

LING419 Homework 2

Categorial Grammar

When you give derivations, label each step with the rule you're using to make that step. You can use whatever notation you want for this, as long as it's understandable, but the simplest might be to adopt Steedman's: $<$ and $>$ for function application, $>B$ and $<B$ for function composition, and T for type-raising.

A Semantics for CONJ (4 points)

Recall from our discussion in class that given a system of logic with the four rules in (1), it makes sense to adopt the meanings in (2) for the English words 'and' and 'or'.

$$(1) \quad \frac{A \star B}{A} \quad \frac{A \star B}{B} \quad \frac{A}{A \sim B} \quad \frac{B}{A \sim B} \quad (2) \quad \text{'and': } \lambda p \lambda q . p \star q \quad \text{'or': } \lambda p \lambda q . p \sim q$$

For example, the last step in the derivation of 'John walks or Mary sleeps' would be as in (3), using the rule on the right.

$$(3) \quad \boxed{\begin{array}{c} \frac{\frac{\text{John walks}}{S} \quad \frac{\text{or}}{\text{CONJ}} \quad \frac{\text{Mary sleeps}}{S}}{\text{walks}'(john') \quad \lambda p \lambda q . p \sim q \quad \text{sleeps}'(mary')}_{\Phi} \\ \frac{S}{\text{walks}'(john') \sim \text{sleeps}'(mary')} \end{array}} \quad \boxed{\begin{array}{c} \frac{S \quad \text{CONJ} \quad S}{m \quad f \quad n}_{\Phi} \\ \frac{S}{f(m)(n)} \end{array}}$$

Now suppose there is another symbol, namely ' \times ', which can be used to glue two "sentences of logic" together just like ' \star ' and ' \sim ' can. The rule for making inferences involving this new symbol is given in (4). Effectively, it says that if we know that $A \times B$ is true, and we also know that B is true, then we can infer that A is true.

$$(4) \quad \frac{A \times B}{\frac{B}{A}}$$

Let's treat 'only if' as just a single word, with category CONJ. Assuming the role of 'only if' in inferences is as illustrated in (5), which of the two meanings in (6) should we adopt?

$$(5) \quad \frac{\text{Bill laughs only if Fred jumps}}{\frac{\text{Bill laughs}}{\text{Fred jumps}}}$$

$$(6) \quad \text{'only if': } \lambda p \lambda q . p \times q \quad \text{'only if': } \lambda p \lambda q . q \times p$$

Explain why. Include in your explanation: (i) a derivation showing the semantics of 'John walks only if Mary sleeps', and (ii) an example of an inference involving this English sentence.

B Quantifier scope ambiguities (6 points)

Show two distinct derivations, including semantics, for the string of words in (7). One derivation should end up pairing it with the meaning in (8a),¹ and the other should end up pairing it with the meaning in (8b).

- (7) Some witness said that John knew every victim.
- (8) a. $\exists x[\textit{witness}'(x) \wedge \forall y[\textit{victim}'(y) \rightarrow \textit{said}'(\textit{knew}'(y)(\textit{john}'))(x)]]$
 b. $\forall y[\textit{victim}'(y) \rightarrow \exists x[\textit{witness}'(x) \wedge \textit{said}'(\textit{knew}'(y)(\textit{john}'))(x)]]$

(If you're having trouble getting your mind around the two different meanings, imagine John is accused of murdering a number of people and the issue at hand is whether or not he knew all of them. Perhaps there was a single witness who can settle the question (8a), or perhaps we need to put together the testimonies of a number of different witnesses to establish that John did indeed know all the victims (8b).)

