

# **Semantics and First-Order Predicate Calculus**

---

11-711 Algorithms for NLP

17 October 2019

(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".
- Today we'll use **first-order logic**. Also called First-Order Predicate Calculus. Logical form.

# A MRL Should Be Able To ...

---

- Verify a query against a knowledge base: *Do CMU students follow politics?*
- Eliminate ambiguity: *CMU students enjoy visiting Senators.*
- Cope with vagueness: *Sally heard the news.*
- Cope with many ways of expressing the same meaning (canonical forms): *The candidate evaded the question* vs. *The question was evaded by the candidate.*
- Draw conclusions based on the knowledge base: *Who could become the 46th president?*
- Represent all of the meanings we care about

# 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**).
- Logical vocabulary: like reserved words in PL
- Non-logical vocabulary
  - Each element **denotes** (maps to) a well-defined part of the model
  - Such a mapping is called an **interpretation**

# 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	$A \wedge B$	$A \vee B$	$A \Rightarrow B$
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	1

- Only really need  $\wedge$  and  $\neg$

“ $A \vee B$ ” is “ $(\neg A) \wedge (\neg B)$ ”

“ $A \Rightarrow B$ ” is “ $\neg (A \wedge \neg B)$ ” or “ $\neg A \vee B$ ”

# Quantifiers in FOL

---

- Two ways to use variables:
  - refer to one anonymous object from the domain (**existential**;  $\exists$ ; “there exists”)
  - refer to all objects in the domain (**universal**;  $\forall$ ; “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)

$\forall x \text{ Restaurant}(x) \Rightarrow \text{Likes}(\text{Noah}, x)$

- 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?

# Peano Arithmetic

---

- P1:  $\forall n (\neg(0=n+1))$
- P2:  $\forall n \forall m (n+1=m+1 \rightarrow n=m)$
- P3: For any formula  $\varphi$  with one free variable  $n$ ,  
 $(\varphi[0/n] \wedge (\forall n (\varphi \rightarrow \varphi[n+1/n]))) \rightarrow \forall n \varphi$
- P4:  $\forall n (n+0=n)$
- P5:  $\forall n \forall m (n+(m+1) = (n+m) + 1)$
- P6:  $\forall n (n*1=n)$
- P7:  $\forall n \forall m (n*(m+1) = (n*m) + n)$

# 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”!



# Assigning a correspondence to a model: natural language example

---

- What is the meaning of “*Gift*”?

# Assigning a correspondence to a model: natural language example

---

- What is the meaning of “*Gift*”?
  - English: a present

# Assigning a correspondence to a model: natural language example

---

- What is the meaning of “*Gift*”?
  - English: a present
  - German: a poison

# Assigning a correspondence to a model: natural language example

---

- What is the meaning of “*Gift*”?
  - English: a present
  - German: a poison
    - (Both come from the word “*give/geben*”!)
- Logic is **complete** for proving statements that are true in *every* interpretation
  - but **incomplete** for proving all the truths of arithmetic

# 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 its **descriptive** apparatus (with extensions) but not its **inference** mechanisms.

# First-Order Worlds, Then and Now

---

- 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.
  - Lay contributors to these databases may be helping us to solve the knowledge acquisition problem.
- Also, lots of research on using NLP, information extraction, and machine learning to grow and improve knowledge bases from free text data.
  - “Read the Web” project here at CMU.
- And: Semantic embedding/NN/vector approaches.

# 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

# **Connecting Syntax and Semantics**



# Semantic Analysis

---

- Goal: transform a NL statement into MRL (today, FOL).
- Sometimes called “semantic parsing.”
- 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: your 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  $\lambda$ -expressions.

# Extension: Lambda Notation

---

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

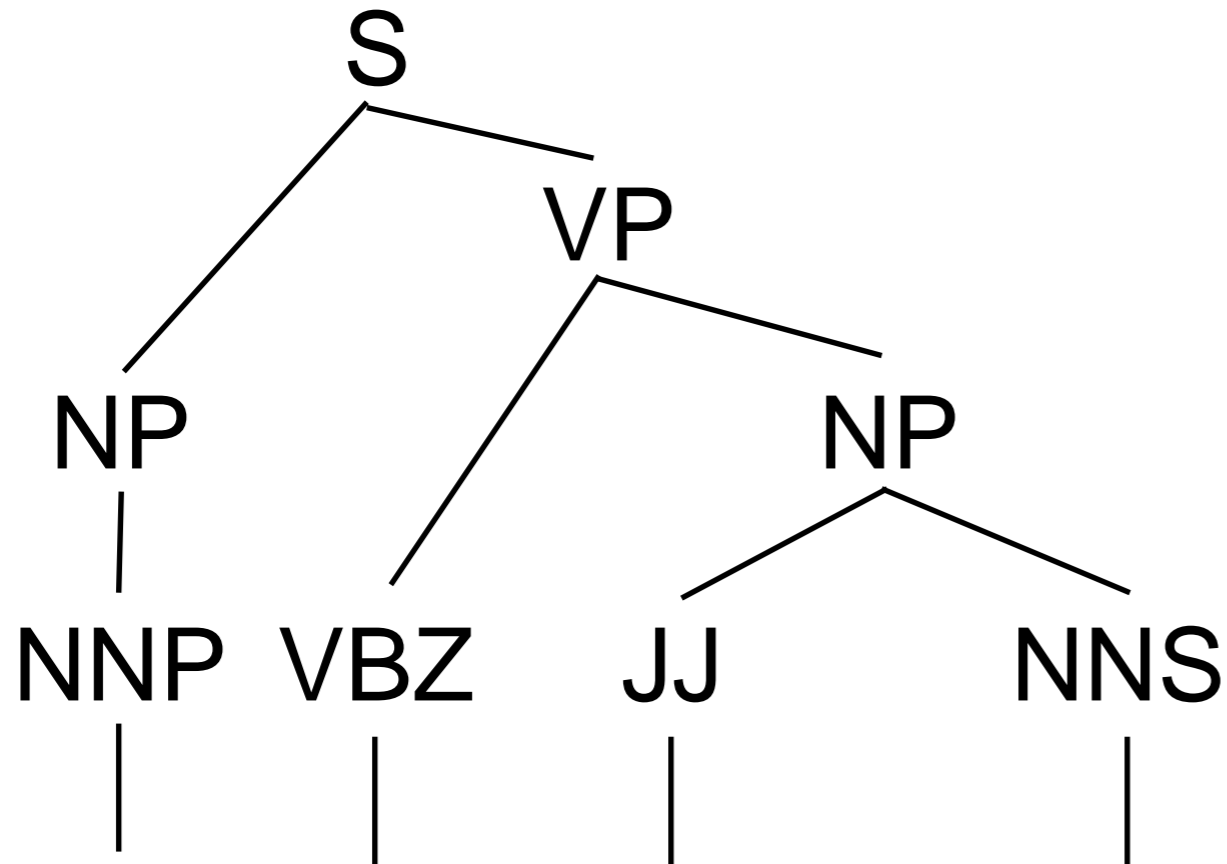
# Lambda reduction: order matters!

