# Compositionally interpreting formal, natural languages

Simon Charlow (simon.charlow@rutgers.edu)

September 23, 2015

## 1 Review from last time

Semantics of propositional logic:

- $\llbracket v \rrbracket^w \coloneqq w(v)$ if $v$ is an atomic formula
- $\llbracket \neg\varphi \rrbracket^w \coloneqq 1 - \llbracket \varphi \rrbracket^w$
- $\llbracket \varphi \wedge \psi \rrbracket^w \coloneqq \texttt{Min}(\llbracket \varphi \rrbracket^w, \llbracket \psi \rrbracket^w)$
- $\llbracket \varphi \vee \psi \rrbracket^w \coloneqq \texttt{Max}(\llbracket \varphi \rrbracket^w, \llbracket \psi \rrbracket^w)$
- $\llbracket \varphi \Rightarrow \psi \rrbracket^w \coloneqq \texttt{Max}(\llbracket \neg\varphi \rrbracket^w, \llbracket \psi \rrbracket^w)$

Equivalently and alternatively, in terms of set-theoretic representations:

- $\llbracket v \rrbracket \coloneqq \{w : w(v) = 1\}$ if $v$ is an atomic formula
- $\llbracket \neg\varphi \rrbracket \coloneqq \overline{\llbracket \varphi \rrbracket}$
- $\llbracket \varphi \wedge \psi \rrbracket \coloneqq \llbracket \varphi \rrbracket \cap \llbracket \psi \rrbracket$
- $\llbracket \varphi \vee \psi \rrbracket \coloneqq \llbracket \varphi \rrbracket \cup \llbracket \psi \rrbracket$
- $\llbracket \varphi \Rightarrow \psi \rrbracket \coloneqq \llbracket \neg\varphi \rrbracket \cup \llbracket \psi \rrbracket$

As emphasized last time, these rules are **syncategorematic**, ergo **non-compositional**: instead of specifying independent meanings for ¬, ∧, ∨, …, their meanings are characterized *in construction*.

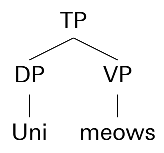
Today we will develop some formal tools for interpreting **intransitive** sentences like Uni meows, **transitive** sentences like Uni licked Porky, and even **ditransitive** sentences like Uni showed Puffy to Porky.

We'll start building a theory but quickly run into some roadblocks. We'll see that we can solve these issues in a way that helps us give a more compositional semantics for propositional logic.

## 2 Base case

### 2.1 Some test cases

Interpreting intransitive sentences like (Uni meows) seems easy. Just suppose meows denotes the set of meowers, and give a simple rule for interpretation in terms of **set membership**:



$$\llbracket \text{DP VP} \rrbracket = 1 \text{ iff } \llbracket \text{DP} \rrbracket \in \llbracket \text{VP} \rrbracket$$

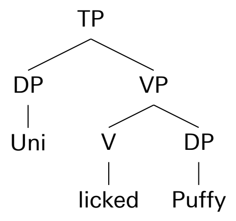
Here, iff Uni $\in \{x : x \text{ meows}\}$. (See H&K's long proof of a similar case.)

How about Uni meows and Uni purrs? OK: Another rule for interpretation:

$$[\![\text{TP}_1 \text{ and } \text{TP}_2]\!] = 1 \text{ iff } [\![\text{TP}_1]\!] = [\![\text{TP}_2]\!] = 1$$

Here, iff Uni $\in \{x : x \text{ meows}\}$, and Uni $\in \{x : x \text{ purrs}\}$. Fine truth conditions, but they come at the price of another rule. What about disjunction?

Transitive cases like Uni licked Puffy are a harder nut to crack. Ideally, we would like to associate every node in the tree with an interpretation.



$[\![\text{V DP}]\!] = ???$

But we don't know how to associate the VP with a set, which would allow us to apply the earlier rule to form the whole sentence. Will we need another interpretation rule?

To say nothing of *ditransitive* cases like Uni showed Puffy to Porky. Will these require *yet another rule*??

### 2.2 Why our theory won't look this way

We'd *really* like to avoid case-by-case rules for how things are composed:

- Don't really get any sense of how things work in the general case.
- I.e. not very explanatory.
- Super **syncategorematic**. Not everything gets assigned a meaning.

Can that really be how things work? We're gonna say NAH. What we would like is as *general* a characterization of the interpretation function $[\![\cdot]\!]$ as we can get.

What we will end up saying: meanings are either functions or arguments. Composing meanings is uniformly a process of *apply functions to arguments*.

We will need a bit of math to get there.

## 3 Some math

### 3.1 Start with a model

(See board)

Some facts about the model:

- Some cats are meowing. Some aren't.
- Some cats are bigger than others.
- Some cats are to the left of others.

## 3.2 Relations refresher

We can use relations to formalize features of this model.

