Teaching type-driven compositional semantics in intro

Sample lesson plan and teaching guide

Ai Taniguchi

Carleton University

(Version as of April 23, 2020; feedback always welcome)

1 Note to instructors

- This document is a sample flow of lectures for the semantics module in an introductory linguistics course.
- In my experience teaching formal semantics in intro has been successful with a class of 200 students; I'm confident this would work in smaller classes as well.
- This lesson plan presupposes that you've taught syntax in the class already.
- This lesson plan uses an active learning and experiential-learning based teaching method where students are encouraged to participate in problem-solving themselves (as opposed to the instructor just "telling the students how semantics works").
- This active learning method comes with a "example first, terminology later" approach. I find it pedagogically more effective to explain concepts via examples first, and THEN give the terminology that goes along with it. Not the other way around where you give the term, define it, and then give an example. Retention is much better if students understand the concept itself first.
- This is a way of teaching semantics in intro that has personally worked for me. I'm definitely not saying that this is the only way to teach semantics in intro; I'm also not saying that this style will work for everyone! What works and what doesn't often depends on your personality, teaching style, and most of all how comfortable with the material you are. Please use this document as a light suggestion rather than a bible.
- Please also note that in order to teach compositionality, you don't necessarily have to use lambda calculus. You can also really teach the same thing using set theoretic notions. One benefit of using type-driven compositional semantics is that it makes talking about functional words like *the* easier later on.

💬 ← This symbol indicates a point of discussion with the students, and a good place to extract input from students (i.e., class participation encouraged).

⚠ → This symbol indicates (in my experience) difficulties that tend to arise with undergraduate students, and ways to navigate that issue.

2 Day 1

2.1 Learning outcome

- The purpose of the first class is to get across to students what linguistic meaning is, and how it can be analyzed.

⚠ Laymen (including students) often come into semantics thinking it is either (i) lexical semantics (e.g., "what does *love* mean?"), (ii) pragmatics (e.g., implicatures), or (iii) some version of non-formal discourse analysis (e.g., "what was the speaker's intention in phrasing this this way?"). Truth condition is not what people typically immediately mentally jump to when they hear "meaning", so you really want to make a convincing case of this on the first day. Wherever possible, extract truth value judgments from students (ask "is this sentence true or false?" "can just a word be true or false?" instead of just spoon-feeding them facts like "sentences have truth values". They should come to the realization that they indeed do have judgments about sentential meaning in this way.

2.2 Sense, reference, denotation

- You want to first get across to students that there are two ways of thinking about meaning: **sense** (what the linguistic expression actually expresses) and **reference** (what the linguistic expression points to in the actual world). (Frege 1997)

💬 One effective way of teaching this experientially is to start with some word that doesn't have an easy lexical semantics, preferably something that students have a strong opinion about. When I taught this in Ottawa, Ontario, Canada, I used *beavertail* as an example. It's a locally loved fried dough pastry. I'll use this as the example below.

1. Start with a question like "What is a beavertail?". Get a student to tell you the basic "definition" of the word (e.g., 'a flat dessert thingie with toppings'). Pretend to not know what it is.

2. Have ready some pictures of things that are NOT beavertails. My first one was a dessert pizza. This really offended the students lol. Get a reaction out of them ("no that is NOT a beavertail!").

3. Now get them to precisify the "definition" of *beavertail* so dessert pizzas can be excluded (e.g., "no it's shaped like a beaver's tail...").

4. Show them another picture of a non-beavertail. I showed them a dessert naan with fruit toppings. "So is this a beavertail?" "No," they say, exasperated with me.

5. Repeat this until they're deepy frustrated with you lol. "No it's deep-fried!" "Oh so it's a beignet." "No it's flatter and bigger!" "Oh a funnel cake..." "No it's one flat piece of dough!"

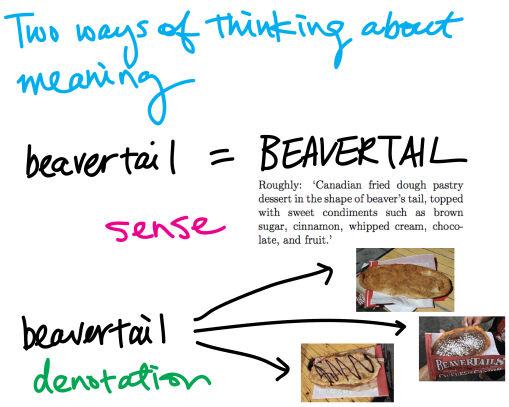
6. Then finally, put a real picture of a beavertail up. **Point to it** and say, "Oh, so THIS is a beavertail?" If you're lucky like I was, you'll get the exasperated students pointing at it too like, "Yes, THAT THING."