---

- $\lambda x.\lambda y.$ Serves( $y, x$ ) (Bill)(Jane) becomes  $\lambda y.$ Serves( $y, \text{Bill}$ )(Jane)  
Then  $\lambda y.$ Serves( $y, \text{Bill}$ ) (Jane) becomes Serves(Jane, Bill)
- $\lambda y.\lambda x.$ Serves( $y, x$ ) (Bill)(Jane) becomes  $\lambda x.$ Serves(Bill,  $x$ )(Jane)  
Then  $\lambda x.$ Serves(Bill,  $x$ ) (Jane) becomes Serves(Bill, Jane)

# An Example

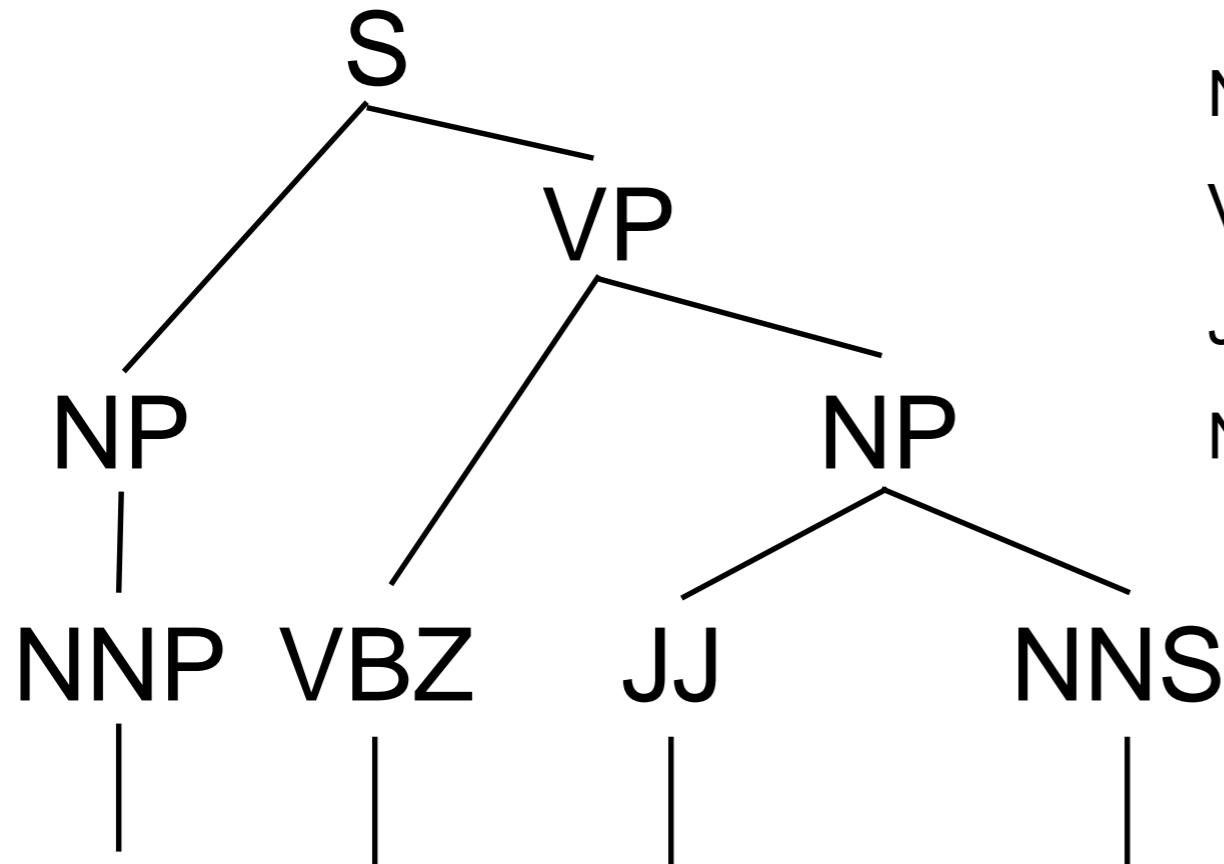
---



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

# An Example

---



NNP  $\rightarrow$  Noah { Noah }

VBZ  $\rightarrow$  likes {  $\lambda f. \lambda y. \forall x f(x) \Rightarrow \text{Likes}(y, x)$  }

