2.7. 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. In particular: without a better understanding of formal sentence structure, we won't be sure how to apply our formal test of validity when faced with complex sentences. Central to developing a more powerful grasp of the formal language, and English-to-Formalese translation, is an understanding of how sentences are **constructed**, and the **chemical** method this construction uses.

1. Construction. Just as a chemist synthesizes new compounds out of chemical building blocks, we build new larger sentences in English or the formal language 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: just as the English negation "It failed to rain yesterday" is constructed from the subject matter sentence "It rained yesterday" by adding the form phrase "fail to," the formal negation "~P" is built up from "P" by adding a tilde.

It **failed to** rain yesterday	~P
|	|
It rained yesterday	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.

We'll have ice cream **and** we'll have cake. $(P \wedge Q)$

We'll have ice cream We'll have cake P Q

But crucially: just as in chemistry, molecules here need not be constructed simply out of atoms. Molecules can be built from *smaller molecules* as well. That means we can have, e.g., negations within negations.

It did **not** fail to rain yesterday $\sim\sim P$

It **fail**ed **to** rain yesterday $\sim P$

It rained yesterday P

And we find conjunctions within conjunctions.

We'll have ice cream and we'll have cake, **and** we'll have champagne $((P \wedge Q) \wedge R)$

We'll have ice cream **and** we'll have cake We'll have champagne $(P \wedge Q)$ R

We'll have ice cream We'll have cake P Q

The ability to have sentences within sentences (within sentences…) threatens to remove any upper limit to the complexity of sentences – meaning there is literally *no end* to the sentence complexity we might encounter. 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, negate each sentence on the list. One of the resulting negations is certainly *not* on that list.)

Happily, however, this unlimited 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 yet 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 tree. No atoms – no molecules.

And since the atoms – the 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 "~~P" we really only perform **one** procedure, *twice* over. We **add a tilde** to "P" to get "~P". But then we just **add a tilde** again to construct "~~P" out of "~P". (If we wanted to build up "~~~P" in turn, we'd do the same thing to "~~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 rule for constructing negations captures this 'recycling' feature.

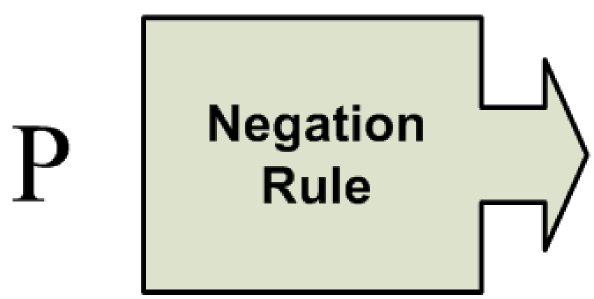
> 2. If ▲ is a formal sentence, then ~▲ is a formal sentence.

The "▲" here is just a blank, where **any** formal sentence can go. (When read aloud, "▲"is pronounced "**blah**".) 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.

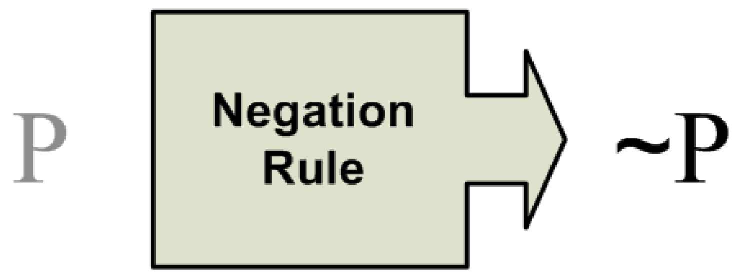
Rather than stopping with this one-story construction, however, we apply Rule 2 to it.



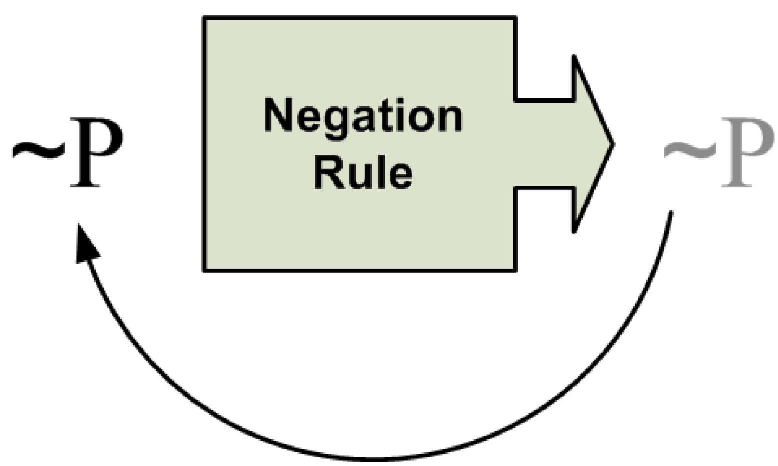
Putting “P” in the “▲” blank, Rule 2 says the following.