Here's the starting point for deriving the meaning in (8a):

some witness	said that	John	knew	every victim
$S/(S\backslash NP)$	$(S\backslash NP)/S$	NP	$(S\backslash NP)/NP$	$(S\backslash NP)\backslash((S\backslash NP)/NP)$
$\lambda p.\exists x[\textit{witness}'(x) \wedge p(x)]$	$\lambda s\lambda x.\textit{said}'(s)(x)$	\textit{john}'	$\lambda y\lambda x.\textit{knew}'(y)(x)$	$\lambda q\lambda z.\forall y[\textit{victim}'(y) \rightarrow q(y)(z)]$

Here's the starting point for deriving the meaning in (8b):

some witness	said that	John	knew	every victim
$S/(S\backslash NP)$	$(S\backslash NP)/S$	NP	$(S\backslash NP)/NP$	$S\backslash(S/NP)$
$\lambda p.\exists x[\textit{witness}'(x) \wedge p(x)]$	$\lambda s\lambda x.\textit{said}'(s)(x)$	\textit{john}'	$\lambda y\lambda x.\textit{knew}'(y)(x)$	$\lambda q.\forall y[\textit{victim}'(y) \rightarrow q(y)]$

Hints: In both cases you'll need to type-raise 'John'. You might find that it helps to abbreviate $S\backslash NP$ as VP, at least sometimes. If you're not sure how to get started, try getting two different meanings for 'Some witness saw every victim'; then try to put 'said that John knew' together into something with the same category as 'saw'.

C Possible and impossible languages (3 points)

From Steedman 1996, chapter 3, pages 53–54:

A subject extraction like (17b) would require the addition of the “forward crossing” composition rule $>B_{\times} \dots$. Although such rules are permitted by the theory, we could not use this rule to devise a grammar for a language identical to English, differing only in freely allowing subject extractions.

Explain, in your own words, what he means by this. What *prediction* about the range of existing human languages — or about the “possible” and “impossible” human languages — is the theory making? What would be an example of an *empirical finding* that would be evidence against this theory of grammar? (I'm not asking about any actual facts here, just the *kind of thing* that one would look for, eg. “This theory predicts that blickets will never be able to snarfle; so if we ever look out in the world and find some blickets that are snarfing, that would be evidence against the theory.”)

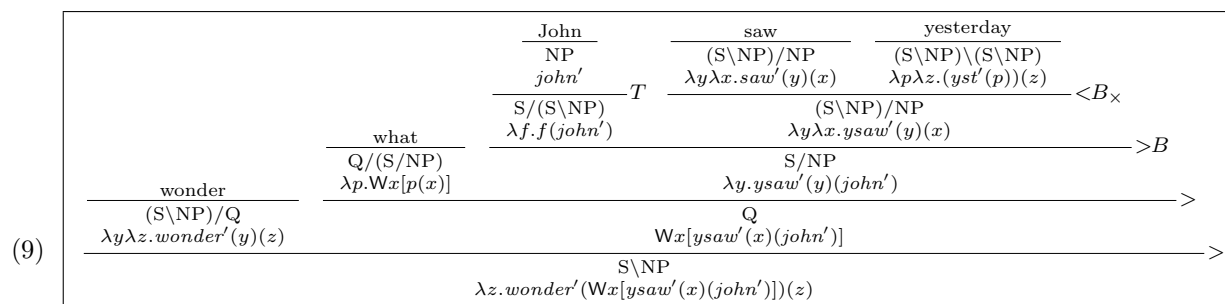
¹Or, this other meaning which is equivalent (for us) to (8a): $\exists x[\textit{witness}'(x) \wedge \textit{said}'(\forall y[\textit{victim}'(y) \rightarrow \textit{knew}'(y)(\textit{john}')])(x)]$. This comes about if we go down the route of putting together 'John knew every victim' into an S. This can be done actually from either of the two starting points given; type-raising John will only be required with the second (less complex) of the two starting points. Try it out if you're curious.

Extra: (Not for extra course credit, just for the fun and satisfaction.) Show how it would be possible to derive ‘I think Keats that likes Chapman’ (see p.54) if we *did* include the $>B_x$ rule in the grammar of English. Assume that ‘that’ has category S/S. Don’t worry about showing the semantics of this derivation.

D English and Whinglish (7 points)

(Embedded) Questions in English

Let’s say that the relevant part of the derivation of an English sentence like ‘I wonder what John saw yesterday’ goes along the lines of the following. (*ysaw’* is an abbreviation for the meaning of ‘saw yesterday’, which we can think of as just a sort of big transitive verb. Don’t worry about the internals of that first step if it’s unclear.)