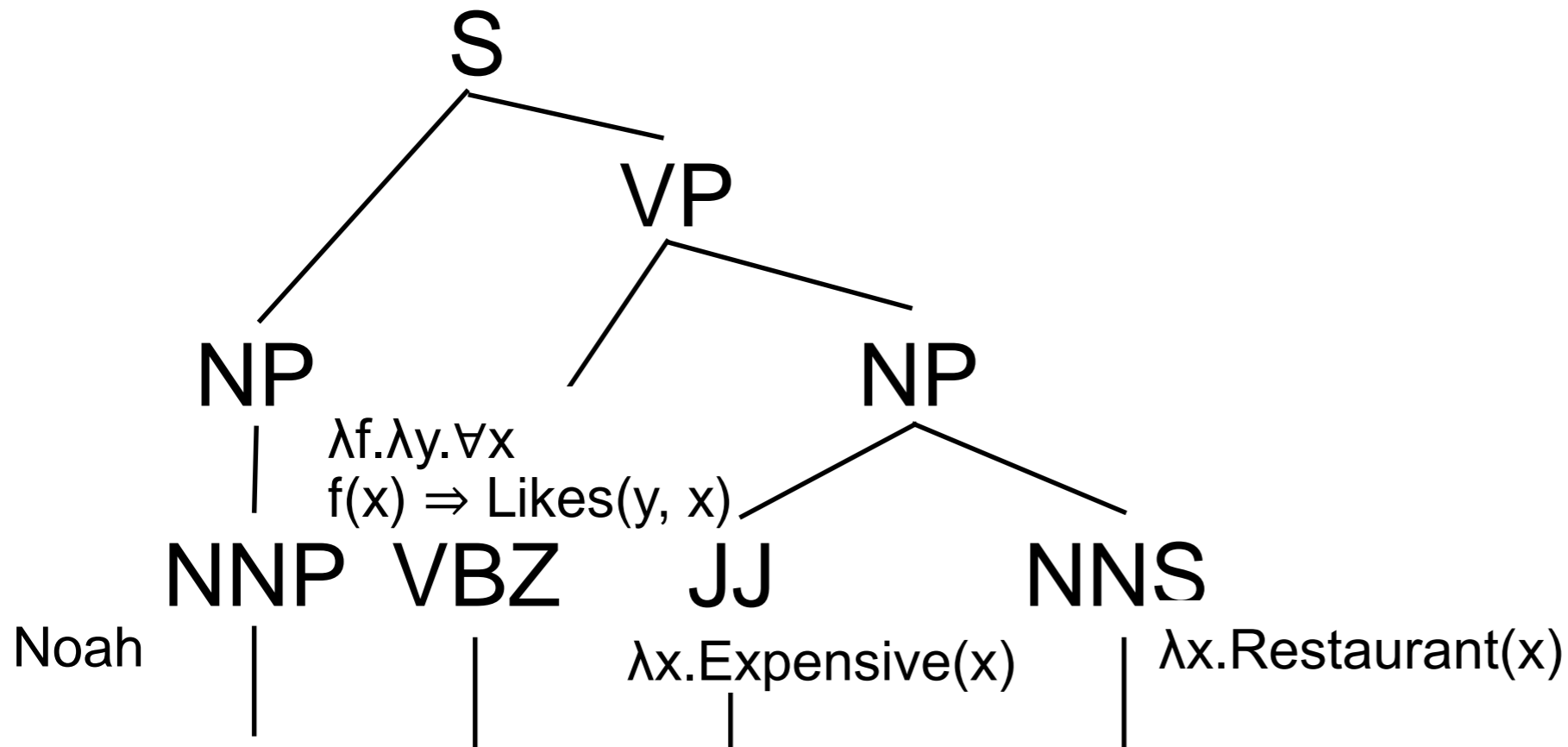
JJ  $\rightarrow$  expensive {  $\lambda x. \text{Expensive}(x)$  }

