Scope ambiguities, continuations and strengths

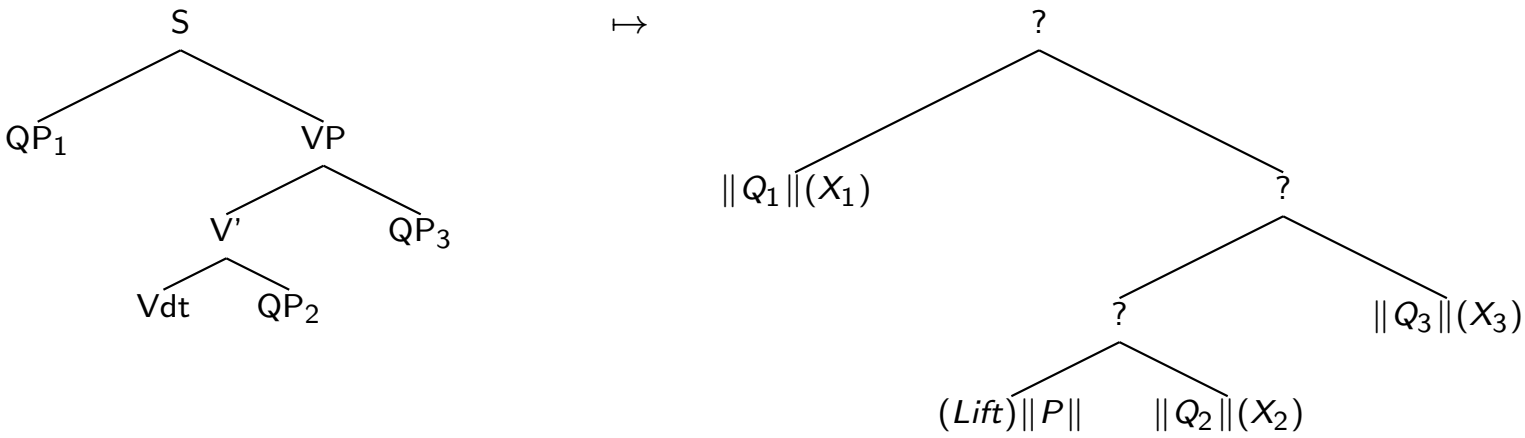
Justyna Grudzińska and Marek Zawadowski

University of Warsaw

Fourth Workshop on Natural Language and Computer Science (NLCS)
New York City, NY
July 10, 2016

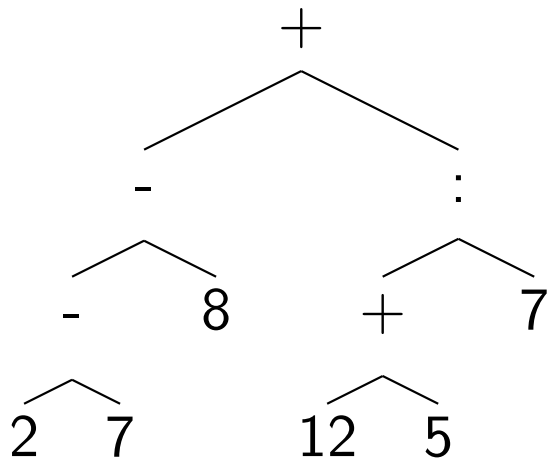
Introduction

Some teacher gave every student most books (6-way ambiguous)