What’s happened here is we’ve built up something that’s looking to combine with an NP to its left; in the final semantics, the entity denoted by that NP will be the one doing the wondering (it will fill the slot indicated by the λz), and “the thing being wondered” is:

$$\text{Wx}[\text{ysaw}'(x)(\text{john'})]$$

which is the semantic value of the embedded question² ‘what John saw yesterday’. Think of that mysterious W thing I’ve invented as roughly parallel to \forall and \exists :

$\forall x[\text{ysaw}'(x)(\text{john'})]$	For all x , John saw x yesterday
$\exists x[\text{ysaw}'(x)(\text{john'})]$	For some x , John saw x yesterday
$\text{Wx}[\text{ysaw}'(x)(\text{john'})]$	For which x , John saw x yesterday

In English, that semantic value for the embedded question comes about when ‘what’ combines with something of type S/NP (i.e. a sentence still missing one of its NPs) which has as semantic value *the property that the answer to the question must have*. In our example, this is the property of being seen by John yesterday, or $\lambda y. \text{ysaw}'(y)(\text{john'})$.

(Embedded) Questions in Whinglish

Now imagine a language just like English, except that it is “*wh*-in-situ”: words like ‘who’ and ‘what’ appear in the same position as a “normal” (non-questioning) argument would. Call this language Whinglish. The Whinglish equivalent of the sentence above, repeated in (10a), uses the word order shown in (10b).

²We’re restricting attention to embedded questions, rather than matrix questions like ‘What did John see yesterday’, so that we don’t have to worry about subject-aux inversion and do-support. But nothing significant hinges on this.

- (10) a. I wonder what John saw yesterday
 b. I WONDER JOHN SAW WHAT YESTERDAY

So the word order of the Whinglish embedded question ‘JOHN SAW WHAT YESTERDAY’ is exactly the same as the word order of a corresponding Whinglish sentence (and the same as a corresponding English sentence for that matter), such as ‘JOHN SAW MARY YESTERDAY’.

This raises the question of how our theory of grammar is going to account for these two languages. The English sentence in (10a) and the Whinglish sentence in (10b) both need to end up with the same semantic value, since they mean the same thing, namely:

$$wonder'(Wx[ysaw'(x)(john')])(i')$$

Having seen above how we can get this semantic value for the English sentence, two linguists, Christopher Ategorial and Terrence Ransformational, propose two different ways to account for the Whinglish sentence.

Chris Ategorial’s analysis

Chris figures that since in Whinglish embedded questions have word order just like embedded sentences do in both English and Whinglish, we may as well try to treat them in the same way. First he scribbles down how things would work for a simple embedded sentence, something like ‘I THINK JOHN SAW MARY YESTERDAY’:

		SAW	MARY	
		(S\NP)/NP	NP	
		$\lambda y \lambda x. saw'(y)(x)$	$mary'$	
		S\NP		>
		$\lambda x. saw'(mary')(x)$		<
	JOHN	S\NP		
	NP	$\lambda x. ysaw'(mary')(x)$		
	$john'$	S		<
THINK	S		>	
(S\NP)/S	$ysaw'(mary')(john')$		<	
$\lambda y \lambda z. think'(y)(z)$	S\NP		>	
	$\lambda z. think'(ysaw'(mary')(john'))(z)$			

That’s pretty simple, and doesn’t even require anything fancier than function application. If we can keep things looking like that, we’ll be doing well. In the embedded question, ‘WHAT’ seems to be in the position corresponding to ‘MARY’, so Chris supposes that ‘WHAT’ is an NP in Whinglish; similarly, ‘WONDER’ seems to be in the position corresponding to ‘THINK’, so Chris supposes that ‘WONDER’ is an (S\NP)/S in Whinglish.

