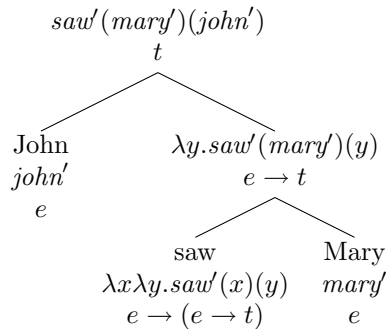


LING419 Homework 3

Trees and movement with semantics

For parts B, C and D of this assignment, when asked to give a derivation in the style of Heim and Kratzer (H&K), you should give a semantic value and a semantic type for each node of the tree, like this:

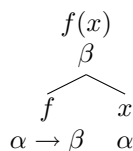


There are two rules in the H&K system (for now). Wherever two constituents combine to form a larger one, you need to use one of these rules to determine the semantic value of the new, larger constituent. In the simple example above, function application applies at each node. Remember that trees are like mobiles, so left-to-right order is insignificant. In the following descriptions, α and β are placeholders for semantic types like e , t , $e \rightarrow t$, etc.

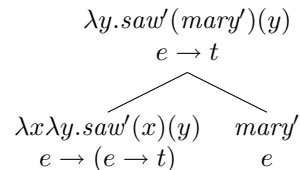
In this system, we take the semantic value of a trace t_i , for any index i , to be a sort of “dummy entity” of type e , written just as t_i (or $A(i)$ if you prefer).

- **Function application (FA):** If you combine something denoting f , which is a function from type α to type β , with something denoting x , which is of type α , the result denotes $f(x)$ — which is, of course, of type β .

Recipe:

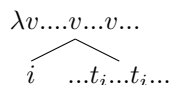


Example:

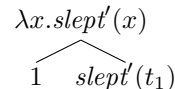


- **Predicate abstraction (PA):** If you combine something denoting Φ with an index i , the result denotes what you get when you: (a) choose a new variable which doesn't appear anywhere in Φ , say v ; (b) put λv on the front of Φ ; and (c) wherever t_i (or $A(i)$ if you prefer) appears in Φ , replace it with v .

Recipe:



Example:



A Successive cyclic movement (4 points)

Using the lexical items in (1) we can easily derive the sentence in (2). We can also derive the question in (3a) if we just put aside subject-auxiliary inversion, i.e. we actually derive the string in (3b) with the auxiliary ‘will’ in the T position, without it having moved to the C position.¹ In each case we end up with a tree that has only a *c* feature on its head; this is sort of the equivalent to ending up with an S in categorial grammar.

(1)	Mary :: d	see :: =d V	will :: =V =d t	$\epsilon_C :: =t c$
	John :: d	say :: =c V		$\epsilon_C :: =t +wh c$
	Fred :: d			that :: =t c
	who :: d -wh			

(2) Mary will say that John will see Fred.

- (3) a. Who will John see?
b. Who John will see?

Recall that when we looked at subjacency accounts of island effects, it made sense to suppose that a *wh*-phrase moves “successive cyclically” through all the SpecCP positions it passes by; see Haegeman 1994, pp. 402–405. So in the derivation of (4a)/(4b), ‘who’ makes two movement steps: first, out of its base position (complement of ‘see’) to an intermediate SpecCP position, and second, from this intermediate position to its pronounced position.

- (4) a. Who will Mary say that John will see?
b. Who Mary will say that John will see?

What goes wrong if we try to use the lexicon in (1) to produce this “two step” derivation of (4b) (still ignoring head movement)? What do we need to add to the lexicon in (1) in order to produce it? Show a derivation of (4b) using your modified lexicon.

NB: Don’t change anything about the merge and move *rules* that we have introduced in class, just add new entries to the list in (1) for these rules to work on.

B Relative clauses (6 points)

Relative clauses in English

Give the full H&K-style derivation for the expression in (5), using the tree in (6) and the lexical semantic values in (7).

