Constructing Formal Sentences: Atoms and Molecules

Our casual understanding of the formal language sufficed for the simple English examples examined so far. But the limits of such an intuitive grasp are by now familiar: it will easily be overwhelmed by complexity. So despite our informal grasp of formal language, we may, for example, have no good idea which formal sentence properly translates the following complex bit of English.

> Either we won't have both a tax cut and increased spending, or either The House or the Senate will block the spending bill and the President and Fed chair won't both suggest a rate hike.

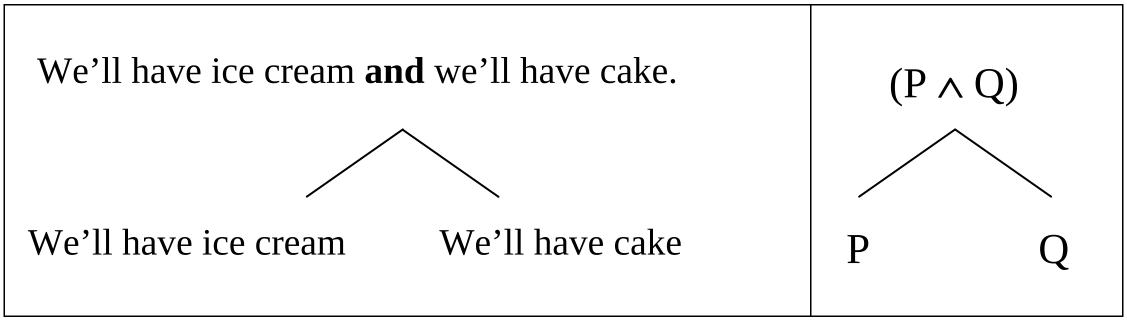
Central to developing a more powerful grasp of the formal language, and English-Formalese translation, is an understanding of how sentences are **constructed**, and the **chemical** method this construction uses.

Just as a chemist synthesizes new compounds out of chemical building blocks, we build new larger sentences in English or Formalese out of smaller, ultimately atomic, parts. Here the **atoms** are the subject matter sentences of English and their formal counterparts, the sentence letters. Larger molecular sentences are built up from these by adding bits of logical form.

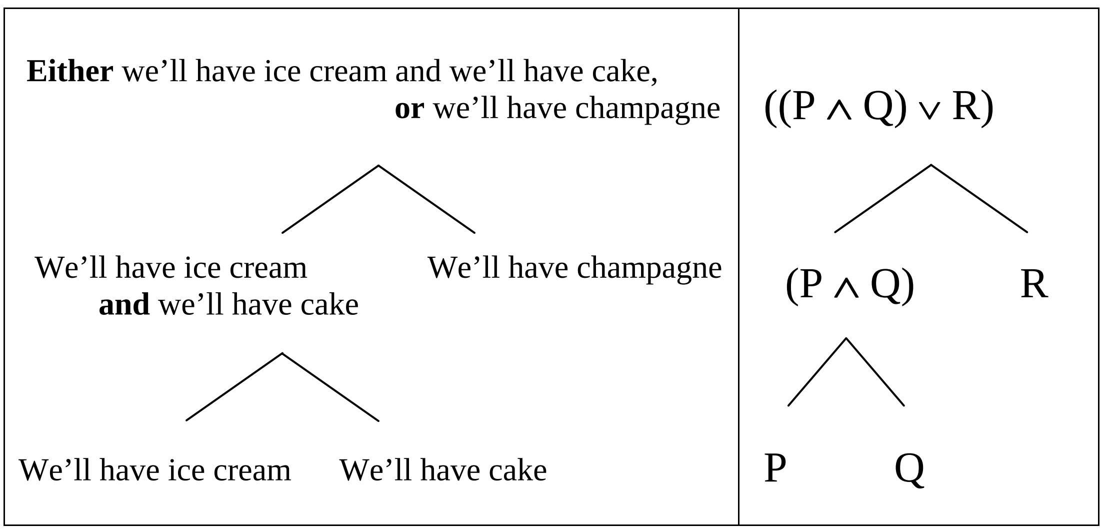
So: where the English negation "It is not sunny" is constructed from the subject matter sentence "It is sunny" by adding the form phrase "not," the formal negation "~P" is built up from the "P" by adding a tilde.

It is **not** sunny	~P
|	|
It is sunny	P

Likewise an English conjunction can be constructed from two subject matter sentences; and this is mirrored in the construction of a formal conjunction out of two sentence letters.



And just as in chemistry, molecules need not be constructed simply out of atoms. Molecules can be built from smaller molecules as well.



Recognizing this potential for molecules-within-molecules, we see that the list of molecular combinations is without limit.

$\sim\sim\sim P$
$(\sim P \vee \sim(Q \wedge \sim\sim R))$
$(P \wedge Q) \vee (P \wedge \sim Q)$
$(P \vee (Q \vee (R \vee (S \vee T))))$
….

And for that reason writing out an exhaustive list of all the different combinations will prove quite hopeless. Given any finite list of combinations, we can always find another sentence not on that list. (We could, for example, add a tilde to the front of every sentence on the list.)

But this endless variety can still be captured in a finite number of general laws – a small number, in fact. Two basic observations are important here: one about atoms, the other about molecules.

First: note that even when building molecules out of smaller molecules (out of smaller molecules…) the process must begin, at bottom, with atomic sentences. Retracing each of the 'construction trees' above, we always find sentence letters as the first step, at the bottom of the trees. No atoms – no molecules.

And since sentence letters are the first step in building any formal sentence, we list them first in our rules for sentence construction.