7. You have now exemplified sense vs. reference.

- What they were trying to articulate was the **sense** of *beavertail*. When they pointed at it and said THAT thing is a beavertail, that is **reference**.

- You use the **sense** of a linguistic expression to determine **reference**. It is because you have the sense of *beavertail* in your head that you can say "this is (not) a beavertail" for any thing in the world you're given.

- Thus framing meaning in terms of the sense (e.g., *whatever* ***beavertail*** *means*) and denotation (e.g., the set of all beavertails in the actual world) are two ways of talking about meaning. See note about reference vs. denotation in the box to the right.



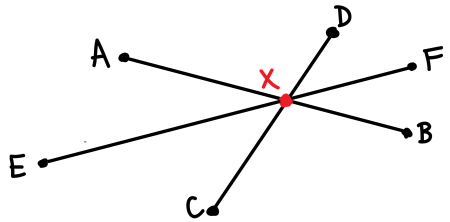
- Explicitly introduce the terms **sense** (what the linguistic expression expresses) and **denotation** (what the linguistic expression points to) now.

- Discuss what names (e.g., *Celine Dion*) denotes. That expression points to a particular individual.

- Explicitly introduce the term **reference** (unique denotation) now. Names and definite descriptions refer to unique individuals.

💬 To demonstrate the contrast between sense and reference, give examples of things with a unique reference that *can be described in different ways*. e.g., Give a picture of Luke Skywalker; this individual can be described as *the brother of Leia Organa* or *the son of Anakin Skywalker*. These two definite descriptions have different *senses*, but they *refer/denote* the same individual.

💬 Another good example of this is the diagram below. Ask students to describe point X as an intersection of two lines. There's multiple ways to do that. *The intersection of AB and CD* and *the intersection of CD and EF* have different senses, but they have the same reference/denotation. It's different ways of describing the same thing.



Three senses of point X =
1. intersection of $\overline{AB}$ & $\overline{CD}$
2. intersection of $\overline{CD}$ & $\overline{EF}$
3. intersection of $\overline{AB}$ & $\overline{EF}$

- Do the opposite as well: give an example of something with a particular sense that points to different individuals (depending on when it's uttered). Titles are good: *the Prime Minister of Canada* in 2020 points to Justin Trudeau, but in 2014 it pointed to Stephen Harper. One sense, two different references.

FYI: Reference, denotation, extension
Usage of these terms slightly vary depending on who's using it sometimes. Or: the way that it is used has evolved since Frege's time, potentially. Some people (and textbooks) use *reference*, *denotation*, and *extension* interchangeably. From my understanding, they don't actually exactly mean the same thing.

Denotation has the widest meaning of 'things that a linguistic expression points to'. It was originally interpreted as 'things that a linguistic expression points to in the actual world' (e.g., a set of beavertails in the actual world for *beavertail*), but in formal semantics today it's used in a wider sense to point to things like possible worlds (e.g., semanticists say things like "sentences denote a set of possible worlds").

Extension is more restricted: the extension of a linguistic expression is what it points to in the actual world, with emphasis on actual world. It contrasts with the **intension** of linguistic expressions, which points to things across all possible worlds. The discussion of intensions will likely not come up in intro.

Reference, from how I understand it from Frege, is unique denotation/extension. It's THE particular thing that a linguistic expression points to. So this speaks to names and definite descriptions (e.g., *Simone Biles*, *the gold medal winner of the 2016 Olympics in gymnastics*, etc.), which point to particular individuals in the actual world. The *Simone Biles* refers to the individual Simone Biles in the actual world. In more recent years, I've seen semanticists say things like "the reference of *beavertail* is the set of all beavertails in the actual world", however.

- Make them realize at this point that the sense of a linguistic expression is actually quite hard to articulate into words (re: *beavertail*). This is why semanticists talk about meaning in terms of denotation instead a lot, and this is what you'll be assuming for the next few lectures. If you're doing some lexical semantics later, I'd flag that you'll come back to sense later too.
- Useful at this point to introduce a notation for denotations: the double brackets ⟦ ⟧. ⟦beavertail⟧ reads as 'the denotation of the linguistic

expression *beavertail*. ⟦Justin Trudeau⟧ reads as 'the denotation of the linguistic expression (name) *Justin Trudeau*. You could notate this right now as:

- ⟦beavertail⟧ = [insert picture of a bunch of beavertails] and/or
- ⟦beavertail⟧ = the set of all beavertails in the actual world
- ⟦Justin Trudeau⟧ = [insert picture of Justin Trudeau] and/or
- ⟦Justin Trudeau⟧ = the individual Justin Trudeau in the actual world

