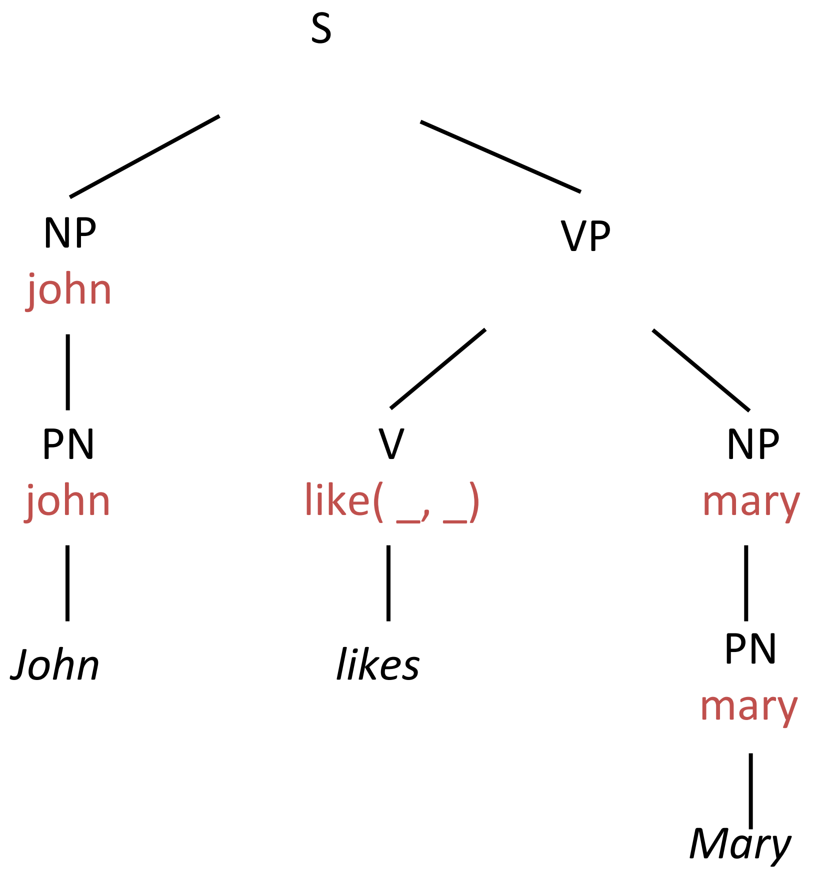
FLST: Semantics IV

Vera Demberg
vera@coli.uni-sb.de

http://www.coli.uni-saarland.de/courses/FLST/2011/

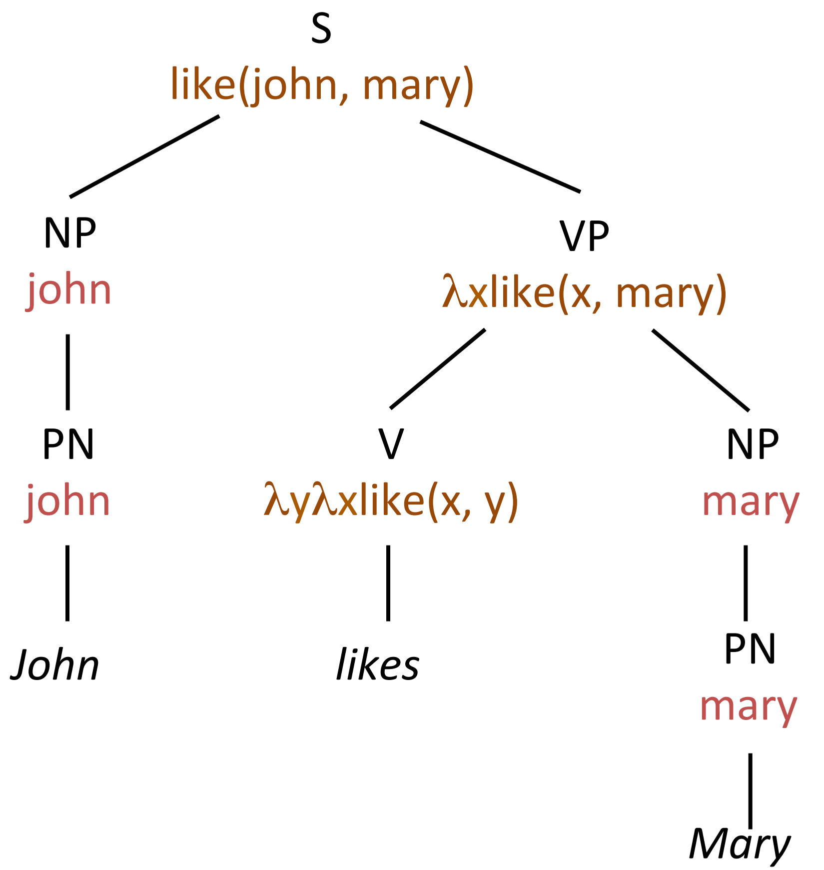
Semantic Composition

- *John likes Mary* ⇒ like(john, mary)



Semantic Composition: Reduction

- *John likes Mary* ⇒ like(john, mary)



A Challenge for Compositional Semantics

Every student presented a paper

$$\forall d\ (student(d) \rightarrow \exists p\ (paper(p) \wedge present(d,p)))$$

A Challenge for Compositional Semantics

Every student presented a paper

$$\forall d\ (\text{student}(d) \rightarrow \exists p\ (\text{paper}(p) \wedge \text{present}(d,p)))$$