NNS  $\rightarrow$  restaurants {  $\lambda x. \text{Restaurant}(x)$  }

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

# An Example

---

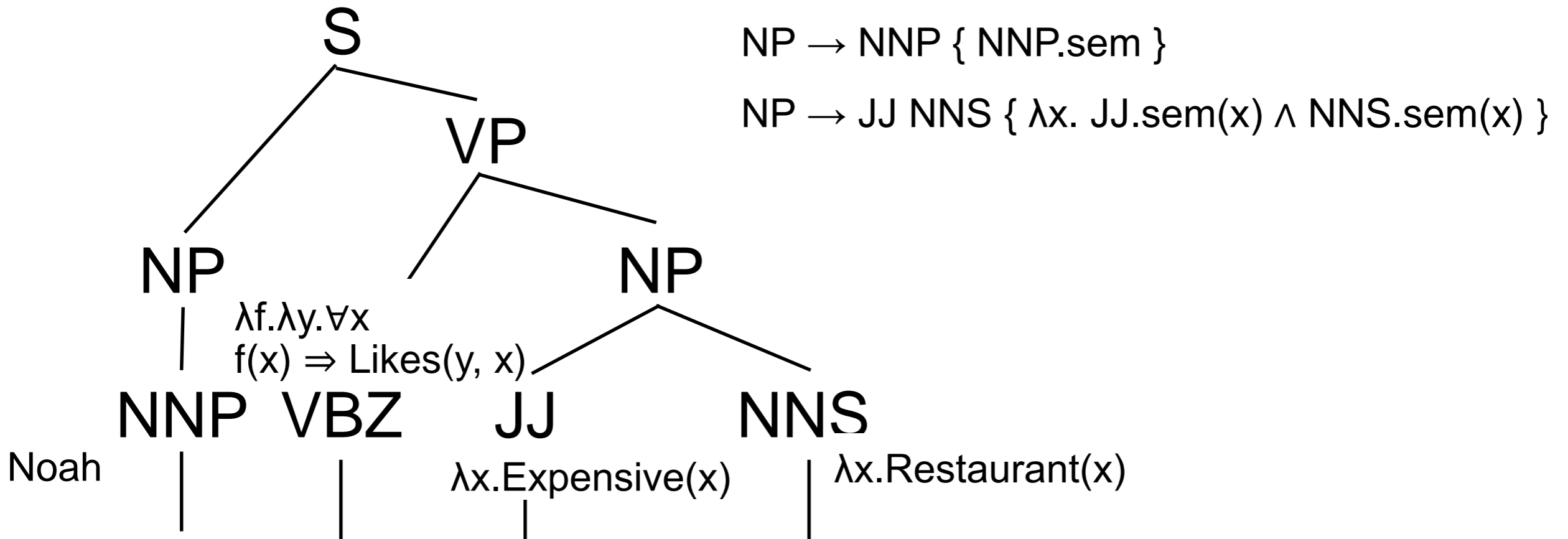


- *Noah likes expensive restaurants.*
- $\forall x \text{ Restaurant}(x) \wedge \text{Expensive}(x) \Rightarrow \text{Likes}(\text{Noah}, x)$