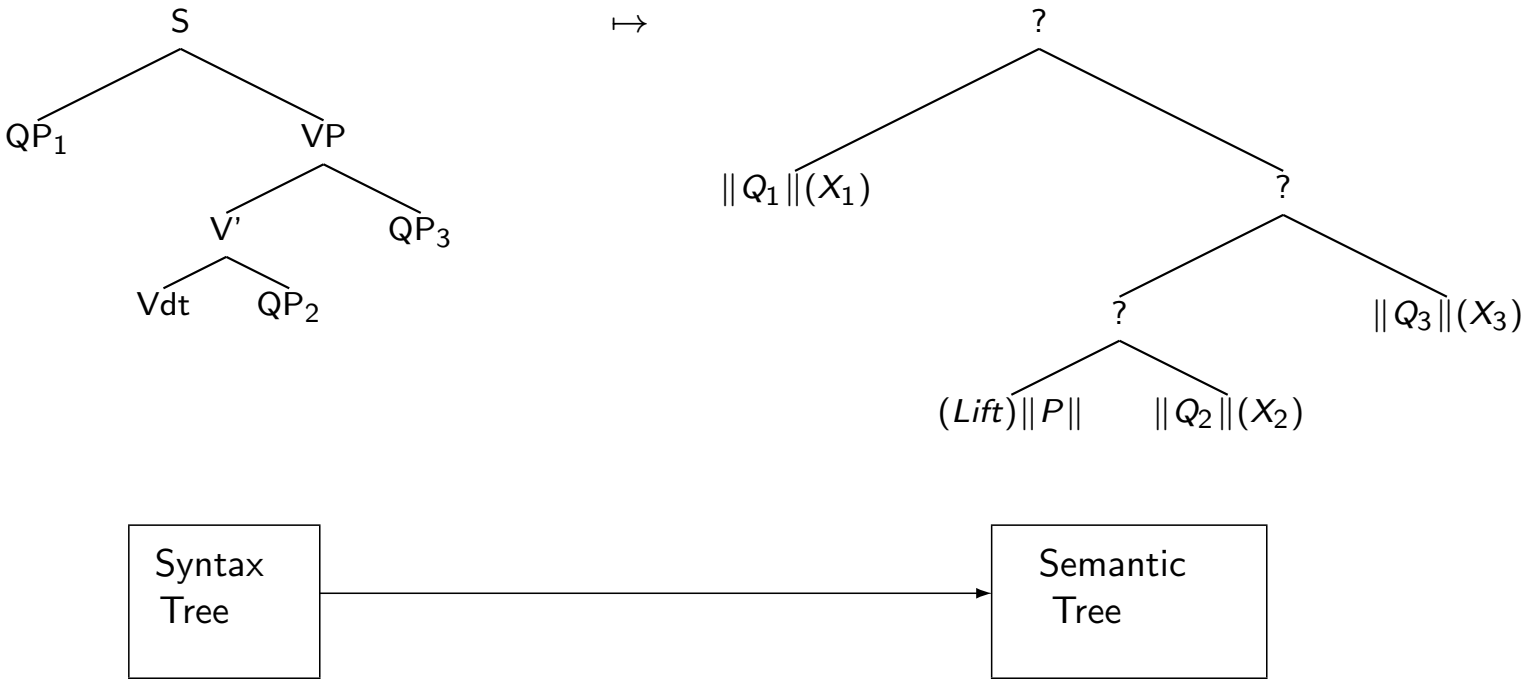
Introduction

$$((2 - 7) - 8) + ((12 + 5) : 7)$$



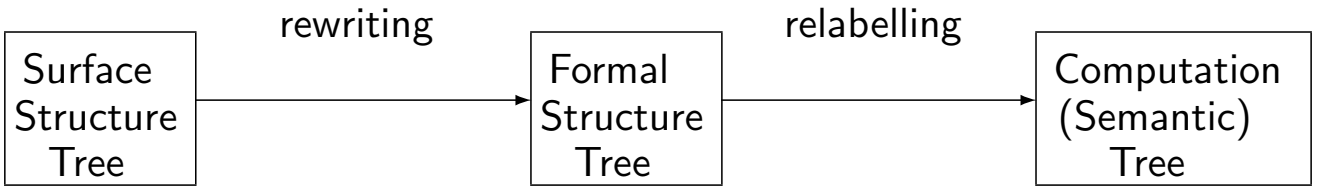
Introduction

Some teacher gave every student most books (6-way ambiguous)



Challenges

- calculate the truth-value in each of sentence's readings
- leaving the shape of the syntactic tree as intact as possible
- keep the operations as simple as possible



Mainstream Scope Assignment Strategies

Strategy	**A** *Movement Strategy (May 77)*	**B** *Polyadic Approach (May 85)*	**C** *Continuation-based Approach (Barker 2002)*
Rewrite rules	*QR*	*QR*, *Rotation*	*No rewrite rules* (**in situ**)
Relabelling inner nodes (operations)	**mos**	**mos** **pile′up**	**CPS**
Relabelling leaves (semantic values)	*Predicate* $\mapsto$ *relation* $QP \mapsto$ $\mathcal{C}$*-comp.*	*Predicate* $\mapsto$ *relation* $QP \mapsto$ $\mathcal{C}$*-comp.*	*Predicate* $\mapsto$ *continuized relation* $QP \mapsto$ $\mathcal{C}$*-comp.*

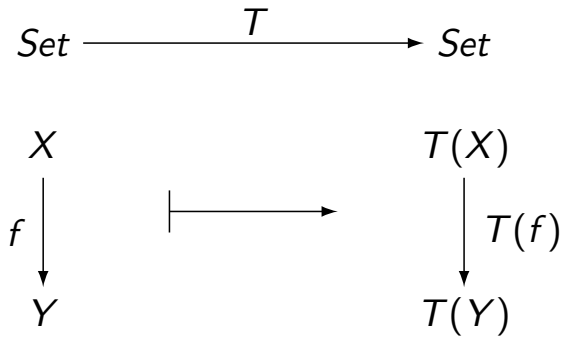
Strategy	**A** *Movement Strategy (May 77)*	**B** *Polyadic Approach (May 85)*	**C** *Cont-based Approach (Barker 02)*	**D** **Strat C** *& new* **shuffle** *operations*
6 readings	✓	✓	-	✓
in situ	-	-	✓	✓
simple operations	✓	✓?	✓??	*???*

Outline

1. Monads
   1. definition
   2. strength(s) and derived operations
2. Continuation monad
   1. definition
   2. strength(s) and derived operations
3. Scope-Assignment Strategies
   1. Strategy A: traditional movement strategy (May 77, Montague 74);
   2. Strategy B: polyadic approach (May 85, Keenan 87, Zawadowski 89);
   3. Strategy C: continuation-based (in situ) approach (Barker 2002).
4. Empirically adequate in situ Strategy D: **shuffle**-operations.

Monad

A **monad** on *Set* is a triple (+ conditions ...):
T-**computations** (endofunctor)



unit (return) (natural transformation)

$$\eta : 1_{Set} \longrightarrow T$$
$$\eta_X : X \to T(X)$$

lifts elements of X as T-computations.

multiplication (natural transformation)

$$\mu : T^2 \longrightarrow T$$
$$\mu_X : T^2(X) = T(T(X)) \to T(X)$$

flattens T-computations on T-computations to T-computations.

Let (T, η, μ) be a monad on *Set*.

The **left (right) strength** is a natural transformation with components

$$\mathbf{st}^l{}_{X,Y} : T(X) \times Y \longrightarrow T(X \times Y)$$

$$(\mathbf{st}^r{}_{X,Y} : X \times T(Y) \longrightarrow T(X \times Y))$$

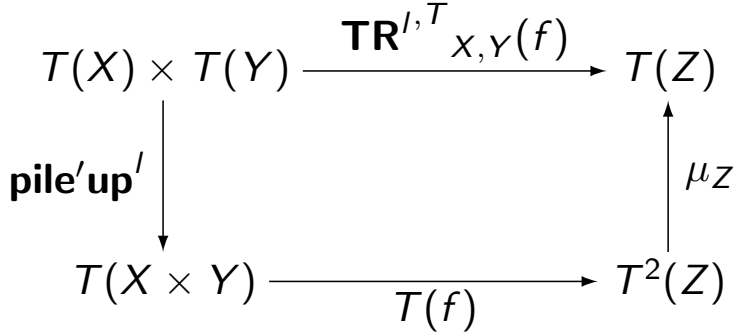
for sets X and Y (plus some conditions).