⚠ It's worthwhile to really clarify to the students the difference between e.g., *beavertail*, the **linguistic expression**, vs. beavertails, the **things in the actual world**. The linguistic expressions are the words, phrases, and sentences: the things that we say. These expressions point to actual things in the actual world we live in. This contrast is easy to grasp but potentially confusing if not explicitly clarified (especially because you will wind up writing things like *the set of all beavertails in the actual world*, which looks like a linguistic expression, as the denotation of the linguistic expression. This has the danger of misleading students to think that semantics is a theory of paraphrases (linguistic expression to linguistic expression), which it is not — so remind them overtly that we're actually talking about real things in the real world as the denotation. Initially using pictures and photos to show the denotation of words is helpful for emphasizing this point.

- Provide interim conclusion here that different kinds of linguistic expressions point to

1. Nouns (*beavertail*), (intransitive) verbs(*sneeze*), and adjectives (*orange*) denote sets of things with that property in the actual world (give some examples; e.g., ⟦orange⟧ = the set of all orange things in the actual world, etc.)

⚠ Some students struggle with the idea that noun phrases are properties, just like adjectives. In other words, they wonder why ⟦beavertail⟧ isn't just a unique beavertail: it's "that thing" you're talking about. Nouns feel "concrete" to them, I think. Stress the difference between *beavertail* vs. *the beavertail* to mitigate this.

2.3 Denotation of sentences

💬 Transition the discussion to what a sentence denotes. Elicit from students how individual words and sentences differ (e.g., *win* vs. *Celine Dion won the Grammy in 1999*)

- I always like to say this to them: What if I walked into the room and just said "won"? What's off about that? They very reliably reply to this with, I want to know who won what. This is a good transition to talk about the fact that predicates are "incomplete", but sentences are "complete".
- Transition the student response into the concept of **truth values** (the value **true** and **false**). Sentences have truth values, words don't.
- You can say *It is true that Celine Dion won the Grammy in 1999*, but not **It is true that won*, **It is true that Celine Dion*, etc.
- Now you need to talk about **truth conditions**. Pick a sentence where the actual truth value of the sentence isn't immediately obvious (I did *This room has 195 seats*). Ask if this is true or false. Students obviously won't know. But press them and ask could you find out if you really wanted to — the answer is yes. Ask them what it would take for this sentence to be true. They'll say, there needs to be 195 seats in this room. That's the condition that would make this sentence true. That's the **truth condition** of a sentence.
- This means that the denotation of any sentence can be thought of as a truth condition. With the relevant world facts, you can assign the sentence a truth value. It's ok if you don't actually know if the sentence is true or false; what you do know is what it would take for it to be true.
- Interim conclusion here (pick whatever examples you've used):

1. ⟦beavertail⟧ = the set of all beavertails in the actual world
2. ⟦American⟧ = the set of all American entities (things/people) in the actual world

⚠ To some students *entity* means inanimate things only. It may be worthwhile to clarify that entity means things and people. You can also use the term *individual*, but that also needs to be clarified since many people think of individuals as animate only. You can also just use the word *thing*, too imho.

3. ⟦Justin Trudeau⟧ = the individual Justin Trudeau in the actual world
4. ⟦Justin Trudeau is the PM of Canada⟧ = TRUE if JT is the PM of Canada, FALSE otherwise (so, TRUE)
5. ⟦Justin Trudeau is American⟧ = TRUE if JT is American, FALSE otherwise (so, FALSE)

⚠ I've found that on exams and homework, first-year students really struggle with the idea that sentences denote an abstract truth value. Even though they seem to get it in class, a handful of them will suddenly say that everything, including sentences, are sets of things. It's common, for example, for them to incorrectly say that *JT is Trudeau is Canadian* denotes the set of all Canadians in the actual world. Make sure several examples are covered in class.

2.4 Introduction to set theory

- Start with a reminder of what something like (the adjective) *orange* denotes (elicit from students): the set of all orange entities in the actual world. What about *Garfield*? The individual Garfield.
- Show both via pictures (rather than words) for the purpose of demonstrating set theory.
- What we want to do is to build up the meaning of the sentence *Garfield is orange* out of the meaning of *Garfield* and *orange* (**compositionality**).

💬 Now think about the sentence *Garfield is orange*. Using the denotation of *orange* and the denotation of *Garfield*, how can we characterize what *Garfield is orange* means? This is easy to elicit from students: they'll say, it means that Garfield is in this set of orange things.