# An Example

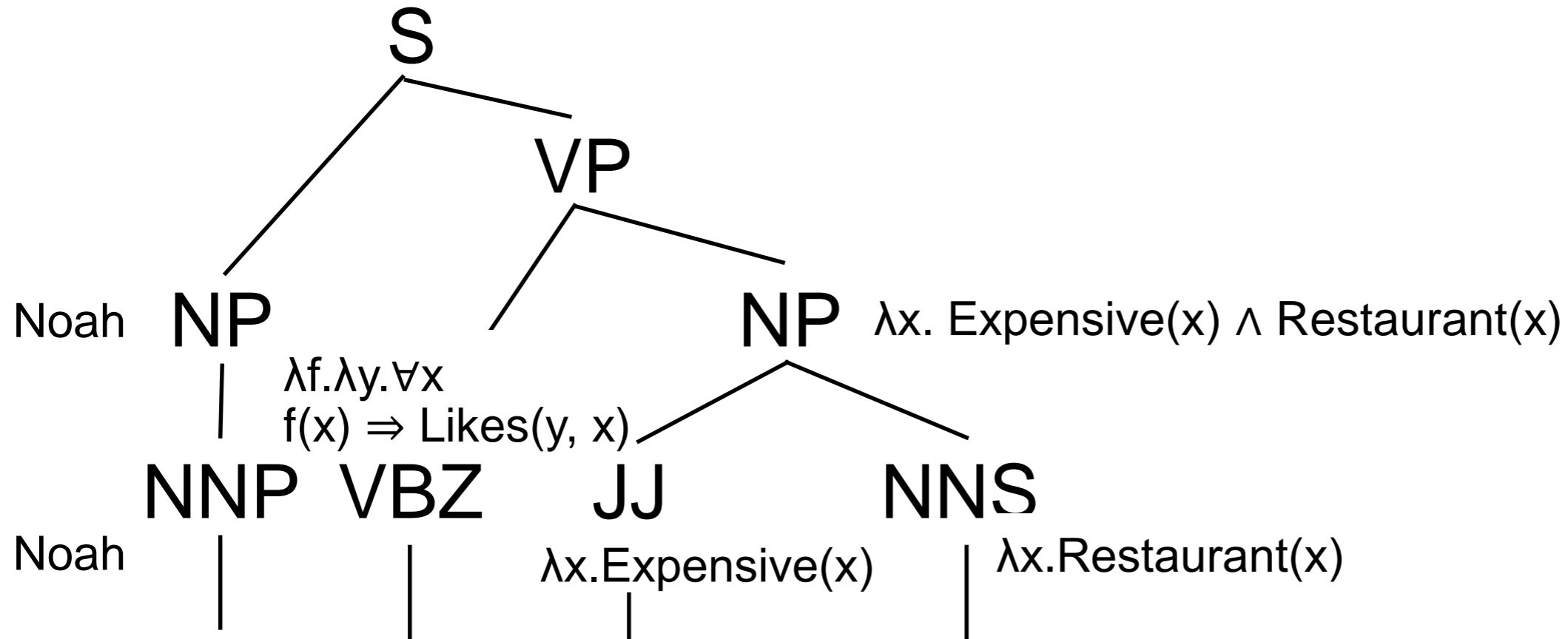
---



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

# An Example

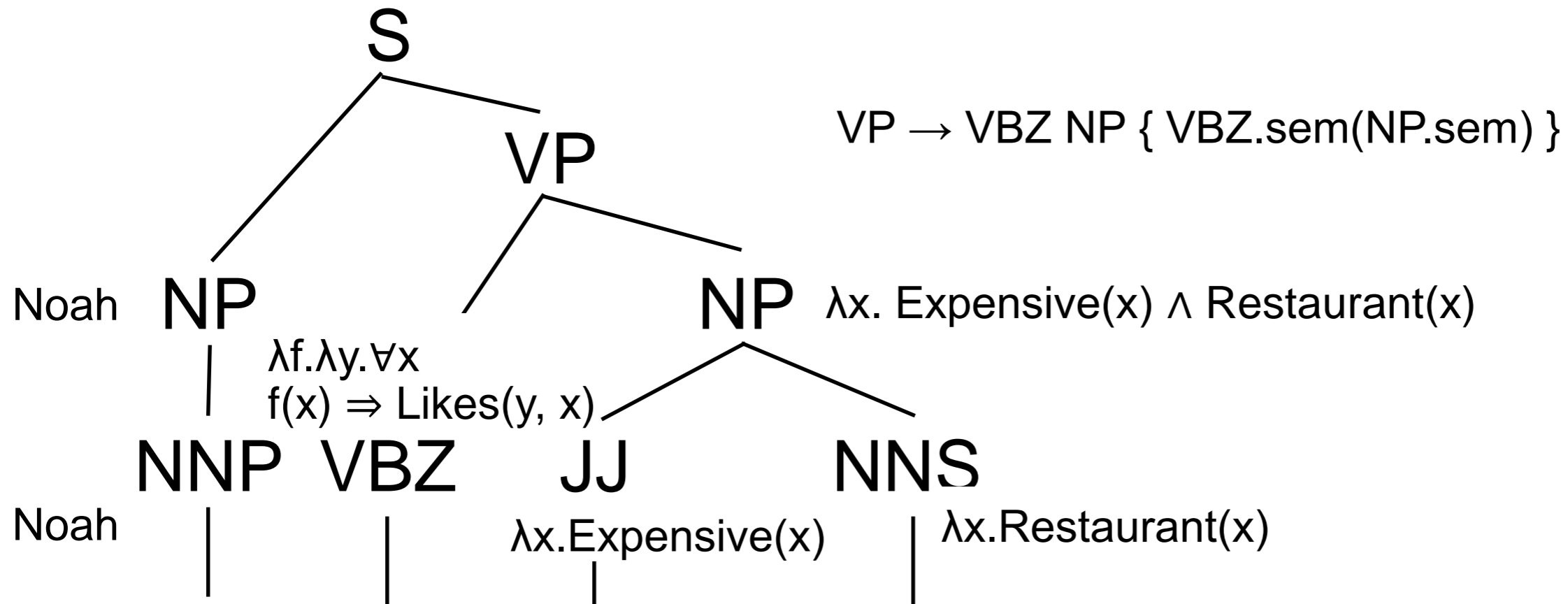
---



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

