Lambda abstraction and its uses

PHIL 43916
November 12, 2012

1. The lambda operator

Our goal today is to introduce a new operator into our language — the lambda operator, symbolized by the Greek letter 'λ' — and show some of the natural language constructions it can be used to analyze.

We've discussed a few different sorts of operators so far. Negation combines with a sentence to form a sentence; conjunction and disjunction combine with a pair of sentences to form a sentence; and the complementizer 'that' combines with a sentence to form a complementizer phrase. 'λ' combines with sentences to form predicates (verb phrases).

Intuitively, and very roughly, the meaning of

λx(ψ)

has as its meaning the property of being such that ψ is true. So, for example,

λx(Grass is green)

is the property of being such that grass is green — in our semantics, the set of individuals which are such that grass is green. Why do we write 'λx' rather than just 'λ'? Because the principal purpose of this operator is to combine with formulae which contain an unbound (free) variable, like 'x is green or x is blue.' The formula

λx(x is green or x is blue)

stands for, intuitively, the property of being green or being blue.

How, exactly, should we understand the semantics of this operator? One way to state the semantic value of expressions containing it is the following:

$$\lambda x[\psi]t \leftrightarrow \psi[t/x]$$

where ‘$\psi[t/x]$’ stands for the sentence obtained by replacing every occurrence of ‘x’ with the term ‘t’. So, applying this to our example above would give us

$$\lambda x(\text{x is green or x is blue})t \leftrightarrow (\text{t is green or t is blue})$$

We can divide this biconditional into its two directions:

Lambda reduction
$\lambda x[\psi]t \rightarrow \psi[t/x]$

Lambda abstraction
$\psi \rightarrow \lambda x[\psi[x/t]]t$

These are schemata; when we say that these are rules governing ‘λ’, what we are saying is that every instance of these schemata are true.

We can also write out its semantics in more familiar terms as

$$[\![\lambda x S]\!]^{M,w,i,g} = \{u \in U: [\![S]\!]^{M,w,i,g[u/x]} = 1\}$$

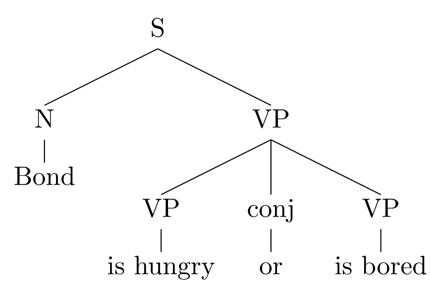
Now that we have this operator on the table, let’s see what it might be good for.

2. VP conjunction and disjunction

Consider the sentence

Bond is hungry or is bored.

From the point of view of the semantic theory we have been developing, this sentence is a little puzzling. One wants to provide a tree something like



But the problem is that the meaning of ‘or’ is a function from a pair of truth-values to a truth value; this simply gives us no way of computing the semantic value of the highest VP node on the basis of ⟦is hungry⟧ and ⟦is bored⟧, which are sets of individuals.

A natural thought is that we can reduce predicate disjunction and conjunction to ordinary sentence-level disjunction and conjunction, by taking the logical form of the above sentence to be

> Bond is hungry or Bond is bored.

which has an unproblematic structure. The problem is that this sort of move seems not to be always available. Consider the sentences

> Everyone is hungry or is bored.
> Someone is bored and is hungry.

Pursuing the strategy sketched above would give us the wrong truth conditions for each.

This suggests that we need some way of understanding how conjunction and disjunction can be used to form complex VPs; ideally, we should be able to do this while avoiding the surprising conclusion that conjunction as it occurs in predicates is wholly unrelated to the meaning of ‘and’ when used as a sentence connective.

Our lambda operator provides us with one way to do this. We might provide something like the following rules for predicate conjunction and disjunction:

$$[\![_{\mathrm{VP}}\ \mathrm{VP}_1 \text{ or } \mathrm{VP}_2]\!] = [\![\lambda x[\mathrm{VP}_1(x) \vee \mathrm{VP}_2(x)]]\!]$$
$$[\![_{\mathrm{VP}}\ \mathrm{VP}_1 \text{ or } \mathrm{VP}_2]\!] = [\![\lambda x[\mathrm{VP}_1(x)\ \&\ \mathrm{VP}_2(x)]]\!]$$

Here’s a complication with this story. Consider a case in which we have a complex predicate one part of which contains a quantifier, as in

> Bond is boring and eats everything.

So far we’ve understood quantifier phrases like ‘everything’ as combining with sentences to form sentences. But the only way to fit the above sentence into that form would be for the quantifier to move out of the complex predicate to take wide scope over the whole sentence; and this would seem to violate our constraints about quantifiers moving out of conjuncts, which we used to explain the ungrammaticality of sentences like

> Everyone went to the store and he was disappointed.

This means that we have to treat ‘everything’ as combining with the verb ‘eats’. What rule might you give for ⟦[every β] VP⟧?

