Lambda: the ultimate syntax-semantics interface

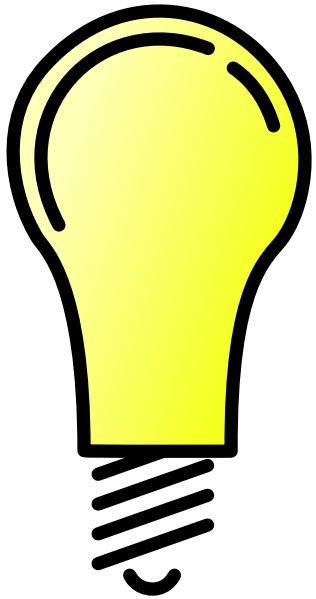
Chung-chieh Shan
(with Jacques Carette and Oleg Kiselyov)

14 August 2013

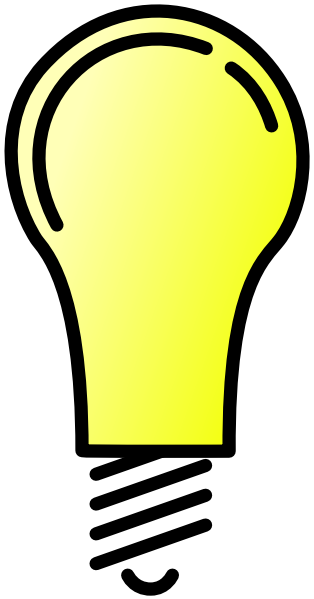
Human concepts

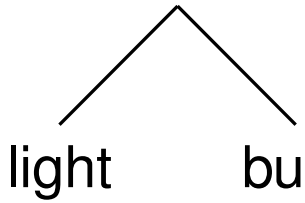
Line up representations and what they represent

Multiple compositional interpretations



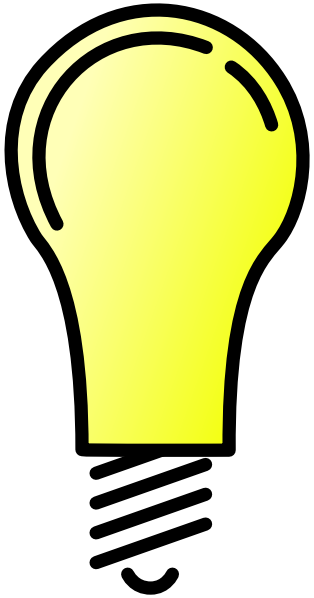
Multiple compositional interpretations

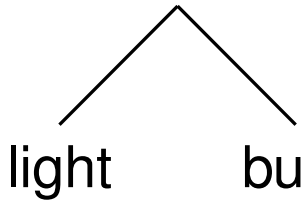




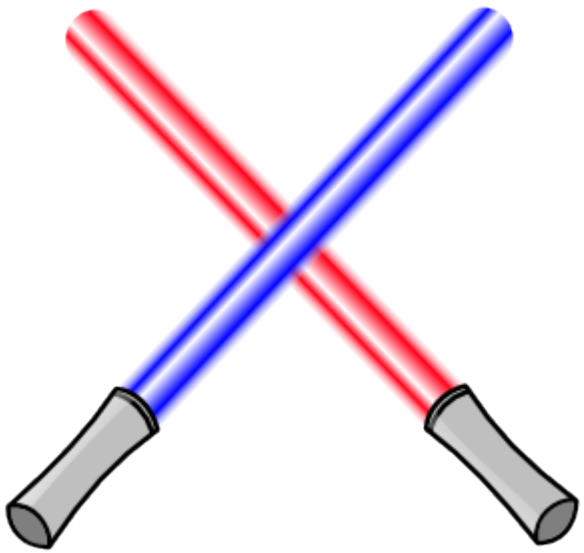

```
object {sphere ...
  interior {media {emission <.4,.3,.2>}}}
object {cylinder ...}
```

