Homework 2: Semantics

LING 5001 Introduction to Linguistics

Due on October 19th

1 Compositionality

Below is a sample grammar of English, including the CFG rules to generate the structures below and the semantic values of each of the lexical items. For simplicity's sake, we will treat certain phrases as one single word with its own semantic value, even though we know that they are composed of further components.

(1) **CFG Rules:**

a. S $\rightarrow$ NP VP
b. S $\rightarrow$ Neg S
c. Neg $\rightarrow$ It's-not-the-case-that
d. NP $\rightarrow$ Dale
e. NP $\rightarrow$ Shelly
f. NP $\rightarrow$ the-pie
g. NP $\rightarrow$ D N
h. D $\rightarrow$ every
i. N $\rightarrow$ agent
j. VP $\rightarrow$ was-delicious
k. VP $\rightarrow$ V NP
l. V $\rightarrow$ made
m. V $\rightarrow$ loves

(2) **Semantic Values:**

a. $[\![$It's-not-the-case-that$]\!] = \lambda p.\neg p$[1]
b. $[\![$Dale$]\!] =$ **dale**
c. $[\![$Shelly$]\!] =$ **shelly**
d. $[\![$the-pie$]\!] =$ **pie**
e. $[\![$every$]\!] = \lambda P.\lambda Q.\forall x[P(x) \rightarrow Q(x)]$[2]
f. $[\![$agent$]\!] = \lambda x.agent(x)$
g. $[\![$was-delicious$]\!] = \lambda x.delicious(x)$

[1]This is read as 'it is not the case that p. This is a function of type $\langle t, t\rangle$; i.e., it accepts a truth-value, and returns a truth value. Function application will take an entire formula and rewrite in place of p to the right of the negation symbol $\neg$.

[2]This is read as 'for all x, if x is P, then x is Q. This is type $\langle\langle e, t\rangle, \langle\langle e, t\rangle, t\rangle\rangle$. In other words, it accepts a function of type $\langle e, t\rangle$, then outputs a function of type $\langle\langle e, t\rangle, t\rangle$; This output function than accepts a function of type $\langle e, t\rangle$ and outputs a truth value.

h. $[\![\text{made}]\!] = \lambda x.\lambda y.make(y, x)$
i. $[\![\text{loves}]\!] = \lambda x.\lambda y.loves(y, x)$

Here is a sentence that this grammar generates, along with the semantic value of each node. Remember that *function application* applies each time two phrases combine in the syntax:

(3) $[\![\text{S}]\!] = love(\textbf{dale}, \textbf{pie})$

$[\![\text{NP}]\!] = \textbf{dale}$ — Dale

$[\![\text{VP}]\!] = \lambda y.love(y, \textbf{pie})$

$[\![\text{V}]\!] = \lambda x.\lambda y.love(y, x)$ — loves

$[\![\text{NP}]\!] = \textbf{pie}$ — the pie

By standard rules of inference, we know can show that, for instance, this sentence entails that *Somebody loves the pie* ($\exists x[love(x, \textbf{pie})]$).

A.) (40 points). For sentence, give the syntactic tree that the CFG above generates, and the semantic value of each node. Remember that the CFG above simplifies some things, i.e., VP can expand to the single word *was-delicious*.

a. The-pie was-delicious
b. Shelly made the-pie
c. Every agent loves the-pie
d. It's-not-the-case-that Dale made the-pie

2 Semantics of Adjectives

In class, we modeled the meaning of adjectives as functions of type $\langle e, t\rangle$, which could combine with nouns (also type $\langle e, t\rangle$) to create a composed function of type $\langle e, t\rangle$ with the predicates conjoined:

(4) a. $[\![\text{grey}]\!] = \lambda x.grey(x)$
b. $[\![\text{cat}]\!] = \lambda x.cat(x)$
c. $[\![[_{\text{NP}}\ \text{grey cat}]]\!] = \lambda x.grey(x)\ \&\ cat(x)$

In other words, the meaning of *grey cat* takes an entity, and gives you **true** just in case that entity is both in the set of grey things and in the set of cat things:

(5) $grey(x)\ \&\ cat(x) = \textbf{true}$ iff $x \in [\![grey]\!] \cap [\![cat]\!]$

This models the fact that, e.g., *Ernie is a grey cat* entails *Ernie is grey* and *Ernie is a cat.* The name of this inference rule is **conjunctin reduction**. Hooray semantics!

B.) (15 points). Let's consider the following two sentences:

(6) a. Zirconium is a fake diamond.
b. Obama is a former president.