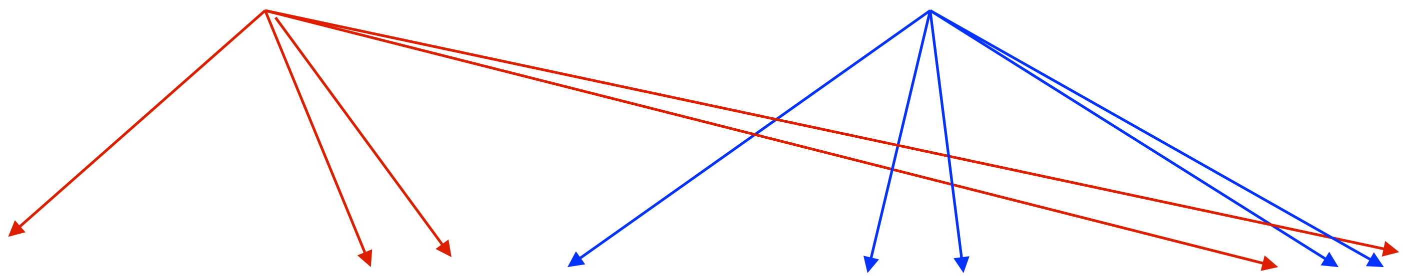
A Challenge for Compositional Semantics

Every student presented a paper

$$\forall d\ (student(d) \rightarrow \exists p\ (paper(p) \wedge present(d,p)))$$

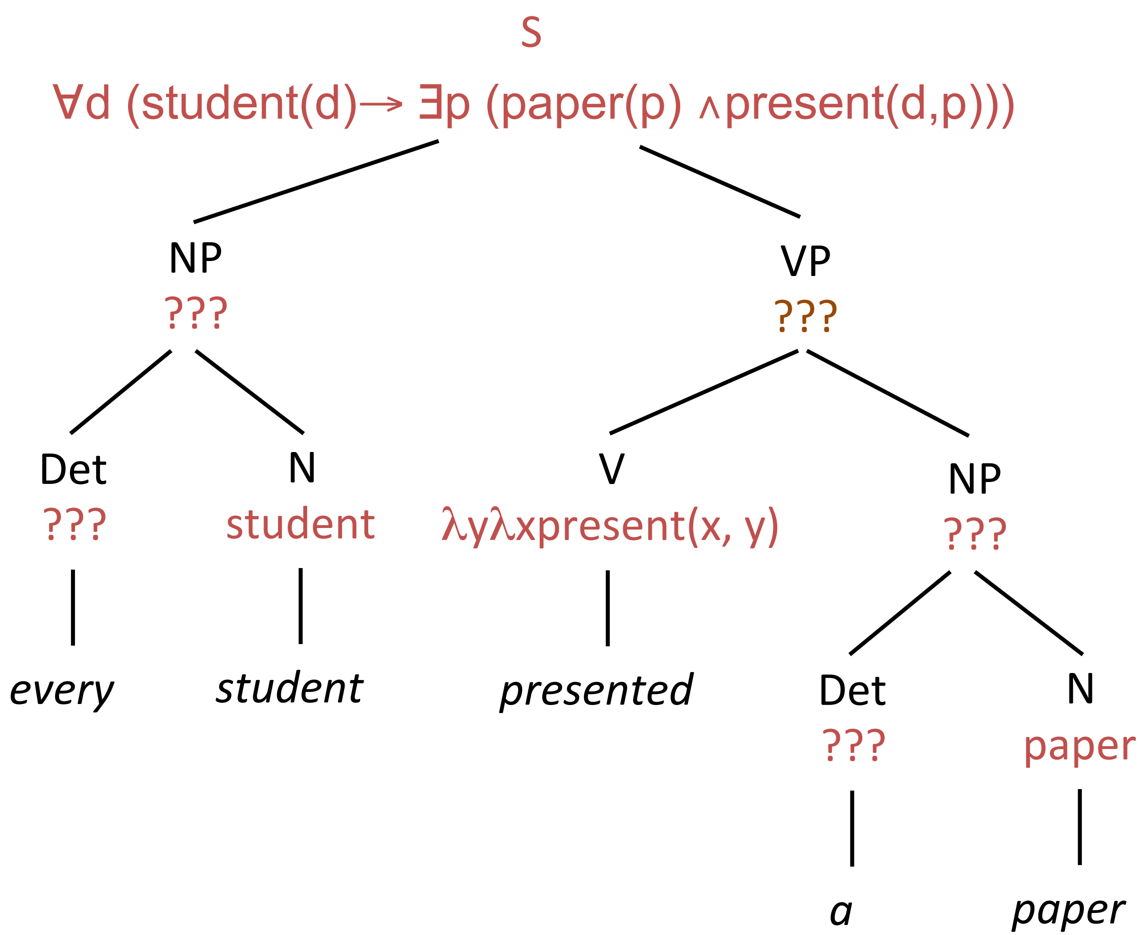
A Challenge for Compositional Semantics

Every student presented a paper



$\forall d\ (student(d) \rightarrow \exists p\ (paper(p) \wedge present(d,p)))$

A Challenge for Compositional Semantics



Attributive Adjectives

- *Mary owns a rusty bicycle*
  ∃b (own(mary, b) ∧ rusty(b) ∧ bicycle(b))
- *Mary owns an expensive bicycle*
  ∃b (own(mary, b) ∧ expensive(b) ∧ bicycle(b)) ??
- *Mary owns an expensive vehicle*
- *Bill is a poor piano player* ???
  poor(bill) ∧ piano_player(bill)

Attributive Adjectives

- For subclass of adjectives (*rusty, married, blond*), the semantics of the attributive use can be defined as **intersection** between two set-denoting predicates: **a rusty bicycle is rusty and a bicycle**.
  → We call them intersective or referential adjectives.

- For another class of adjectives (*expensive, fast, tall, good*), we can only state that the adjective modifier **restricts** the denotation of the nominal argument**: a poor piano player is a piano player** (after all), **but not necessarily poor**.
  → We call them restrictive or relative adjectives.