This might seem a bit ad hoc. But some evidence that in sentences like this the quantifier does attach to the VP is given by the fact that whereas

> A napkin was next to every plate.

is ambiguous (and favors the reading on which ‘every plate’ has wide scope), the sentence

> A napkin was next to every plate but had already been used.

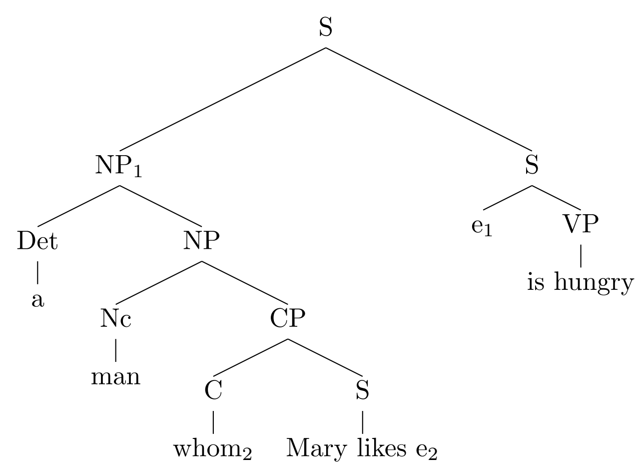
forces the reading on which ‘a napkin’ has wide scope. The idea would be that this is explained by the constraint on movement of ‘every plate’ out of conjuncts.

3. Relative clauses

The following are some example sentences involving relative clauses:

> A man *whom Mary likes* is hungry.
> A man *that likes Mary* is bored.

It looks like these clauses are functioning as predicates; in particular, they look like predicates derived from sentences. This makes it very natural to use λ in the analysis of these clauses. Consider the first sentence above. We might (simplifying a bit the view defended in the text) take its tree to be something like



There are two steps the computation of the semantic value of this tree that need comment. The first is the way in which we derive the semantic value of the CP from ‘whom’ and the S. Let’s suppose that

$[\![\text{Mary likes e2}]\!]^{M,w,i,g}=1$ iff Mary $\in \{x{:}<x,g(e2)>\in[\![\text{likes}]\!]^{M,w,i,g}\}$

then

$$[\![\text{whom Mary likes e2}]\!]^{M,w,i,g} = [\![\lambda y[\text{Mary likes e2}[y/e2]]]\!]$$
$$= \{y{:}\ \text{Mary} \in \{x{:}<x,y>\in[\![\text{likes}]\!]^{M,w,i,g}\}\}$$

This gives us a set of individuals — the set of individuals that Mary likes — as the semantic value of the CP. How do we combine this with $[\![\text{man}]\!]$ to get the semantic value of the parent NP?

We again use lambda abstraction, in the way just illustrated by our discussion of predicate conjunction and disjunction. The idea is, very roughly, that

$$[\![\text{NP}]\!]=[\![\lambda x(\text{man}(x)\ \&\ \text{whom Mary likes}(x))$$
$$=\{x{:}x \in [\![\text{man}]\!] \text{ and } x \in \{y{:}\ \text{Mary} \in \{x{:}<x,y>\in[\![\text{likes}]\!]^{M,w,i,g}\}\}\}$$

4. VP anaphora

Here's an example of VP anaphora (also called in the text 'VP deletion' and 'VP ellipsis'):

John went to the store. Bob did too.

A natural first thought is that this is *simply* a case of ellipsis — we are just saving time by not repeating words. On this interpretation, these sentences mean the same thing as

John went to the store. Bob went to the store too.

But this can't be quite right, as is shown by

John went to the store. Bob will too.

Our initial idea would lead to interpreting this pair as either of

John went to the store. Bob went to the store too.
John went to the store. Bob will went to the store too.

But the first interpretation gives the wrong truth conditions, and the second is ungrammatical. This indicates that we have to look to the logical form of the relevant sentences — where we can find an untensed verb to serve as the antecedent for the second sentences — rather than just to the surface form.

Now consider the following pair:

> Bob thinks that he is the cat's pajamas.
> John does too.

Contrast two readings of this pair: the 'strict reading' and the 'sloppy reading.' A possible explanation for this distinction in terms of the distinction between a reading of the first sentence in which 'he' is bound, and a reading of that sentence in which it is unbound and assigned a value by the context.

Here is a residual puzzle involving the interaction of scope with VP ellipsis. It is a fact familiar from our discussion of quantifier phrases that

> A student showed the campus to every visitor.

is ambiguous depending on which quantifier takes wide scope. This ambiguity remains if we add to the discourse another sentence anaphoric on this one, as in

> A student showed the campus to every visitor. And a staff member did too.

This is as it should be. The odd thing is that if we use a proper name rather than 'a staff member', we (arguably) force the reading in which 'a student' has wide scope:

> A student showed the campus to every visitor. And Bob did too.

But it's not obvious why both interpretations of the first sentence shouldn't still be available.