Unlike *grey cat*, the meaning of *fake diamond* and *former president* cannot be modeled as the conjunction of the two predicates, i.e., $\lambda x.fake(x)$ & $diamond(x)$, and $\lambda x.former(x)$ & $president(x)$. Explain why these are inappropriate analyses for the meanings of these NPs. (**Hint:** Do the same patterns of entailment apply for *fake diamond* and *former president* as *grey cat*?)

Logicians have typically been interested in the way that sets of premises can license inferences. For instance:

(7) a. Ernie is grey
b. Ernie is a cat
c. Therefore, Ernie is a grey cat.

That is, if I assent to the proposition *Ernie is grey* and the proposition *Ernie is a cat*, then I can infer that *Ernie is a grey cat.* This is "reversing" the same kind of inference that we make when we infer *Ernie is grey* from *Ernie is a grey cat.*

However, the following inference seems ill-formed:

(8) a. France is a hexagon.
b. France is a republic.
c. #Therefore, France is a hexagonal republic.

C.) (15 points). How might we explain why the inference in (8) is ill-formed, but the inference in (7) is well-formed? (**Hint:** Maybe $[\![\text{France}]\!] = \textbf{france}$ is an inadequate analysis. Does *France* serve the same function in the meaning of these sentences?)

3 Semantic Types of Verbs

We have been modeling the meaning of verbs as either the set of entities that perform the action, or the set of relations:

a. $[\![meow]\!] = \lambda x.meow(x) = \textbf{true}$ iff $x \in \{\textbf{ernie}, \textbf{linus}, \ldots\}$
b. $[\![like]\!] = \lambda x.\lambda y.like(y, x) = \textbf{true}$ iff $\langle y, x\rangle \in \{\langle \textbf{dustin}, \textbf{ernie}\rangle, \langle \textbf{chomsky}, \textbf{linguistics}\rangle, \ldots\}$

Next, examine the following set of sentences:

(9) a. Dale ate the pie
b. Dale ate the pie with the fork
c. Dale ate the pie with the fork at the diner
d. Dale at the pie with the fork at the diner with Harry
e. Dale ate the pie with the fork at the diner with Harry on Saturday

The grammar allows us to keep stacking adjuncts onto the VP, and each adjunct ends up adding a new entity to the relation that the verb denotes. That is, the putative semantic representation for these sentences are as below:

(10) a. $ate(\textbf{dale}, \textbf{pie})$
b. $ate(\textbf{dale}, \textbf{pie}, \textbf{fork})$

c. $ate(\textbf{dale}, \textbf{pie}, \textbf{fork}, \textbf{diner})$
d. $ate(\textbf{dale}, \textbf{pie}, \textbf{fork}, \textbf{diner}, \textbf{harry})$
e. $ate(\textbf{dale}, \textbf{pie}, \textbf{fork}, \textbf{diner}, \textbf{harry}, \textbf{saturday})$

D.) (10 points) This analysis makes it impossible to determine what the semantic value of the word *eat* is. Explain what the problem is. (**Hint:** What kind of function is $[\![\text{eat}]\!]$?)

Perhaps more importantly, this analysis does not capture the entailment relationships between the sentences above. In (10), the sentences with more adjuncts entail the sentences with fewer adjuncts, i.e., *Dale ate the pie with the fork at the diner* entails *Dale ate the pie with the fork*, which in turn entails *Dale ate the pie*. However, postulating indefinitely many sets of eating relations does not guarantee that these entailments will be captured.

To fix this, semanticists have postulated that verbs and VP adjuncts are predicates of an event variable instead of a relation. That is, the meaning of the verb *eat* is something more like $\lambda x.\lambda y.\lambda e.eat(y, x, e)$, i.e., true just in case the event e is an event of Dale eating the pie. If so, then VP adjuncts combine with this predicate much in the same way that adjectives combine with nouns. Thus, the meaning of the sentences above are:

(11) a. $\exists e[eat(\textbf{dale}, \textbf{pie}, e)]$
b. $\exists e[eat(\textbf{dale}, \textbf{pie}, e)\ \&\ with(e, \textbf{fork})]$
c. $\exists e[eat(\textbf{dale}, \textbf{pie}, e)\ \&\ with(e, \textbf{fork})\ \&\ at(e, \textbf{diner})] \ldots$

Thus, the meaning for (11-c) would be something like "there is an event e, such that e is an event of Dale eating the pie, and e was with a fork, and e was at the diner."

E.) (10 points) Explain how the "event analysis" can capture the entailment facts in (10).

F.) (10 points) Do you have any further questions about semantics?