# An Example

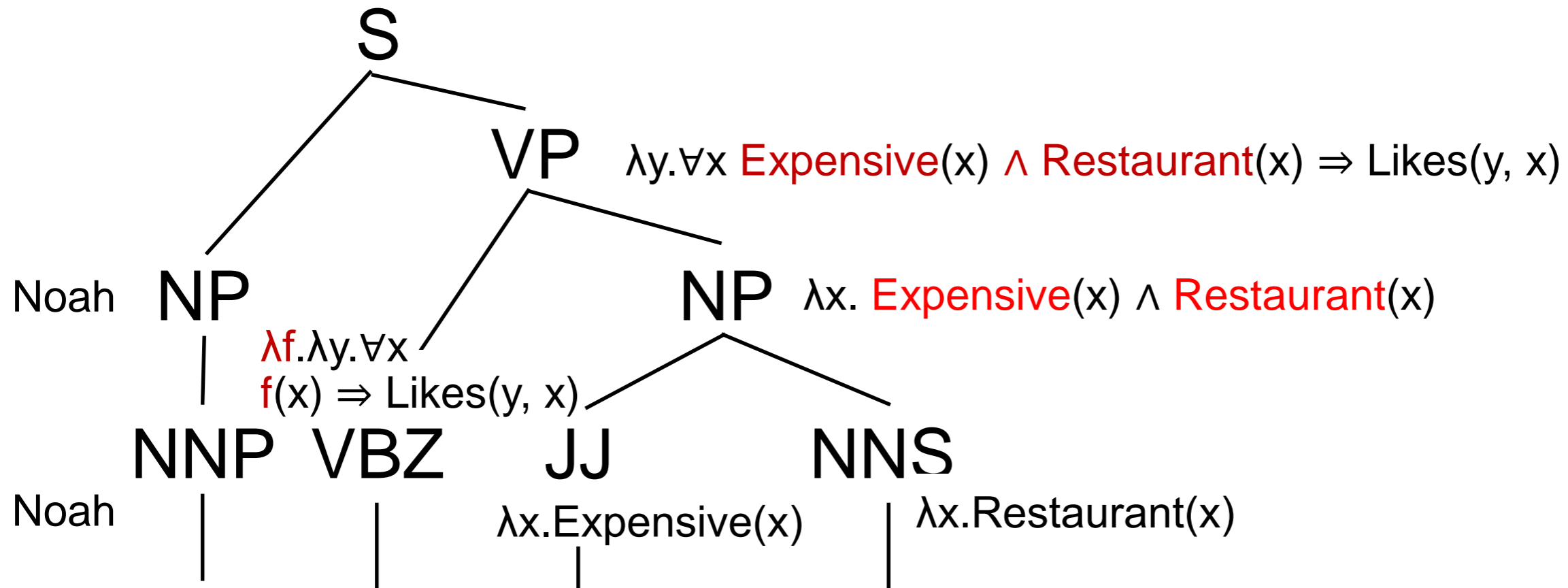
---



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

# An Example

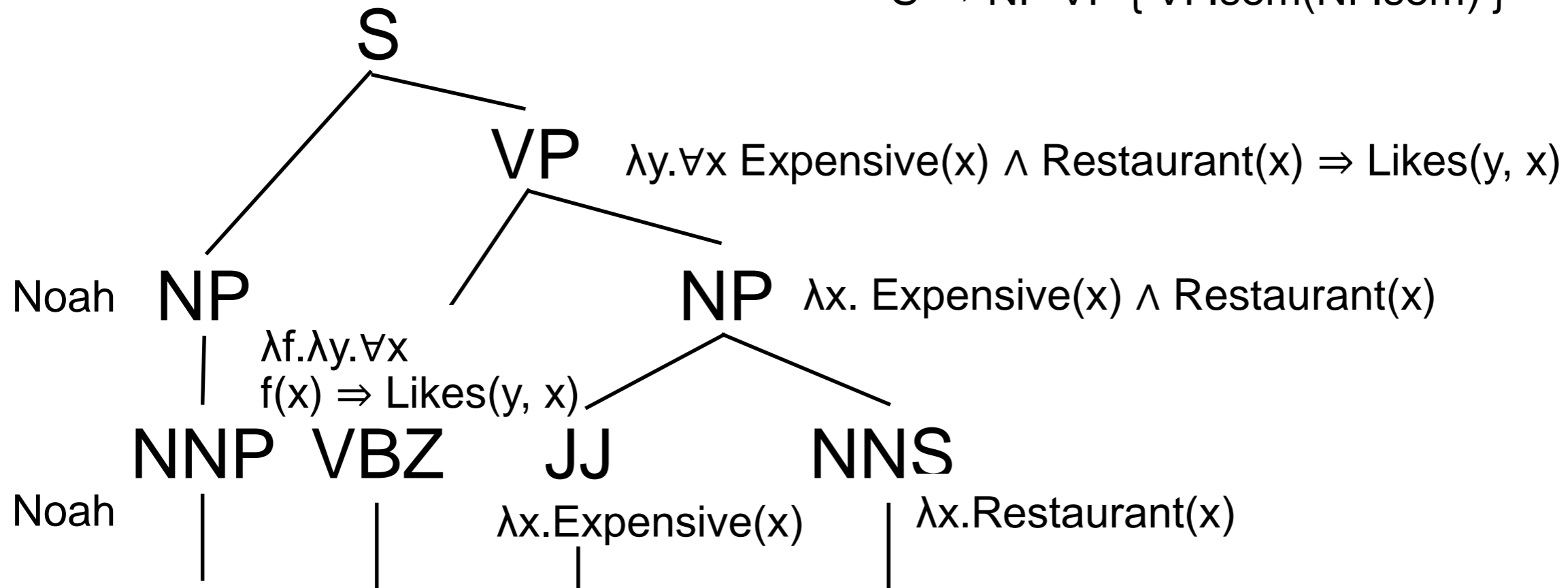
---



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

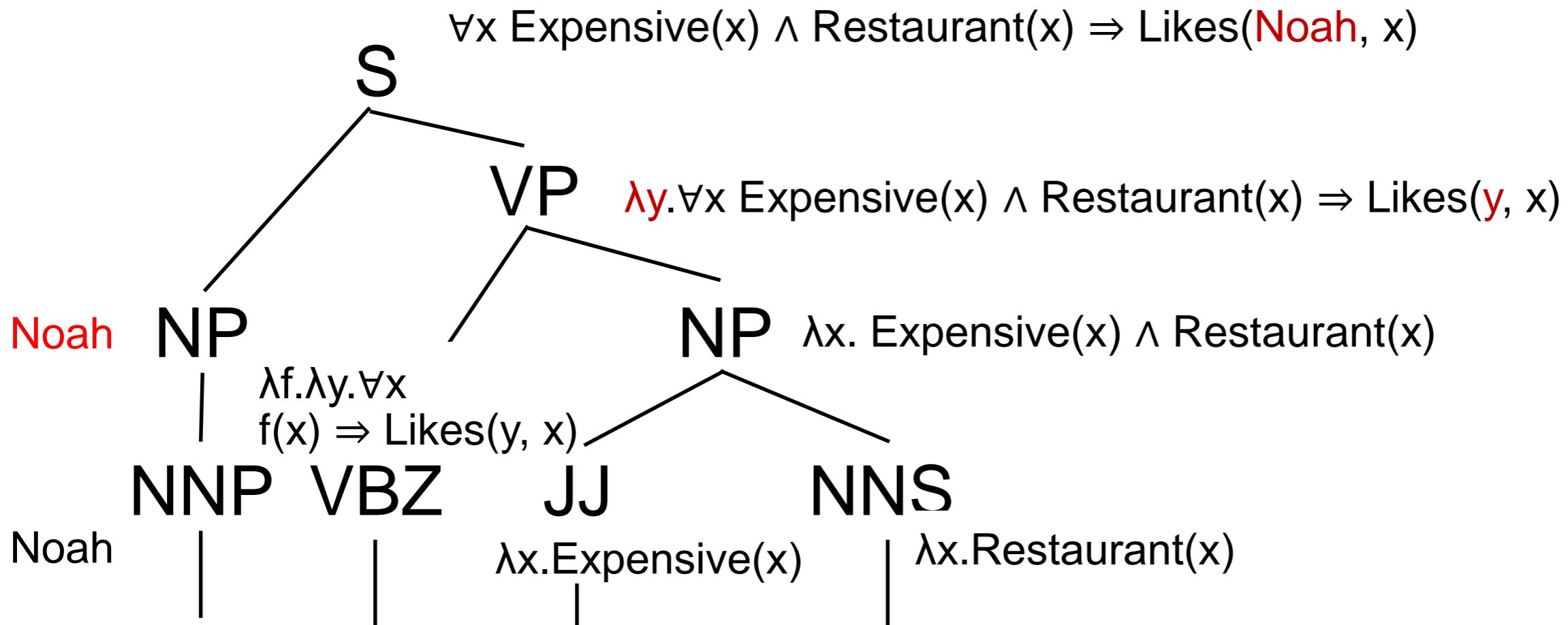
