## 3.8. Construction Trees Revisited (and Reversed)

Construction rules serve as a building code for formal sentences: just as city building codes set out the conditions for constructing a house that passes legal muster, so our construction rules state how to build a string of symbols that counts as a genuine sentence in the formal language (rather than just a string of formal gibberish).

But building codes can also be used to inspect a house that's already constructed – say, as part of selling that house. And our four construction rules can be applied likewise to assess a finished sentence whose construction we may not have witnessed. In that case we begin with the finished sentence, and hang its construction tree under it to prove it was constructed legally.

Recovering the construction tree in this way involves 'un-building' the sentence – **performing the construction process in reverse**. Since construction began with atoms and subsequently used three molecule-building rules, un-building uses those same construction rules in reverse as *molecule-dissolving* procedures, leading back to the original atoms.



1. Sentence letters are formal sentences.
2. If ● is a formal sentence, then ~● is a formal sentence.
3. If ● and ▲ are formal sentences, then **(● ∧ ▲)** is a formal sentence.
4. If ● and ▲ are formal sentences, then **(● ∨ ▲)** is a formal sentence.

Specifically: since each molecular rule adds a connective (and – in the case of wedges and vels – a pair of parentheses), in reverse each molecular rule **removes a connective** (and with wedges and vels, a pair of parentheses).

The trick here is to decide where the un-building should begin – that is, which connective should be removed first. Let us call the last connective added in the construction process the **main connective** of that sentence. So in the following sentence the tilde is the main connective, as it was the last one added in construction of "~(P ∧ Q)".

$$\sim(P \wedge Q) \qquad \text{Rule 2}$$

$$(P \wedge Q) \qquad \text{Rule 3}$$

$$P \qquad Q \qquad \text{Rule 1 (twice)}$$

The **left-most symbol** turns out to be a reliable clue as to which rule applied last in construction (and hence which connective is the main connective of the sentence). For the output of Construction Rule 2 has a tilde as left-most symbol; whereas Construction Rules 3 and 4 leave a left parenthesis as left-most symbol.

> 2. If ● is a formal sentence, then ~● is a formal sentence.
>
> 3. If ● and ▲ are formal sentences, then **(**● ∧ ▲**)** is a formal sentence.
>
> 4. If ● and ▲ are formal sentences, then **(**● ∨ ▲**)** is a formal sentence.

The above sentence “$\sim(P \wedge Q)$” has a tilde as left-most symbol; and that alone tells us it’s a negation, the output of Rule 2. Reading its construction tree from top to bottom outlines its un-building: Rule 2 in reverse removes a tilde, yielding “$(P \wedge Q)$,” a conjunction produced by Rule 3; and Rule 3 in reverse removes a wedge and parentheses, leaving the two sentence letters “P” and “Q”. (Sentence letters can’t be un-built with any molecular rule in reverse, since they have no connectives to remove.)

Consider next a sentence without a construction tree.

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

Being a molecular sentence, “$((P \wedge Q) \vee \sim R)$” must have been the output of one of the three molecular rules. And since the left-most symbol here is a parenthesis, it could only be the output of Rule 3 or Rule 4.

As a matter of fact this sentence is a **disjunction**, the product of Rule 4.[1]

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

Its main connective is thus a vel – the very connective that brought those parentheses with it.

$$\underline{\mathbf{(}}\,(P \wedge Q)\ \underline{\vee}\ {\sim}R\,\underline{\mathbf{)}}$$

From this output we work back to the two inputs, by applying Rule 4 in reverse. Since Rule 4 adds a vel and outer parentheses, Rule 4 in reverse **removes** a vel and outer parentheses.

> **Rule 4 in Reverse**: remove a vel, and the outermost pair of parentheses.

This leads us back to the two sentences being linked together by the vel.

$$((P \wedge Q) \vee {\sim}R)$$

$$(P \wedge Q) \qquad\qquad {\sim}R$$

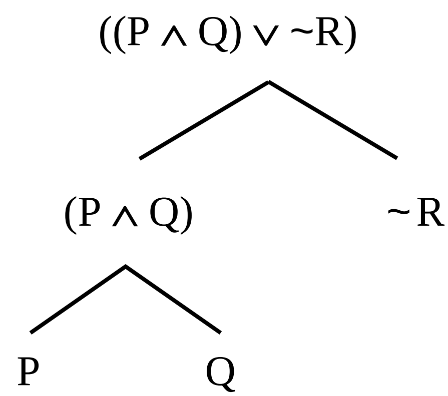
[1] A procedure for showing mechanically *why* the vel, and not the wedge, is the main connective here is addressed in 3.8.1. Problem C.