- In the general case, attributive adjectives take a predicate as argument and return a modified predicate.
  We would like to write:

- ∃b (own(mary, b) ∧ expensive(bicycle)(b))

Second-order Predicates

- *Flipper is a dolphin, A dolphin is a mammal* ⊨ *Flipper is a mammal*
- *Bill is blond, Blond is a hair colour* ⊨ *???*
- "*colour*" is a second-order predicate: a predicate that takes first-order predicates as argument. We would like to write:
  - blond(bill) and hair_colour(blond) both are formulas
  - From blond(bill) and hair_colour(blond), nothing follows.

Higher-Order Logic

- *Most students passed (the exam)*

- *"Most"* is a second-order relation that takes two first-order relations as arguments. *Most students passed* expresses a relation between the predicates *student* and *pass*, which applies if the number of passing students exceeds the number of non-passing students. We would like to write:

- most(student, pass)

Higher-Order Logic

- "Predicate Logic", as we have defined it so far, is actually First-Order Logic (FOL in short): All predicates and relations take terms as their denotations, which denote entities (members of U_M) in the model structure. Quantification is over variables, which also take entities as values.

- There are phenomena in NL semantics, which cannot be described with FOL, but require some kind of "Higher-Order Logic".

The Language of Type Theory

- The set of basic types is {e, t} :
  - e (for entity) is the type of individual terms
  - t (for truth value) is the type of formulas

- All pairs (σ, τ) made up of (basic or complex) types σ, τ are types. (σ, τ) is the type of functions which map arguments of type σ to values of type τ.

- In short: The set of types is the smallest set **T** such that $e,t \in \mathbf{T}$, and if $\sigma,\tau \in \mathbf{T}$, then also $(\sigma,\tau) \in \mathbf{T}$.

Some Types for NL Expressions

- Proper name bill: e
- Sentence it_rains: t
- One-place predicate constant:
  work, student: (e,t)
- Attributive adjective:
  married, poor: ((e,t),(e,t))
- Degree modifier:
  very, relatively: (((e,t),(e,t)),((e,t),(e,t)))
- Second-order predicate
  hair_colour: ((e,t),t)
- Two-place relation:
  like, own: (e,(e,t))

Function Application

- The syntax of type theory is very similar to FOL syntax.

- **Important change:** The clause for combining relational expressions with individual expressions is replaced by the more general rule of functional application, where "WE_σ" refers to the set of well-formed expressions of type σ:

- If $\alpha \in WE_{<\sigma, \tau>}$, $\beta \in WE_\sigma$, then $\alpha(\beta) \in WE_\tau$.

- Note: A functor expression of a complex type applied to an appropriate argument yields a (more complex) expression of less complex type.

Attributive Adjectives

Bill is a poor piano player

poor: ((e,t),(e,t))   piano_player: (e,t)

poor(piano_player): (e,t)   bill: e

poor(piano_player)(bill): t

Second-order predicates

Bill is blond.

$$\frac{\text{bill: e} \quad \text{blond: (e,t)}}{\text{blond(bill): t}}$$

Blond is a hair colour.

$$\frac{\text{blond: (e,t)} \quad \text{hair\_colour : ((e,t),t)}}{\text{hair\_colour (blond): t}}$$

Bill is a hair colour ???

- hair_colour(bill) is not even a well-formed expression.

Higher-Order Variables

Bill has the same hair colour as John.

$\exists \mathcal{G}\ (\text{hair\_colour}(\mathcal{G}) \wedge \mathcal{G}\ (\text{bill}) \wedge \mathcal{G}\ (\text{john}))$

Type-Theoretic Model Structure

- Let U be a non-empty set of entities.
- The domain of possible denotations for every type τ, D_τ , is given by:
  - $D_e = U$
  - $D_t = \{0,1\}$
  - $D_{<\sigma, \tau>}$ is the set of all functions from D_σ to D_τ
- A type-theoretic model structure is a pair M = <U, V>, where
  - U is a non-empty domain of individuals
  - V is an interpretation function, which assigns to each non-logical constant of type σ a member of D_σ.

Denotation of One-Place Predicates

- Let U consist of John, Bill, Mary, Paul, and Sally (persons, not proper names!)
  - $D_t = \{0,1\}$
  - $D_e = U = \{j, b, m, p, s\}$
  - $D_{<e,t>} = \{ \begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 0 \\ s \rightarrow 1 \end{bmatrix}, \begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 1 \\ s \rightarrow 1 \end{bmatrix}, \begin{bmatrix} j \rightarrow 0 \\ b \rightarrow 1 \\ m \rightarrow 0 \\ p \rightarrow 0 \\ s \rightarrow 1 \end{bmatrix}, \ldots\}$
- Functions into {0,1} are called "characteristic functions": They provide an equivalent way to describe sets (in the above example, $D_{<e,t>}$ could be written as { {j,m,s}, {j, m, p, s}, {b, s}, ...})

A Member of $D_{<<e,t>, <e,t>>}$

$$\begin{bmatrix} \begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 0 \\ s \rightarrow 1 \end{bmatrix} & \rightarrow & \begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 0 \\ s \rightarrow 0 \end{bmatrix} \\ \begin{bmatrix} j \rightarrow 0 \\ b \rightarrow 1 \\ m \rightarrow 0 \\ p \rightarrow 0 \\ s \rightarrow 1 \end{bmatrix} & \rightarrow & \begin{bmatrix} j \rightarrow 0 \\ b \rightarrow 0 \\ m \rightarrow 0 \\ p \rightarrow 0 \\ s \rightarrow 1 \end{bmatrix} \\ \begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 1 \\ s \rightarrow 1 \end{bmatrix} & \rightarrow & \begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 0 \\ s \rightarrow 0 \end{bmatrix} \\ & \ldots & \end{bmatrix}$$