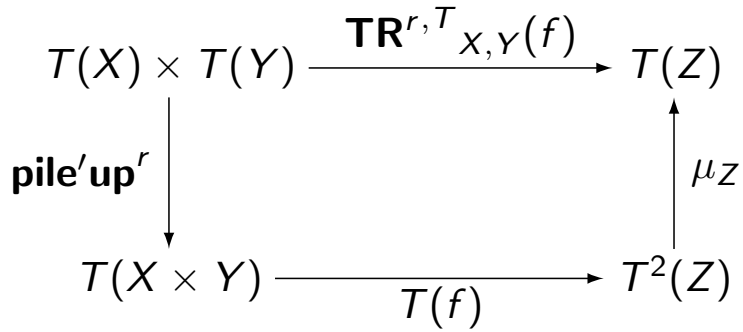
Strengths allow to lift pairs of T-computations to T-computations on products!

For any sets X and Y, the *left (right) pile up* $\mathbf{pile'up}^{l}{}_{X,Y}$ ($\mathbf{pile'up}^{r}{}_{X,Y}$) is defined as the composition

$$\begin{array}{ccc} T(X)\times T(Y) & \xrightarrow{\mathbf{pile'up}^{l}{}_{X,Y}} & T(X\times Y) \\ \downarrow{\scriptstyle \mathbf{st}^{l}{}_{X,T(Y)}} & & \uparrow{\scriptstyle \mu_{X\times Y}} \\ T(X\times T(Y)) & \xrightarrow[T(\mathbf{st}^{r}{}_{X,Y})]{} & T^2(X\times Y) \end{array} \qquad \begin{array}{ccc} T(X)\times T(Y) & \xrightarrow{\mathbf{pile'up}^{r}{}_{X,Y}} & T(X\times Y) \\ \downarrow{\scriptstyle \mathbf{st}^{r}{}_{T(X),Y}} & & \uparrow{\scriptstyle \mu_{X\times Y}} \\ T(T(X)\times Y) & \xrightarrow[T(\mathbf{st}^{l}{}_{X,Y})]{} & T^2(X\times Y) \end{array}$$

There are two (binary) T-transformations, right and left. For a function $f : X \times Y \longrightarrow T(Z)$, the left and right T-transform is defined as the composition





Continuation monad $\mathcal{C}$ (endofunctor)

- $\mathbf{t} = \{0, 1\}$, $\mathcal{P}(X) = X \Rightarrow \mathbf{t}$ - powerset of X;
- function $f : X \to Y$ induces an inverse image function between powersets

$$\mathcal{P}(f) = f^{-1} : \mathcal{P}(Y) \to \mathcal{P}(X)$$

$$h \mapsto h \circ f, \qquad \mathcal{P}(f) = \lambda h_{:\mathcal{P}(Y)}.\lambda x_{:X}.h(f\ x)$$

- taking again an inverse image function

$$\mathcal{C}(f) = \mathcal{P}(f^{-1}) : \mathcal{C}(X) = \mathcal{P}^2(X) \to \mathcal{P}^2(Y) = \mathcal{C}(Y)$$

$$Q \mapsto Q \circ f^{-1}, \qquad \mathcal{C}(f)(Q) = \lambda h_{:\mathcal{P}(Y)}.Q(\lambda x_{:X}.h(f\ x))$$

for $Q \in \mathcal{C}(X)$.

Now we can look at the notion of computation related to $\mathcal{C}$

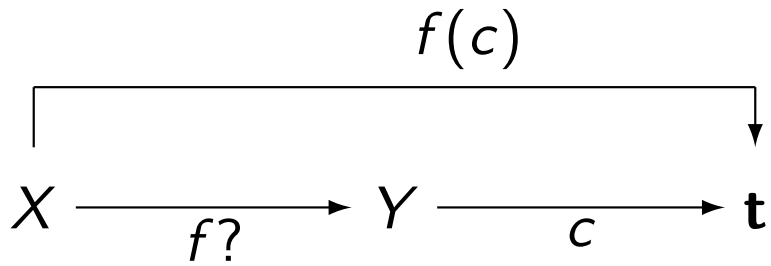
$$f : X \longrightarrow \mathcal{C}(Y) = \mathcal{P}(Y) \Rightarrow \mathbf{t}$$

By exponential adjunction (uncurrying) it corresponds to a function

$$f' : \mathcal{P}(Y) \times X \longrightarrow \mathbf{t}$$

and again by exponential adjunction (currying) it corresponds to a function

$$f'' : \mathcal{P}(Y) \longrightarrow \mathcal{P}(X)$$



Continuation Monad

Continuation monad $\mathcal{C}$ (natural transformations)

The **unit**

$$\eta_X : X \to \mathcal{C}(X)$$

is given by

$$\eta_X(x) = \lambda h_{:\mathcal{P}(X)}.h(x)$$

for $x \in X$.
The **multiplication**

$$\mu_X : \mathcal{C}^2(X) \longrightarrow \mathcal{C}(X)$$

is given by

$$\mu_X(\mathcal{F})(h) = \mathcal{F}(\lambda D_{:\mathcal{C}(X)}.D(h))$$

for $\mathcal{F} \in \mathcal{C}^2(X)$ and $h \in \mathcal{P}(X)$.