# An Example

$S \rightarrow NP VP \{ VP.sem(NP.sem) \}$



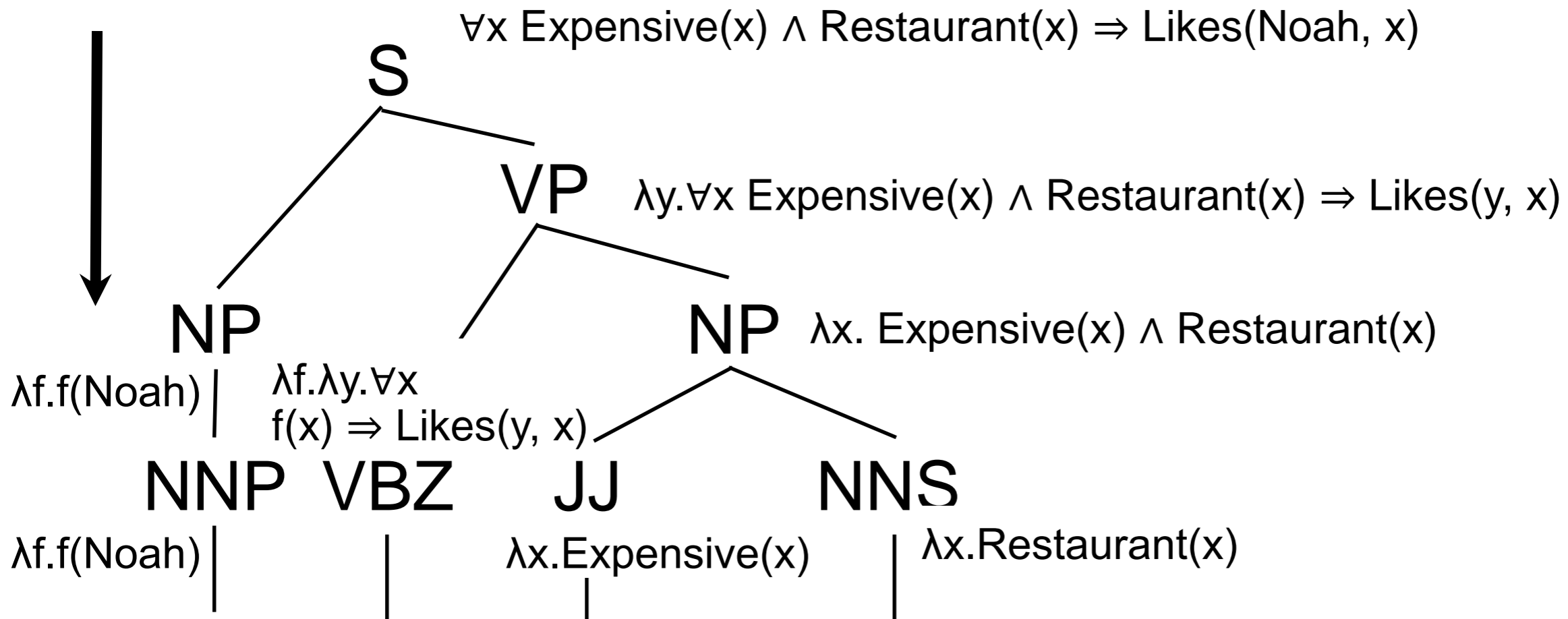
- *Noah likes expensive restaurants.*
- $\forall x \text{ Restaurant}(x) \wedge \text{Expensive}(x) \Rightarrow \text{Likes}(\text{Noah}, x)$

