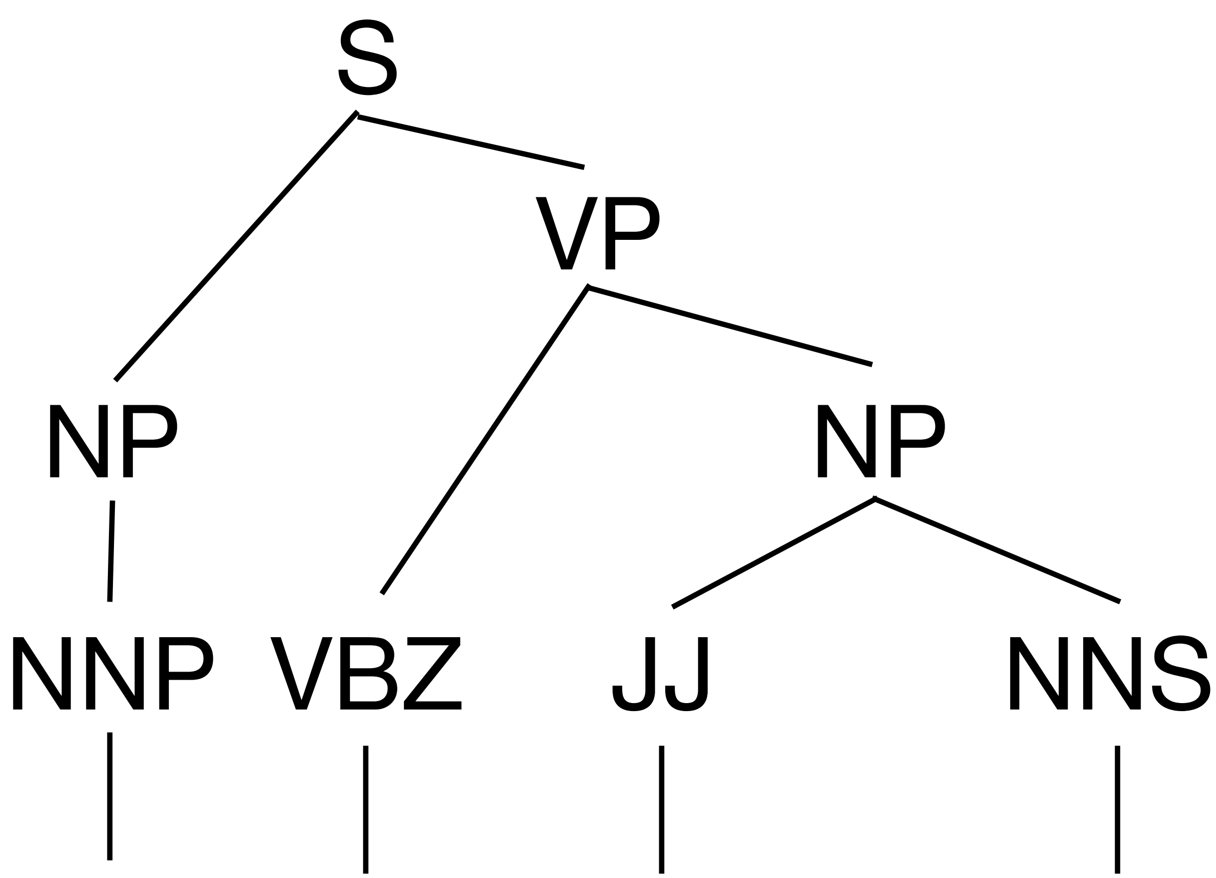
- **λx.λy.**Serves(y, x) (Bill)(Jane) becomes λy.Serves(y, Bill)(Jane)

  Then λy.Serves(y, Bill) (Jane) becomes Serves(Jane, Bill)

- **λy.λx.**Serves(y, x) (Bill)(Jane) becomes λx.Serves(Bill, x)(Jane)