Interpretation Function, Examples

$V_M(\textit{john}) = j$

$V_M(\textit{mary}) = m$

$V_M(\textit{piano player})$: $\begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 0 \\ s \rightarrow 1 \end{bmatrix}$

$V_M(\textit{semanticist})$: $\begin{bmatrix} j \rightarrow 0 \\ b \rightarrow 1 \\ m \rightarrow 0 \\ p \rightarrow 0 \\ s \rightarrow 1 \end{bmatrix}$

$V_M(\textit{skier})$: $\begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 1 \\ s \rightarrow 1 \end{bmatrix}$

Interpretation Function, Examples

$$V_M(\textit{talented}): \begin{bmatrix} \begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 0 \\ s \rightarrow 1 \end{bmatrix} \rightarrow \begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 0 \\ s \rightarrow 0 \end{bmatrix} \\ \begin{bmatrix} j \rightarrow 0 \\ b \rightarrow 1 \\ m \rightarrow 0 \\ p \rightarrow 0 \\ s \rightarrow 1 \end{bmatrix} \rightarrow \begin{bmatrix} j \rightarrow 0 \\ b \rightarrow 0 \\ m \rightarrow 0 \\ p \rightarrow 0 \\ s \rightarrow 1 \end{bmatrix} \\ \begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 1 \\ s \rightarrow 1 \end{bmatrix} \rightarrow \begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 0 \\ s \rightarrow 0 \end{bmatrix} \\ \ldots \end{bmatrix}$$

Interpretation of Functor-Argument Structures

$$[\![\alpha(\beta)]\!]^{M,g} = [\![\alpha]\!]^{M,g}([\![\beta]\!]^{M,g})$$

Example

John is a talented piano-player

$\Rightarrow$ talented(piano-player)(john)

$[\![\text{talented(piano-player)(john)}]\!]^{M,g} =$

$[\![\text{talented(piano-player)}]\!]^{M,g}([\![\text{john})]\!]^{M,g} =$

$[\![\text{talented}]\!]^{M,g}([\![\text{piano-player})]\!]^{M,g})([\![\text{john}]\!]^{M,g}) = V_M$

$V_M(\text{talented})(V_M(\text{piano-player}))\ (V_M(\text{john}))$

Example continued:

V_M(*piano-player*)

V_M(*talented*):

$$\begin{bmatrix} \begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 0 \\ s \rightarrow 1 \end{bmatrix} \rightarrow \begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 0 \\ s \rightarrow 0 \end{bmatrix} \\ \begin{bmatrix} j \rightarrow 0 \\ b \rightarrow 1 \\ m \rightarrow 0 \\ p \rightarrow 0 \\ s \rightarrow 1 \end{bmatrix} \rightarrow \begin{bmatrix} j \rightarrow 0 \\ b \rightarrow 0 \\ m \rightarrow 0 \\ p \rightarrow 0 \\ s \rightarrow 1 \end{bmatrix} \\ \begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 1 \\ s \rightarrow 1 \end{bmatrix} \rightarrow \begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 0 \\ s \rightarrow 0 \end{bmatrix} \\ \ldots \end{bmatrix}$$

Example continued:

V_M(*piano-player*)

V_M(*talented*):

V_M(*talented*)(V_M(*piano-player*))

$$\begin{bmatrix} \begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 0 \\ s \rightarrow 1 \end{bmatrix} & \rightarrow & \begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 0 \\ s \rightarrow 0 \end{bmatrix} \\ \begin{bmatrix} j \rightarrow 0 \\ b \rightarrow 1 \\ m \rightarrow 0 \\ p \rightarrow 0 \\ s \rightarrow 1 \end{bmatrix} & \rightarrow & \begin{bmatrix} j \rightarrow 0 \\ b \rightarrow 0 \\ m \rightarrow 0 \\ p \rightarrow 0 \\ s \rightarrow 1 \end{bmatrix} \\ \begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 1 \\ s \rightarrow 1 \end{bmatrix} & \rightarrow & \begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 0 \\ s \rightarrow 0 \end{bmatrix} \\ & \ldots & \end{bmatrix}$$

Example continued:

$V_M(\textit{john})$

$V_M(\textit{piano-player})$

$V_M(\textit{talented})(V_M(\textit{piano-player}))(j)$

$V_M(\textit{talented})(V_M(\textit{piano-player}))$

$V_M(\textit{talented})$:

