Semantics 564 **Lecture 15** Oct. 14, 1999

1 A quick note on what we're doing

What we're doing is (trying to) make the natural language English a fully-defined formal language. For any interpretable string of English (as defined by native speakers, who know the secrets of English, but won't tell), the goal is to be able to interpret it fully, using our formal system, and come up with just the meaning(s) that it has, and no others. If we get it right, we'll have a good model of what goes on in the mind of speakers when they're decoding strings of English. And we can tell how we're doing by comparing the meanings we get with the meanings understood by English speakers.

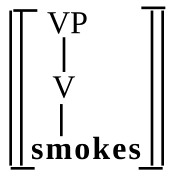
In any sense, semantics is the hardest thing to model, of all the aspects of language. Phonology is relatively easy: you look at the strings of sounds, and observe their relationship to one another, and how they change, and there isn't anything intrangible about it. Morphology is slightly trickier, but still fairly straighforward. (American structuralists, in the first half of this century, concluded that the job of describing a language was finished when they'd described its phonology and morphology. In an interesting sense, they were right.. more on that in a minute). Syntax wasn't investigated except at a very superficial level until after *Syntactic Structures*, because nobody really had an idea of how to go about investigating it -- it just wasn't somthing you could formulate a coherent question about. Well, now we can formulate coherent questions about syntax, and we have a very good idea of what the syntax of natural languges looks like -- and there's far less variation than you might think. In fact, the farther up the structural ladder you go, the less variation there is (which makes a certain amount of sense). In semantics, it's perhaps arguable that there's *no* variation in how languages go about encoding and decoding messages; given a certain syntactic structure, any human will compose its meaning in the same way. (Lexical items may differ, of course, as will Gricean implicatures and cultural conventions, but the actual process of finding out what the pieces of structure mean is identical). So in some sense the structuralists were right -- it's possible that a semantic model is not something that should be described for an individual language, but for all languages. The process of composition we describe for English should in principle apply to syntactic structures generated by any language's grammar.

2 Some review

Everybody's seems to have a fairly good handle on what's going on, but everybody's got some small point or otherthat they're confused about, so we're going to spend a little more time on types and functions just to hammer it all home. First off, here's some clarifications:

1. *denotation* = fully specified meaning; the [[ ]] interpretation function applied to a lexical item or piece of structure.

The denotation of [[**Jan**]] is Jan

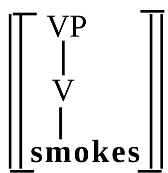
The denotation of  is f: $D_{<e>} \text{ --> } D_{<t>}$

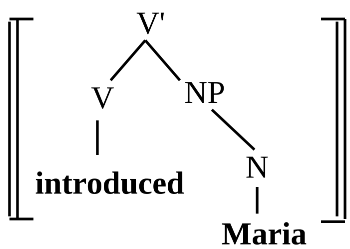
for any x $D_{<e>}$, f(x)=1 iff x smokes

or [λx $D_{<e>}$ **.** x smokes]

2. *type* = the class that a denotation belongs to, i.e. a thing, a truth value, or a function of some sort.

The type of [[**Jan**]] is <e>.

The type of  is <e,t>

The type of  is <e,<e,t>>.

The type of [[**and**]] is <t, <t,t>>

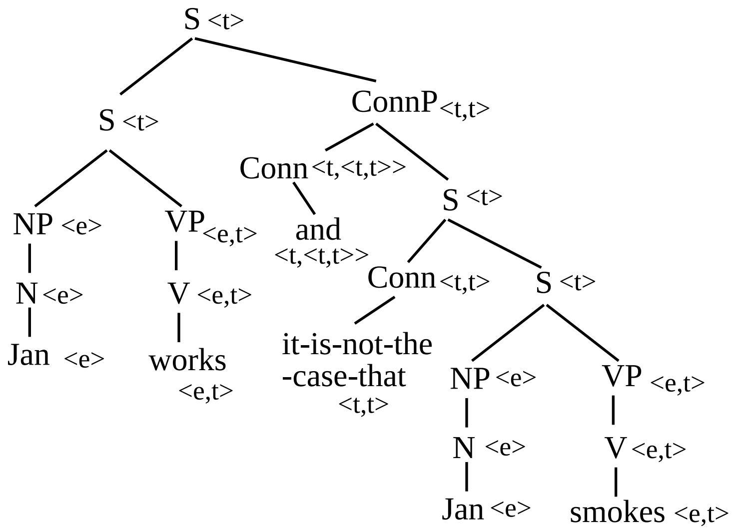
3. The set of all things of a given type is $D_{<type>}$

Now, let's look at the part of the first exercise I didn't do (oops!) last time.
Here's what H&K say:

"Using the labelling system introduced at the end of section 2.3, specify the type of denotation for each node in your binary branching structure for the sentence, "Jan works, and it is not the case that Jan smokes."

