1 Overview

• This handout describes our core theory of semantic composition. It’s a bare-bones theory, but still powerful in the sense that generalizing it to a wider range of sentences is straightforward.

• The theory is guaranteed compositional even by the strictest interpretation of the principle.

• The most important conceptual move is to interpret lexical items as sets and functions. Meanings might not literally be sets and functions, but, following Lewis's advice, we're hypothesizing that at least these do what meanings do.

• We need a bunch of rules in order to respect the syntactic structures. However, there are just a few rules of semantic composition and they are very simple.

• So, in essence, if you know the lexical meanings of your language and you can put them together according to the syntactic rules, then the only other concepts you need to be a full-fledged interpreter are a few simple semantic composition rules.

• The entire grammar presented here is implemented in very simple Python code here: https://web.stanford.edu/class/linguist130a/materials/semgrammar130a.py

2 Notation for describing functions

2.1 Functions

\[ \text{IS\_EVEN}(x) = \begin{cases} \text{T} & x \mod 2 = 0 \\ \text{F} & x \mod 2 \neq 0 \end{cases} \]

\[ \lambda x (\text{T if } x \mod 2 = 0 \text{ else } \text{F}) \]

2.2 Function application

\[ \text{IS\_EVEN}(1) = \text{F} \]

\[ \begin{pmatrix} 0 \rightarrow \text{T} \\ 1 \rightarrow \text{F} \\ 2 \rightarrow \text{T} \\ 3 \rightarrow \text{F} \\ \vdots \end{pmatrix} (1) = \text{F} \]

\[ \left( \lambda x (\text{T if } x \mod 2 = 0, \text{ else } \text{F}) \right)(1) \]

\[ (\text{T if } 1 \mod 2 = 0, \text{ else } \text{F}) \text{ } \text{F} \]
3 Basic semantic objects

3.1 Truth values

T for truth and F for falsity.

3.2 Universe

The set of entities in our tiny possible world:

\[ U = \{ \text{Maggie}, \text{Lisa}, \text{Bart}, \text{Homer} \} \]

Apologies to Marge! Her hair is so tall that she would make this handout very long!

4 Semantic lexicon

4.1 PNs

Proper names are directly referential:

- \([\text{Maggie}] = \text{Maggie}\)
- \([\text{Lisa}] = \text{Lisa}\)
- \([\text{Bart}] = \text{Bart}\)
- \([\text{Homer}] = \text{Homer}\)

4.2 Ns

Nouns denote sets of entities (subsets of \(U\)):

1. \([\text{Simpson}] = \{ \text{Maggie}, \text{Lisa}, \text{Bart}, \text{Homer} \}\)
2. \([\text{child}] = \{ \text{Maggie}, \text{Lisa}, \text{Bart} \}\)
3. \([\text{student}] = \{ \text{Bart} \}\)
4. \([\text{parent}] = \{ \text{Homer} \}\)
4.3 Intransitive Vs

Intransitive verbs also denote sets of entities (subsets of $U$):

\[(5) \quad [\text{skateboards}] = \{\text{Bart, Maggie}\}\]

\[(6) \quad [\text{studies}] = \{\text{Bart}\}\]

\[(7) \quad [\text{introspects}] = \{\text{Bart, Maggie}\}\]

\[(8) \quad [\text{speaks}] = \{\text{Bart, Maggie}\}\]

In our grammar, intransitive Vs will combine with the subject of the sentence to produce a truth-valued claim by testing whether the subject’s denotation is a member of the verb’s denotation.

```
Bart  skateboards

Maggie  skateboards
```
4.4 Transitive Vs

Transitive verbs denote functions from entities into sets of entities:

\[
\text{teases} = \lambda y \begin{cases} 
\text{if } y = \text{Bart, Lisa, and Homer all tease each other.} \\
\text{if } y = \text{Maggie neither teases nor is teased. Note: the object comes in first as } y, \text{ so the return values are the people who tease } y. \\
\text{if } y = \text{Bart} \\
\text{if } y = \text{Lisa, and Homer are teased by each other.} \\
\end{cases}
\]

\[
\text{admires} = \lambda y \begin{cases} 
\text{if } y = \text{Everyone admires only Lisa and Maggie, but no one admires themselves.} \\
\text{if } y = \text{Bart only admires Lisa and Maggie.} \\
\text{if } y = \text{Lisa and Maggie are not admired by anyone.} \\
\end{cases}
\]

Important insight: once a transitive V combines with its object, it denotes a set of entities – semantically, it’s just like an intransitive verb.
4.5 Adjectives

Adjectives combine with noun meanings to produce new noun meanings. The core of it is this very basic constituent structure:

You can see that we’re treating the following as intersective adjectives:

\( \text{[scholarly]} = \lambda X \left( \left\{ \begin{array}{c} \text{scholar} \, , \\
\text{child} \end{array} \right\} \cap X \right) \)

\( \text{[distractible]} = \lambda X \left( \left\{ \begin{array}{c} \text{distract} \, , \\
\text{child} \end{array} \right\} \cap X \right) \)