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.

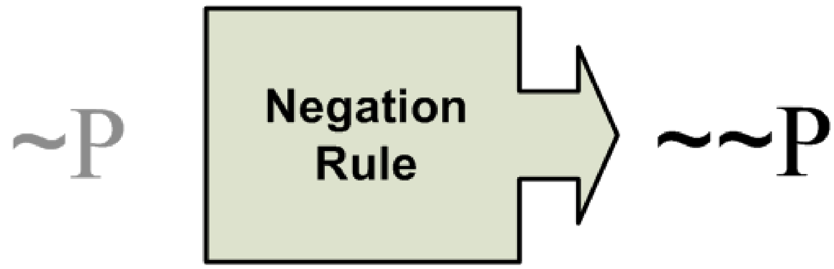


But (here comes the ‘recycling’): since “~P” is itself a formal sentence, and **any** formal sentence can go in the “▲” blank of Rule 2, we can place “~P” in that blank.

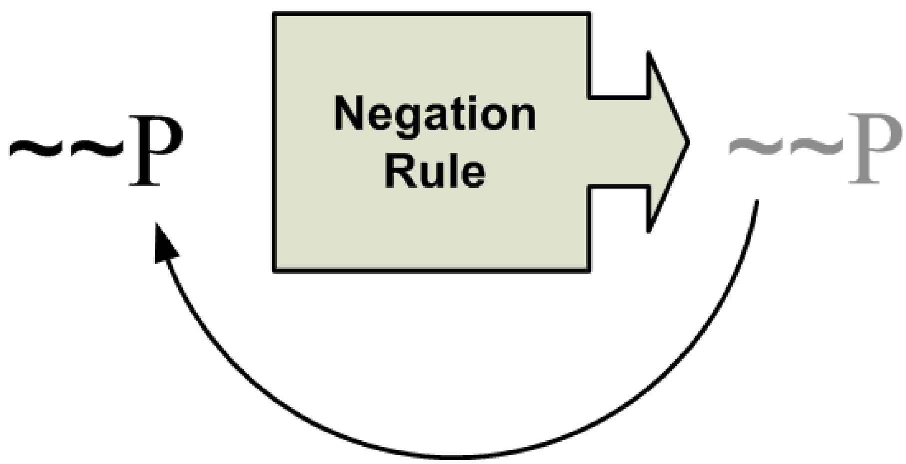


Since the Negation Rule always does the same thing – add a tilde to the left – with “~P” as input the Negation Rule yields “~~P” as output.

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

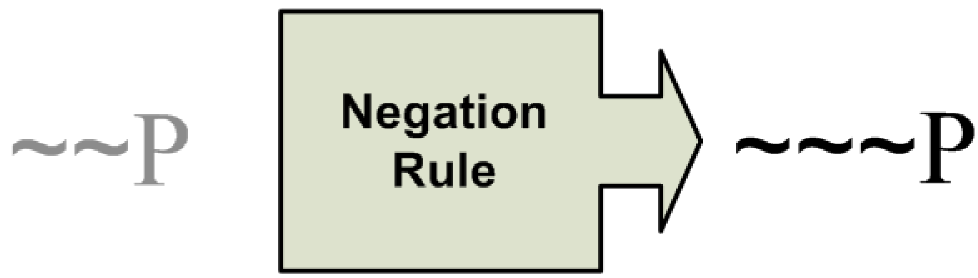


So “~~P” is a legal formal sentence as well. And we can just keep going: since “~~**P**” is a formal sentence, it **can serve as fresh input** for the Negation Rule.



The Negation Rule adds a tilde to the left, as always. With “~~P” as input that yields “~~~P” as output.

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



All these applications of construction Rules 1 and 2 can be diagrammed much more briefly in the familiar ‘construction tree’ format, like so.

4.	~~~P	(From 3, by Rule 2)
3.	~~P	(From 2, by Rule 2)
2.	~P	(From 1, by Rule 2)
1.	P	(By Rule 1)

And note: though we constructed four different sentences here, we didn’t need four 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 **(**● $\wedge$ ▲**)** 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. (“●”is pronounced “**bling**”.)

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

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

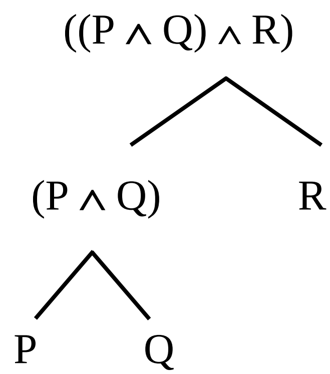
Since these sentence letters *are* indeed formal sentences (from Rule 1), we now have “(P $\wedge$ Q)” as a formal sentence.