# An Example



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

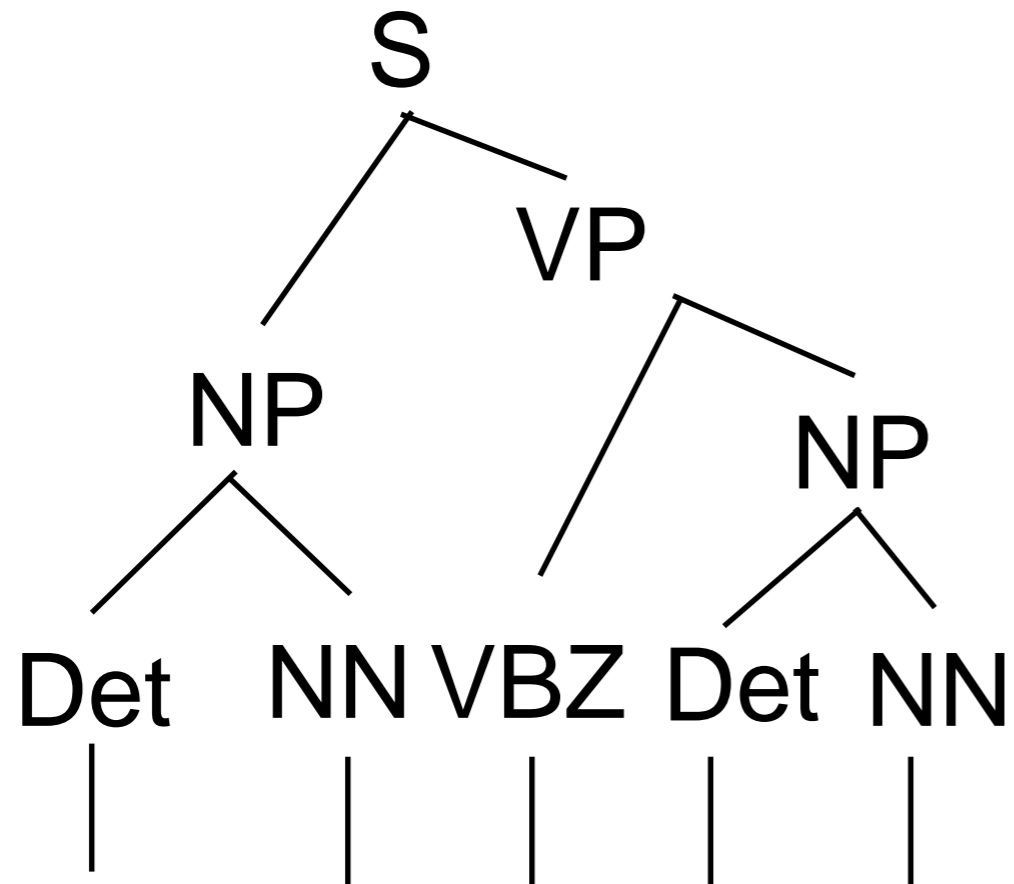
# Alternative (Following *SLP*)



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

$S \rightarrow NP VP \{ NP.sem(VP.sem) \}$

# Quantifier Scope Ambiguity



$S \rightarrow NP VP \{ NP.sem(VP.sem) \}$

$NP \rightarrow Det NN \{ Det.sem(NN.sem) \}$

$VP \rightarrow VBZ NP \{ VBZ.sem(NP.sem) \}$

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

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

$NN \rightarrow \text{man} \{ \lambda v.Man(v) \}$

$NN \rightarrow \text{woman} \{ \lambda y.Woman(y) \}$

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

- *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.
  - $\forall u \text{ Man}(u) \Rightarrow \exists x \text{ Woman}(x) \wedge \text{Loves}(u, x)$
  - $\exists x \text{ Woman}(x) \wedge \forall u \text{ Man}(u) \Rightarrow \text{Loves}(u, x)$
- 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 things 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
- Happily, philosophers/logicians have thought about this too

# Generalized Quantifiers

---

- In FOL, we only have universal and existential quantifiers
- One formal extension is type-restriction of the quantified variable: *Everyone likes Udipi*:

$\forall x \text{ Person}(x) \Rightarrow \text{Likes}(x, \text{Udipi})$  becomes

$\forall x | \text{Person}(x). \text{Likes}(x, \text{Udipi})$

- English and other languages have a much larger set of quantifiers: *all, some, most, many, a few, the, ...*
- These have the same *form* as the original FOL quantifiers with type restrictions:

$\langle \text{quant} \rangle \langle \text{var} \rangle | \langle \text{restriction} \rangle . \langle \text{body} \rangle$

# Generalized Quantifier examples

---

- *Most dogs bark*

Most  $x \mid \text{Dog}(x) . \text{Barks}(x)$

- *Most barking things are dogs*

Most  $x \mid \text{Barks}(x) . \text{Dog}(x)$

- *The dog barks*

The  $x \mid \text{Dog}(x) . \text{Barks}(x)$

- *The happy dog barks*

The  $x \mid (\text{Happy}(x) \wedge \text{Dog}(x)) . \text{Barks}(x)$

- ***Interpretation*** and ***inference*** using these are harder...

# Speech Acts

---

- ***Mood*** of a sentence indicates relation between speaker and the concept (proposition) defined by the LF
- There can be operators that represent these relations:
  - ASSERT: the proposition is proposed as a fact
  - YN-QUERY: the truth of the proposition is queried
  - COMMAND: the proposition describes a requested action
  - WH-QUERY: the proposition describes an object to be identified

# ASSERT (Declarative mood)

---

- *The man eats a peach*

ASSERT(The x | Man(x) . (A y | Peach(y) . Eat(x,y)))



# YN-QUERY (Interrogative mood)

---

- *Does the man eat a peach?*

YN-QUERY(The x | Man(x) . (A y | Peach(y) . Eat(x,y)))

# COMMAND (Imperative mood)

---

- *Eat a peach, (man).*

COMMAND(A y | Peach(y) . Eat(\*HEARER\*,y))

# WH-QUERY

---

- *What did the man eat?*

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

- One of a whole set of new quantifiers for wh-questions:
  - *What*: WH x | Thing(x)
  - *Which dog*: WH x | Dog(x)
  - *Who*: WH x | Person(x)
  - *How many men*: HOW-MANY x | Man(x)

# Other complications

---

- 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
- Etc.

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

```
(*A-BE
(FORM FINITE)
(TENSE PRESENT)
(MOOD DECLARATIVE)
(PUNCTUATION PERIOD)
(IMPERSOAL -)
(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)))))))))
```