- cf., Language Files for an introduction to set theory. I would do enough to cover what a set is, set membership, and intersective vs. subsective adjectives. I personally didn't teach formal set theory notation (to save time for more compositional semantics), but it would be possible to do it here.

3 Day 2

3.1 Learning outcome

- Relating set theory to a formal theory of compositional meaning, particularly lambda calculus
- What they're learning on this day is **compositionality**: the meaning of a larger linguistic unit is composed of the meaning of is subparts. The meaning of words add up to make up the meaning of sentences.
- Sentences denote truth values, but this results from combining things that DON'T have truth values... how does that happen? (Answer: functions.)

3.2 From sets to functions

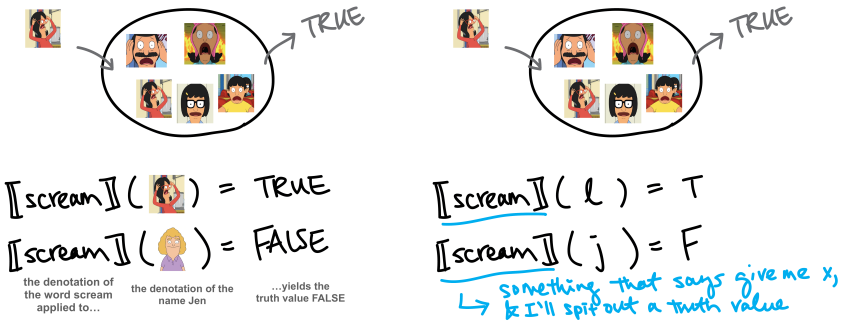
- Start with the baby set-theoretic notion of a sentence with an intransitive verb, like *Linda screamed*.
- ⟦scream⟧ is the set of all entities that scream in the actual world. Have a visual representation of this set (I used pictures of Bob's burgers characters). Assume for the purposes of this class that this is the complete set of screamers.
- To say *Linda screamed* is to say that Linda is in this set. Display Linda's picture separately from this set and ask, is Linda in this set? Students say yes. So we get the truth value TRUE for this sentence.
- Now display a picture of Jen, or anyone else who is not in this set. Now the sentence is *Jen screamed*. Ask again: is Jen is this set? Now they will say no. This set tells us that this sentence is FALSE.
- Now we're seeing something here: this set of things is a very powerful and knowledgeable object — for any entity in the world, it's able to tell you if that thing is in this set or not (i.e., whether it screamed or not). It's like a magical bag of screamers that you can look in in, and it'll tell you "yep!" or "nope!" if you give it an entity and ask, is this thing in there?
- In other words, you can give this set an individual, and it'll reliably spit out a truth value (TRUE or FALSE) for you.
- Your big transition is here: so one way of thinking about the denotation of something like *scream* is that it's a set of things (set talk), but another way of thinking about it is that it's a *function*.
- A **function** is anything that has a unique output for any input. You put something in, something comes out. A meat grinder is a function. You put beef in, you get ground beef out (never ground pork — it's always reliably ground beef; this is what is meant by 'has a unique

output'). You put chicken in, you get ground chicken out. You can also show mathematical functions: 1 + x = y is a "add 1" function. You put in 2 for x, you get 3 as the output for y.

- Terminology: a function **applies** to its input and yields an output. The meat grinder applies to the meat and yields ground meat. The *scream* function applies to Linda and yields TRUE.

- Notationally, you write: $[\![\alpha]\!](\beta)$. This reads as 'the denotation of α (a function) applied to β'.

- I recommend pictorally representing this first. Since no one has time to say "the individual Linda" or find a picture of Linda every time, tell them we will abbreviate "the individual Linda" as bolded lowercase **l**. (Note to instructor: formally, constants (like **l** for Linda) should be bolded; I wouldn't be picky about this with how students write it, but FYI). You can abbreviate TRUE and FALSE as T and F if you haven't already at this point.



3.3 One-place predicates

- So how do you express the denotation of a verb like *scream* as a function? Introduce the λ (lambda) notation.