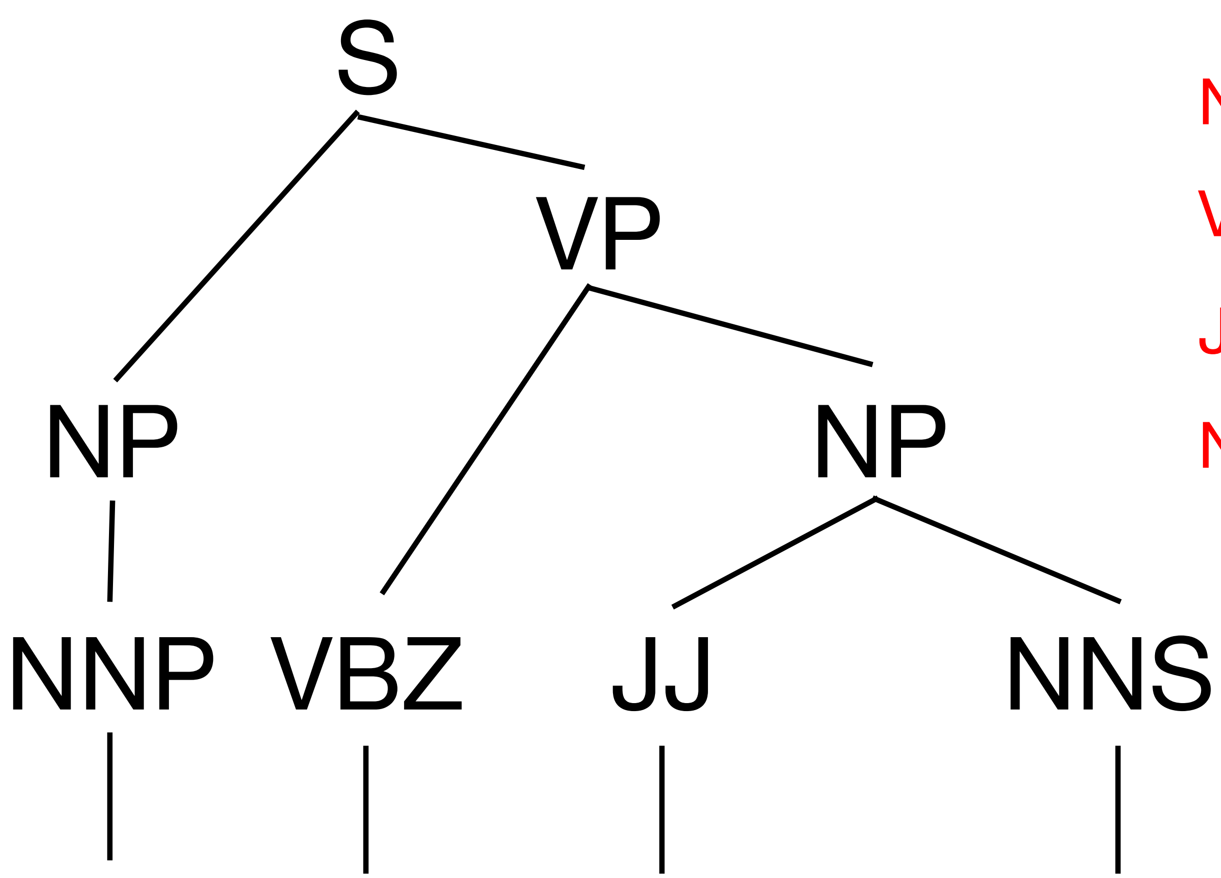
  Then λx.Serves(Bill, x) (Jane) becomes Serves(Bill, Jane)

# An Example




- *Noah likes expensive restaurants*.