For the continuation monad, the **left strength** is

$$\mathbf{st}^l : \mathcal{C}(X) \times Y \longrightarrow \mathcal{C}(X \times Y)$$

$$\mathbf{st}^l(N, y) = \lambda c_{:\mathcal{P}(X \times Y)}.N(\lambda x_{:X}.c(x, y))$$

for $N \in \mathcal{C}(X)$ and $y \in Y$.
and the **right strength** is

$$\mathbf{st}^r : X \times \mathcal{C}(Y) \longrightarrow \mathcal{C}(X \times Y)$$

$$\mathbf{st}^r(x, M) = \lambda c_{:\mathcal{P}(X \times Y)}.M(\lambda y_{:Y}.c(x, y))$$

for $x \in X$ and $M \in \mathcal{C}(Y)$.

For the continuation monad, both **pile′up**-operations can be defined by lambda terms as follows. For $M \in \mathcal{C}(X)$ and $N \in \mathcal{C}(Y)$ we have

$$\mathbf{pile'up}^{l} : \mathcal{C}(X) \times \mathcal{C}(Y) \to \mathcal{C}(X \times Y)$$

$$\mathbf{pile'up}^{l}(M, N) = \lambda c_{:\mathcal{P}(X\times Y)}.M(\lambda x_{:X}.N(\lambda y_{:Y}\, c(x, y)))$$

$$\mathbf{pile'up}^{r} : \mathcal{C}(X) \times \mathcal{C}(Y) \to \mathcal{C}(X \times Y).$$

$$\mathbf{pile'up}^{r}(M, N) = \lambda c_{:\mathcal{P}(X\times Y)}.N(\lambda y_{:Y}.M(\lambda x_{:X}\, c(x, y))).$$

Thus in the case of the continuation monad ‘piling up’ computations one on top of the other is nothing but putting (interpretations of) quantifiers in order, either first before the second or the second before the first.

Transforms for monad $\mathcal{C}$ are called **CPS**-transforms.
For $f : X \times Y \to Z$ we have

$$\mathbf{CPS}^{l}(f) = \mathcal{C}(f) \circ \mathbf{pile'up}^{l}{}_{X,Y} : \mathcal{C}(X) \times \mathcal{C}(Y) \longrightarrow \mathcal{C}(Z)$$

given for $M \in \mathcal{C}(X)$ and $N \in \mathcal{C}(Y)$ by

$$\mathbf{CPS}^{l}(f)(M, N) = \lambda h_{:\mathcal{P}(Z)}.M(\lambda x_{:X}.N(\lambda y_{:Y}.h(f(x, y)))).$$

Right version is similar.

The most popular **CPS**-transforms are those for the evaluation (application) $ev : X \times (X \Rightarrow Y) \to Y$

$$\mathbf{CPS}^{\prime}(ev) = \mathcal{C}(ev) \circ \mathbf{pile}^{\prime}\mathbf{up}^{\prime}{}_{X,X\Rightarrow Y} : \mathcal{C}(X) \times \mathcal{C}(X \Rightarrow Y) \longrightarrow \mathcal{C}(Y)$$

given for $M \in \mathcal{C}(X)$ and $N \in \mathcal{C}(X \Rightarrow Y)$ by

$$\mathbf{CPS}^{\prime}(ev)(M, N) = \lambda h_{:\mathcal{P}(Y)}.M(\lambda x_{:X}.N(\lambda g_{:X\Rightarrow Y}.h(g\,x))).$$

Right version is similar.

Left evaluations

$$\mathbf{eps}^{l}{}_{X} = \lambda h_{:\mathcal{P}(X)}.\lambda x_{:X}.h(x) : \mathcal{P}(X)\times X \to \mathbf{t};$$

$$\mathbf{eps}^{l,X}_{Y} = \mathbf{eps}^{l}{}_{Y} = \lambda c_{:\mathcal{P}(X\times Y)}.\lambda y_{:Y}.\lambda x_{:X}.c(x,y) : \mathcal{P}(X\times Y)\times Y \to \mathcal{P}(X);$$

and right evaluations ...

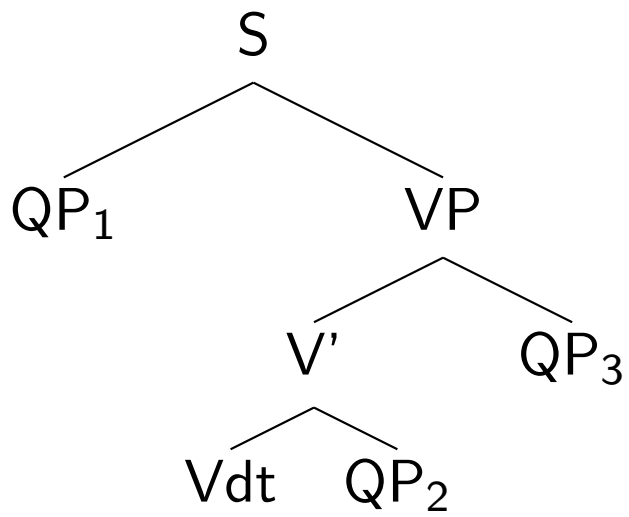
Left **mos**'es

$$\mathbf{mos}^{l}{}_{X} = \lambda Q_{:\mathcal{C}(X)}.\lambda c_{:\mathcal{P}(X)}.Q(c) : \mathcal{C}(X)\times\mathcal{P}(X) \to \mathbf{t};$$

