Semantics

Greg Kobele

June 11, 2018

Review

Type Driven Application

$$\alpha \otimes \beta = \begin{cases} [\![\alpha]\!]\,([\![\beta]\!]) & \textit{if } \alpha : ab \textit{ and } \beta : a \\ [\![\beta]\!]\,([\![\alpha]\!]) & \textit{if } \alpha : a \textit{ and } \beta : ab \end{cases}$$

$\alpha \otimes \beta$ **is**

1. $\alpha(\beta)$ if that makes sense
2. $\beta(\alpha)$ if that makes sense
3. nothing otherwise

Semantic Interpretation Rules

$$\left[\!\!\left[\begin{array}{ccc} & \bullet & \\ / & & \backslash \\ \alpha & & \beta \end{array} \right]\!\!\right] = [\![\alpha]\!] \otimes [\![\beta]\!]$$

$$\left[\!\!\left[\begin{array}{c} \bullet \\ | \\ \alpha \end{array} \right]\!\!\right] = [\![\alpha]\!]$$

Problems

1. Our syntax is richer than the above allow
2. DPs in object position

   transitive verbs *eet*

   DPs *(et)t*

   $V \oplus DP$ doesn't make sense!

Lambda Calculus

Previously

We introduced the λ calculus in the first week of class

- in the context of interpreting *parts* of sentences

Given the meaning of the whole

- we wanted to break it up into parts
- so that we could assign meanings to words
- and use these word meanings
- to predict the meanings of new sentences

What are the meanings of parts?

Sentence meanings

truth conditions descriptions of how the world must be like for the sentence to be true

logical formulae structured objects that support inference

Parts

truth conditions sets, functions

logical formulae parts of formulae with holes

Every boy will laugh

truth conditions true iff every individual which is a boy laughs

logical formula $\forall x.\textsc{boy}(x) \rightarrow \textsc{laugh}(x)$

$\forall$ "for all"

$\rightarrow$ "if ... then"

need some way to break this up into pieces:

word	meaning
boy	BOY
laugh	LAUGH
every	$\forall x.\square_1(x) \rightarrow \square_2(x)$

Lambda calculus

$\forall x.\Box_1(x) \rightarrow \Box_2(x)$
a formula that is missing parts

- two ’holes’

Open questions

- how do I use this thing?
- if I have one object, which hole should I put it in?

The λ-calculus

a language for talking about decomposing structured objects

holes have names

$\lambda P.\lambda Q.\forall x.P(x) \rightarrow Q(x)$

- here, P is the name for the first hole
- and Q the name for the second

λ tells you how to use it

- the first object goes in the P hole
- the second into the Q hole

compare:

$$\lambda Q.\lambda P.\forall x.P(x) \rightarrow Q(x)$$

Plugging in Holes

The basic thing you do with holes is fill them up

$\lambda x.M$
an object with a hole named x

N
some object

$(\lambda x.M)\ N$
plugging N into the hole named x in M

β equivalence

$(\lambda x.M)\ N$
plugging N into the hole in M

Basically
M, where all x's have been replaced by N

write this as $M[x := N]$

We want these to mean the same thing
$(\lambda x.M)\ N \equiv M[x := N]$

Example

$$(\lambda P.\lambda Q.\forall x.P(x) \rightarrow Q(x))\ \textsc{boy}$$

Equivalent to:

$$(\lambda Q.\forall x.P(x) \rightarrow Q(x))[P := \textsc{boy}]$$

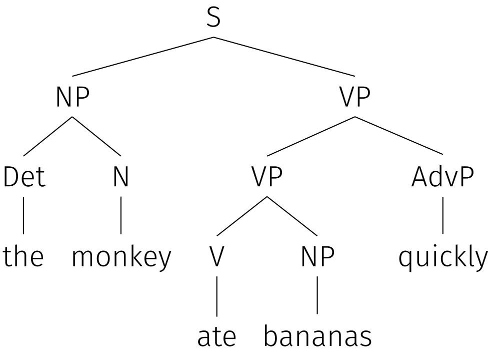
in words:

- replace all P's by BOY

$$\lambda Q.\forall x.\textsc{boy}(x) \rightarrow Q(x)$$

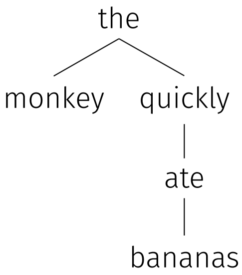
Examples

A tree



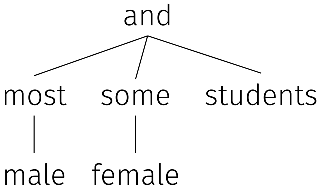
A logical formula

THE(MONKEY)(QUICKLY(ATE(BANANAS)))



Another one

(AND(MOST(MALE))(SOME(FEMALE)))(STUDENTS)



Types

Basic Types

There are different kinds of things

semantics propositions t

individuals e

trees T

a hole might restrict what it can be filled by

Not all holes are the same

$\lambda\phi.\lambda\psi.\phi \rightarrow \psi$

$\rightarrow$ "if … then"

connects two *propositions*

- what makes sense?
  - *if* John *then* Mary
  - *if* praises Susan *then* sleeps
  - *if* John steals *then* his mother cries

$\lambda x.\textbf{LAUGH}(x)$

- needs an *individual*
- and then becomes a *proposition*
- John *laughs*
- not John steals *laughs*

Type notation

$$\alpha\beta$$

- means that the next hole requires an α
- and that the result of filling it will be a β

$\lambda P.\lambda Q.\forall x.P(x) \to Q(x)$
has type

$$(et)(et)t$$

because

- the next hole (P) requires something of type (et)
- after P is filled, the next hole will be Q
- after Q is filled, we have a proposition

Examples

$a(b)(c(e)(f))(d)$

$\lambda x.a(b)(x)(d)$

$\lambda x.a(b)(x(e))(d)$

$\lambda x.x(\lambda y.c(y)(f))$