Lambda calculus

(1) a. SCREAM(x)
'x screamed' (i.e., 'x is in the scream set')

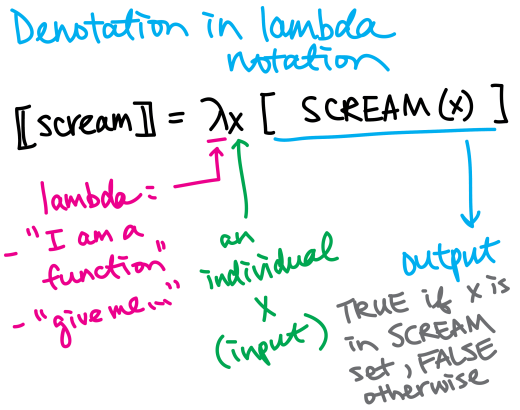
b. $\lambda x[SCREAM(x)]$ 'Give me an individual x, and I'll give you TRUE if this x sneezed, FALSE otherwise'

(2) *Linda screamed*

a. $\lambda \underline{x}$ [SCREAM(**l**)] **unsaturated** function
↑ ↑
input output

b. λx[SCREAM(x)] (**l**) saturating a function

c. SCREAM(**l**)



- Give the lexical entries (denotation) of *Linda* and *scream*:

1. [[Linda]] = **l** (Remind them: the individual Linda)
2. [[scream]] = λx[SCREAM(x)] (Remind them: the set of all screamers, or the function from individuals to truth values that yields TRUE if the individual screamed, FALSE otherwise)

- Do a step-by-step computation of [[Linda screamed]]. Explicitly tell them to ignore tense (don't have the tools to talk about it yet).

⟦Linda screamed⟧

1. = ⟦Linda scream⟧ (ignore tense)
2. = ⟦scream⟧(⟦Linda⟧) (the function applies to input)
3. = ⟦scream⟧(**l**) (give denotation of *Linda*)
4. = $\lambda x[SCREAM(x)](\mathbf{l})$ (give denotation of *scream*)
5. = SCREAM(**l**) (apply the function to input)
6. = T if SCREAM(**l**), F otherwise (state it as truth condition)

- Do similar practice in class.
- At a timing that it seems necessary, it's good to summarize what things are functions/sets and what things are not.

  1. Words like *scream*, *Canadian*, and *beavertail* (**predicates**) are INCOMPLETE. These things are sets of things and want to tell you if something is in this set or not. Without this something, it can't form a full complete thought. Sets of things are functions.
  2. Names like *Linda* are THINGS THAT COMPLETE INCOMPLETE THINGS (**arguments**).
  3. Sentences like *Linda screamed* are COMPLETE. It's done. It doesn't want to combine with anything else. It denotes a truth value.

4 Day 3

4.1 Learning outcome

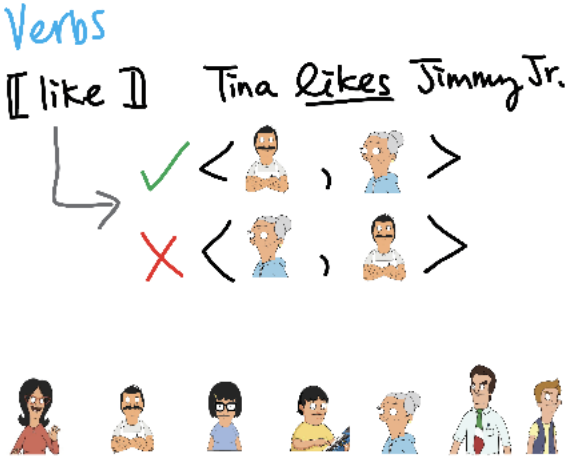
- Compositionally analyzing two-place predicates
- Basic semantic types
- Applying basic knowledge to novel data with semantic types

4.2 Two-place predicates: sets

- Note to instructor: you have the option of stopping at one place predicates in an intro class. You don't necessarily have to do the computation of two-place predicates. You could opt to just go straight to semantic types after this (and e.g., have them just figure out the semantic type of two-place predicates instead).
- Start discussing two-place predicates. I would just do transitive verbs for an intro class. e.g., *Tina likes Jimmy Jr.*.

Ask students what's different about a verb like *like* compared to *scream*. They should notice quickly that *like* requires "two things" to be a complete sentence.

- It's helpful to have a Model you can work with here. Pick your favorite pop culture reference. Have a finite set of individuals you're working with. This is your Model.
- In this Model, you can establish who likes who. What's going to be important is that just because A likes B doesn't mean B likes A back.
- This will help set up the idea that transitive verbs like *like* is a relation between two individuals. But it matters who the subject is and who the object is.
- Introduce **ordered pairs**. It's a set of two things in a certain order. It's literally a pair of things that is ordered. Ordered pairs are written as $\langle x, y \rangle$. For the mathematically inclined, it's helpful to talk about graphs and ordered pairs.
- We can think of *like* as denoting ordered pairs. There are certain pairs of people that fall in this "like" relation. For example, Bob likes Edith, so $\langle b, e \rangle$ is a "like" relation. But Edith doesn't like Bob, so $\langle e, b \rangle$ is not a "like" relation. Tina likes Jimmy Jr, so $\langle t, j \rangle$ is a "like" relation, and so on.