- $\forall x$ Restaurant(x) $\wedge$ Expensive(x) $\Rightarrow$ Likes(Noah, x)

# An Example




NNP → Noah { Noah }

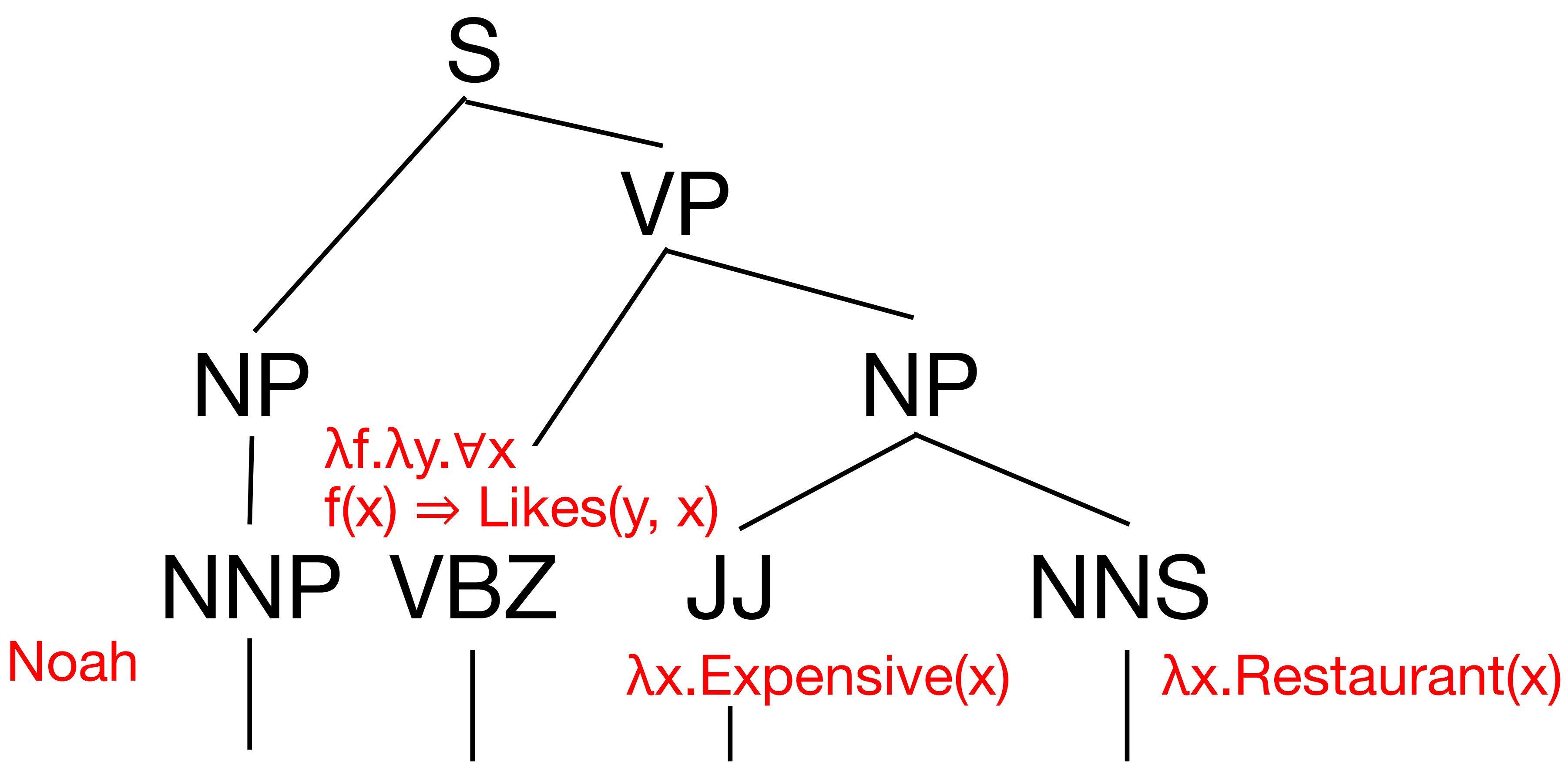
VBZ → likes { λf.λy.∀x f(x) ⇒ Likes(y, x) }