What do they want here? Well, they ask for the *type* of the denotation for each node. They don't actually want you to calculate the actual denotation for each node, rather, they just want you to give its type. Here's the answer:

3. "Jan works, and it is not the case that Jan smokes."



4. Now, more clarifications: the relationship between the different notations for functions.

Let's say we're in the following world.

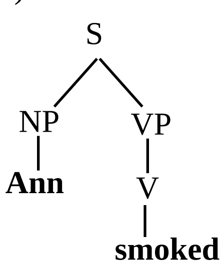
(a) there are three individuals, Ann, Jan, and Sue

(b) there are three things that they do: smoking, hugging, and introducing.

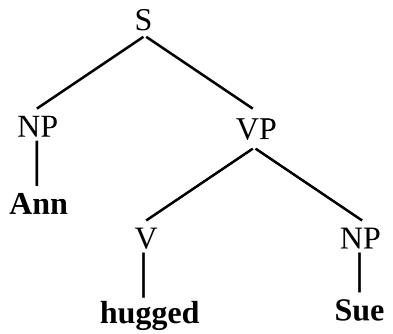
(c) The following things occurred:
Ann smoked, and Sue smoked, but Jan didn't smoke.
Ann hugged Sue, and Jan hugged Sue, and no other huggings happened.
Ann introduced Jan to Sue, and Ann introduced Sue to Jan, and no other introductions happened.

5. So, we've got the following tree structures for our three events:

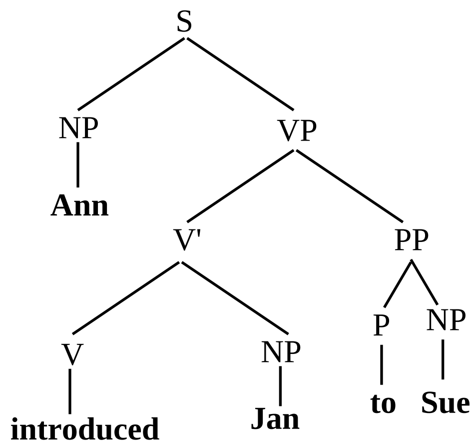
(a)



(b)



(c)



6. What are the various ways we've written the denotations of our various functions?

(a) [[**smoked**]]

= f: $D_{<e>}$ --> {0,1}
for any x $D_{<e>}$, f(x) = 1 iff x smoked

=[λx $D_{<e>}$ **.** x smokes]

(Note that this isn't the *reference* of **smoked** in this world, just its *sense*.

= f:

Ann -->	1
Sue -->	1
Jan -->	0

= {<Ann, 1>, <Sue, 1>, <Jan,0>}

(b) [[**hugged**]]

(now, we know this has to be a function from individuals to a function from individuals to truth values, because of the structure we know it has in 5b above):

= f: $D_{<e>}$ --> {g: $D_{<e>}$ --> {0,1}}
for any x $D_{<e>}$, f(x) = g_x: $D_{<e>}$ --> {0,1}
for any y $D_{<e>}$, g(y) = 1 iff y hugged x.

= f: $D_{<e>}$ --> $D_{<e,t>}$
for any x,y $D_{<e>}$, f(x)(y) = 1 iff y hugged x.

