Lambda: the ultimate syntax-semantics interface

Chung-chieh Shan
(with Jacques Carette and Oleg Kiselyov)

14 August 2013
Human concepts
Line up representations and what they represent
Multiple compositional interpretations
Multiple compositional interpretations

object {sphere ...
  interior {media {emission <.4,.3,.2>}}}
object {cylinder ...}
Multiple compositional interpretations

object {sphere ...
interior {media {emission <.4,.3,.2>}}}

object {cylinder ...}
Multiple compositional interpretations

- Light bulb
  - Object {sphere ...}
    - Interior {media {emission <.4,.3,.2>}}}
  - Object {cylinder ...} 

- Light saber
  - Object {cone ...}
    - Interior {media {emission <.4,.3,.2>}}}
  - Object {sphere ...}
    - Interior {media {emission <.4,.3,.2>}}}
  - Object {cylinder ...}
Interpretations are everywhere

contract  recipe  sentence  formula  …

value  cooking  spelling  evaluate  …
schedule  nutrition  meaning  simplify
Interpretations are everywhere

We taught linguists and programmers at conferences and schools

The will  Collaborative tasks for mutual acculturation

The way  Modular, higher-order programming languages
Montague grammar fragments

“I reject the contention that an important theoretical difference exists between formal and natural languages . . .
I regard the construction of a theory of truth—or rather, of the more general notion of truth under an interpretation—as the basic goal of serious syntax and semantics.”
Montague grammar fragments

“I reject the contention that an important theoretical difference exists between formal and natural languages . . .

I regard the construction of a theory of truth—or rather, of the more general notion of truth under an interpretation—as the basic goal of serious syntax and semantics.”


\[
\begin{align*}
S &::= \text{NP VP} & [S] &= [VP]([NP]) \\
\text{VP} &::= \text{TV NP} & [VP] &= [TV]([NP]) \\
\text{NP} &::= \text{John} & [NP] &= j \\
\text{NP} &::= \text{Mary} & [NP] &= m \\
\text{TV} &::= \text{likes} & [TV] &= (o \mapsto (s \mapsto (s, o) \in \{(j, m), (m, j), \ldots\}))
\end{align*}
\]

Calculemus.
Understanding type classes

<table>
<thead>
<tr>
<th></th>
<th>class</th>
<th>instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>math</td>
<td>signature</td>
<td>model</td>
</tr>
<tr>
<td>linguistics</td>
<td>language</td>
<td>interpretation</td>
</tr>
<tr>
<td>computer science</td>
<td>interface</td>
<td>implementation</td>
</tr>
</tbody>
</table>
Domain-specific languages for collaboration

Multiple communities. Multiple interpretations.

The will  Calculemus: automation, application, understanding
The way  Expressions abstract over interpretations, rather than the other way around

Potential exchanges: quotation, types, pragmatics

Other domains:
- Probability distributions
- Braid drawings
- Context-free grammars
- Optimized code generators
- Geometric objects

Let’s talk!