JJ → expensive { λx.Expensive(x) }

NNS → restaurants { λx.Restaurant(x) }

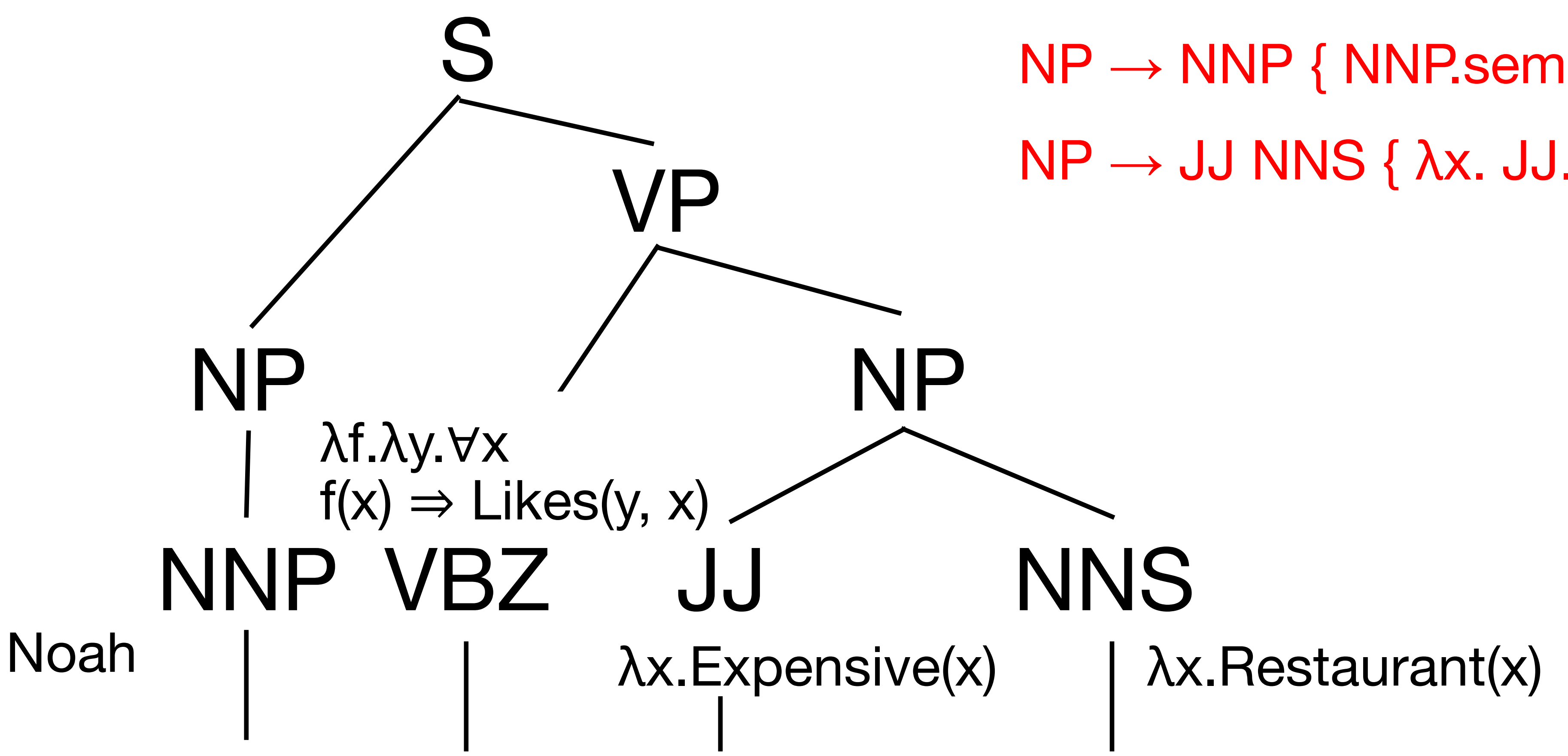
- *Noah likes expensive restaurants*.
- ∀x Restaurant(x) ∧ Expensive(x) ⇒ Likes(Noah, x)