Relations are sets of ordered pairs. The members of the ordered pairs stand in a certain relationship. E.g.:

- $\{(1,2), (1,3), (2,3), (3,4), \cdots\}$
- $\{$(Veneeta, semantics), (Simon, semantics), (Ken, syntax), $\cdots\}$

Diagram displays the licked relation (for convenience, the same as the bigger-than relation):

$$\{(A, B), (A, C), (B, C)\}$$

Other ways of specifying this relation:

- $\{(x, y) : x \text{ licked } y\}$
- $xRy$ iff $x$ licked $y$

## 3.3 Functions refresher

A function $f$ is any relation such that for any $x \in \text{Dom}(f)$, $(x, y) \in f$. Instead of $(x, y) \in f$, we write $f(x) = y$.

Functions can be **partial**. Given some $x$, if $f$ doesn't include any pair of the form $(x, y)$, then $f(x)$ is **undefined**.

**Tabular notation** for functions. Here, a partial square root function:

$$\begin{bmatrix} 1 \to 1 \\ 4 \to 2 \\ 9 \to 3 \end{bmatrix}$$

Here is a function that pairs each cat (that has a cat to its left) with the cat to its immediate left:

$$\begin{bmatrix} B \to A \\ C \to B \end{bmatrix}$$

## 3.4 Characteristic functions refresher

A function $f_S$ is the **characteristic function** of a set $S$ iff, for any $x$ in $S$, $f(s) = 1$, and for any $x$ not in $S$, $f(s) = 0$.

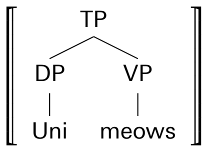
Example: call the characteristic function of the set of people in this room $f$. What is $f$(Veneeta)? What is $f$(Augustina)?

We can now rethink our entries for an intransitive verb like *meows*. Instead of denoting a set $S$, it'll denote the characteristic function on $S$:

$$[\![\text{meow}]\!](x) = 1 \text{ iff } x \in \{y : y \text{ meows}\}$$

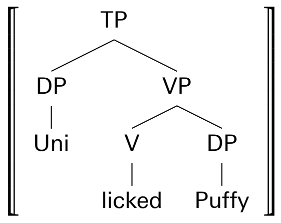
In other words, $[\![\text{meows}]\!]$ is that function $f$ such that for any $x$, $f(x) = 1$ iff $x$ meows.

Payoff: interpreting intransitives via functional application.



$= [\![\text{meows}]\!]([\![\text{Uni}]\!])$

But what, now, about transitives? By analogy with the case we just did, we expect something like the following:



$= [\![\text{licked Puffy}]\!]([\![\text{Uni}]\!])$

In other words, since VPs denote functions from individuals to truth values, so should licked puffy.

The rest of the class will be fleshing this out.

## 3.5 Functions into functions and Currying

Functions don't always have to return a value like 1 or 0. Functions might also return *other functions*.

Think about the addition operation. A natural way to think of it is as an operation that takes two numbers $m$ and $n$ at once and then gives you back $m + n$ — i.e. as a function from pairs of numbers into a third number. But it could just as well take the things to be added *one at a time*!

The same goes for relations like the licked relation, the is-taller-than relation, etc. Each can be given in terms of a function, for example one that takes the licker and lick-ee one at a time.

First: you can just as well think of a relation in terms of the corresponding characteristic function, i.e. a function from pairs into 1 or 0:

$$\begin{bmatrix} (A, A) \to 0 \\ (A, B) \to 1 \\ (A, C) \to 1 \\ (B, A) \to 0 \\ (B, B) \to 0 \\ (B, C) \to 1 \\ (C, A) \to 0 \\ (C, B) \to 0 \\ (C, C) \to 0 \end{bmatrix}$$

From there it's a small step to **Currying/Schönfinkelization**: any $n$-ary relation can be turned into an $n$-place function. There are two ways to do this:

$$\begin{bmatrix} A \to \begin{bmatrix} A \to 0 \\ B \to 1 \\ C \to 1 \end{bmatrix} \\ B \to \begin{bmatrix} A \to 0 \\ B \to 0 \\ C \to 1 \end{bmatrix} \\ C \to \begin{bmatrix} A \to 0 \\ B \to 0 \\ C \to 0 \end{bmatrix} \end{bmatrix} \qquad \begin{bmatrix} A \to \begin{bmatrix} A \to 0 \\ B \to 0 \\ C \to 0 \end{bmatrix} \\ B \to \begin{bmatrix} A \to 1 \\ B \to 0 \\ C \to 0 \end{bmatrix} \\ C \to \begin{bmatrix} A \to 1 \\ B \to 1 \\ C \to 0 \end{bmatrix} \end{bmatrix}$$

**Left-to-right** $l \to r \to (l, r) \in R$ **Right-to-left** $r \to l \to (l, r) \in R$

The L position in a relation is (by convention) associated with subjects. The R position is (by convention) associated with objects. Thus:

- Left-to-right Currying is faithful to the order of *terminals*.