= [λx $D_{<e>}$ **.** [λy $D_{<e>}$ **.** x hugged y]]

= f:

Ann -->	Ann -->	0
	Jan -->	0
	Sue -->	0
Sue -->	Ann -->	1
	Jan -->	1
	Sue -->	0
Jan -->	Ann -->	0
	Jan -->	0
	Sue -->	0

= {<Ann, <Ann, 0>> , <Ann, <Jan, 0>> , <Ann, <Sue, 0>> , <Sue, <Ann,1>>, <Sue, <Jan, 1>> , <Sue, <Sue,0>>, <Jan, <Ann,0>> , <Jan, <Jan,0>>, <Jan, <Sue,0>>}

(c) [[**introduced**]]

= f: $D_{<e>}$ --> {g: $D_{<e>}$ --> {h: $D_{<e>}$ --> {0,1}}}
for any x $D_{<e>}$, f(x) = g_x: $D_{<e>}$ --> {h: $D_{<e>}$ --> {0,1}}
for any y $D_{<e>}$, g(y) = h_{xy}: $D_{<e>}$ --> {0,1}
for any z $D_{<e>}$, h(z)=1 iff z introduced x to y.

= f: $D_{<e>}$ --> $D_{<e, <e,t>>}$
for any x,y,z x $D_{<e>}$, f(x)(y)(z) = 1 iff z introduced x to y.

$= [\lambda x \;\; D_{<e>} \,.\, [\lambda y \;\; D_{<e>} \,.\, [\lambda z \;\; D_{<e>} \,.\,$ z introduced x to y]]]

= f

Ann -->	Ann -->	Ann -->	0
		Jan -->	0
		Sue -->	0
	Jan -->	Ann -->	0
		Jan -->	0
		Sue -->	0
	Sue -->	Ann -->	0
		Jan -->	0
		Sue -->	0
Jan -->	Ann-->	Ann -->	0
		Jan -->	0
		Sue -->	0
	Jan -->	Ann -->	0
		Jan -->	0
		Sue -->	0
	Sue -->	Ann -->	1
		Jan -->	0
		Sue -->	0
Sue -->	Ann -->	Ann -->	0
		Jan -->	0
		Sue -->	0
	Jan-->	Ann -->	1
		Jan -->	0
		Sue -->	0
	Sue -->	Ann -->	0
		Jan -->	0
		Sue -->	0

= {<Sue, < Ann, <Ann,0>>>, <Sue, < Ann, <Jan,0>>>, <Sue, < Ann, <Sue,0>>>,
<Sue, < Jan, <Ann,1>>>, <Sue, < Jan, <Jan,0>>>, <Sue, < Jan, <Sue,0>>>,
<Sue, < Sue, <Ann,0>>>, <Sue, < Sue, <Jan,0>>>, <Sue, < Sue, <Sue,0>>>,
<Jan, < Sue, <Ann,1>>>, <Jan, < Sue, <Jan,0>>>, <Jan, < Sue, <Sue,0>>>,
<Jan, < Jan, <Ann,0>>>, <Jan, < Jan, <Jan,0>>>, <Jan, < Jan, <Sue,0>>>,

<Jan, < Ann, <Ann,0>>>, <Jan, < Ann, <Jan,0>>>, <Jan, < Ann, <Sue,0>>>,
<Ann, < Sue, <Ann,1>>>, <Ann, < Sue, <Jan,0>>>, <Ann, < Sue, <Sue,0>>>,
<Ann, < Jan, <Ann,0>>>, <Ann, < Jan, <Jan,0>>>, <Ann, < Jan, <Sue,0>>>,
<Ann, < Ann, <Ann,0>>>, <Ann, < Ann, <Jan,0>>>, <Ann, < Ann, <Sue,0>>>}

7. Notes about the final exercise:

(c) $\lambda X \quad Pow(D)\,.\,\{y \quad D_{<e>}: y \quad X\}$

The term $\{y \quad D_{<e>}: y \quad X\}$ is a *set* , not just one element. This is the regular old set predicate notation that we used for the whole beginning of term — that colon means "such that"; it's not the same as the **.** in the lambda-notation. Watch for the squiggly brackets vs. square brackets -- if it has squiggly brackets around it, it's a set; square brackets + a λ mean a function. In particular, some of you thought that this mapped an element X of PowD onto some one y that wasn't a member of X: that's not right. It maps an element X of PowD onto the *set* of elements of D that aren't in X.

Note, further, that they've chosen a capital X to represent a variable that ranges over sets (elements of Pow(D)).