\( \text{[hungry]} = \lambda X \left( \left\{ \begin{array}{c} \text{hungry} \, , \\
\text{child} \end{array} \right\} \cap X \right) \)

\( \text{[Springfieldian]} = \)

The same semantic types work for the other adjective types, but they don’t use \( \cap \) or commit to the incoming \( X \) being true of the entities in the resulting set:

\( \text{[alleged]} = \lambda X : \{ y \in U : \text{someone claimed that } y \in X \} \)

Basic semantic composition:

\( \text{[scholarly(child)]} = \text{[scholarly]} \left( \text{[child]} \right) = \left\{ \begin{array}{c} \text{scholar} \, , \\
\text{child} \end{array} \right\} \cap \left\{ \begin{array}{c} \text{scholar} \, , \\
\text{child} \end{array} \right\} \)

\( \text{[hungry(scholarly(child))]} = \)
4.6 Negation

We would like a negation that operates on verb phrases like *skateboards* and *admires Maggie*:

(20) \[ [\text{never}] = \lambda X \left( \right) \]

4.7 Quantificational determiners

\begin{align*}
&\text{every} & &\text{child} & &\text{studies} \\
&\text{determiner} & &\text{restriction} & &\text{scope}
\end{align*}

(21) \[ [\text{every}] = \lambda X \left( \lambda Y \left( \text{T if } X \subseteq Y, \text{ else } F \right) \right) \]

(22) \[ [\text{every} (\text{child})] = \lambda Y \left( \text{T if } [\text{child}] \subseteq Y, \text{ else } F \right) \]

(23) \[ [\text{every} (\text{child}) (\text{studies})] = \text{T if } [\text{child}] \subseteq [\text{studies}], \text{ else } F \]

(24) \[ [\text{some}] = \lambda X \left( \lambda Y \left( \text{T if } X \cap Y \neq \emptyset, \text{ else } F \right) \right) \]

(25) \[ [\text{no}] = \lambda X \left( \lambda Y \left( \text{T if } X \cap Y = \emptyset, \text{ else } F \right) \right) \]

(26) \[ [\text{at least three}] = \lambda X \left( \lambda Y \left( \text{T if } |X \cap Y| \geq 3, \text{ else } F \right) \right) \]

(27) \[ [\text{at most three}] = \lambda X \left( \lambda Y \left( \text{T if } \frac{|X \cap Y|}{|X|} > \frac{1}{2}, \text{ else } F \right) \right) \]

(28) \[ [\text{most}] = \lambda X \left( \lambda Y \left( \text{T if } \frac{|X \cap Y|}{|X|} > \frac{1}{2}, \text{ else } F \right) \right) \]

(29) \[ [\text{between five and ten}] = \]

6
An imagined dialogue about quantificational determiners

(30) You defined quantificational determiners as denoting relations between sets of entities. Isn’t that too complicated?

It is complicated, but it’s not too complicated! It’s the least complicated thing we could do! We really need the determiner to control both its restriction and its scope:

(31) But couldn’t every just denote the universe \( U \)?

No way! We need to consider the role of the restriction: every child, every scholarly parent, and so forth.

(32) Ok, then let’s say that every student picks out the set of students, and every parent the set of parents, and so forth. That would at least be somewhat simpler.

That still won’t work! Suppose \([\text{every student}]\) was the set of students, for example. What would we do about the verb phrase? We need \([\text{every student skateboards}]\) to be false and \([\text{every student speaks}]\) to be true. What are the criteria for making that distinction?

(33) The criteria could be subset, as you gave it. \([\text{every student skateboards}]\) is \( F \) because the set of students is not a subset of the set of skateboarders, but \([\text{every student speaks}]\) is \( T \) because the set of students is a subset of the set of things that speak. That’s just like “if \( x \) is a student, then \( x \) skateboards”. It seems intuitively correct.

Exactly! But that’s just a rephrasing of the analysis we gave. You start with

\[
T \text{ if } [\text{student}] \subseteq Y, \text{ else } F
\]

We explicitly bind the variable \( Y \), as in

\[
\lambda Y \left( T \text{ if } [\text{student}] \subseteq Y, \text{ else } F \right)
\]

This is intuitively a set of sets. That captures the variation we just noted in truth values for different verb phrases. And this is the meaning of every student. To get all the way back to \([\text{every}]\), we just bind the slot filled by \([\text{student}]\):

\[
\lambda X \left( \lambda Y \left( T \text{ if } X \subseteq Y, \text{ else } F \right) \right)
\]

This is what’s in (21).

(34) Okay, you convinced me for every. But surely no, no student, etc., can all just denote the empty set. That seems intuitively like what no means: nothingness.

