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
Some teacher gave every student most books (6-way ambiguous)
\[
((2 - 7) - 8) + ((12 + 5) : 7)
\]
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

<table>
<thead>
<tr>
<th>Strategy</th>
<th>A (Movement Strategy (May 77))</th>
<th>B (Polyadic Approach (May 85))</th>
<th>C (Continuation-based Approach (Barker 2002))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rewrite rules</td>
<td>QR</td>
<td>QR, Rotation</td>
<td>No rewrite rules (in situ)</td>
</tr>
<tr>
<td>Relabelling inner nodes (operations)</td>
<td>mos</td>
<td>mos</td>
<td>CPS</td>
</tr>
<tr>
<td>Relabelling leaves (semantic values)</td>
<td>Predicate $\mapsto$ C-comp.</td>
<td>Predicate $\mapsto$ C-comp.</td>
<td>Predicate $\mapsto$ continuized relation</td>
</tr>
</tbody>
</table>

Justyna Grudzińska and Marek Zawadowski
<table>
<thead>
<tr>
<th>Strategy</th>
<th>A Movement Strategy (May 77)</th>
<th>B Polyadic Approach (May 85)</th>
<th>C Cont-based Approach (Barker 02)</th>
<th>D Strat C &amp; new shuffle operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 readings</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>in situ</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>simple operations</td>
<td>✓</td>
<td>✓?</td>
<td>✓??</td>
<td>???</td>
</tr>
</tbody>
</table>
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.
A **monad** on **Set** is a triple (+ conditions ...):

$T$-computations (endofunctor)

\[
\begin{align*}
\text{Set} & \xrightarrow{T} \text{Set} \\
X & \xrightarrow{f} Y & T(X) & \xrightarrow{T(f)} T(Y)
\end{align*}
\]

**unit (return)** (natural transformation)

\[
\eta : 1_{\text{Set}} \rightarrow T \\
\eta_X : X \rightarrow T(X)
\]

lifts elements of $X$ as $T$-computations.

**multiplication** (natural transformation)

\[
\mu : T^2 \rightarrow T \\
\mu_X : T^2(X) = T(T(X)) \rightarrow T(X)
\]

flattens $T$-computations on $T$-computations to $T$-computations.
Let \((T, \eta, \mu)\) be a monad on \(\textbf{Set}\).

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

\[
\text{st}^l_{X, Y} : T(X) \times Y \rightarrow T(X \times Y)
\]

\[
(\text{st}^r_{X, Y} : X \times T(Y) \rightarrow T(X \times Y))
\]

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 $\text{pile'}\text{up}^l_{X,Y}$ ($\text{pile'}\text{up}^r_{X,Y}$) is defined as the composition
There are two (binary) $T$-transformations, right and left. For a function $f : X \times Y \rightarrow T(Z)$, the left and right $T$-transform is defined as the composition

$$T(X) \times T(Y) \xrightarrow{\text{pile'up}^l} T(X \times Y) \xrightarrow{T(f)} T^2(Z) \xrightarrow{\mu_Z} T(Z)$$

and

$$T(X) \times T(Y) \xrightarrow{\text{pile'up}^r} T(X \times Y) \xrightarrow{T(f)} T^2(Z) \xrightarrow{\mu_Z} T(Z)$$
Continuation monad $\mathcal{C}$ (endofunctor)

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

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

$$h \mapsto h \circ f,$$

$$\mathcal{P}(f) = \lambda h : \mathcal{P}(Y). \lambda x : X. h(f \ x)$$

- taking again an inverse image function

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

$$Q \mapsto Q \circ f^{-1},$$

$$\mathcal{C}(f)(Q) = \lambda h : \mathcal{P}(Y). Q(\lambda x : X. h(f \ x))$$

for $Q \in \mathcal{C}(X)$. 
Now we can look at the notion of computation related to $C$

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

By exponential adjunction (uncurrying) it corresponds to a function

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

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

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

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

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

\[ \text{st}^l : \mathcal{C}(X) \times Y \longrightarrow \mathcal{C}(X \times Y) \]

\[ \text{st}^l(N, y) = \lambda c : \mathcal{P}(X \times Y). N(\lambda x : X . c(x, y)) \]

for \( N \in \mathcal{C}(X) \) and \( y \in Y \).

and the **right strength** is

\[ \text{st}^r : X \times \mathcal{C}(Y) \longrightarrow \mathcal{C}(X \times Y) \]

\[ \text{st}^r(x, M) = \lambda c : \mathcal{P}(X \times Y). M(\lambda y : Y . c(x, y)) \]

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

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

$$\text{pile’up}^l(M, N) = \lambda c: \mathcal{P}(X \times Y). M(\lambda x:X. N(\lambda y:Y. c(x, y)))$$