In ‘construction tree’ format, those moves look like this.

	3. (P $\wedge$ Q)	From 1, 2, by Rule 3
1. P	2. Q	Rule 1 (twice)

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 very conjunction just produced.



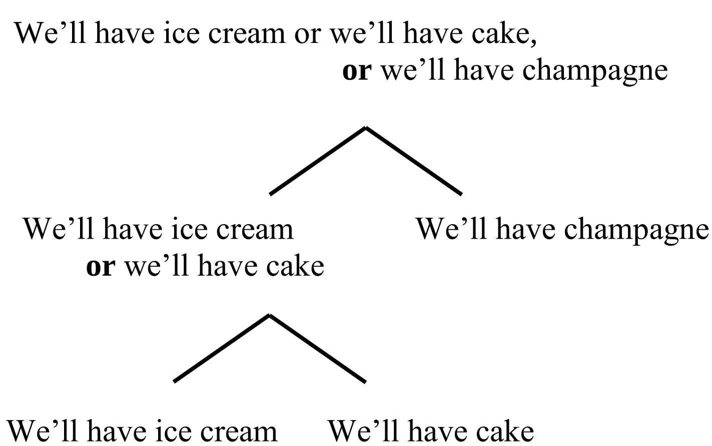
The formal language thus has a ready counterpart for ‘triple-barreled’ conjunctions of English like the following.

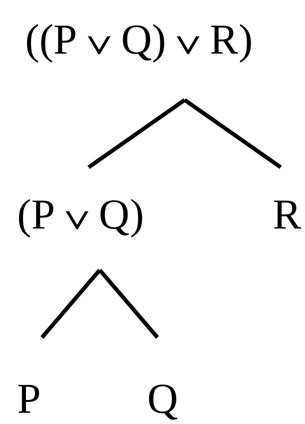
> Barbie is happy and Kitty is sad and Trixie is bored.

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.

Two ‘cycles’ on this construction rule provide ‘triple-barreled’ disjunctions – something we find in English as well.





2. Grouping and Associativity. 'Triple-barreled' English conjunctions and disjunctions pose a potential puzzle about formal translation. Consider again the earlier 'triple-barreled' conjunction.

Barbie is happy and Kitty is sad and Trixie is bored.

We translated this sentence as a formal conjunction, with a smaller conjunction "$(P \wedge Q)$" as its left part.

$$((P \wedge Q) \wedge R)$$

But should we instead have treated it as a conjunction with a smaller conjunction "$(Q \wedge R)$" as its right part?

$$(P \wedge (Q \wedge R))$$

It turns out either translation is equally acceptable – for it is a happy feature of 'triple-barreled' conjunctions that **how we group the parts will not affect truth or falsehood of the conjunction** (and so will not affect the validity of an argument featuring that conjunction). English examples make this intuitively clear. Whenever it's true that Rex attended and both-Nick-and-Nora attended, it's true that both-Rex-and-Nick attended and Nora attended (and vice versa).

The same moral holds for 'triple-barreled' disjunctions: whenever it's true that we're either having ice cream, or we're having cake or pie, it will be true that we're having ice cream or cake, or we're having pie (and vice versa). So the 'triple-barreled' English disjunction can equally well be translated as either of the following two formal sentences.

We're having ice cream or cake or pie.

$$((P \vee Q) \vee R)$$
$$(P \vee (Q \vee R))$$

This welcome feature of conjunctions and disjunctions has the technical name "**associativity**". But we could just as well call it the "irrelevance of grouping" for conjunction and disjunctions.

3. Conclusion: Recursion. As a matter of professional trivia: the marvelous 'recycling' feature found in our three molecular rules goes by the technical name "**recursion"**. Our molecule-building rules are thus **recursive rules**. In what follows we'll treat as interchangeable the phrases "molecular rules" and "recursive rules," and likewise the phrases "recursive" and "recycling".

And finally, a confession: while we've spoken throughout of sentence construction rules, these are traditionally called rules of **grammar**.[1] 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".

Construction Rules for the Chapter Two Formal Language[2]

Atomic Sentences:

1. Sentence letters are formal sentences.

Molecular Sentences:

2. If ▲ is a formal sentence, then ~▲ is a formal sentence.

3. If ● and ▲ are formal sentences, then **(**● $\wedge$ ▲**)** is a formal sentence.

4. If ● and ▲ are formal sentences, then **(**● $\vee$ ▲**)** is a formal sentence.

---

[1] Grammar is in turn sometimes called "**syntax**," and construction rules are then called "**syntactic rules**".

[2] It is traditional when stating recursive construction rules for a language to add a 'closure clause' stating that **only** sentences made by these rules count as legal sentences of the language. Here and in the chapters to follow such a clause is assumed without mention.