Multiple compositional interpretations

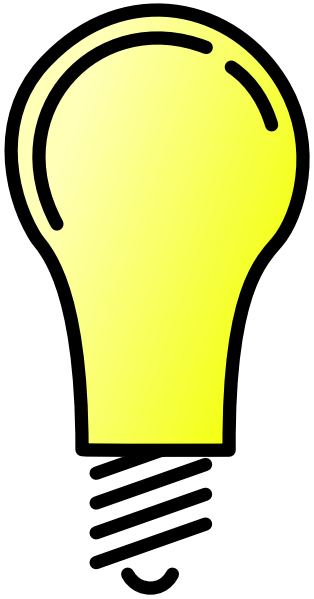


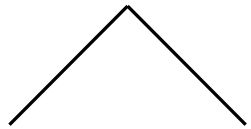



```
object {sphere ...
  interior {media {emission <.4,.3,.2>}}}
object {cylinder ...}
```

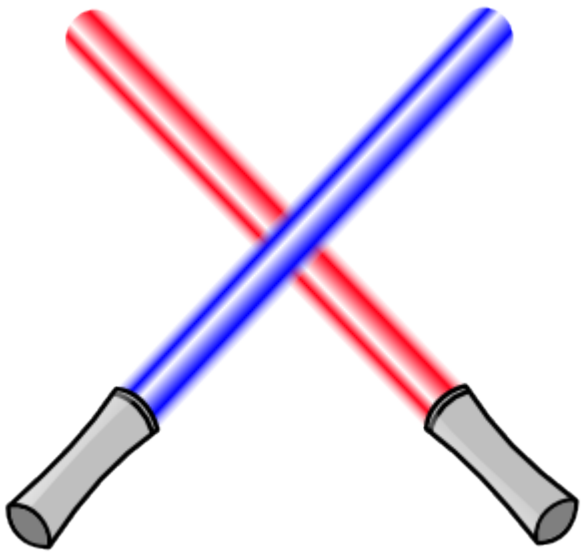


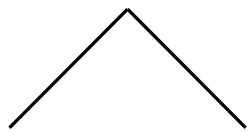
Multiple compositional interpretations





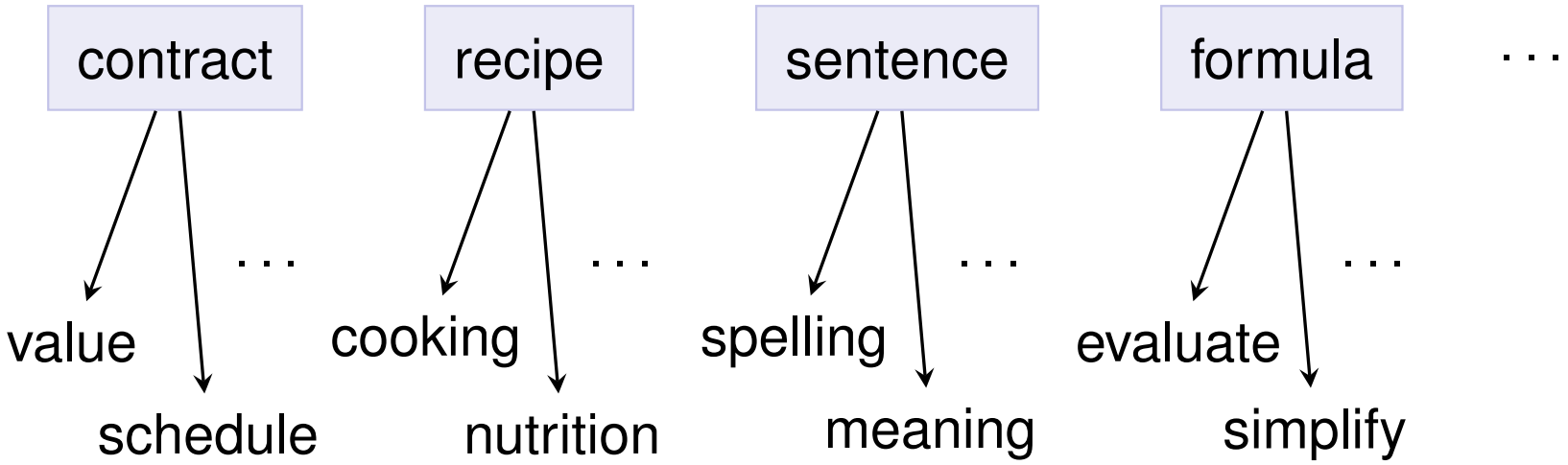

```
object {sphere ...
  interior {media {emission <.4,.3,.2>}}}
object {cylinder ...}
```