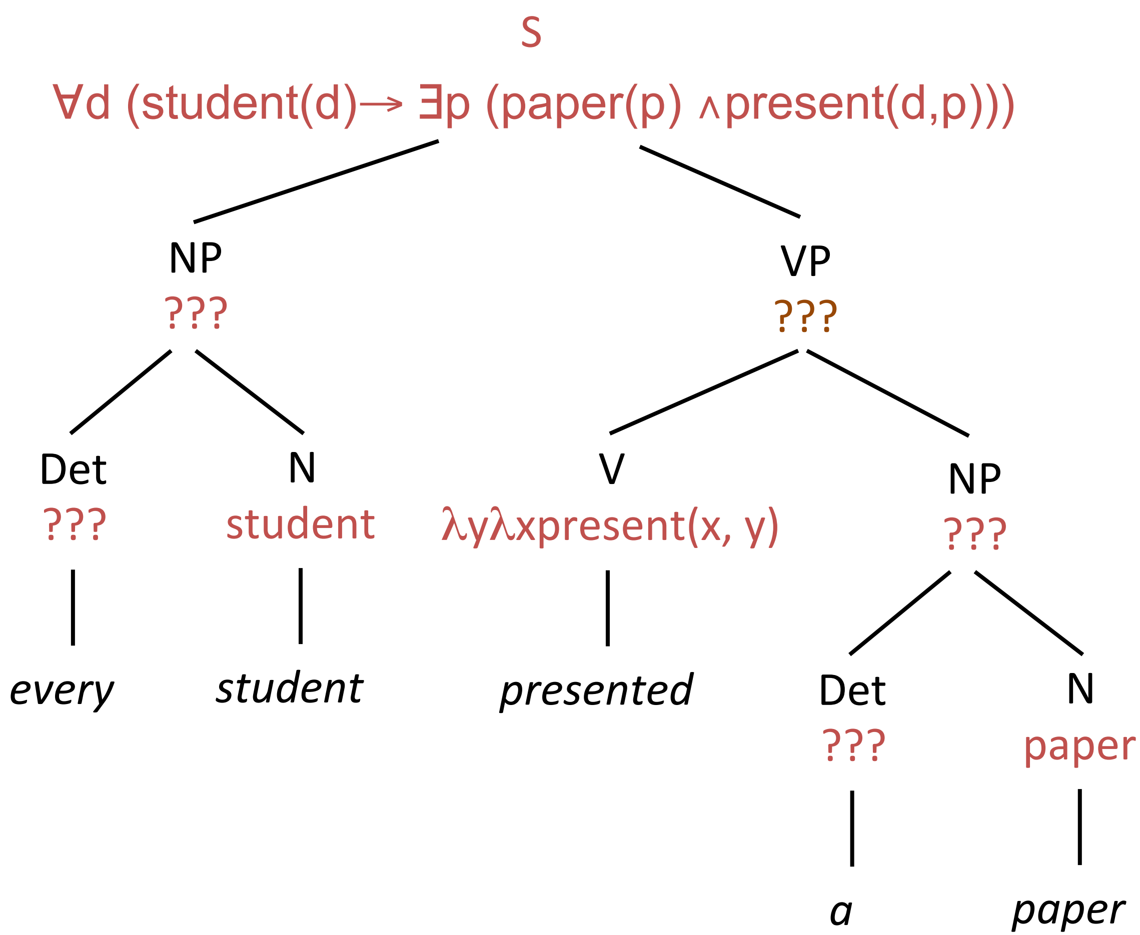
$$
\begin{bmatrix}
\begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 0 \\ s \rightarrow 1 \end{bmatrix} & \rightarrow & \begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 0 \\ s \rightarrow 0 \end{bmatrix} \\
\begin{bmatrix} j \rightarrow 0 \\ b \rightarrow 1 \\ m \rightarrow 0 \\ p \rightarrow 0 \\ s \rightarrow 1 \end{bmatrix} & \rightarrow & \begin{bmatrix} j \rightarrow 0 \\ b \rightarrow 0 \\ m \rightarrow 0 \\ p \rightarrow 0 \\ s \rightarrow 1 \end{bmatrix} \\
\begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 1 \\ s \rightarrow 1 \end{bmatrix} & \rightarrow & \begin{bmatrix} j \rightarrow 1 \\ b \rightarrow 0 \\ m \rightarrow 1 \\ p \rightarrow 0 \\ s \rightarrow 0 \end{bmatrix} \\
& \ldots &
\end{bmatrix}
$$

Back to "Every student presented a paper"

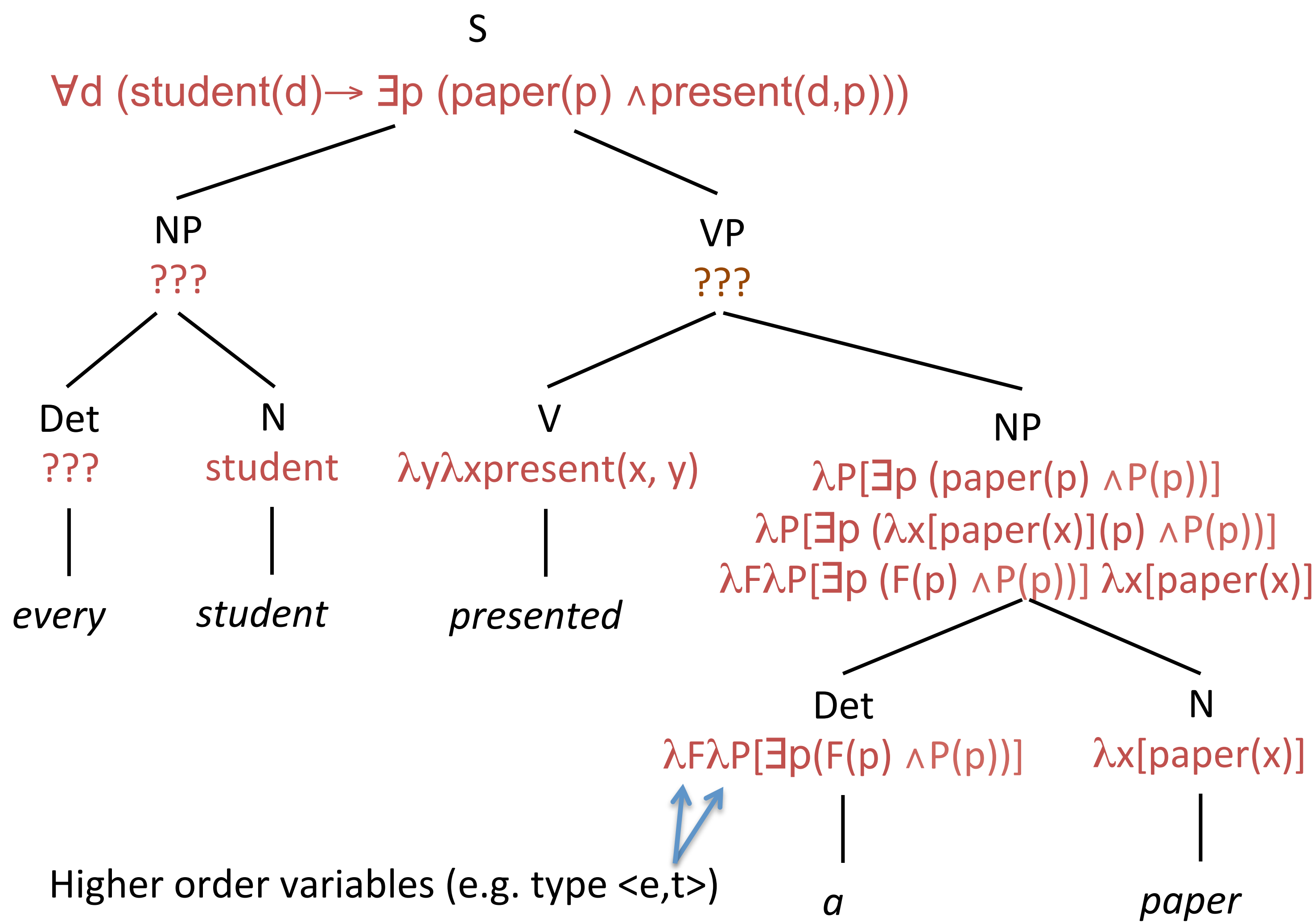
- Now that we've looked at higher order logic, let's get back to our earlier problem.