(d) $\lambda X \quad Pow(D)\,.\,[\lambda y \quad D_{<e>}\,.\,y \quad X]$

Now, here, we've still got the domain of sets of Pow(D), but the range is no longer a set. Now, the range (the element after the dot) is a function. And it's a function that maps any element y of $D_{<e>}$ onto truth value 1 iff y is not a member of X. So this time, the term $[\lambda y \quad D_{<e>}\,.\,y \quad X]$ is the characteristic function of the term $\{y \quad D_{<e>}: y \quad X\}$.

3 **Argument Structure and Linking** (section 3.5 of H&K)

For completeness, let's finish off chapter 3, and talk about H&K's thoughts on argument structure and linking. Recall that last time, we were contrasting a theory where the lexical entry for a predicate contained a theta-grid with one where it just consisted of the H&K-style function:

8.

(a) θ-criterion lexical entry:

hug (agent, theme)
introduce (agent, goal, theme)

(+ of course specification of its meaning)

(b) H&K-style lexical entry

hug: $[\lambda x \quad D_{<e>} . [\lambda y \quad D_{<e>} . \text{ y hugs x}]]$

introduce: $[\lambda x \quad D_{<e>} . [\lambda y \quad D_{<e>} . [\lambda z \quad D_{<e>} . \text{ z introduced x to y}]]]$

Argument structure addition to theta-grid:

9. **hug** (agent, (theme))
   introduce (agent, (goal, (theme)))

In addition to specifying what theta-roles a verb has, on this conception, the theta-grid can tell you where to put them. The least prominent/embedded argument goes in the least embedded syntactic position, etc. Note: this is a theory about *projection*.

H&K say, rightly, that this type of hierarchical ordering of theta-roles is also redundant on their theory, because the lexical entry for **introduce**, for instance, already has all this hierarchy built into it. There could never be a situation where the most prominent argument of **introduce** was not interpreted as the agent.

Now, it's a fact that languages don't allow random shuffling of their arguments. So, H&K's example,

10. a) Sue hugged Jan.
    b) Sue hugged Jan.

can't mean, in one use, that Sue hugged Jan, and in the second use, that Jan hugged Sue. That is, the argument structure of **hug** always has to map the agent role to the subject position, and the theme role to the object position. The notion of prominence, in combination with a principle like UTAH

11. UTAH (Uniformity of Theta Assignment Hypothesis)
    Identical thematic relationships between items are represented by identical structural relationships.

ensure this, on a theory with argument structures.

On a theory like H&K's, this situation, they claim, could never arise, because of the way functional application applies. The first argument to combine with the verb **hug** will always be interpreted as the internal argument, simply because of the definition of hug: a function that takes an argument x to give a function that takes an argument y to give the truth value 1 if y hugs x. No separate principle is needed to guarantee that a structure with **hug** in it cannot receive a pseudo-passive interpretation.

However, a separate principle *is* still needed. There's nothing, on H&K's theory, that would prevent a language from having a lexical entry like 12 a or b:

12. (a) **guh**: $[\lambda x \quad D_{<e>} . [\lambda y \quad D_{<e>} . x \text{ hugs } y]]$

$[_S$ Ann $[_{VP}$ guhs Jan]]

(b) **ecudortni**: $[\lambda x \quad D_{<e>} . [\lambda y \quad D_{<e>} . [\lambda z \quad D_{<e>} . y \text{ introduced } x \text{ to } z]]]$

$[_S$ Ann $[_{VP}$ $[_{V'}$ introduced Jan] $[_{PP}$ to Sue]]]

That is, in a fictional language, there could be a verb "guh" which in the structure given, "Ann guhs Jan" would mean "Jan hugs Ann". Ditto for introduce. There are no such languages, it seems, and UTAH, on a strong version, predicts this. H&K's lexical entries don't. That's ok, though; it seems reasonable to assume that in fact the *frame* is the source of the "theta-roles", and the verb just specifies how the particular arguments relate to each other. That is, the interpretation of a syntactic structure by the brain involves some structural equivalent of a strong UTAH, and separate argument structures aren't necessary.

In fact, the difference between one-place verbs like **dance** and verbs like **fall**, will have to be taken care of by something like UTAH, at least on the version of the semantics and asumptions about their structures that we have so far.