The left part, “(P ∧ Q),” is a smaller molecule with a wedge as it main connective. “(P ∧ Q)” is the product of Rule 3, the conjunction rule.

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

Rule 3 in reverse **removes** a wedge and outer parentheses.

> **Rule 3, in Reverse**: remove the outermost pair of parentheses, and take a conjunction sign from between the two parts.



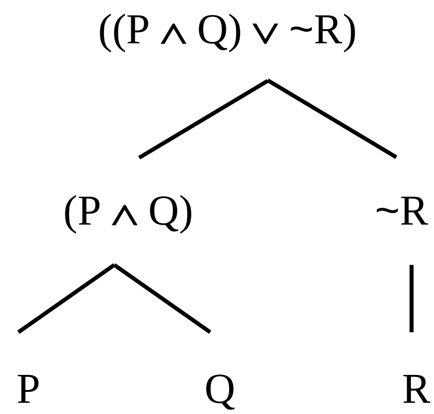

Since “P” and “Q” are atoms, they cannot be un-built by any molecular rule in reverse.

But “~R,” on the right of the tree, is a molecule susceptible to disassembly. “~R” has a tilde as its left-most symbol – meaning it’s a negation, built by Rule 2.

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

Rule 2 in Reverse removes a tilde from the left.

**Rule 2 In Reverse**: remove a tilde from the left of the sentence.

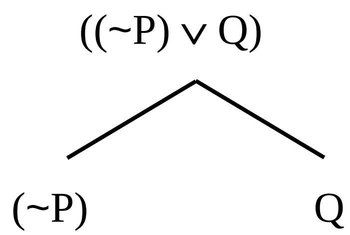



This illustrates the general strategy for recovering the construction tree for any formal sentence: break down the sentence using the three molecular rules in reverse, until only atomic sentences (sentence letters) remain.

It turns out that any genuine formal sentence can be un-built, by the molecular rules in reverse, to just sentence letters. And anything which isn't a formal sentence can't be un-built back to sentence letters in this way.[2]

[2] If we simply apply the molecular rules blindly in search of a proper construction tree (without determining which connective is the **main** connective), then we must rather say: a genuine formal sentence will yield a construction tree with sentence letters at the bottom – along with however many (bogus) trees that don't have sentence letters at the bottom. (For instance, by using Rule 3 in reverse to remove a wedge and outermost parentheses, we could un-build the perfectly fine formal sentence "$((P \wedge Q) \vee R)$" into "(P" and "$Q) \vee R$" – neither a sentence letter, and neither susceptible of further un-building.) By contrast, on this approach anything which isn't a formal sentence will yield **not even one** tree resolving it into sentence letters. How the reverse construction procedure could be sharpened to apply less blindly – yielding just the one correct tree for a formal sentence – is addressed in 3.18.1, Problem C.

For instance, the following string of symbols is a piece of formal gibberish, as a reverse construction tree shows.




By removing a wedge and outer parentheses, Rule 3 in reverse can certainly un-build “((~P) ∨ Q)” into parts “(~P)” and “Q”. But no molecular rule in reverse can un-build “(~P)”. Because the left-most symbol is a left parenthesis, Rule 2 can’t apply. Rule 3 in reverse removes a pair of parentheses and a wedge – meaning it can’t apply to “(~P),” which has no wedge to remove. For the same reason Rule 4 can’t be used here, since in reverse it removes parentheses and a vel. Since molecular rules in reverse can’t un-build it entirely to sentence letters, “((~P) ∨ Q)” is shown not to be a genuine formal sentence, but mere symbolic gibberish.

Reverse construction trees have a variety of applications. Separating genuine formal sentences from their pseudo-sentence imposters is, as we’ve just seen, one use. And recovering the construction tree for a formal sentence will later prove essential to the truth table test of validity.

But we stress finally that understanding how a formal sentence is constructed also makes clear that sentence’s **logical meaning** – whether, for instance, “~(P ∧ Q)” is denying a conjunction, or asserting one with a denial as a part.[3] For that is crucial to proper translation from English to Formalese – a topic to which we now return.

[3] It’s the denial of a conjunction – because its main connective is a tilde.