No, that won’t work! Consider no parent studies. This is true in our possible world, but neither \([\text{parent}]\) nor \([\text{studies}]\) is the empty set in our possible world. It’s their intersection
that is empty if this sentence is true. And we want to say that in general, and that’s what our theory does. We can again start with a specific claim:

\[ T \text{ if } [\text{parent}] \cap [\text{studies}] = \emptyset, \text{ else } F \]

And then we back off to get \([\text{no parent}]:\)

\[ \lambda Y \left( T \text{ if } [\text{parent}] \cap Y = \emptyset, \text{ else } F \right) \]

And once more to get the meaning we defined in (25):

\[ \lambda X \left( \lambda Y \left( T \text{ if } X \cap Y = \emptyset, \text{ else } F \right) \right) \]

(35) I am starting to see that this is the least complicated thing we can do. And I also see that this basic set-up can work for lots of determiners. We start with our framework

\[ \lambda X \left( \lambda Y \left( T \text{ if } \frac{|\text{student} \cap Y|}{|\text{student}|} > \frac{1}{2}, \text{ else } F \right) \right) \]

and then we just need to specify what the relation is for any given determiner.

Yes!

(36) And, if I want to, I can start with a specific instance of what I want to capture, like

\[ [\text{most students skateboard}] = T \text{ if } \frac{|\text{student} \cap \text{skateboards}|}{|\text{student}|} > \frac{1}{2}, \text{ else } F \]

and then just back out the variables with lambda binders, first for the scope:

\[ [\text{most students}] = \lambda Y \left( T \text{ if } \frac{|\text{student} \cap Y|}{|\text{student}|} > \frac{1}{2}, \text{ else } F \right) \]

and then for the restriction:

\[ \lambda X \left( \lambda Y \left( T \text{ if } \frac{|X \cap Y|}{|X|} > \frac{1}{2}, \text{ else } F \right) \right) \]

Beautiful! And remember to always do the binding in the order you did it: restriction outer (comes in first) and scope inside (comes in second). Some determiners are order-sensitive, like most and every.
5 Semantic grammar

(Lex) Given a leaf node X, \([X]\) is looked up in the lexicon.

(NB) Given a syntactic structure \(X \rightarrow Y\), \([X] = [Y]\)

(S) Given a syntactic structure \(S = PN \rightarrow VP\), \([S] = T\) if \([PN] \in [VP]\), else \(F\)

(A) Given a syntactic structure \(NP_i \rightarrow AP \rightarrow NP_j\), \([NP_j] = [AP](\([NP_i]\))\)

(N) Given a syntactic structure \(VP_i \rightarrow never \rightarrow VP_j\), \([VP_j] = [never](\([VP_i]\))\)

(TV) Given a syntactic structure \(VP \rightarrow V \rightarrow PN\), \([VP] = [V](\([PN]\))\)

(Q1) Given a syntactic structure \(QP \rightarrow D \rightarrow NP\), \([QP] = [D](\([NP]\))\)

(Q2) Given a syntactic structure \(S \rightarrow QP \rightarrow VP\), \([S] = [QP](\([VP]\))\)
6 Illustrations

(37) \[
S \quad \xrightarrow{PN \ VP} \quad T \ if \ \{ Bart, V \} \in F[S], \ else \ F[S]
\]

(38) \[
NP \quad \xrightarrow{AP \ NP} \quad \{ \ \} \cap \{ \ \} \quad [A]
\]

\[
\lambda X \left( \{ \ \} \cap X \right) \left( \{ \ \} \right)
\]

\[
\lambda X \left( \{ \ \} \cap X \right) \left( \{ \ \} \right) \quad [NB]
\]

\[
\lambda X \left( \{ \ \} \cap X \right) \left( \{ \ \} \right) \quad [NB]
\]

\[
\lambda X \left( \{ \ \} \cap X \right) \left( \{ \ \} \right) \quad [NB]
\]

\[
\lambda X \left( \{ \ \} \cap X \right) \left( \{ \ \} \right) \quad [NB]
\]

\[
\lambda X \left( \{ \ \} \cap X \right) \left( \{ \ \} \right) \quad [NB]
\]

\[
\lambda X \left( \{ \ \} \cap X \right) \left( \{ \ \} \right) \quad [NB]
\]

\[
\lambda X \left( \{ \ \} \cap X \right) \left( \{ \ \} \right) \quad [NB]
\]

\[
\lambda X \left( \{ \ \} \cap X \right) \left( \{ \ \} \right) \quad [NB]
\]
(41) S
  QP
  D every NP
  N child
  V skateboards

F
  T if [child] ⊆ [skateboards], else F
  λY( T if [child] ⊆ Y, else F ) [Q2]
  [skateboards] [NB]
  [skateboards] [NB]
  [skateboards] [LEX]

  λX( λY( T if X ⊆ Y, else F ) ) [NB]
  [child] [NB]
  [child] [LEX]

  [every] [LEX]
(45) S
     QP   VP
     no   V
     N   PN
     parent

(46) S
     QP   VP
     every V
     D   NP
     A   NP
     A    N
     scholarly child

(47) S
     QP       VP
     some     never V
     D   NP    VP
     A   NP
     A     V
     A    N
     hungry child