"Every student presented a paper."

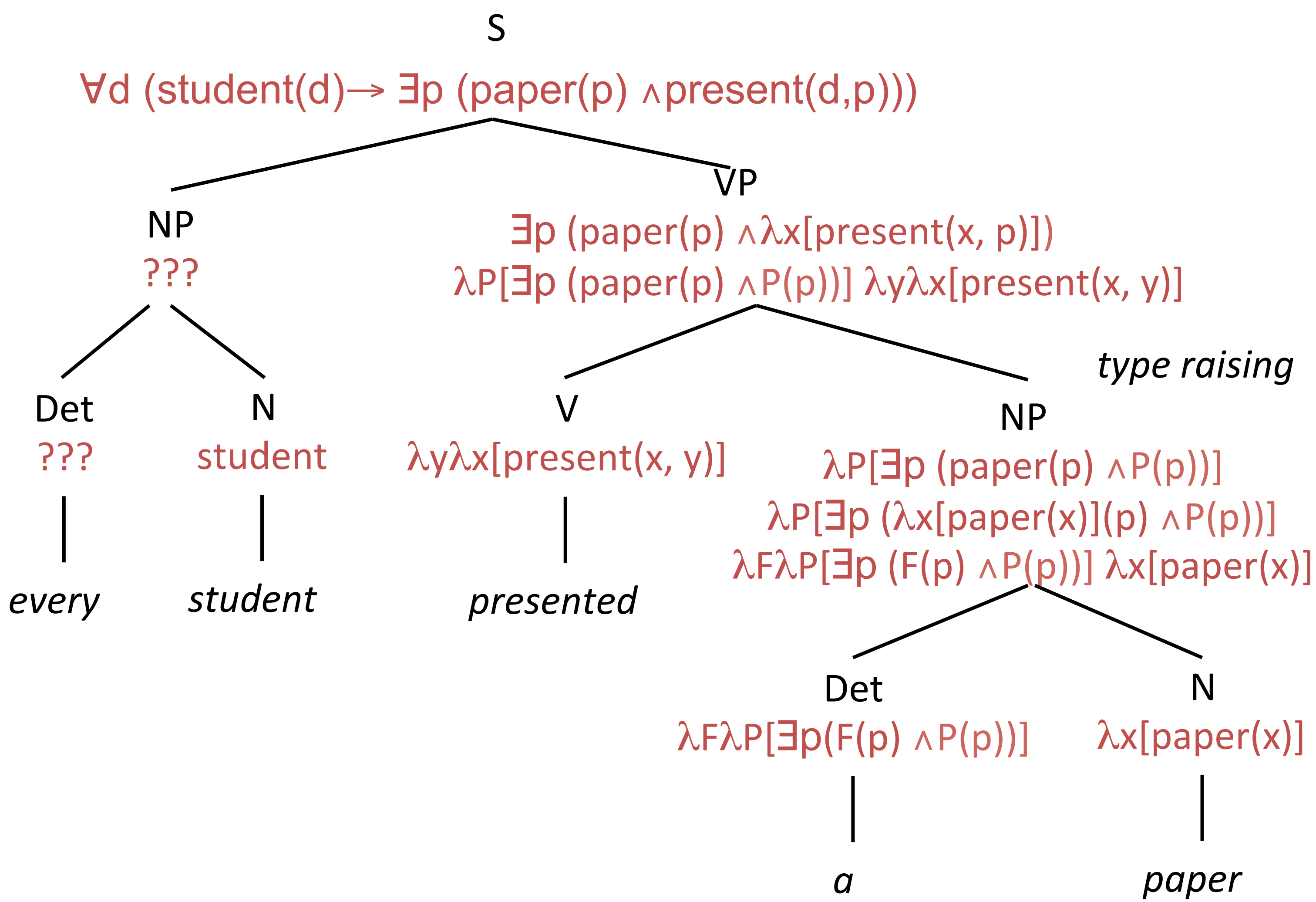
A Challenge for Compositional Semantics



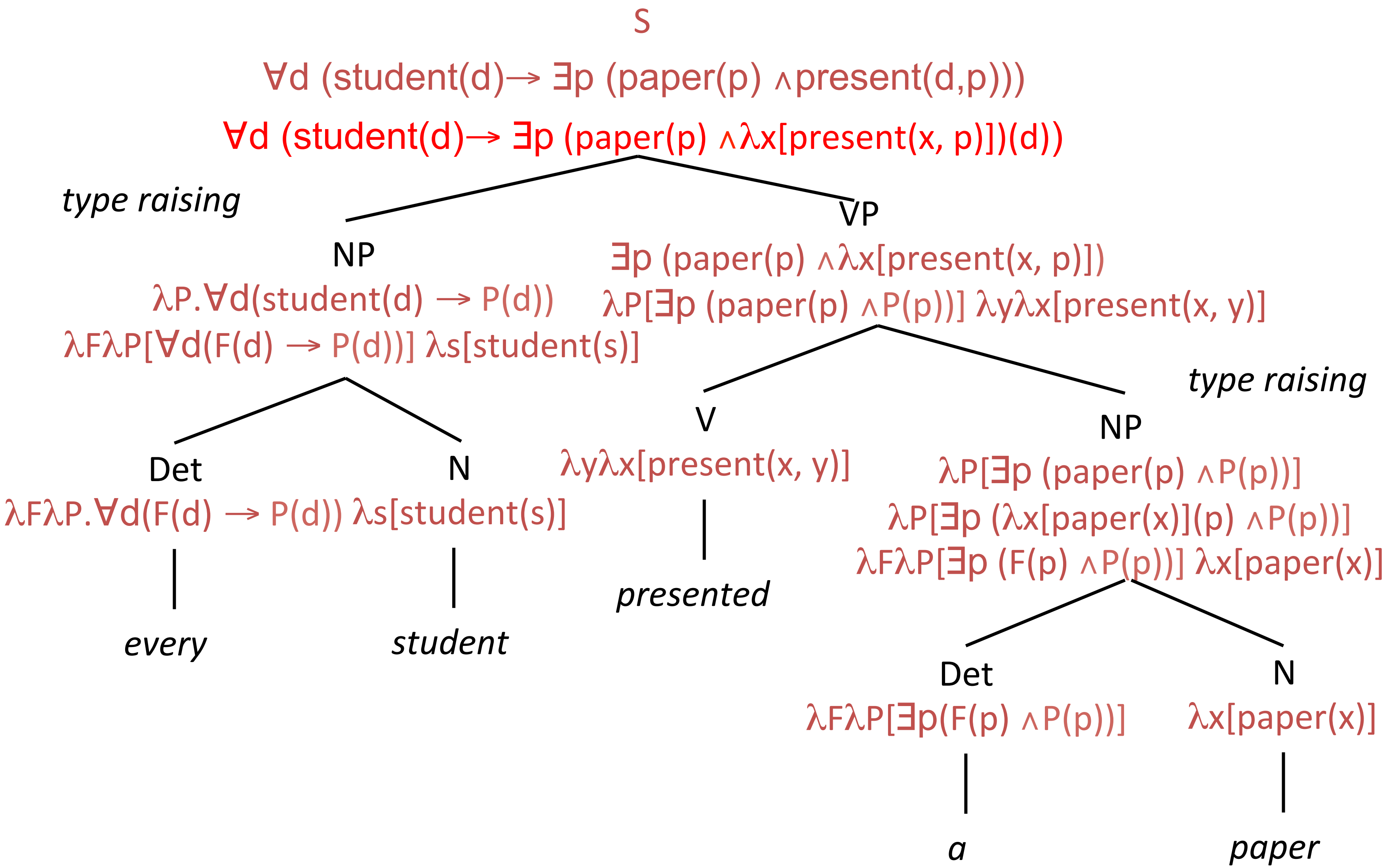
A Challenge for Compositional Semantics



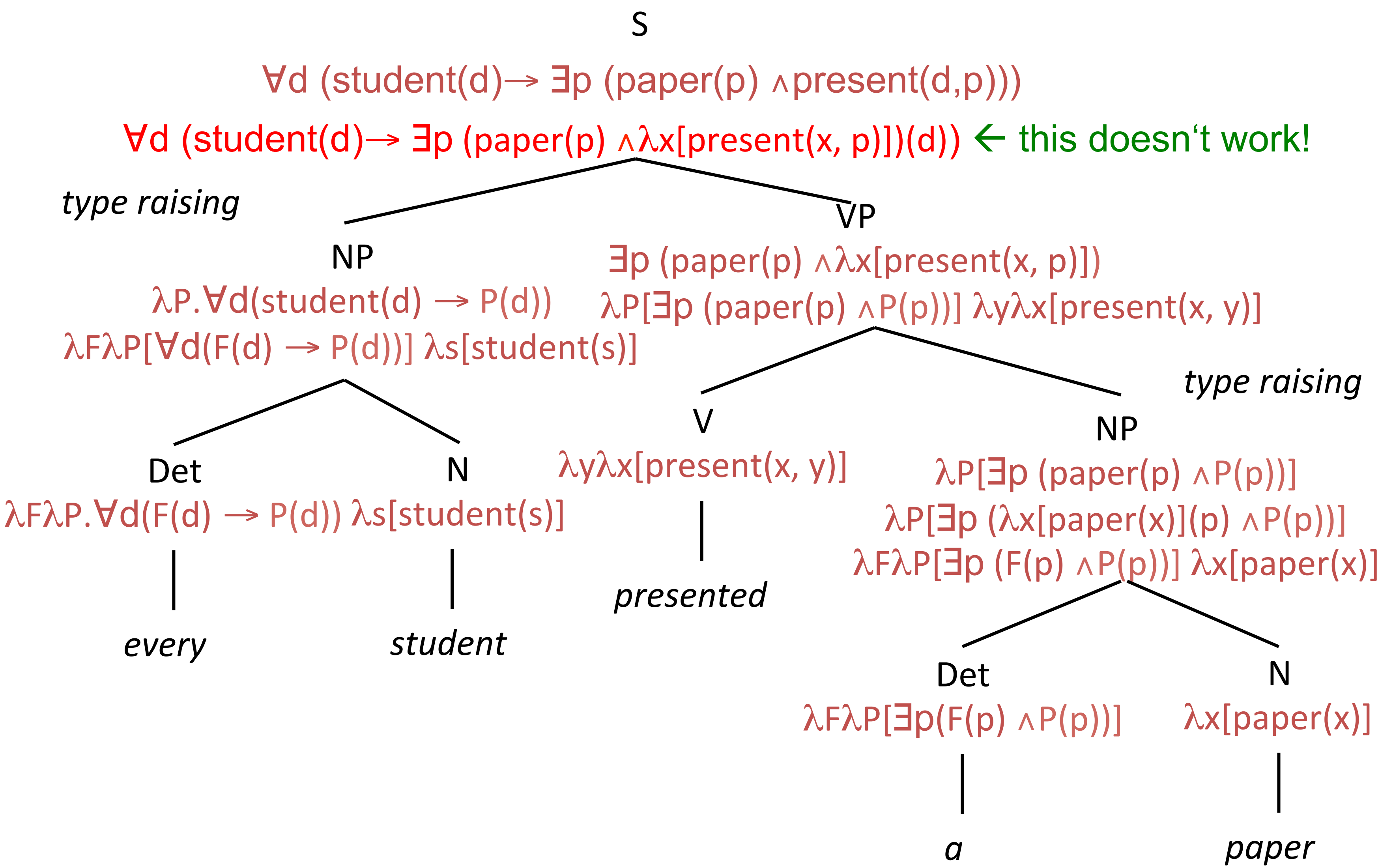
A Challenge for Compositional Semantics