# An Example




- *Noah likes expensive restaurants*.
- ∀x Restaurant(x) ∧ Expensive(x) ⇒ Likes(Noah, x)

# An Example

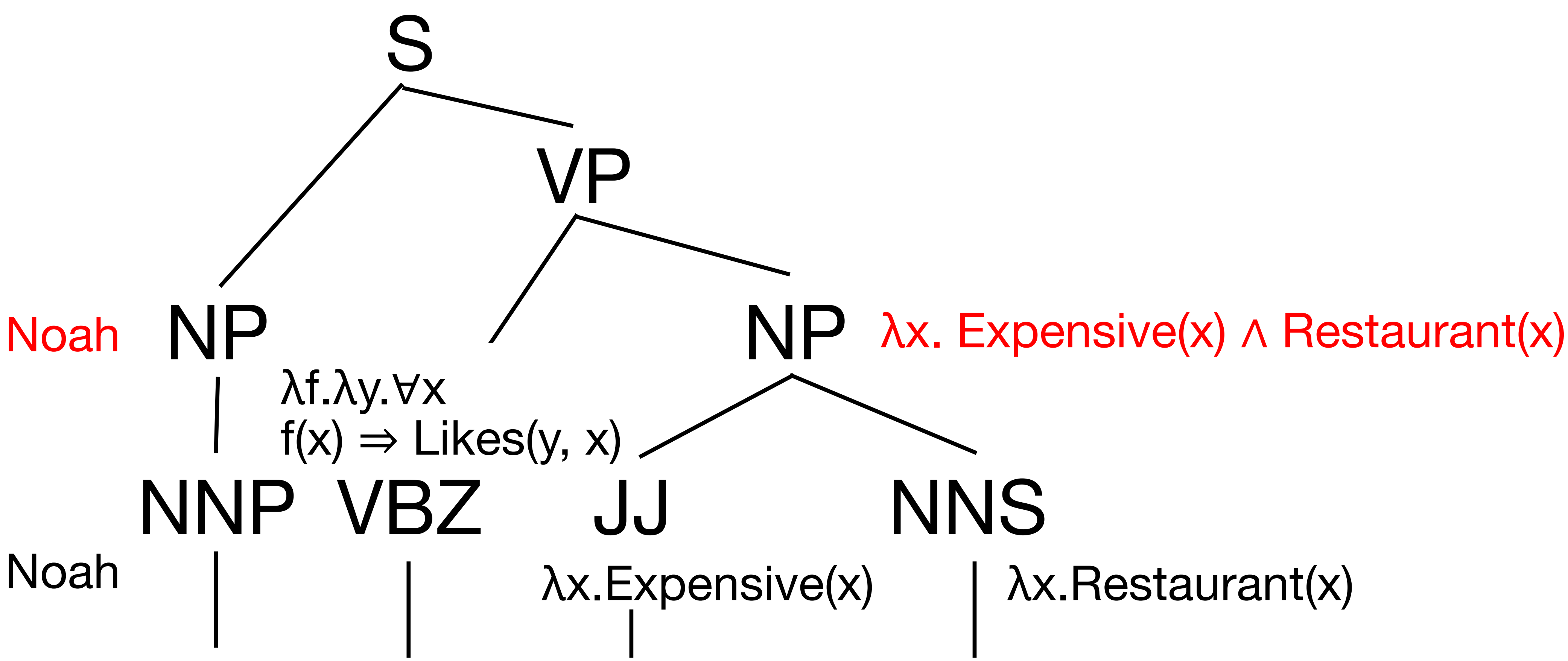



NP → NNP { NNP.sem }

NP → JJ NNS { λx. JJ.sem(x) ∧ NNS.sem(x) }

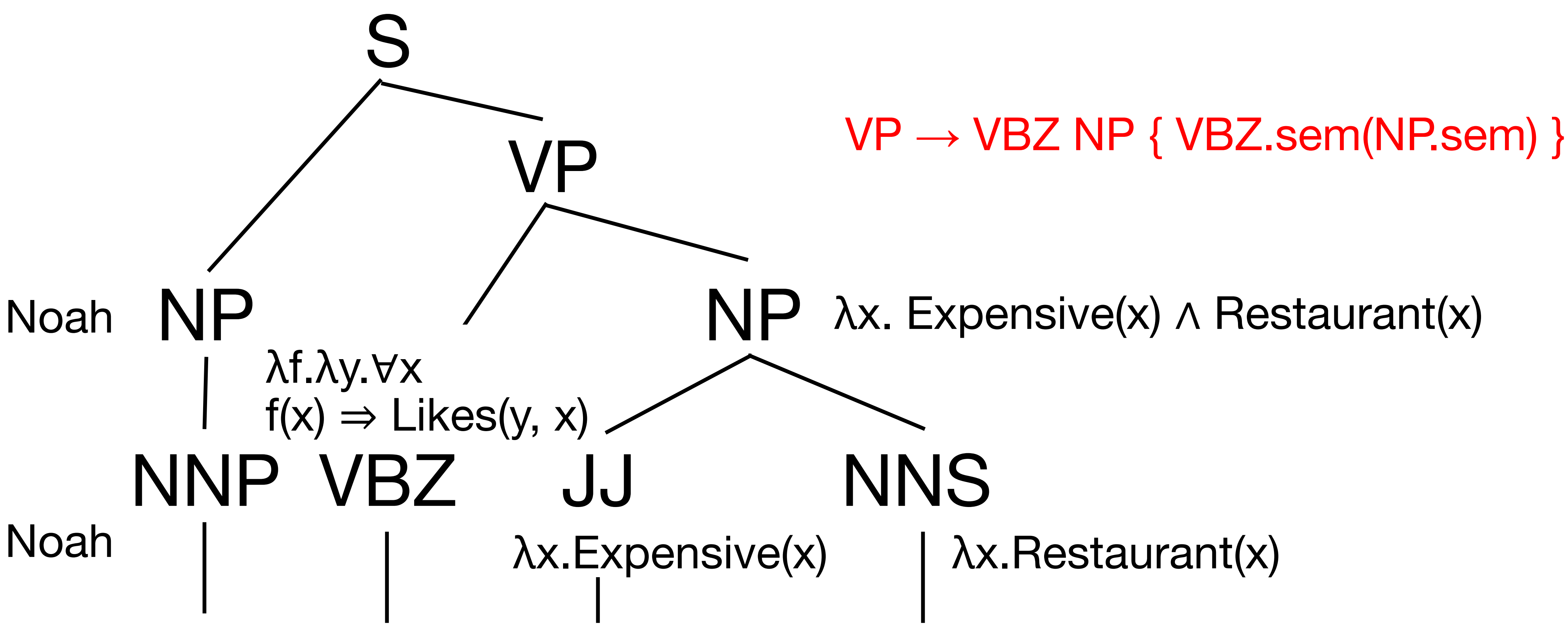
- *Noah likes expensive restaurants*.
- ∀x Restaurant(x) ∧ Expensive(x) ⇒ Likes(Noah, x)