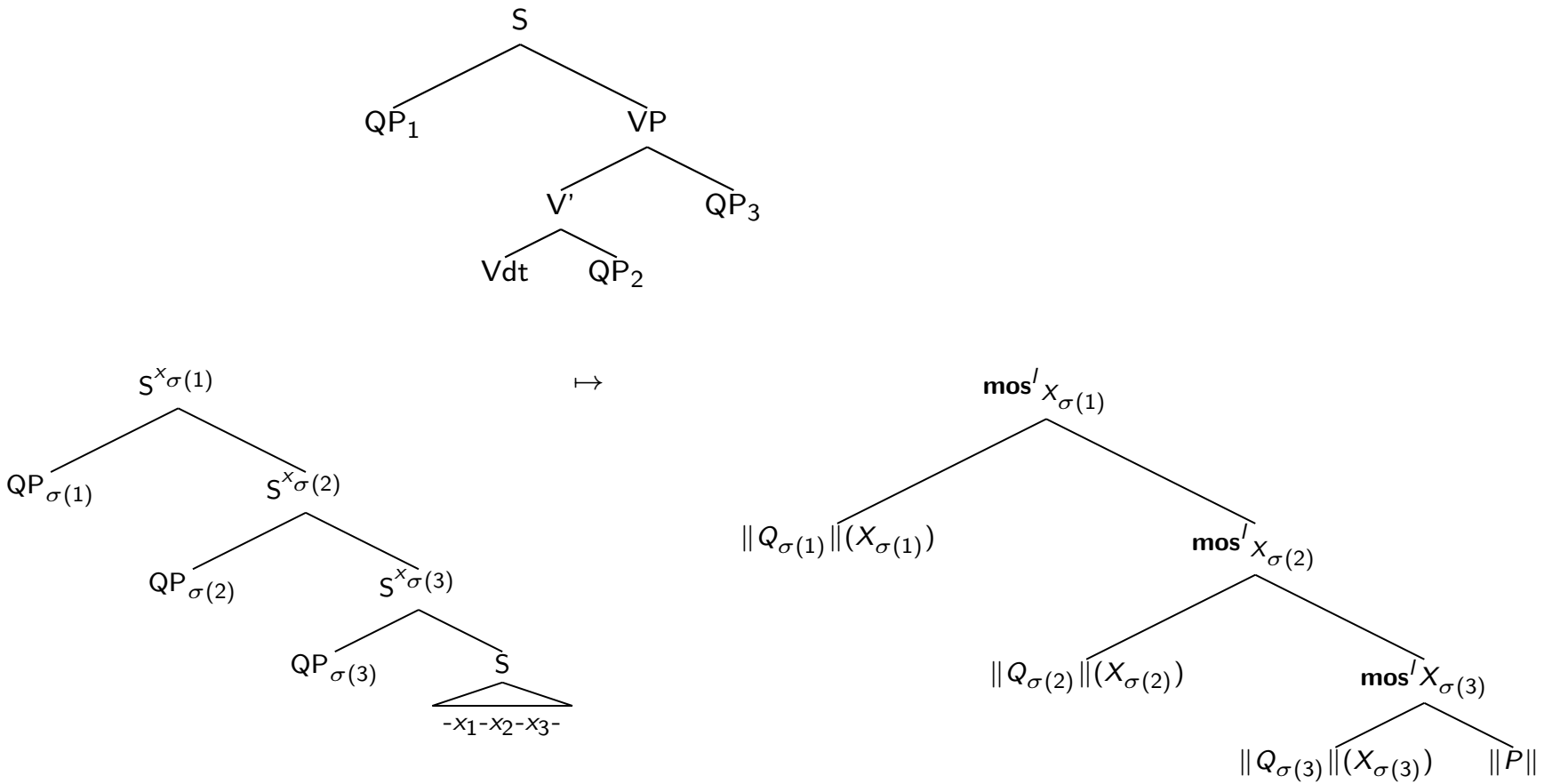
$$\mathbf{mos}^{l}{}_{Y} = \lambda Q_{:\mathcal{C}(Y)}.\lambda c_{:\mathcal{P}(X\times Y)}.\lambda x_{:X}.Q(\lambda y_{:Y}.c(x,y)) : \mathcal{C}(Y)\times\mathcal{P}(X\times Y) \to \mathcal{P}(X);$$

and right **mos**'es...

Sentence with three QPs, e.g.
Some teacher gave every student most books.