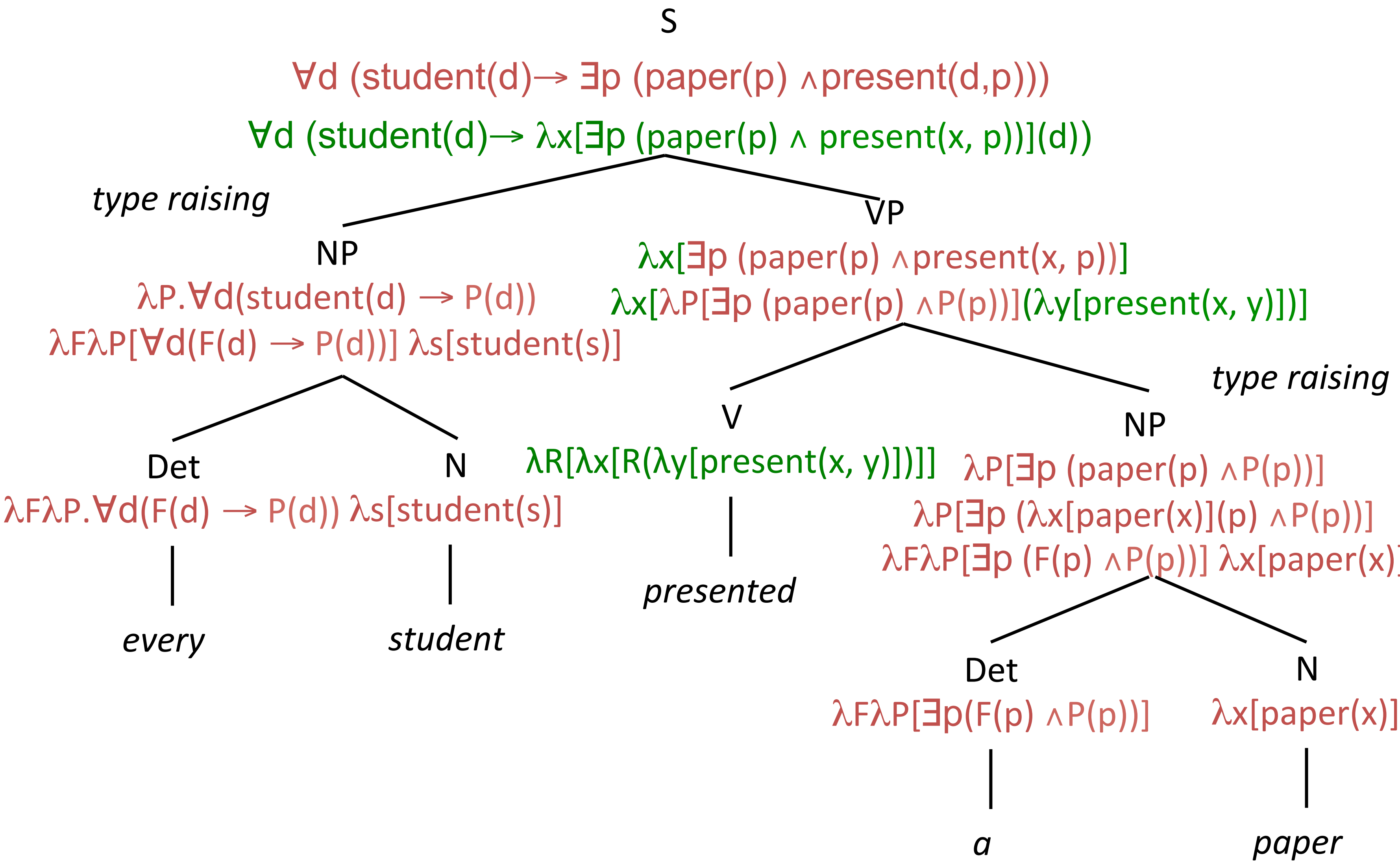
A Challenge for Compositional Semantics



A Challenge for Compositional Semantics



A Challenge for Compositional Semantics



A Challenge for Distributional Semantics

- We‘ve seen how to get reading
  $\forall d\ (student(d) \rightarrow \exists p\ (paper(p) \wedge present(d,p)))$
  for sentence „Every student presented a paper“

- but this was quite complicated

- plus, there is the other reading:
  $\exists p\ (paper(p) \wedge \forall d\ (student(d) \rightarrow present(d,p)))$

→ take Semantic Theory