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

$$\text{pile’up}^r(M, N) = \lambda c: \mathcal{P}(X \times Y). N(\lambda y:Y. M(\lambda x:X. c(x, y))).$$

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.
Continuation monad

CPS-transforms

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

\[
\text{CPS}^l(f) = C(f) \circ \text{pile}'\text{up}^l_{X,Y} : C(X) \times C(Y) \rightarrow C(Z)
\]

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

\[
\text{CPS}^l(f)(M, N) = \lambda h: P(Z). M(\lambda x: X. N(\lambda y: Y. h(f(x, y))))
\]

Right version is similar.
The most popular CPS-transforms are those for the evaluation (application) operation \( ev : X \times (X \Rightarrow Y) \rightarrow Y \)

\[
\text{CPS}^l(ev) = C(ev) \circ \text{pile}'\text{up}^l_{X,X\Rightarrow Y} : C(X) \times C(X \Rightarrow Y) \rightarrow C(Y)
\]

given for \( M \in C(X) \) and \( N \in C(X \Rightarrow Y) \) by

\[
\text{CPS}^l(ev)(M, N) = \lambda h : P(Y). M(\lambda x : X. N(\lambda g : X \Rightarrow Y. h(g \ x))).
\]

Right version is similar.
Left evaluations

$$\text{eps}^l_X = \lambda h : \mathcal{P}(X). \lambda x : X. h(x) : \mathcal{P}(X) \times X \to t;$$

$$\text{eps}^l_X = \text{eps}^l_Y = \lambda c : \mathcal{P}(X \times Y). \lambda y : Y. \lambda x : X. c(x, y) : \mathcal{P}(X \times Y) \times Y \to \mathcal{P}(X);$$

and right evaluations ...

Left mos’es

$$\text{mos}^l_X = \lambda Q : \mathcal{C}(X). \lambda c : \mathcal{P}(X). Q(c) : \mathcal{C}(X) \times \mathcal{P}(X) \to t;$$

$$\text{mos}^l_Y = \lambda Q : \mathcal{C}(Y). \lambda c : \mathcal{P}(X \times Y). \lambda x : X. Q(\lambda y : Y. c(x, y)) : \mathcal{C}(Y) \times \mathcal{P}(X \times Y) \to \mathcal{P}(X);$$

and right mos’es...
**Sentence with three QPs**, e.g.

*Some teacher gave every student most books.*

```
S
  /\  
QP₁   VP
     /\  
    V'  QP₃
       /\  
      Vdt QP₂
```
Scope assignment strategies

Strategy A

Justyna Grudzińska and Marek Zawadowski
The Computation Tree gives rise to the following general map, with $\sigma \in S_3$

$$\text{strat}_A^{3,\sigma} : C(X_1) \times \mathcal{P}(X_1 \times X_2 \times X_3) \times C(X_2) \times C(X_3) \xrightarrow{\bar{\pi}_\sigma(i)} C(X_\sigma(i))$$

$$C(X_\sigma(1)) \times C(X_\sigma(2)) \times C(X_\sigma(3)) \times \mathcal{P}(X_1 \times X_2 \times X_3)$$

$$1 \times 1 \times \text{mos}^I X_\sigma(3)$$

$$C(X_\sigma(1)) \times C(X_\sigma(2)) \times \mathcal{P}(\ldots \times X_\sigma(3) \times \ldots)$$

$$1 \times \text{mos}^I X_\sigma(2)$$

$$C(X_\sigma(1)) \times \mathcal{P}(X_\sigma(1))$$

$$\text{mos}^I X_\sigma(1)$$

2
Scope assignment strategies

Strategy B

\[ S \xrightarrow{QP_1} VP \xrightarrow{V'} QP_3 \xrightarrow{Vdt} QP_2 \]

\[ S^\sigma(1) \rightarrow S^\sigma(2) \rightarrow S^\sigma(3) \rightarrow -x_1-x_2-x_3- \]

\[ S^\sigma(1) \times S^\sigma(2) \times S^\sigma(3) \rightarrow mos/ x_1 \times x_2 \times x_3 \rightarrow \parallel P \parallel \]

\[ \parallel Q_\sigma(1) \parallel (X_\sigma(1)) \rightarrow pile/ up\]

\[ \parallel Q_\sigma(2) \parallel (X_\sigma(2)) \quad \parallel Q_\sigma(3) \parallel (X_\sigma(3)) \rightarrow pile/ up\]
The Computation Tree gives rise to the following general map