Scope assignment strategies
Strategy A

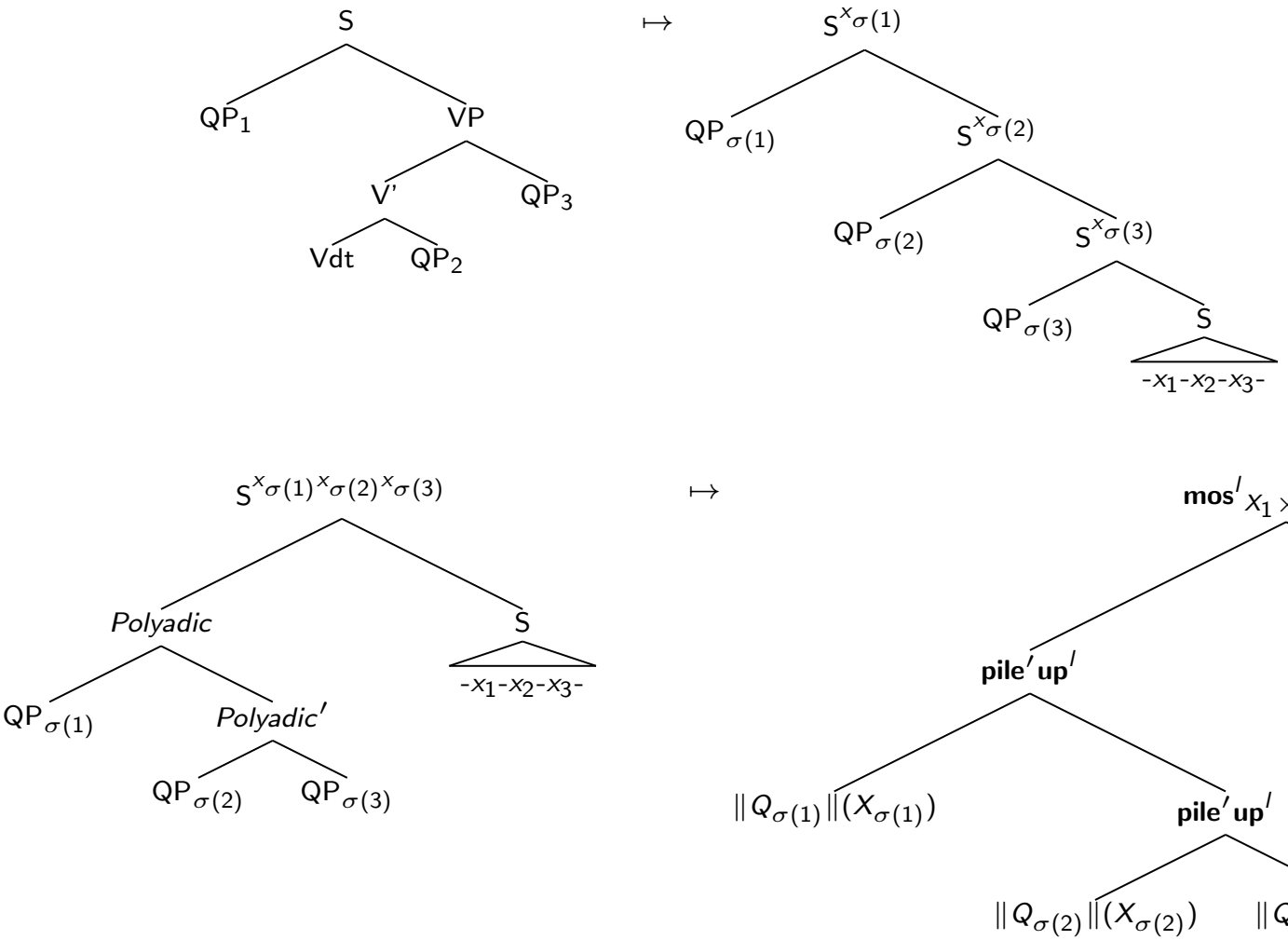


The Computation Tree gives rise to the following general map, with $\sigma \in S_3$

$$\begin{array}{l}
\mathcal{C}(X_1) \times \mathcal{P}(X_1 \times X_2 \times X_3) \times \mathcal{C}(X_2) \times \mathcal{C}(X_3) \xrightarrow[\bar{\pi}_{\sigma(i)}]{} \mathcal{C}(X_{\sigma(i)}) \\
\quad \downarrow \langle \bar{\pi}_{\sigma(1)}, \bar{\pi}_{\sigma(2)}, \bar{\pi}_{\sigma(3)}, \pi_2 \rangle \\
\mathcal{C}(X_{\sigma(1)}) \times \mathcal{C}(X_{\sigma(2)}) \times \mathcal{C}(X_{\sigma(3)}) \times \mathcal{P}(X_1 \times X_2 \times X_3) \\
\quad \downarrow 1 \times 1 \times \mathbf{mos}^I{}_{X_{\sigma(3)}} \\
\mathcal{C}(X_{\sigma(1)}) \times \mathcal{C}(X_{\sigma(2)}) \times \mathcal{P}(\ldots \times \widehat{X_{\sigma(3)}} \times \ldots) \\
\quad \downarrow 1 \times \mathbf{mos}^I{}_{X_{\sigma(2)}} \\
\mathcal{C}(X_{\sigma(1)}) \times \mathcal{P}(X_{\sigma(1)}) \\
\quad \downarrow \mathbf{mos}^I{}_{X_{\sigma(1)}} \\
2
\end{array}$$

$\mathbf{strat}_A^{3,\sigma}$:

Scope assignment strategies
Strategy B



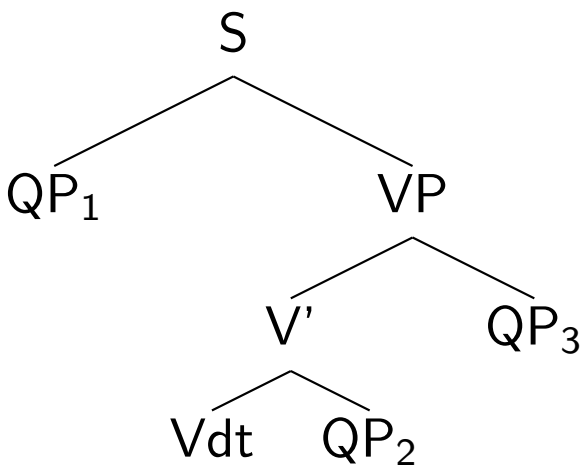
The Computation Tree gives rise to the following general map

$$\mathbf{strat}_B^{3,\sigma} :$$

$$\begin{array}{c}
\mathcal{C}(X_1) \times \mathcal{P}(X_1 \times X_2 \times X_3) \times \mathcal{C}(X_2) \times \mathcal{C}(X_3) \\
\downarrow \langle \bar{\pi}_{\sigma(1)}, \bar{\pi}_{\sigma(2)}, \bar{\pi}_{\sigma(3)}, \pi_2 \rangle \\
\mathcal{C}(X_{\sigma(1)}) \times \mathcal{C}(X_{\sigma(2)}) \times \mathcal{C}(X_{\sigma(3)}) \times \mathcal{P}(X_1 \times X_2 \times X_3) \\
\downarrow 1 \times \mathbf{pile'up}^I \times 1 \\
\mathcal{C}(X_{\sigma(1)}) \times \mathcal{C}(X_{\sigma(2)} \times X_{\sigma(3)}) \times \mathcal{P}(X_1 \times X_2 \times X_3) \\
\downarrow \mathbf{pile'up}^I \times 1 \\
\mathcal{C}(X_{\sigma(1)} \times X_{\sigma(2)} \times X_{\sigma(3)}) \times \mathcal{P}(X_1 \times X_2 \times X_3) \\
\downarrow \mathcal{C}(\pi_{\sigma^{-1}}) \times 1 \\
\mathcal{C}(X_1 \times X_2 \times X_3) \times \mathcal{P}(X_1 \times X_2 \times X_3) \\
\downarrow \mathbf{mos}^I{}_{X_1 \times X_2 \times X_3} \\
2
\end{array}$$