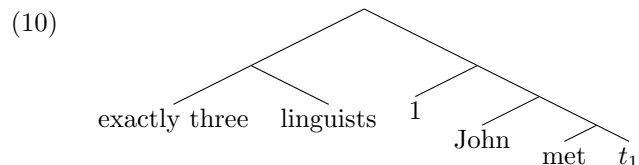
Notice that this expression is just a phrase — *not* a complete sentence — so it will not end up with a semantic value of type *t*. Rather, since the distribution of this phrase is just the same as the distribution of the word ‘girl’, the final semantic value of the phrase should be of type $e \rightarrow t$, like ‘girl’ is.

(5) girl who John met

¹If you’re unsure about subject-auxiliary inversion, or T-to-C movement, see Carnie 2007, pp. 260–263. If it seems careless to be putting it aside, just think of what we’re deriving as an embedded question with ‘I wonder’ in front of it, like we did when talking about island effects in categorial grammar: ‘I wonder who John will see’. This removes the complication of subject-auxiliary inversion. There are also some dialects of English (eg. Indian English) which, it has been claimed, don’t do subject-auxiliary inversion even in non-embedded questions, so (3b) is exactly what’s pronounced in these dialects.

Recall that if P is some property, then $\{x : P(x)\}$ is the set of all things that have that property. For example, $\{x : girl'(x)\}$ is the set of all girls.

(9) John met exactly three linguists



(11) $\llbracket \text{John} \rrbracket := john'$ $\llbracket \text{met} \rrbracket := \lambda x \lambda y. met'(x)(y)$
 $\llbracket \text{linguists} \rrbracket := \lambda x. linguist'(x)$ $\llbracket \text{exactly three} \rrbracket := \lambda P \lambda Q. |\{x : P(x)\} \cap \{x : Q(x)\}| = 3$

For comparison, fill in the semantics of the following CCG derivation for the same sentence:

<u>John</u>				
NP				
$john'$				
$S/(S \setminus NP)$	T	<u>met</u>	<u>exactly three</u>	<u>linguists</u>
...		$(S \setminus NP)/NP$	$(S \setminus (S/NP))/N$	N
		$\lambda x \lambda y. met'(x)(y)$	$\lambda P \lambda Q. \{x : P(x)\} \cap \{x : Q(x)\} = 3$	$\lambda x. linguist'(x)$
		S/NP	$S \setminus (S/NP)$	
		
S				
...				

D Quantifier scope ambiguities (5 points)

Give two distinct H&K-style derivations for the string of words in (12), using the lexical semantic values in (14). One derivation should end up pairing it with the meaning in (13a), and the other should end up pairing it with the meaning in (13b).

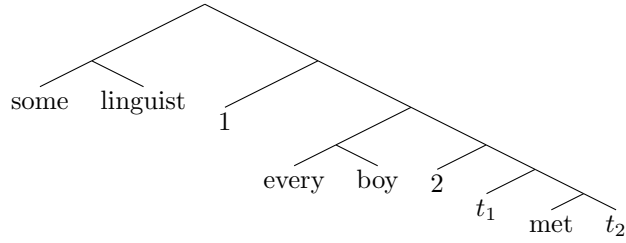
(12) Some linguist met every boy

(13) a. $\exists x[linguist'(x) \wedge \forall y[boy'(y) \rightarrow met'(y)(x)]]$
 b. $\forall y[boy'(y) \rightarrow \exists x[linguist'(x) \wedge met'(y)(x)]]$

(14) $\llbracket \text{linguist} \rrbracket := \lambda x. linguist'(x)$ $\llbracket \text{some} \rrbracket := \lambda P \lambda Q. \exists x[P(x) \wedge Q(x)]$
 $\llbracket \text{boy} \rrbracket := \lambda x. boy'(x)$ $\llbracket \text{every} \rrbracket := \lambda P \lambda Q. \forall x[P(x) \rightarrow Q(x)]$
 $\llbracket \text{met} \rrbracket := \lambda x \lambda y. met'(x)(y)$

One of the derivations will use the tree in (15); the other will use a different tree.

(15)



Extra: (Not for extra course credit, just for the fun and satisfaction.) We've written the semantic value of 'every' and 'some' using logical symbols like \forall and \exists and so on, without talking about sets; when talking about conservativity, and for 'exactly three' above, we used sets. We can also write the semantics for 'every' and 'some' using sets. Can we write the semantics for 'exactly three' (or 'exactly one' or 'exactly two') using logical symbols like \forall and \exists and so on? How about 'most'?