- (11) a. I THINK JOHN SAW MARY YESTERDAY
 b. I WONDER JOHN SAW WHAT YESTERDAY

Putting semantics aside for now, Chris writes the syntax out for the embedded question using these new categories he’s figured out:

		SAW	WHAT	
		(S\NP)/NP	NP	
		S\NP		>
		(S\NP)\(S\NP)		<
	JOHN	S\NP		
	NP	S		<
WONDER	S		>	
(S\NP)/S	S\NP			

Chris now wonders how he’s going to get the weird W thing into the semantics of the sentence. It seems pretty reasonable to suppose that this comes from the word ‘WONDER’; it doesn’t seem to make sense to

suppose that it comes from the word ‘WHAT’ in Whinglish, because that has category NP and so it just denotes some simple entity, and it can’t come from any other word, because all the other words in (11b) are also in (11a), but the meaning of (11a) doesn’t have a W anywhere in it. So Chris decides that the semantic value for ‘WONDER’ must look something like this:

$$\lambda s \lambda z. wonder'(W\dots s\dots)(z)$$

The thing of category S (the embedded question, basically) that it combines with first will somehow link up with the W to make up “the thing wondered”; and then it will combine with a subject NP, as usual.

What the embedded question really has to contribute is a *property*, just like the chunk ‘John saw yesterday’ in the English example contributed the property of “being seen by John yesterday”, or $\lambda y. ysaw'(y)(john')$. But the Whinglish chunk ‘JOHN SAW WHAT YESTERDAY’ is of type S, just like ‘JOHN SAW MARY YESTERDAY’, so it’s a fully-satisfied statement or proposition, with no empty slots. It’s going to have to look something like this:

	$\frac{\text{SAW}}{(S \setminus NP) / NP}$ $\lambda y \lambda x. saw'(y)(x)$	$\frac{\text{WHAT}}{NP}$ $what'$	>	$\frac{\text{YESTERDAY}}{(S \setminus NP) \setminus (S \setminus NP)}$ $\lambda p \lambda z. (yst'(p))(z)$	<
	$\frac{\text{JOHN}}{NP}$ $john'$	$\frac{S \setminus NP}{\lambda x. saw'(what')(x)}$		$\frac{S \setminus NP}{\lambda x. ysaw'(what')(x)}$	<
$\frac{\text{WONDER}}{(S \setminus NP) / S}$ $\lambda s \lambda z. wonder'(W\dots s\dots)(z)$		$\frac{S}{ysaw'(what')(john')}$			<
		$\frac{S \setminus NP}{??}$			>

Chris thinks up a way around this though: we can sort of understand $ysaw'(what')(john')$ as semantically contributing a property, despite being an S, because it has this special value $what'$ in it. We can understand chunks like this as representing the property that something in the position of $what'$ is stated to have: so $ysaw'(what')(john')$ contributes the property of being seen by John yesterday, whereas $ysaw'(mary')(what')$ would contribute the property of seeing Mary yesterday.

To get this to work out, all Chris needs to do is have the W in ‘WONDER’ associate itself with $what'$. So if ‘WONDER’ combines with $ysaw'(what')(john')$, the result is about wondering “for which x , John saw x yesterday”, and if ‘WONDER’ combines with $ysaw'(mary')(what')$, the result is about wondering “for which x , x saw Mary yesterday”. And now that $what'$ is going to behave like this, being associated with the W, Chris realises that it’s actually a variable (like x and y), not a constant (like $john'$ and $mary'$), so he just writes it as w instead.

	$\frac{\text{SAW}}{(S \setminus NP) / NP}$ $\lambda y \lambda x. saw'(y)(x)$	$\frac{\text{WHAT}}{NP}$ w	>	$\frac{\text{YESTERDAY}}{(S \setminus NP) \setminus (S \setminus NP)}$ $\lambda p \lambda z. (yst'(p))(z)$	<
	$\frac{\text{JOHN}}{NP}$ $john'$	$\frac{S \setminus NP}{\lambda x. saw'(w)(x)}$		$\frac{S \setminus NP}{\lambda x. ysaw'(w)(x)}$	<
$\frac{\text{WONDER}}{(S \setminus NP) / S}$ $\lambda s \lambda z. wonder'(Ww[s])(z)$		$\frac{S}{ysaw'(w)(john')}$			<
		$\frac{S \setminus NP}{\lambda z. wonder'(Ww[ysaw'(w)(john')])}(z)$			>

(12)

