Gradually Transforming Syntax to Semantics

Oleg Kiselyov    Leo Tingchen Hsu

Tohoku University, Japan

New Landscapes in Theoretical Computational Linguistics
OSU, October 16, 2016
Summary: Semantics-by-Transformations

QR

- Transformation to semantics (covert movement), ...
- Negative predictions

Now

- restrained, rigorous, type preserving
- mostly deterministic

- quantifier ambiguity, scoping islands and binding, crossover, topicalization, *inverse linking*

- The product of long evolution (of my views)
- Precisely specified and can be carried out mechanically: Semantic calculator
- Compositionality: not just meanings but transformations
Broader Context

meaning from some (abstract) form

Proof search

- Logically insightful
- Hard to get negative predictions
- Hard to characterize the space of derivations
Broader Context

meaning from some (abstract) form

Evaluation
Chung-chieh Shan: Linguistic side-effects
Barker et al.: Monads in natural languages

DRT

- Algorithmic; possible claim to real life
- Mostly deterministic (as real programs)
- Inherently partial
- (Usually) precisely specified and *mechanized*
- Too rigid
- Too easy to get bogged down in technical details
Broader Context

meaning from some (abstract) form

History

- 2007-2008 multi-prompt delimited control
- 2009 ACG with multi-prompt delimited control
- 2011-2012 ACG with monads, then applicatives
- 2015 ACG with staging and applicatives
- 2015 LENLS talk (still applicatives)
- 2015 LENLS paper (starting to abstract the details away)
Problems

(1) Every girl\_i’s father loves her\_i mother.
(1a) *Every girl\_i’s father loves its\_i mother.
(1b) *Her\_i father loves every girl\_i’s mother.
(1c) A girl\_i met every boy who liked her\_i.
(2a) That John\_i left upset his\_i teacher.
(2b) *That every boy\_i left upset his\_i teacher.
(3a) Alice’s present for him\_i, every boy\_i saw.
(3b) *Every boy\_i, his\_i mother likes.
(4) Two politicians spy on someone from every city.
Problems

(1) Every girl’s father loves her mother.
(1a) *Every girl’s father loves its mother.
(1b) *Her father loves every girl’s mother.
(1c) A girl met every boy who liked her.
(2a) That John left upset his teacher.
(2b) *That every boy left upset his teacher.
(3a) Alice’s present for him, every boy saw.
(3b) *Every boy, his mother likes.
(4) Two politicians spy on someone from every city.
(Concrete) Terms

"John" · "loves" · "Mary"

Algebraic structure

Carrier : string
"John" : string
"loves" : string
"mary" : string
· : string → string → string
(Concrete) Terms

"John"."loves"."Mary"

Algebraic structure

Carrier : string
"John" : string
"loves" : string
"mary" : string
· : string → string → string

Too concrete. Too little typed
Abstract (Tecto) Terms

cl john (love mary)

Multisorted Algebraic structure

Carriers: $S, NP, N, VP, PP$

cl : $NP \rightarrow VP \rightarrow S$

john : $NP$

mary : $NP$

love : $NP \rightarrow VP$
Logic Terms

love john mary

First-Order Multisorted Algebraic structure

Types : $e$, $t$
mary : $e$
john : $e$
love : $e \to e \to t$
conj, disj, ... : $t \to t \to t$
$x, y, z, ...$ : $e$
$\forall x$ : $t \to t$
$\exists y$ : $t \to t$

Not $\lambda$-calculus
More than one Abstract Language

\[
\begin{align*}
\text{every}_x & : \ N \rightarrow NP \\
\text{a}_x & : \ N \rightarrow NP \\
\text{var}_x, \text{var}_y, \ldots & : \ NP \\
\text{U}_x, \text{U}_y, \ldots & : \ N \rightarrow S \rightarrow S \\
\text{E}_x, \text{E}_y, \ldots & : \ N \rightarrow S \rightarrow S \\
\text{he}, \text{she}, \text{it} & : \ NP
\end{align*}
\]
Transformation Approach Overview

Abstract

\( cl \ john \ (love \ mary) \)

Syntax

"John".("loves"."Mary")

Semantics

love mary john

(Context-sensitive) re-writing
Quantifier ambiguity

\[
cl (a_y \text{ woman}) (\text{love} (\text{every}_x \text{ man}))
\]

\[
\downarrow \mathcal{L}_E
\]

\[
(E_y \text{ woman})
\]

\[
(cl \text{ var}_y (\text{love} (\text{every} \text{ man})))
\]

\[
\downarrow \mathcal{L}_U
\]

\[
(E_y \text{ woman})(U_x \text{ man})
\]

\[
(cl \text{ var}_y (\text{love} \text{ var}_x))
\]

\[
\Downarrow
\]