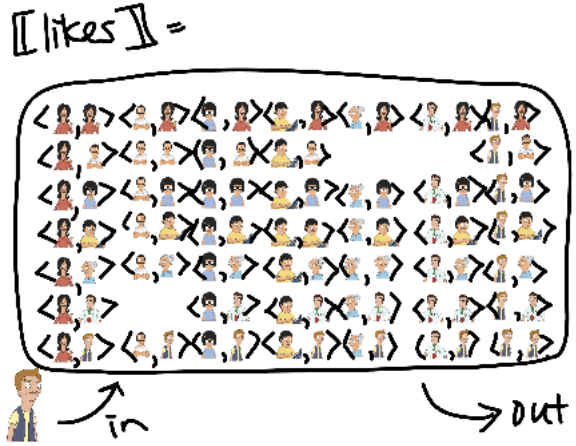
- With a finite set of individuals in your Model, you can have a concrete set of ordered pairs that constitute a "like" relation. You could visually show this.

- This is a set of things again: a set of ordered pairs, this time.

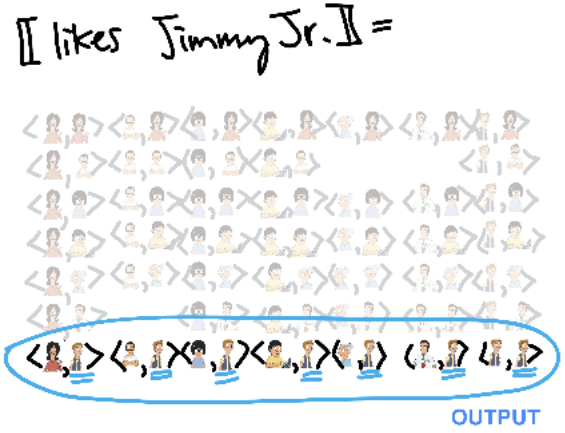
- If something is a set, it's a function too (re: last class).

- Maybe worthwhile to remind them the syntax of a sentence with a transitive verb. Remind them that the verb syntactically combines with the object first. Give the tree of the sentence you are going to analyze (e.g., *Tina likes Jimmy Jr.*)

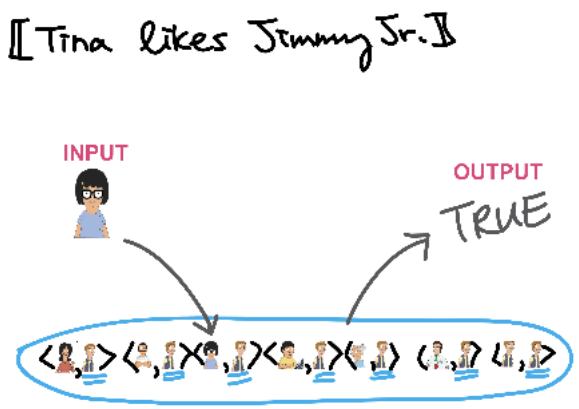
- So given this set of ordered pairs, the first thing that this verb *like* wants to know is who is liked (the object). It wants the object as its input. Let's feed *Jimmy Jr.* into this set.



- Now this set has narrowed the original set down into the set of people that like Jimmy Jr. This is the denotation of *like Jimmy Jr.*. Note that it's still a set, meaning it's still incomplete. This is because this expression is "not done yet."



- Now we have to give this set/function the subject, Tina. If we feed Tina into this function, this function can tell you TRUE she does like Jimmy Jr. (i.e., she's in this set), or FALSE she doesn't (i.e., she's not in this set). In this case it yields TRUE.



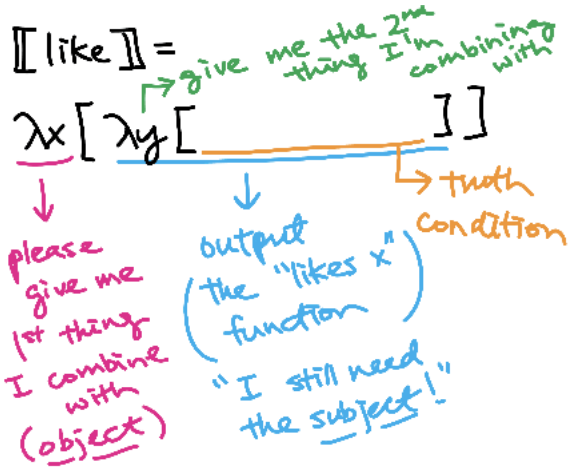
- So a transitive verb is a two-step process.