Chris thinks this a pretty good analysis. It seems pretty clear that the two sentences in (11) have almost exactly the same structure, so he’s happy to have found a way to get their different semantics to come out correctly while still having the overall structure basically the same. The only trick is to get ‘WHAT’ to introduce a variable, that the W inside ‘WONDER’ is automatically associated with, rather than a constant

— but we were already making use of both variables and constants anyway, so he doesn't think this is much of a problem. There's no avoiding the fact that we're going to have to do *something* a bit differently from the way things work in (9) to deal with these Whinglish embedded questions, since the S chunk 'JOHN SAW WHAT YESTERDAY' has to tell 'WONDER' which *property* the answer to the question must have.

Chris is very curious to see if his rival Terry can think up anything as elegant as this.

Terry Ransformational's analysis

Terry doesn't really like the idea of having the S chunk 'JOHN SAW WHAT YESTERDAY' indicating a property. He wants properties to be indicated by chunks with category S/NP or S\NP, as usual, which worked fine for English in (9). But 'JOHN SAW WHAT YESTERDAY' sure looks like an S.

Terry has read syntax books where they talked about *wh*-words in English "moving" from one position to another: from the position straight after 'saw' in this case, to the position where they're pronounced at the front of the sentence. (Chris has always thought that these books were pretty ridiculous.) In Whinglish, these *wh*-words don't seem to have moved anywhere. They get pronounced in the position that, according to those syntax books, they start in and then move out of in English.

This gives Terry an idea. He proposes that, even though the Whinglish sentence is pronounced with a different word order from the English version, the *wh*-words have sort of moved in Whinglish too — they've moved invisibly, in some way that affects semantics but not word order, or something — so the sentences in the two languages look exactly the same for the purposes of semantics. So he doesn't have to deal with the problem of the S chunk 'JOHN SAW WHAT YESTERDAY' indicating a property, because he just puts the 'WHAT' where it is in English before doing his semantics, and so there's an S/NP chunk representing the property just like there was in English in (9). This way Terry's account for the Whinglish sentence just looks exactly the same as (9).

(13)

		JOHN	SAW	YESTERDAY	
		NP	(S\NP)/NP	(S\NP)\(S\NP)	
		<i>john'</i>	$\lambda y \lambda x. saw'(y)(x)$	$\lambda p \lambda z. (yest'(p))(z)$	$<B_x$
		S/(S\NP)	(S\NP)/NP		
		$\lambda f.f(john')$	$\lambda y \lambda x. ysaw'(y)(x)$		$>B$
	WHAT		S/NP		
	Q/(S/NP)		$\lambda y. ysaw'(y)(john')$		
	$\lambda p.Wx[p(x)]$				
WONDER			Q		$>$
(S\NP)/Q			$Wx[ysaw'(x)(john')]$		
$\lambda y \lambda z. wonder'(y)(z)$					$>$
		S\NP			
		$\lambda z. wonder'(Wx[ysaw'(x)(john')]) (z)$			

Chris thinks this is quite comical. The stuff about *wh*-words moving in English was already crazy enough, but at least in those cases they were meant to end up in the position where we hear them pronounced. "But in Whinglish, the movement is even spookier — it's 'invisible', or something!" Chris chuckles.

Your task

Think of the two analyses of Whinglish embedded questions as two extensions of the basic theory that we've been discussing so far this semester, which accounted neatly for the English sentence in (9).

- a. According to Chris, what is the central difference between the way English embedded questions work and the way Whinglish embedded questions work? What "new stuff" did he need to add to the theory for his analysis?

- b. According to Terry, what is the central difference between the way English embedded questions work and the way Whinglish embedded questions work? What “new stuff” did he need to add to the theory for his analysis?
- c. Based on just what we know about Whinglish so far, do we have reason to prefer one extension over the other? Why or why not — and if so, which one? Which of the two extensions do you think has added “more new stuff” to the theory? Why?
- d. Can you imagine any new discoveries about Whinglish or English that would count as evidence in favour of either of the two competing extensions? (You don’t need to be specific here, just think about the *kind of thing* that we might look for to help us decide which version to prefer.) Or is there really no reason we could ever have to prefer one over the other?