\[
\text{strat}_{B}^{3,\sigma} : \\
C(X_1) \times \mathcal{P}(X_1 \times X_2 \times X_3) \times C(X_2) \times C(X_3) \\
\langle \bar{\pi}_{\sigma}(1), \bar{\pi}_{\sigma}(2), \bar{\pi}_{\sigma}(3), \pi_2 \rangle \\
C(X_{\sigma(1)}) \times C(X_{\sigma(2)}) \times C(X_{\sigma(3)}) \times \mathcal{P}(X_1 \times X_2 \times X_3) \\
1 \times \text{pile’up}^I \times 1 \\
C(X_{\sigma(1)}) \times C(X_{\sigma(2)} \times X_{\sigma(3)}) \times \mathcal{P}(X_1 \times X_2 \times X_3) \\
\text{pile’up}^I \times 1 \\
C(X_{\sigma(1)} \times X_{\sigma(2)} \times X_{\sigma(3)}) \times \mathcal{P}(X_1 \times X_2 \times X_3) \\
C(\pi_{\sigma-1}) \times 1 \\
C(X_1 \times X_2 \times X_3) \times \mathcal{P}(X_1 \times X_2 \times X_3) \\
\text{mos}^I_{X_1 \times X_2 \times X_3} \\
2
\]
Scope assignment strategies
Strategy C

Surface Structure Tree

Scope ambiguities, continuations and strengths

25 / 32
The Computation Tree gives rise to the following general map

\[
\text{strat}^3_{C}: \text{C}(X_1) \times \mathcal{P}(X_1 \times X_2 \times X_3) \times C(X_2) \times C(X_3) \\
\downarrow \\
C(X_1) \times CP(X_1 \times X_2 \times X_3) \times C(X_2) \times C(X_3) \\
\downarrow \\
1 \times CPS^7(\epsilon_{X_2}) \times 1 \\
\downarrow \\
C(X_1) \times CP(X_1 \times X_3) \times C(X_3) \\
\downarrow \\
1 \times CPS^{\epsilon'}(\epsilon_{X_3}) \\
\downarrow \\
C(X_1) \times CP(X_1) \\
\downarrow \\
CPS^{\epsilon}(\epsilon_{X_1}) \\
\downarrow \\
C(2) \xrightarrow{ev_{id_2}} 2
\]
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

\[ \text{shuf}^l : \mathcal{CP}(X_1 \times X_3) \times \mathcal{C}(X_3) \to \mathcal{PC}(X_1) \]

\[ \text{shuf}^l(S_2, S_3) = \]

\[ = \lambda S_1: \mathcal{C}(X_1). \text{CPS}^l(\text{eps}^l X_3)(\text{CPS}^l(\text{eps}^l X_1)(S_2, S_1), S_3)(id_t) = \]

\[ = \lambda S_1: \mathcal{C}(X_1). \text{CPS}^l(\text{eps}^l X_1 \times X_3)(S_2, \text{pile'up}^l(S_1, S_3))(id_t) \]

for \( S_2 \in \mathcal{CP}(X_1 \times X_3) \) and \( S_3 \in \mathcal{C}(X_3) \).
Scope assignment strategies
Strategy D

Computation Tree

\[ \text{eps}^r \]

\[ \| Q_1 \| (X_1) \]

\[ \text{shuf}^l \]

\[ \text{CPS}^? (\text{eps}^l, X_2) \quad \| Q_3 \| (X_3) \]

\[ \text{Lift} \| P \| \quad \| Q_2 \| (X_2) \]
To get the second of the two missing readings

\[ QP_3 > QP_1 > QP_2, \]

we define a new operation

\[ \text{shuf}^r : CP(X_1 \times X_3) \times C(X_3) \rightarrow PC(X_1) \]

\[ \text{shuf}^r(S_2, S_3) = \]

\[ = \lambda S_1 : C(X_1) \cdot CPS^l(\text{eps}^r_{X_3})(S_3, CPS^l(\text{eps}^r_{X_1})(S_1, S_2))(id_t) = \]

\[ = \lambda S_1 : C(X_1) \cdot CPS^l(\text{eps}^r_{X_1 \times X_3})(\text{pile'up}^l(S_3, S_1), S_2)(id_t) \]

for \( S_2 \in CP(X_1 \times X_3) \) and \( S_3 \in C(X_3) \).
Scope assignment strategies
Strategy D

Computation Tree

\[ \text{eps}^r \]

\[ \parallel Q_1 \parallel (X_1) \quad \text{shuf}^r \]

\[ \text{CPS}^? (\text{eps}' X_2) \quad \parallel Q_3 \parallel (X_3) \]

\[ \text{Lift} \parallel P \parallel \quad \parallel Q_2 \parallel (X_2) \]
Thank You for Your Attention!