Syntax-driven semantic composition

(Lecture notes for COGS 543)

Umut Özge

November 7, 2017

We start by endorsing *the principle of compositionality*, which says that:

(1) The meaning of an expression is a function of the meanings of its parts and the way they combine.

We use "is a function of" in the mathematical sense: if you know the meanings of basic components and the way they come together, then you know the meaning. Leave aside "the meaning of basic components" for now, but what is meant by "the way they combine?" We all have an intuitive grasp of the principle of compositionality. Take the sequence,

(2) black, coffee, maker

Here are three ways to combine them into a whole:

(3) a. ((black coffee) maker)
b. (black (coffee maker))
c. (black coffee maker)

What the first two means is clear, the third one is a mystery, unless you are smuggling in some inner parentheses. In this tiny example there is more than bringing items together; there is also order in coming together. Now we get a little technical and have a closer look.

Call a process which takes as input something like (2) and produces an output like those in (3) a "combinatory process". If you want to talk about just the output, call it "combination". They are different things. See below.

In characterizing combinatory processes, one crucial parameter is the number of items that you can combine in a single step. We agree on this number to be two,

(4) A combinatory process can combine at most two items at a time,[1]

eliminating (3c), which combines three items in a single step.

[1] Don't get bothered for now by why I wrote "at most" instead of "exactly".

With these conventions at hand, given a sequence of three items, represented as the ordered triple (a, b, c), there are three distinct combinatory processes:[2]

(5) a. First combine a with b, then combine c with the result: $\{\{a, b\}, c\}$;
b. First combine a with c, then combine b with the result: $\{\{a, c\}, b\}$;
c. First combine b with c, then combine a with the result: $\{\{b, c\}, a\}$.

I used set notation in representing the results of combinatory processes, because, for the moment, we do not see combining x with y as different from combining y with x; the arguments to the combination procedure are unordered. But be careful that the combinatory steps are ordered – first combine so and so, *then* so and so.

Also observe that (5b) is different from the other two processes. It combines two items that are not adjacent in the input sequence. Combinatory processes, as we define them for now, are capable of scanning sequences and picking any elements they like.

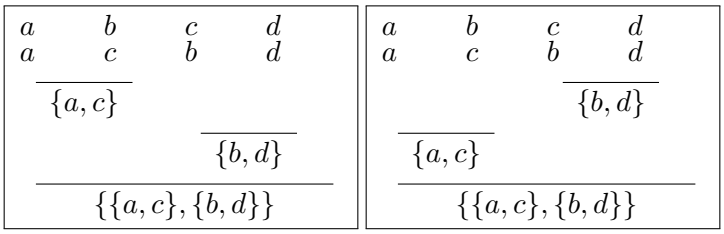
For the three item case, the number of distinct combinatory processes and the number of distinct combinations are equal. It starts to get interesting, when you have a more crowded string of items as input to a combinatory process. Take (a, b, c, d) for instance. Below are two distinct combinatory processes resulting in identical combinations,

(6) a. First combine a with c; then b with d; then $\{a, c\}$ with $\{b, d\}$: $\{\{a, c\}, \{b, d\}\}$.
b. First combine b with d; then a with c; then $\{a, c\}$ with $\{b, d\}$: $\{\{a, c\}, \{b, d\}\}$.

It is in this sense that combinatory processes and combinations are different things; the mapping from processes to combinations is many-to-one.[3]

Now comes some useful notational machinery. First we get rid of this "first combine…then…" idiom. Here is how we represent combinatory processes in (6).

(7)

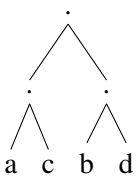


The device we use to be able to combine non-adjacent items is permuting items as we wish, like we did while swapping b and c above.

Combinations can be represented as binary trees, linearly non-ordered (= flipping the branches of any node from left to right does not result in a distinct tree). Here is how we represent (6):

[2] **Question:** How many distinct combinatory processes are there for n items?
[3] **Question:** How many distinct combinations are there for n items?

(8)



Now it is time to answer some questions in the text.

Question: How many distinct combinatory processes are there for n items?

Answer: Given a set of n items, atomic or complex, give them a certain order $\{a_1, a_2, \ldots a_n\}$. If you pick a_1 there are $n-1$ items to combine it with; then pick a_2, don't look back because you already combined a_1 and a_2, so there are $n-2$ possibilities left. Proceeding like this you will end up with $n-1+n-2+\ldots+1$ possibilities, which is $\frac{n(n-1)}{2}$ by Gauss' law. This was the first pass of combinations. After this pass, picking one of the possible combinations, you are left with $n-1$ items in your set. By the same reasoning as above you have $\frac{(n-1)(n-2)}{2}$ possibilities for the second pass. For the first two passes you have $\frac{n(n-1)}{2}\frac{(n-1)(n-2)}{2}$ possibilities. You need to proceed until you have only one item left in your set. Therefore, the number we are after is:

$$\frac{n\prod_{i=1}^{n-1} i^2}{2^{n-1}} = \frac{n!(n-1)!}{2^{n-1}}$$