- Right-to-left Currying is faithful to the order of *combination*.

Compositionality dictates right-to-left Currying.

Notice you can also think of relations as functions into sets

## 4 The great payoff

A single rule for interpretation can get us everything we need today:

$$[\![\text{X Y}]\!] = [\![\text{X}]\!]([\![\text{Y}]\!]) \text{ or } [\![\text{Y}]\!]([\![\text{X}]\!]), \text{ whichever is defined}$$

### 4.1 Node by node compositionality

We can give meanings to every node in every tree we've considered up to now.

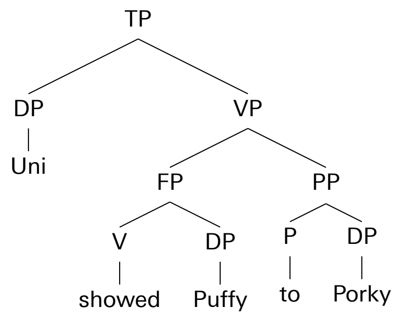
As before, intransitive verbs denote one-place characteristic functions:

$$[\![\text{meow}]\!](x) = 1 \text{iff } x \in \{y : y \text{ meows}\}$$

New piece: transitive verbs denote two-place functions. Saturating one spot gives you something with the same sort of meaning as an intransitive.

$$[\![\text{licked}]\!](y)(x) = 1 \text{ iff } (x, y) \in \{(x, y) : x \text{ licked } y\}$$

Even a ditransitive(!): three-place functions. Saturating one spot gives you something like a transitive. Saturating two gives you something like an intransitive.



$$[\![\text{showed}]\!](z)(y)(x) = 1 \text{iff } (x, z, y) \in \{(x, z, y) : x \text{ showed } z \text{ to } y\}$$

Exercise: if *showed* is a 3-place relation on individuals (as below), what does this suggest about the meaning of *to*?

### 4.2 Back to propositional logic

Currying allows us to think of the propositional logic connectives as functions, which allows us to restate the semantics in a categorematic way! We begin by assigning meanings to our terminals:

- $[\![v]\!]^w \coloneqq w(v)$ if $v$ is an atomic formula
- $[\![\neg]\!]^w(x) \coloneqq 1 - x$
- $[\![\wedge]\!]^w(y)(x) \coloneqq \texttt{Min}(x, y)$
- $[\![\vee]\!]^w(y)(x) \coloneqq \texttt{Max}(x, y)$

- $[\![\Rightarrow]\!]^w(y)(x) \coloneqq \texttt{Max}(1-x,y)$

Each of these values is the functionalization of some set. In the case of negation, the meaning posited is the characteristic function some set (Exercise: which set?). In the case of the binary connectives, the meaning is the Currying of a relation. For example, the case of $\wedge$:

$$\{(x,y) : x = y = 1\} \qquad \begin{bmatrix} (1,1) \to 1 \\ (0,1) \to 0 \\ (1,0) \to 0 \\ (0,0) \to 0 \end{bmatrix} \qquad \begin{bmatrix} 1 \to \begin{bmatrix} 1 \to 1 \\ 0 \to 0 \end{bmatrix} \\ 0 \to \begin{bmatrix} 1 \to 0 \\ 0 \to 0 \end{bmatrix} \end{bmatrix}$$

$R_\wedge$, the $\wedge$-relation  $R_\wedge$ as a characteristic function  $R_\wedge$ as a Curry'd function

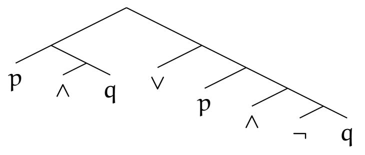
Notice that we invoke the world $w$ in every case, even though the choice of world has absolutely nothing to do with the semantics of the connectives. This is a strategy of **generalizing to the worst case**. It'll make it easier to state our semantics, but it's not crazy to think of it as a cost.

Because our meanings are all specified in terms of the world parameter $w$, the interpretation function for predicate logic needs to be parametrized to $w$. This looks as follows:

$$[\![X\,Y]\!]^w \coloneqq [\![X]\!]^w([\![Y]\!]^w)$$

(We can avoid generalizing to the worst case, as well as the need for a parametrized interpretation function, if we go with the set-theoretic semantics. Exercise: give a compositional, set-theoretic semantics for propositional logic.)

And that's it! We're done. Because the meanings of the connectives take two arguments, the structures implied for complex formulas are necessarily less flat than the structures mooted last class. For example, $(p \wedge q) \vee (p \wedge \neg q)$ would actually receive the following analysis:



We now have all we need to assign a semantics for this structure, relative to any valuation/possible world.

Exercise: do it!