1. Sentence letters are formal sentences.

Second: for all their variety, our grammatical molecules are really only *recycling* a few tricks over and over. For example, the double-negation "~~P" is a molecule built out of a smaller molecule, "~P".

It did **not** fail to rain yesterday	~~P
|	|
It **fail**ed **to** rain yesterday	~P
|	|
It rained yesterday	P

But in constructing this sentence we really only perform **one** procedure, *twice* over. We **add a tilde** to "P" to get "~P". But then we just **add a tilde** all over again to construct "~~P" out of "~P". (If we wanted to build up "~~~P" in turn, we'd do the same thing with "~~P": **add a tilde**.)

So we don't need *one* rule for constructing single negations, a *second* rule for double negations, and so on. We just need one procedure – adding a tilde – which can be **recycled** as often as we please.

Our second rule for constructing formal sentences captures this 'recycling' feature of negations.

2. If ● is a formal sentence, then **~**● is a formal sentence.

The "●" here is just a blank, where **any** formal sentence can go. (When reading "●" aloud, I pronounce it "**bling**".) Allowing *any* formal sentence as input to this rule is crucial for achieving the desired 'recycling' feature.

Think of sentence construction on analogy with house construction, where all the steps in the construction have to be approved by the city building code. Then "P" is a legal construction, thanks to Rule 1.

1. Sentence letters are formal sentences.

Rather than stopping with this one-story construction, however, we could apply Rule 2 to it. Putting "P" in the "●" blank, Rule 2 says the following.

If **P** is a formal sentence, then **~P** is a formal sentence.

And since "P" is a formal sentence (by Rule 1), Rule 2 approves "~P" as a formal sentence as well.

But (here comes the 'recycling'): since "~P" is itself a formal sentence, and *any* formal sentence can go in the "●" blank of Rule 2, we can place "~P" in that blank – with the following result.

If **~P** is a formal sentence, then **~~P** is a formal sentence.

So "~~P" is an acceptable formal sentence as well.

With these applications of construction Rules 1 and 2, we have simply repeated the construction procedure from our earlier tree.

3. ~~P	From 2, by Rule 2
2 ~P	From 1, by Rule 2
1. P	By Rule 1

But note: though we constructed three different sentences here, we didn't need three different rules to cover these steps. Indeed, we could continue this tree to yield "~~~P," "~~~~P," and all their negated descendants, and still do so using only Rules 1 and 2.

The same 'recycling' approach is used to construct conjunctions and disjunctions. The only difference is that these sorts of sentences have both a left part and right part, while the negation doesn't.

Rule 3 governs construction of conjunctions in the formal language.

> 3. If ● and ▲ are formal sentences, then **(**● ∧ ▲**)** is a formal sentence.

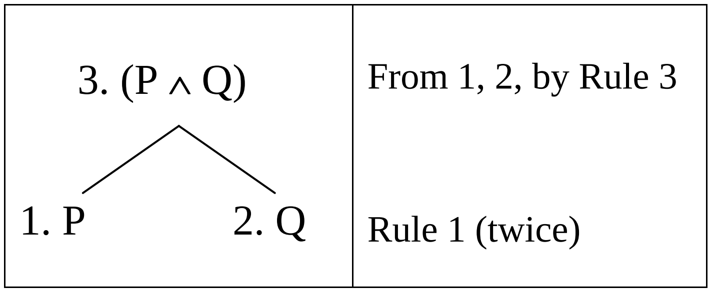
As before, "●" and "▲" are blanks where any formal sentence, big or small, can go. We use two different blank-symbols to allow the left and right parts to be different sentences. (When reading "▲" aloud, I pronounce it "**blah**".)

Putting "P" and "Q" into these blanks, Rule 3 says the following.

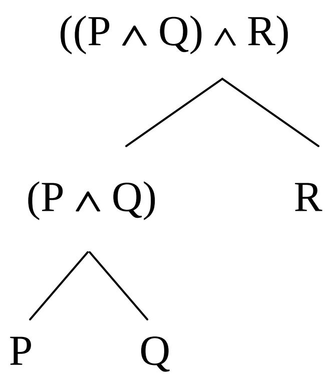
> If P and Q are formal sentences, then (P ∧ Q) is a formal sentence.

Since these sentence letters are indeed formal sentences (from Rule 1), we now have "(P ∧ Q)" as a formal sentence.

In tree format, those moves look like this.



The procedure applied in the last step amounts to (i) putting a wedge between the two 'input' sentences and (ii) wrapping the result in parentheses. Of course we can *recycle* that procedure, applying it to the conjunction just produced.



The formal language thus has a ready counterpart for 'triple-barreled' conjunctions of English like the following.

> Ace is happy and Rex is sad and Zeke is bored.

Understanding how to construct conjunctions, we find no surprises in the rule for constructing **disjunctions**: it differs from the conjunction rule only in the predictable matter of inserting a **vel** rather than a wedge.

> 4. If ● and ▲ are formal sentences, then **(● ∨ ▲)** is a formal sentence.

In combination these four rules allow us to construct an infinite number of different sentences – **all** the sentences of the formal language.

As a matter of professional trivia: the marvelous ‘recycling’ feature found in our three molecular rules usually goes by the technical name “**recursion”**. Our molecule-building rules would thus be called **“recursive rules”**. But we’ll continue to call them “**molecular rules**”.

Finally, a confession: while we’ve spoken throughout of sentence *construction* rules, these are traditionally called rules of **grammar**. And though it’s equally traditional for “grammar” to trigger fear and disgust, we see now that it’s not so bad – just a matter of building by the rules. But to encourage images of Legos® and Lincoln Logs®, rather than eighth grade English class, we’ll continue to speak in terms of “construction”.