Problem set #2

PHIL 43916

September 12, 2012

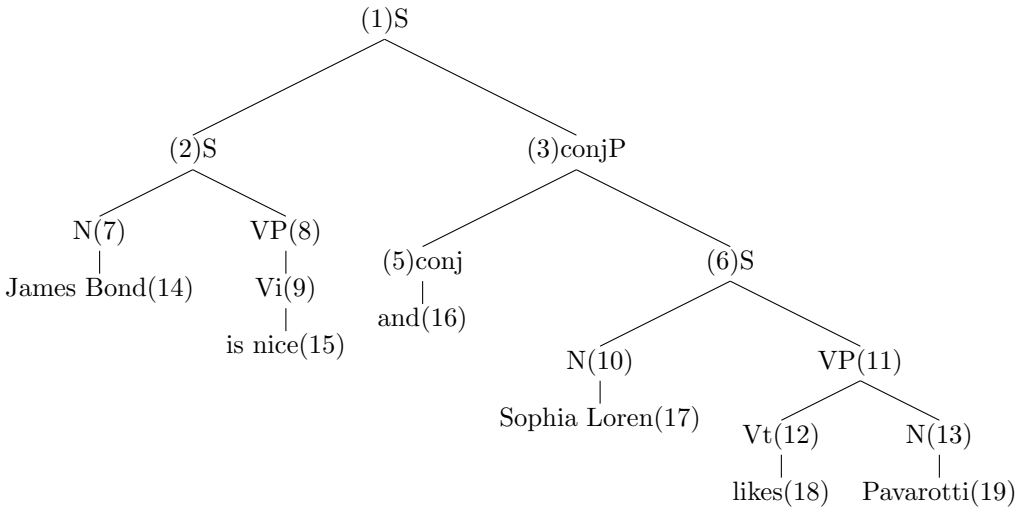
1. Rule-to-rule semantics

Everyone did pretty well on this one. Some fairly minor common things:

- You should make clear when you are using the lexicon, just as you are clear about what rule of the semantic theory you are using.
- Our semantic rules tell you how to get from one node to another. Hence you should always be clear, not just about which rules you are using, but about which prior results about the semantic values of other nodes you are using. So, e.g., you shouldn't just say that a given claim follows from rule (b), but rather than it follows from ($[\![3]\!]^v$, $[\![5]\!]^v$, (b)).
- Despite the fact that I often do this on the board to save time, you don't need to draw little pictures of people. You can just write $[\![\text{Pavarotti}]\!]^v$=Pavarotti. (You also don't need to write anything more complicated, like $[\![\text{Pavarotti}]\!]^v$=Pavarotti the man.)

2. Type driven semantics

On this semantics, the tree should look like this:



And we can calculate its truths conditions as follows:

Node	Value	From
$[\![14]\!]^v=$	James Bond	lexicon
$[\![7]\!]^v=$	James Bond	$[\![14]\!]^v$, pass up
$[\![15]\!]^v=$	f: f(x)=1 if x∈{y:y is nice in v} and 0 otherwise	lexicon
$[\![9]\!]^v=$	f: f(x)=1 if x∈{y:y is nice in v} and 0 otherwise	$[\![15]\!]^v$, pass up
$[\![8]\!]^v=$	f: f(x)=1 if x∈{y:y is nice in v} and 0 otherwise	$[\![9]\!]^v$, pass up
$[\![2]\!]^v=$	$[\![8]\!]^v([\![7]\!]^v)$	functional application
$[\![2]\!]^v=$	1 if James Bond∈{y:y is nice in v} and 0 otherwise	$[\![2]\!]^v$, $[\![8]\!]^v$, $[\![7]\!]^v$, substitution
$[\![17]\!]^v=$	Sophia Loren	lexicon
$[\![10]\!]^v=$	Sophia Loren	$[\![17]\!]^v$, pass up
$[\![18]\!]^v=$	f: f(x)=the function g which is such that g(y)=1 if y∈{y:y likes x in v} and 0 otherwise	lexicon
$[\![12]\!]^v=$	f: f(x)=the function g which is such that g(y)=1 if y∈{y:y likes x in v} and 0 otherwise	$[\![18]\!]^v$, pass up
$[\![19]\!]^v=$	Pavarotti	lexicon
$[\![13]\!]^v=$	Pavarotti	$[\![19]\!]^v$, pass up
$[\![11]\!]^v=$	$[\![12]\!]^v([\![13]\!]^v)$	functional application
$[\![11]\!]^v=$	f: f(y)=1 if x∈{x:x likes Pavarotti in v} and 0 otherwise	$[\![11]\!]^v$, $[\![12]\!]^v$, $[\![13]\!]^v$, substitution
$[\![6]\!]^v=$	$[\![11]\!]^v([\![10]\!]^v)$	functional application
$[\![6]\!]^v=$	1 if Sophia Loren ∈{x:x likes Pavarotti in v} and 0 otherwise	$[\![6]\!]^v$, $[\![10]\!]^v$, $[\![11]\!]^v$, substitution

$[\![16]\!]^v$=	f: f(x)=[1→1, 0→0] if x=1, and f(x) =[1→0, 0→0] if x=0	lexicon
$[\![5]\!]^v$=	f: f(x)=[1→1, 0→0] if x=1, and f(x) =[1→0, 0→0] if x=1	$[\![16]\!]^v$, pass up
$[\![3]\!]^v$=	$[\![5]\!]^v([\![6]\!]^v)$	functional application
$[\![3]\!]^v$=	f: [1→1, 0→0] if Sophia Loren ∈{x:x likes Pavarotti in v} and [1→0, 0→0] otherwise	$[\![3]\!]^v$, $[\![5]\!]^v$, $[\![6]\!]^v$, substitution
$[\![1]\!]^v$=	$[\![3]\!]^v([\![2]\!]^v)$	functional application
$[\![1]\!]^v$=	1 if Sophia Loren ∈{x:x likes Pavarotti in v} and James Bond∈{y:y is nice in v} and 0 otherwise	$[\![1]\!]^v$, $[\![2]\!]^v$, $[\![3]\!]^v$, substitution

General thoughts:

- In this semantics, the semantic value of many nodes will be a function. Suppose that $[\![1]\!]$ is the function which has value 1 if its argument is nice and 0 otherwise. It is bad form to write this as

  $[\![1]\!]$=f(x)=1 if x is nice and 0 otherwise

  since this implies that the $[\![1]\!]$ sometimes =1 — which it doesn't. (If it's semantic value is a function, it can't also be a truth value.) Better to write

  $[\![1]\!]$=the function f which is such that f(x)=1 if x is nice and 0 otherwise

  or, for short,

  $[\![1]\!]$= f: f(x)=1 if x is nice and 0 otherwise
- When you are deriving the truth conditions, it is OK if you take *some* shortcuts. For example, you can combine the pass-up and lexicon steps into one line, and you can combine the functional application and substitution steps into one line. But I thought that it would be useful to have a derivation laid out with all of the steps made explicit.
- Suppose that you have some sentence $[_S$ neg S*]. Then it is true to say that $[\![S]\!]^v$=$[\![neg]\!]([\![S^*]\!])$. But this is not the most informative statement of the truth conditions for S. What you want to do is to substitute in the semantic values you have already derived — whether from the lexicon or other calculations — for neg and S*, and to say what the value of the function $[\![neg]\!]$ is for argument $[\![S^*]\!]$.