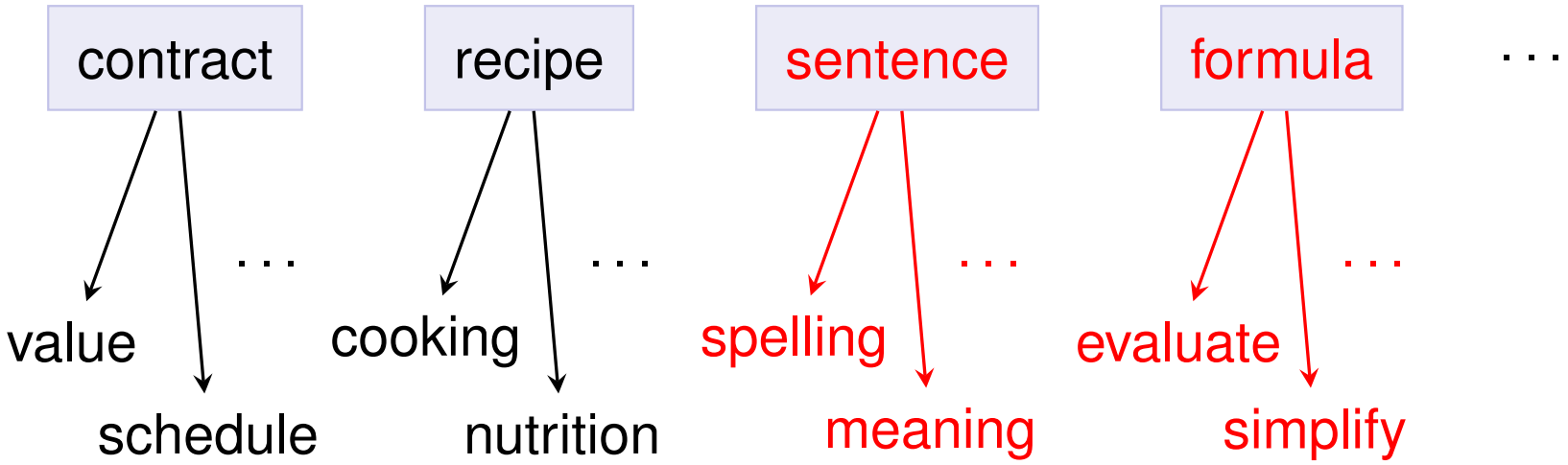



```
object {cone ...
  interior {media {emission <.4,.3,.2>}}}
object {sphere ...
  interior {media {emission <.4,.3,.2>}}}
object {cylinder ...}
```

Interpretations are everywhere



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."
>
> —Richard Montague, "English as a Formal Language" (1970)

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."
>
> —Richard Montague, "English as a Formal Language" (1970)