4.3 Two-place predicates: lambda computation

- Now we translate this into lambda calculus, using function talk.

Elicit the lexical entry of *like* as much as possible from the students. It is best to do this incrementally. First start with, "*like* first says 'I need to know who is liked'; it says 'give me an individual'. So what do we need to start with?" Answer: Lambda x.

Now here's the difference between an intransitive verb and a transitive verb. *Scream* was done after one thing, but *like* is not. It says, "Give me ONE MORE thing, which is thing who likes this other thing." It says "Give me another individual". What do we need? Answer: Lambda y, for a separate individual.



- (In my experience this much is not hard to elicit.)

Now the task is to figure out the truth condition. How should x and y relate, given the predicate LIKE?

⚠ This part is critical; have them think through what order the variables must be in. Helpful to show them the tree again.

- If x is the first thing that the verb combined with, then x is the object. If y is the second thing that the verb combined with, then y is the subject.

- Therefore, the truth condition should read as (T iff) LIKE(y,x).

- The full lexical entry is $\lambda x[\lambda y[LIKE(y, x)]]$. Introduce this two-place predicate notation.

- Some students will be confused as to why it's not (x,y). That's ok. Do a full computation to show that this proposal works.

- This computation will happen in two steps.

⟦Tina likes Jimmy Jr.⟧

1. ⟦likes Jimmy Jr.⟧
   (a) = ⟦like Jimmy Jr.⟧ (ignore tense)
   (b) = ⟦like⟧(⟦Jimmy Jr.⟧) (function applies to input)
   (c) = ⟦like⟧(**j**) (give den. of *Jimmy Jr.*)
   (d) = $\lambda x[\lambda y[LIKE(y, x)]]$(**j**) (give den. of *like*)
   (e) = $\lambda y[LIKE(y, \mathbf{j})]$ (apply function to input)
   ⚠ Still incomplete! No truth value yet.
2. ⟦Tina likes Jimmy Jr.⟧
   (a) = ⟦Tina like Jimmy Jr.⟧ (ignore tense)
   (b) = ⟦like Jimmy Jr.⟧(⟦Tina⟧) (function applies to input)
   (c) = ⟦like Jimmy Jr.⟧(**t**) (give den. of *Tina*)
   (d) = $\lambda y[LIKE(y, \mathbf{j})]$(**t**) (give den. of *like Jimmy Jr.* from previous step)
   (e) = LIKE(**t**, **j**) (apply function to input)
   (f) = T if LIKE(**t**, **j**), F otherwise (state as truth condition)

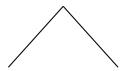
- Do additional practice. Recommended that you make them do one on their own.
- Good to put labels on things, now that you have a contrast: things that require just one argument like *scream* are called **one-place predicates.** Things that require two arguments are called *two-place predicates*.

4.4 Basic semantic types

- You can frame this part as "we've learned something cool: there's actually only two fundamental types of meaning: individuals and truth values". Other words we've seen sort of act as an interface between these two things.
- e.g., ⟦scream⟧ is a function that says "give me an individual, and I'll give you a truth value."
- What we see is that different words have different jobs in their meaning. Some things are functions, some things are not.
- A notion that is useful for talking about this is **semantic types**. It's literally the idea that different words have different kinds of meanings.
- Definition of types:

(3) *Semantic types*
   a. e and t are semantic types
   b. If σ and τ are semantic types, then $\langle \sigma, \tau \rangle$ is a semantic type.
   c. i.e., $\langle$INPUT type, OUTPUT type$\rangle$ = a type
   d. Nothing else is a semantic type.

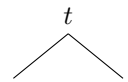
- e is the type of an individual. It's not a function.
- t is the type of anything with a truth value. It's not a function.
- Anything of the format $\langle \sigma, \tau \rangle$ means that it is a function. To the left of the comma is the thing that is the input. To the right is the output.
- Give them a tree to illustrate semantic types.



Linda screamed

💬 Have them fill in this tree with semantic types, as much as possible. Start with the simple ones: which word is type e? What thing has a truth value, and therefore is type t?

- You'll get something like this:



Linda screamed

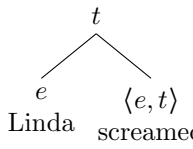
💬 Now what's the type of *screamed*? What does it want as the input? What does it spit out as the output?