Now in many theories, operations can apply to these argument structures to manipulate them and change the mapping of the arguments to the syntax, e.g., Passive:

13. (a) **hug**$_{\text{Passive}}$ (agent, (theme))

|

by-phrase

(b) $[_S$Jan$_i$ (was) $[_{VP}$ $[_{V'}$ hugged t_i] [by Sue]]]

This operation, which happens in the lexicon, changes the mapping of the argument structure to the syntax. The theme argument (the object) still gets mapped to the object position, but the agent now gets mapped to the by-phrase. Syntactic principles will then require the object to move to subject position, giving the surface order you see in 10b.

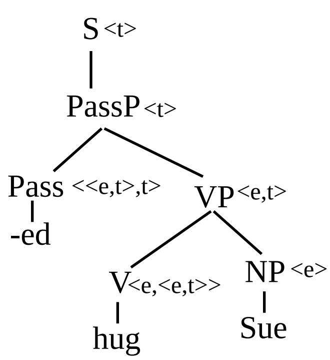
For Heim and Kratzer, no such lexical operation manipulating the interpretation of **hug** is possible. Rather, passive would have to be a syntactic operation. For instance, the passive morpheme could be a function like this:

14. [[**-en**]]: f: $D_{<e,t>}$ --> {0,1}
for any g $D_{<e,t>}$, f(g)=1 iff there is an x $D_{<e>}$ such that g(x)=1

(In order for this to work, we'll have to make certain assumptions, such as that objects of verbs are interpreted in their base postion, not their case position — chains are interpreted in their base position):

So, "Sue was hugged" will have a structure like this:

15.

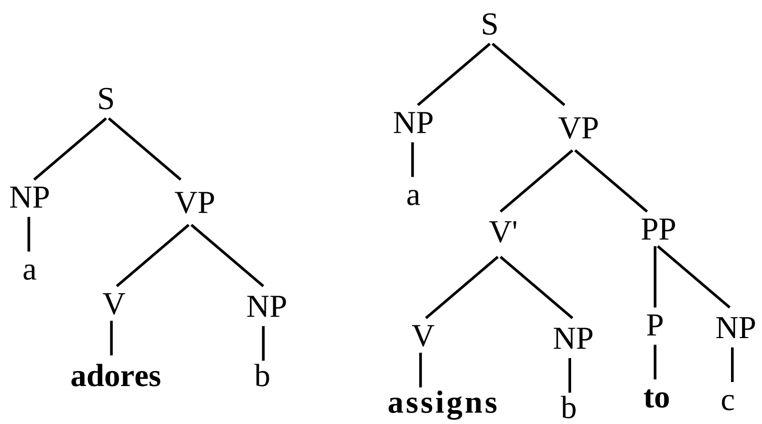


So Passive is a morpheme that itself has a lexical entry and composes with something else by functional application. (Let's not take this proposal too seriously; it's just an example of the sort of approach that is possible within H&K's framework. The main point is that an argument-structure-manipulation approach doesn't make any sense, in their framework.)

Homework:

1. p. 31 exercise 1

When discussing the last part of the question (whether or not your Schönfinkeled functions could be appropriate denotations for the English verbs in this case), assume the following phrase structure trees for the English verbs:



(Note: the answer to this question corresponds to my discussion of Schönfinkeling the characteristic function of the predicate-logical denotation of **love** two classes ago).

2. p.39 exercise 2 (continues onto p. 40).

Bonus question:

3. Look at the two binary-branching functions we have for **and** so far, that allow it to conjoin Ss and VPs (see last time's handout for the latter). Try to construct a third denotation for **and** that will allow it to conjoin two NPs and produce a denotation for the conjunction that will be a suitable argument for any function that takes elements from $D_{<e>}$ -- e.g. "Jan and Sue smoke" and will result in the appropriate truth conditions for the whole sentence. If you get really ambitious, discuss whether or not your denotation gives the appropriate interpretation for the sentence "Ann and Jan carried the chair up the stairs" when it's used to describe each of the following circumstances:

(a) Ann carried the chair up the stairs by herself, and then Jan did the same thing (chair goes up the stairs twice)

(b) Ann and Jan together picked up the chair and carried it up the stairs. (chair goes up the stairs once)

The idea is that the English sentence "Ann and Jan carried the chair up the stairs" can be appropriately used to describe both circumstances, but it's extremely likely that your denotation for "and" will only produce a meaning appropriate to one of the sentences.