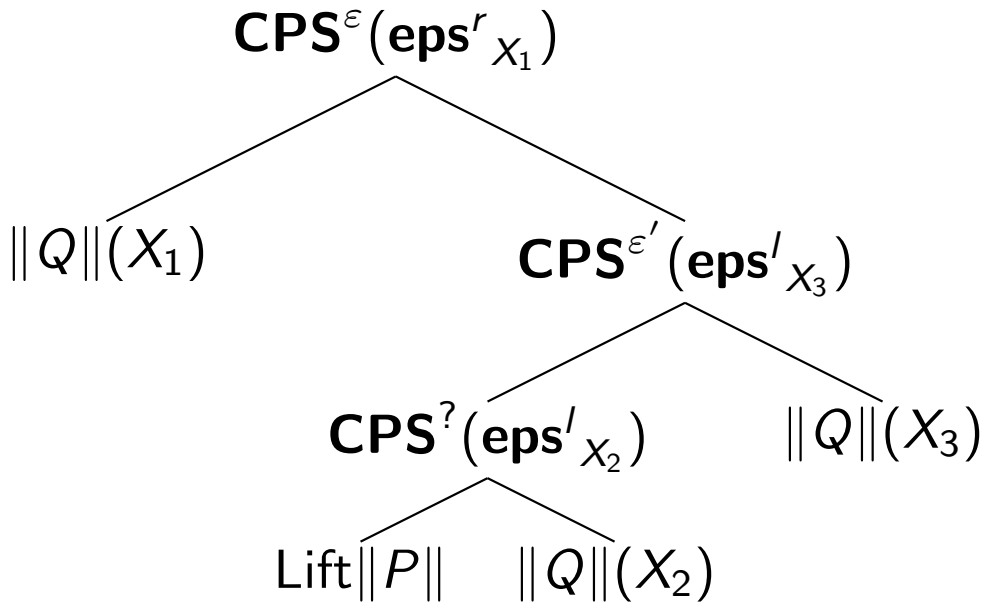
Scope assignment strategies
Strategy C

Surface Structure Tree



$\mapsto$

Computation Tree



The Computation Tree gives rise to the following general map

$$\mathbf{strat}_C^{3,\varepsilon',\varepsilon}:$$

$$\begin{array}{c}
\mathcal{C}(X_1)\times\mathcal{P}(X_1\times X_2\times X_3)\times\mathcal{C}(X_2)\times\mathcal{C}(X_3)\\
\downarrow\ 1\times\eta_{\mathcal{P}(X_1\times X_2\times X_3)\times 1\times 1}\\
\mathcal{C}(X_1)\times\mathcal{CP}(X_1\times X_2\times X_3)\times\mathcal{C}(X_2)\times\mathcal{C}(X_3)\\
\downarrow\ 1\times\mathbf{CPS}^{?}(\mathbf{eps}^{l}{}_{X_2})\times 1\\
\mathcal{C}(X_1)\times\mathcal{CP}(X_1\times X_3)\times\mathcal{C}(X_3)\\
\downarrow\ 1\times\mathbf{CPS}^{\varepsilon'}(\mathbf{eps}^{l}{}_{X_3})\\
\mathcal{C}(X_1)\times\mathcal{CP}(X_1)\\
\downarrow\ \mathbf{CPS}^{\varepsilon}(\mathbf{eps}^{r}{}_{X_1})\\
\mathcal{C}(2)\xrightarrow[ev_{id_2}]{}2
\end{array}$$

Scope assignment strategies
Strategy C

- Strategy C provides a uniform non-movement (in situ) analysis of quantifiers.
- However, it cannot be straightforwardly extended to account for sentences involving 3 QPs - as proved in our ArXiv paper, it only provides **4 out of 6 readings** accounted for by the two other strategies.
- We can augment Strategy C to obtain the missing readings. To this end, we use **CPS**- (and **pile′up**-) operations to define two new operations: **shuf**^{l} and **shuf**^{r}.

To get the first missing reading

$QP_2 > QP_1 > QP_3$,

we define a new operation

$$\mathbf{shuf}^l : \mathcal{CP}(X_1 \times X_3) \times \mathcal{C}(X_3) \to \mathcal{PC}(X_1)$$