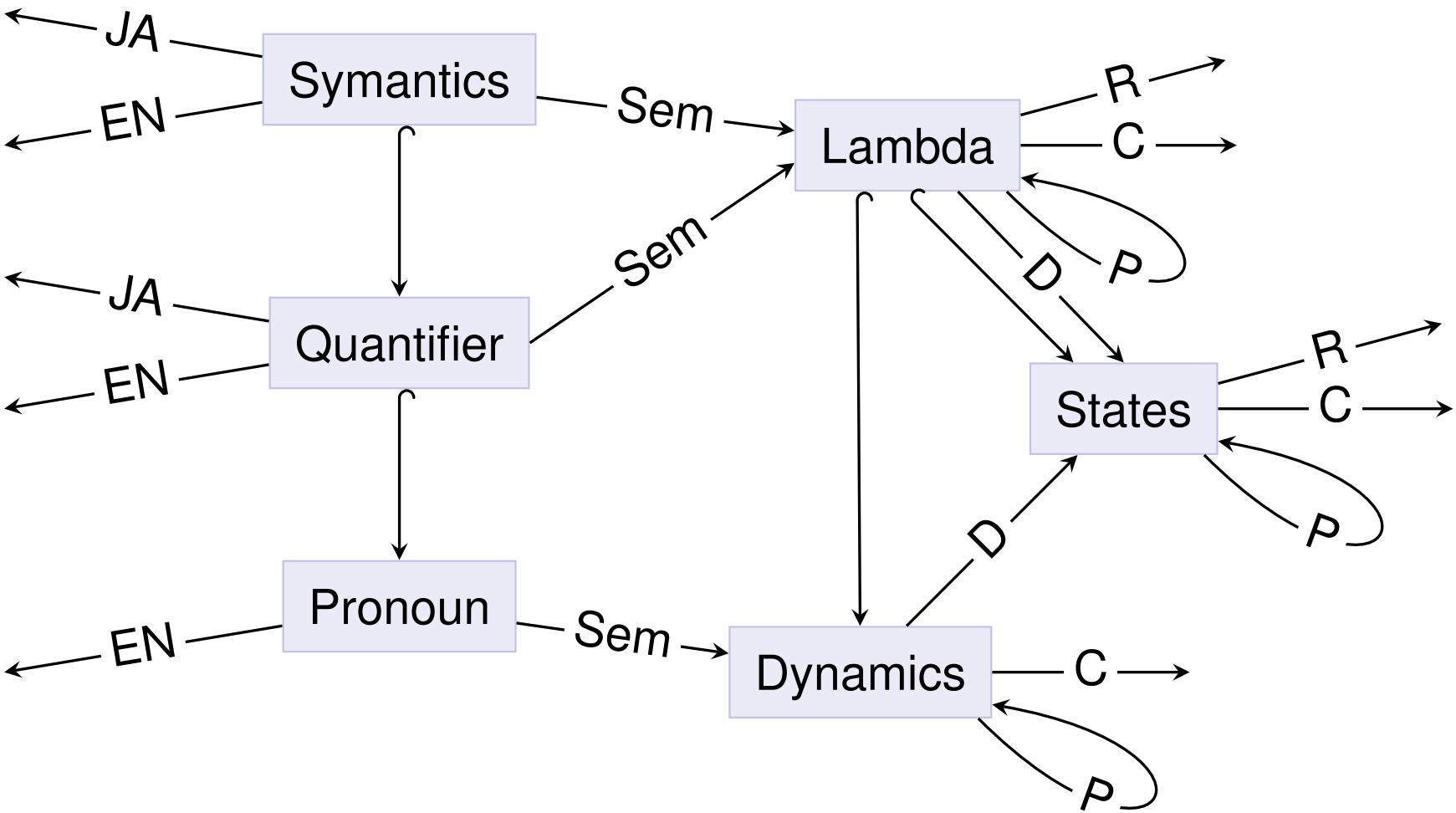
$$
\begin{aligned}
\mathrm{S} &::= \mathrm{NP}\ \mathrm{VP} & [\![\mathrm{S}]\!] &= [\![\mathrm{VP}]\!]([\![\mathrm{NP}]\!]) \\
\mathrm{VP} &::= \mathrm{TV}\ \mathrm{NP} & [\![\mathrm{VP}]\!] &= [\![\mathrm{TV}]\!]([\![\mathrm{NP}]\!]) \\
\mathrm{NP} &::= \mathrm{John} & [\![\mathrm{NP}]\!] &= j \\
\mathrm{NP} &::= \mathrm{Mary} & [\![\mathrm{NP}]\!] &= m \\
\mathrm{TV} &::= \mathrm{likes} & [\![\mathrm{TV}]\!] &= (o \mapsto (s \mapsto (s, o) \in \{(j, m), (m, j), \dots\}))
\end{aligned}
$$

Calculemus.

Understanding type classes

	`class`	`instance`
math	signature	model
linguistics	language	interpretation
computer science	interface	implementation

Map


JA
EN
Symantics
Sem
Lambda
R
C
P
D
JA
EN
Quantifier
Sem
States
R
C
P
D
EN
Pronoun
Sem
Dynamics
C
P

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!