# An Example



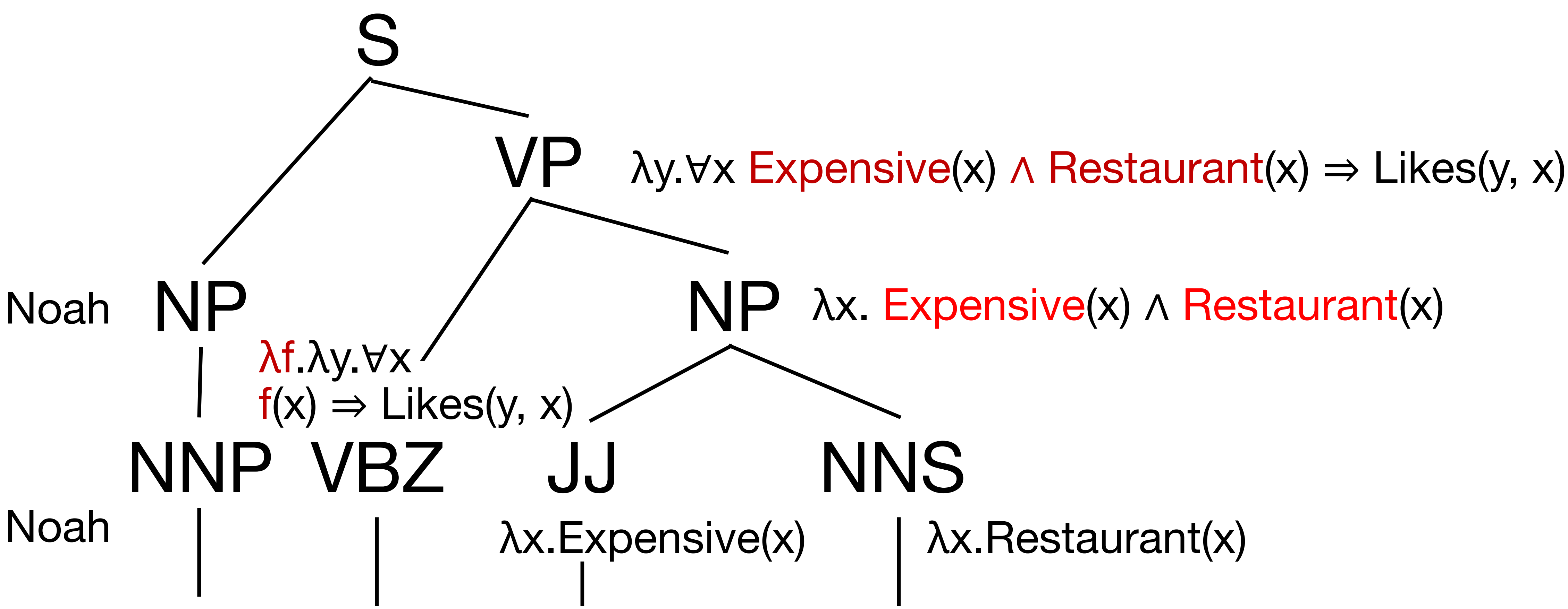

- *Noah likes expensive restaurants*.
- $\forall x$ Restaurant(x) ∧ Expensive(x) ⇒ Likes(Noah, x)

# An Example




- *Noah likes expensive restaurants*.
- ∀x Restaurant(x) ∧ Expensive(x) ⇒ Likes(Noah, x)

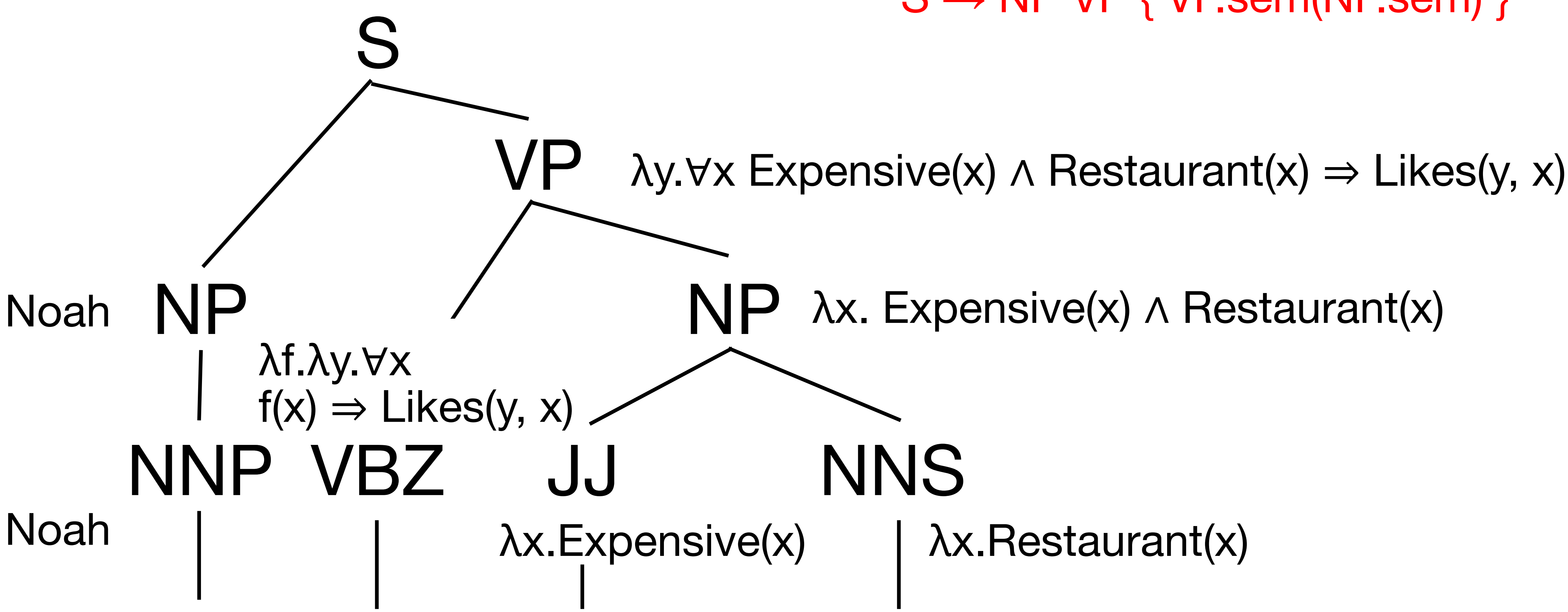
# An Example




- *Noah likes expensive restaurants*.
- $\forall x$ Restaurant(x) $\wedge$ Expensive(x) $\Rightarrow$ Likes(Noah, x)