Semantics

\[
\exists y. \text{ woman } y \land \forall x. \text{ man } x \Rightarrow \text{ love } x y
\]

\[
\mathcal{L}_U[cl \ C[\text{every}_x \ d_r] \ d] \mapsto (U_x \ d_r) (cl \ C[\text{var}_x] \ d)
\]

\[
\mathcal{L}_U[cl \ d \ C[\text{every}_x \ d_r]] \mapsto (U_x \ d_r) (cl \ d \ C[\text{var}_x])
\]
Quantifier ambiguity

\[ cl \ (a_y \ woman) \ (love \ (every_x \ man)) \]

\[ \downarrow \]

\[ \downarrow \mathcal{L}_E \]

\[ (E_y \ woman) \ (cl \ var_y \ (love \ (every \ man))) \]

\[ \downarrow \mathcal{L}_U \]

\[ (E_y \ woman) (U_x \ man) \ (cl \ var_y \ (love \ var_x)) \]

**Semantics**

\[ \exists y. \ woman \ y \land \forall x. \ man \ x \Rightarrow \ love \ x \ y \]

\[ \mathcal{L}_U[cl \ C[\ every_x \ d_r] \ d] \mapsto (U_x \ d_r) \ (cl \ C[\ var_x] \ d) \]

\[ \mathcal{L}_U[cl \ d \ C[\ every_x \ d_r]] \mapsto (U_x \ d_r) \ (cl \ d \ C[\ var_x]) \]

QR, in a precisely specified, and typed-assured way
Implementing re-writing

\[ L_U[cl\ C[every_x\ d_r]\ d] \leftrightarrow (U_x\ d_r)\ (cl\ C[var_x]\ d) \]
\[ L_U[cl\ d\ C[every_x\ d_r]] \leftrightarrow (U_x\ d_r)\ (cl\ d\ C[var_x]) \]

- Shan: delimited continuations
- Barker, Charlow: monads
- ACG: linear lambda-calculus
- AACG: applicative
Implementing re-writing

\[ \mathcal{L}_U[\text{cl } C[\text{every}_x \ d_r] \ d] \mapsto (U_x \ d_r) \ (\text{cl } C[\text{var}_x] \ d) \]
\[ \mathcal{L}_U[\text{cl } d \ C[\text{every}_x \ d_r]] \mapsto (U_x \ d_r) \ (\text{cl } d \ C[\text{var}_x]) \]

- Shan: delimited continuations
- Barker, Charlow: monads
- ACG: linear lambda-calculus
- AACG: applicative
- Us: Whatever
Problems

(1) Every girl$_i$’s father loves her$_i$ mother.
(1a) *Every girl$_i$’s father loves its$_i$ mother.
(1b) *Her$_i$ father loves every girl$_i$’s mother.
(1c) A girl$_i$ met every boy who liked$_i$ her.
(2a) That John$_i$ left upset his$_i$ teacher.
(2b) *That every boy$_i$ left upset his$_i$ teacher.
(3a) Alice’s present for him$_i$, every boy$_i$ saw.
(3b) *Every boy$_i$, his$_i$ mother likes.
Demos
Conclusions

Transformational Formalism

- Abstract $\rightarrow$ Syntax & Semantics, compositionally
- Transformations are composed from smaller ones
- *Transformation are context-sensitive and non-trivial*

Mechanical implementation: semantics calculator

QR, movement, Cooper storage,...

in a precisely specified, and a typed-assured way

http://okmij.org/ftp/gengo/transformational-semantics/
Reflections

Ad hoc and illogical?

But proof system is also sort of re-writing...

Minimalism?

Movements...

What is wrong Lambda-Calculus?

▶ ACG (Lambda-Grammars) are based on it
▶ But it is not a context-sensitive re-writing system by nature
Reflections

Ad hoc and illogical?
But proof system is also sort of re-writing...
Reflections

Ad hoc and illogical?
But proof system is also sort of re-writing...

Minimalism?
Movements...
Reflections

Ad hoc and illogical?
But proof system is also sort of re-writing...

Minimalism?
Movements...

What is wrong Lambda-Calculus?

- ACG (Lambda-Grammars) are based on it
- But it is not a context-sensitive re-writing system by nature