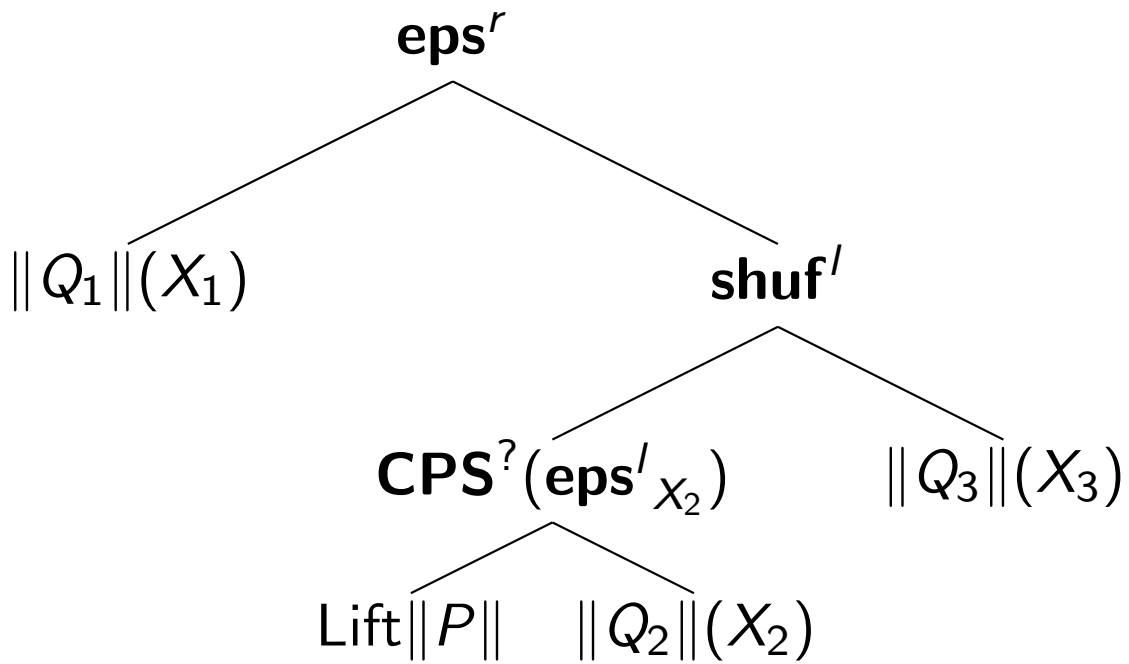
$$\mathbf{shuf}^l(S_2, S_3) =$$

$$= \lambda S_{1:\mathcal{C}(X_1)}.\mathbf{CPS}^l(\mathbf{eps}^l{}_{X_3})(\mathbf{CPS}^l(\mathbf{eps}^l{}_{X_1})(S_2, S_1), S_3)(id_t) =$$

$$= \lambda S_{1:\mathcal{C}(X_1)}.\mathbf{CPS}^l(\mathbf{eps}^l{}_{X_1 \times X_3})(S_2, \mathbf{pile'up}^l(S_1, S_3))(id_t)$$

for $S_2 \in \mathcal{CP}(X_1 \times X_3)$ and $S_3 \in \mathcal{C}(X_3)$.

Computation Tree



To get the second of the two missing readings

$QP_3 > QP_1 > QP_2$,

we define a new operation

$$\mathbf{shuf}^r : \mathcal{CP}(X_1 \times X_3) \times \mathcal{C}(X_3) \to \mathcal{PC}(X_1)$$

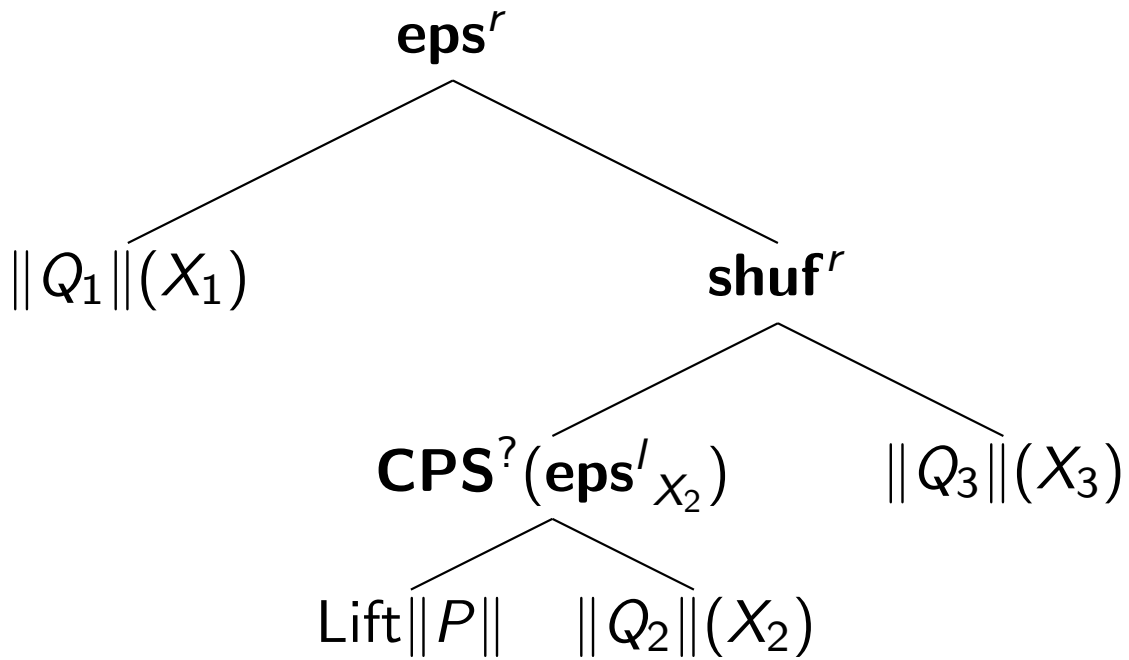
$$\mathbf{shuf}^r(S_2, S_3) =$$

$$= \lambda S_{1:\mathcal{C}(X_1)}.\mathbf{CPS}^l(\mathbf{eps}^r{}_{X_3})(S_3, \mathbf{CPS}^l(\mathbf{eps}^r{}_{X_1})(S_1, S_2))(id_t) =$$

$$= \lambda S_{1:\mathcal{C}(X_1)}.\mathbf{CPS}^l(\mathbf{eps}^r{}_{X_1 \times X_3})(\mathbf{pile'up}^l(S_3, S_1), S_2)(id_t)$$

for $S_2 \in \mathcal{CP}(X_1 \times X_3)$ and $S_3 \in \mathcal{C}(X_3)$.

Computation Tree



The end

Thank You for Your Attention!