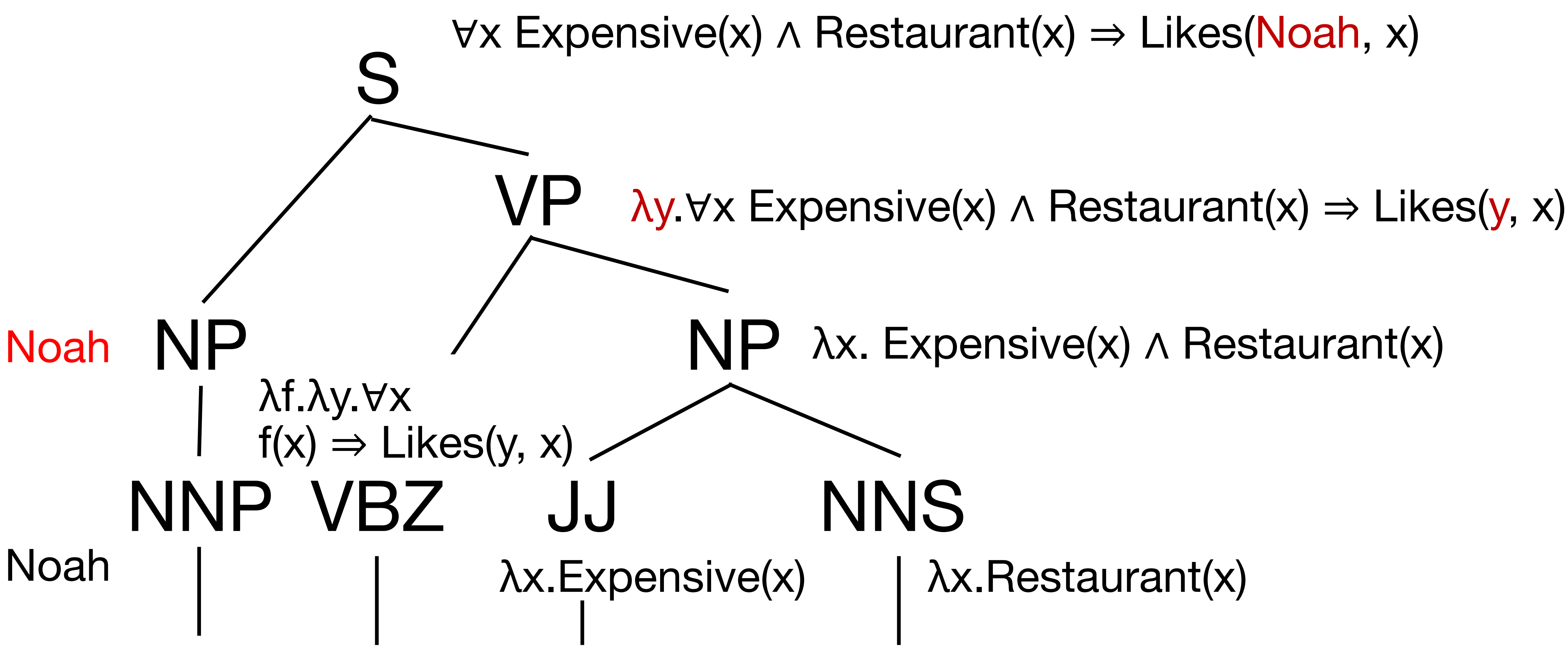
# An Example

S → NP VP { VP.sem(NP.sem) }



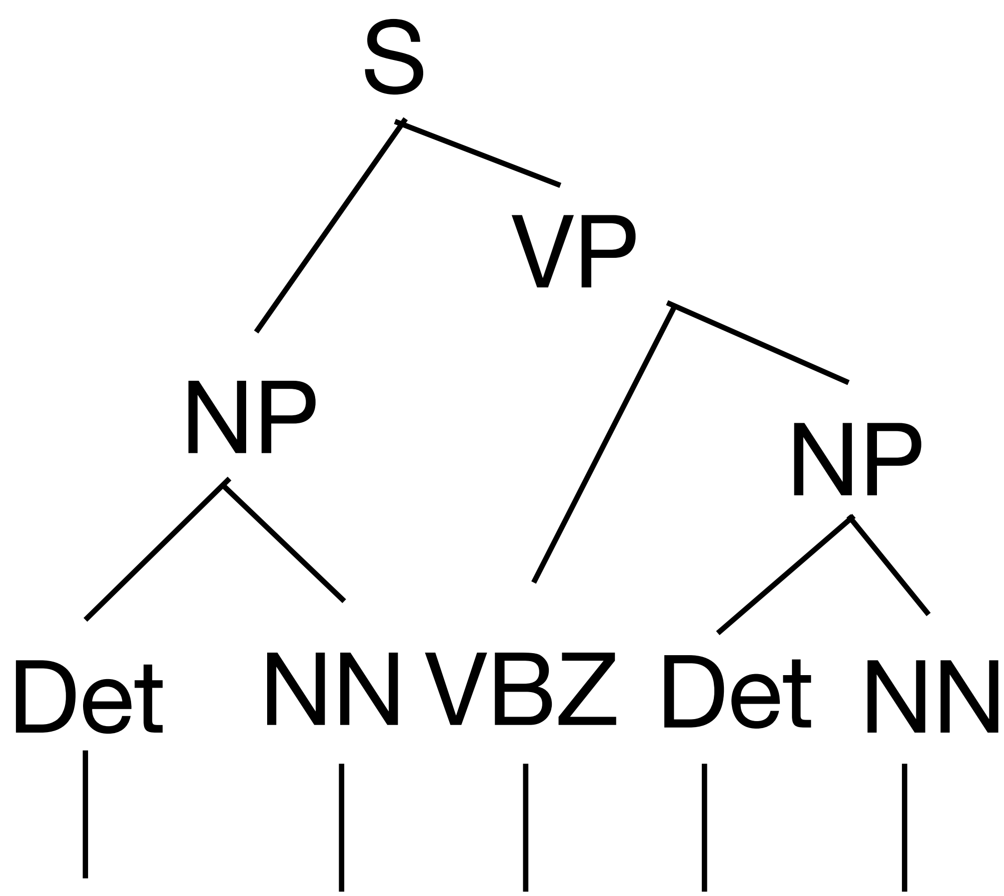

- *Noah likes expensive restaurants*.
- $\forall x$ Restaurant(x) $\wedge$ Expensive(x) $\Rightarrow$ Likes(Noah, x)

# An Example




- *Noah likes expensive restaurants*.
- ∀x Restaurant(x) ∧ Expensive(x) ⇒ Likes(Noah, x)

# Quantifier Scope Ambiguity




S → NP VP { NP.sem(VP.sem) }

NP → Det NN { Det.sem(NN.sem) }

VP → VBZ NP { VBZ.sem(NP.sem) }

Det → every { $\lambda f.\lambda g.\forall u\ f(u) \Rightarrow g(u)$ }

Det → a { $\lambda m.\lambda n.\exists x\ m(x) \wedge n(x)$ }

NN → man { $\lambda v.Man(v)$ }

NN → woman { $\lambda y.Woman(y)$ }

VBZ → loves { $\lambda h.\lambda k.h(\lambda w.\ Loves(k, w))$ }

- *Every man loves a woman*.

- $\forall u\ Man(u) \Rightarrow \exists x\ Woman(x) \wedge Loves(u, x)$

# This Isn't Quite Right!

- "*Every man loves a woman*" really is ambiguous.
  - $\forall u\ Man(u) \Rightarrow \exists x\ Woman(x) \wedge Loves(u, x)$
  - $\exists x\ Woman(x) \wedge \forall u\ Man(u) \Rightarrow Loves(u, x)$

- This gives only one of the two meanings.
  - Extra ambiguity on top of syntactic ambiguity
- One approach is to delay the quantifier processing until the end, then permit any ordering.

# Quantifier Scope

- A seat was available for every customer.
- A toll-free number was available for every customer.

- A secretary called each director.
- A letter was sent to each customer.

- Every man loves a woman who works at the candy store.
- Every 5 minutes a man gets knocked down

  and he’s not too happy about it.

# What Else?

- Chapter 18 discusses how you can get this to work for other parts of English (e.g., prepositional phrases).
- Remember attribute-value structures for parsing with more complex terminals than simple symbols?
  - You can extend those with semantics as well.
- No time for ...