- *Screamed* is type $\langle e, t \rangle$. This is the type of a one-place predicate.
- You can relate this to the lexical entry of *scream(ed)*:

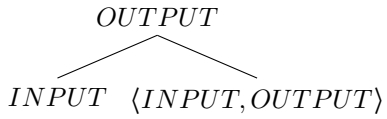
(4) $\lambda \underline{x}$ [SCREAM(**1**)]

↑ input ⟨ e , ↑ output t ⟩

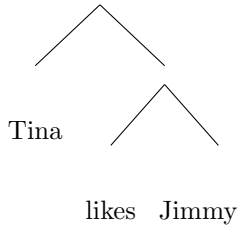
- So here's the tree:



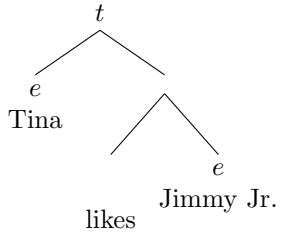
- So really what you're seeing is:



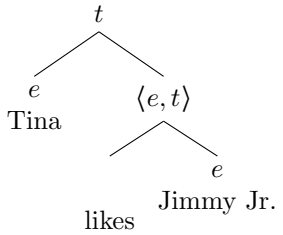
- Trees annotated with types help us visualize what word is combining with what, and how.
- Give them a challenge: figure out the semantic type of *like*.



This much, they can figure out for sure:

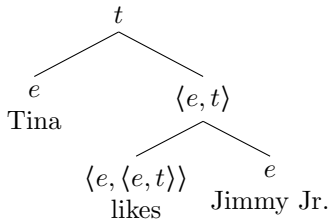


And from that, this as well (this means that *likes Jimmy Jr.* is a one-place predicate, which it is):



- Remind them that *like* is a function! It "wants to combine with *Jimmy Jr.*". So it should be a functional type. It has Jimmy Jr. as its input, and has *likes Jimmy Jr.* as the output. Type-wise, it wants e as the input, and $\langle e, t \rangle$ as the output.

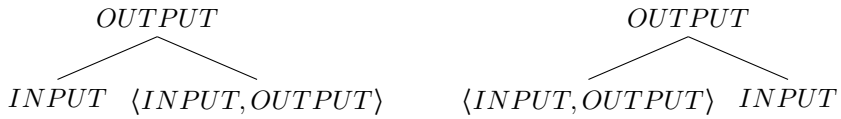
They should land on $\langle e, \langle e, t \rangle \rangle$. This is the type of a two-place predicate.



- $\langle \underline{e}, \langle e, t \rangle \rangle$: INPUT type, OUTPUT type. It's something that applies to an individual and spits out an $\langle e, t \rangle$ function. The output itself is another function that wants to combine with a type e thing to yield a type t thing.
- This is like saying that the word *like* combines with one individual, and saying "thanks for that, but I'm still not done; I still need one more individual before I can give you a truth value." That's why it's still $\langle e, t \rangle$.
- Once *Tina*, type e, combines with *likes Jimmy Jr*, then this function can give you a truth value. T if Tina likes Jimmy Jr., F otherwise.
- $\langle e, \langle e, t \rangle \rangle$ = 'I need two individuals before I can give you a truth value.' This is what it means for something to be a two-place predicate. The

two e's in the semantic type are those two individuals that it needs to combine with.

⚠ Students often find it confusing that functional application can happen in both directions. That is, it can be either of these two:



Worth clarifying that a function just wants an input; it doesn't matter which side it's on.

⚠ The concept of semantic types can be suuuuuper hard for students, especially those fresh out of high school that still think memorizing is learning. This "figure this out based on what you know" exercise is a higher-order cognitive skill. I always like to remind them that this is like word math. They are especially bad at figuring out types of expressions that they haven't seen in class before (e.g., on homework). I wouldn't shy away from this kind of problem though, since it builds very important critical reasoning skills.

5 Day 4

The additional day is for things that are typically covered in the semantics portion of intro: pragmatics, implicatures and such. Standard intro textbooks are fairly good resources for this. For example, most textbooks cover entailments vs. implicatures, and Grice's cooperative principle.

I ideally want to talk more about lexical semantics in intro, but I haven't found a way to incorporate it yet (because of time constraints). I'm personally not a fan of teaching about lexical relations like synonyms, antonyms, etc., because it's not really that cognitively stimulating for students (it's just memorization of terms).

On my website, you can find a video of me teaching actual Generative-Lexicon-style lexical semantics (simplified, of course) in an intro linguistics class for non-majors. I was able to do that for that particular class because the compositional semantics portion was much more condensed than this.

References

Frege, G. (1982/1997). On *Sinn* and *Bedeutung*. In Beaney, M., editor, *The Frege Reader*. Blackwell, Oxford.