  - Statistical models for semantics
  - Parsing algorithms augmented with semantics
  - Handling idioms

# Extending FOL

- To handle sentences in non-mathematical texts, you need to cope with additional NL phenomena:
- Generalized quantifiers:
  - *Most dogs bark* Most x | Dog(x) . Barks(x)
  - *The happy dog barks* The x | (Happy(x) ∧ Dog(x)) . Barks(x)
- Speech Acts: ASSERT, YN-QUERY, COMMAND
  - WH-QUERY: *What did the man eat?*

    WH-QUERY(The x | Man(x) . (WH y | Thing(y) . Eat(x,y)))

# More extensions!

- Relative clauses are propositions embedded in an NP
    - Restrictive versus non-restrictive: *the dog that barked all night* vs. *the dog, which barked all night*
- Modal verbs: non-transparency for truth of subordinate clause: *Sue thinks that John loves Sandy*
- Tense/Aspect
- Plurality
- **Discourse!! vs. *Dialog***

**One of the most successful of these institutions is BancoSol in Bolivia.**

```
(*A-BE
   (FORM FINITE)
   (TENSE PRESENT)
   (MOOD DECLARATIVE)
   (PUNCTUATION PERIOD)
   (IMPERSONAL -)
   (THEME
      (*G-PARTITIVE
         (SUBSTANCE
            (*G-PARTITIVE
                (SUBSTANCE
                   (*O-INSTITUTION
                       (UNIT -)
                       (NUMBER PLURAL)
                       (REFERENCE DEFINITE)
                       (DISTANCE NEAR)
                       (PERSON THIRD)))
                (ADJECTIVE
                   (*P-SUCCESSFUL
                       (DEGREE SUPERLATIVE)))))
         (QUANTIFIER (*QUANT-ONE))))
   (PREDICATE
      (*PROP-BANCOSOL
         (NUMBER SINGULAR)
         (IMPLIED-REFERENCE +)
         (PERSON THIRD)
         (UNIT -)
         (Q-MODIFIER
            (*K-IN
               (OBJECT
                  (*PROP-BOLIVIA
                     (UNIT -)
                     (NUMBER SINGULAR)
                     (IMPLIED-REFERENCE +)
                     (PERSON THIRD))))))))
```