2.37. Deduction Strategy

In chess, knowing how the pieces operate is sufficient to guard against illegal moves, but provides no strategy for winning games. Something similar holds for deductions: if we apply the inference rules in just any old order, each move will be a valid link in a meandering chain to nowhere. But a bit of strategy is all that's needed to reliably find deductions of arguments.

1. The Importance of The *Elim* Rules.

The Intro (✚) rules share a powerful and potentially troublesome feature: each can be applied **an unlimited number of times**.

Given the sentence "P," for instance, ∨✚ can be applied repeatedly, yielding new sentences such as the following.

> (P ∨ Q)
> (P ∨ R)
> ((P ∨ R) ∨ Q)
> ….

Repeated application of ~✚ to "P" likewise yields an unlimited number of sentences.

> P
> ~~P
> ~~~~P
> ….

Even ∧✚ can generate an unlimited number of sentences from "P".

> P
> (P ∧ P)
> ((P ∧ P) ∧ P)
> ….

In light of this 'unlimited applicability,' it would be a strategic disaster to apply Intro rules whenever possible. So as a matter of general deductive strategy we will *not* apply the Intro rules indiscriminately – instead using them only with some specific purpose in mind.

By contrast, when faced with a finite set of sentences the Elim (–) rules **can't** be used an unlimited number of times. So, starting with the sentences "(P ∨ Q)" and "~P," only one instance of an Elim rule is available: an application of ∨–**,** yielding "Q".

1. (P ∨ Q)
2. ~P
3. Q 1, 2, ∨–

There are no ~– or ∧– applicable to sentences (1) and (2), and no ∨– beyond the one already executed. Since Elim rules lack 'unlimited applicability,' applying them automatically incurs no disastrous consequences.

Indeed, it's strategically shrewd to apply Elim rules whenever possible, without bothering about *why* they're being applied. For doing so will often allow us to back unthinkingly into the desired conclusion.

So at the outset of the following deduction we search automatically for any occasion to apply Elim rules.

1. ((P ∨ Q) ∧ ~R)
2. (R ∨ ~Q)

———————— Get: P

Since Premise (1) is a conjunction, we apply ∧– twice to get the left and right parts of the conjunction.

1. ((P ∨ Q) ∧ ~R)	Premise
2. (R ∨ ~Q)	Premise
————————	Get: P
3. (P ∨ Q)	1, ∧–
4. ~R	1, ∧–

No further cases of ∧– are to be had here. But with "~R" now available on line (4), we can use it with line (2) to apply ∨–.

1. ((P ∨ Q) ∧ ~R)	Premise
2. (R ∨ ~Q)	Premise
	Get: P
3. (P ∨ Q)	1, ∧–
4. ~R	1, ∧–
5. ~Q	2, 4, ∨–

And "~Q" on line (5), along with line (3), – yields a new opportunity for ∨–.

1. ((P ∨ Q) ∧ ~R)	Premise
2. (R ∨ ~Q)	Premise
	Get: P
3. (P ∨ Q)	1, ∧–
4. ~R	1, ∧–
5. ~Q	2, 4, ∨–
6. P	3, 5, ∨–

Note two important points here. **First**, we indiscriminately applied the Elim rules wherever possible. No clever strategy was involved, really no thought at all about where the deduction is headed – just a blind, automatic scouring of lines for chances to use an Elim rule.

Second, despite proceeding so automatically, on line (6) we end up with "P" – exactly the sentence we set out to get (as the "Get" line reminds us). So the deduction is complete.

1. ((P ∨ Q) ∧ ~R)	Premise
2. (R ∨ ~Q)	Premise
	~~Get~~: P
3. (P ∨ Q)	1, ∧–
4. ~R	1, ∧–
5. ~Q	2, 4, ∨–
6. P	3, 5, ∨–

This illustrates a general strategy for doing deductions: as soon as a deduction begins, it’s extremely useful to use Elim rules as many times as possible. For often (as in this last example) that’s all we need to back our way into the conclusion.

Deduction Strategy: Automatically use Elim rules whenever possible.

Note further that of the three Elim rules, ∧– is by far the most useful, since ∧– yields *both halves* of the conjunction it’s applied to, and with no help from a second sentence.

By contrast, ∨– can’t do anything with just a disjunction – it needs a second sentence as well (the negation of one of the parts). And even with that second sentence, ∨– yields only one of the parts – not both, like ∧–.

Likewise ~– can’t be applied to just any old negation – only to a *double* negation. And while ~– doesn’t need help from a second sentence, it yields only one new sentence.

Following this ranking, we look *first* for cases of ∧–. Only when these are exhausted do we look for cases of ∨– and ~–.

Deduction Strategy: In the automatic search for cases of the Elim rules, look **first** for cases of ∧–. Do ∨– and ~– afterwards.

To this we add a further, quite general observation: each new sentence obtained from applying an Elim rule can change our deductive opportunities – potentially opening a further opportunity to apply an Elim rule.

This was clear already in our last example. Once "~R" was obtained from ∧– on line (4), it served as input for ∨– on line (5). We couldn't have deduced "~Q" from "(R ∨ ~Q) by ∨– until we had that missing second ingredient, "~R".

1. ((P ∨ Q) ∧ ~R)	Premise
2. (R ∨ ~Q)	Premise
	Get: P
3. (P ∨ Q)	1, ∧–
4. ~R	1, ∧–
5. ~Q	2, 4, ∨–

So after each use of an Elim rule we scan the available lines again, looking for *new* Elim opportunities that may have opened up.

Deduction Strategy: following each use of an Elim rule, scan for new opportunities to use an Elim rule.

2. "Set It Up": Using the *Intro* Rules.

While the Elim rules form the heart of our deductive activity, they're not always sufficient to complete a deduction. Then we need to use Intro rules as well.

Here's a simple example.

1. (P ∨ Q)	Premise
2. (R ∨ S)	Premise
3. (~Q ∧ ~S)	Premise
	Get: (P ∧ R)

Scouring automatically for chances to use Elim rules, we spot an opportunity for $\wedge$– on line (3).

1. $(P \vee Q)$	Premise
2. $(R \vee S)$	Premise
3. $(\sim Q \wedge \sim S)$	Premise
	Get: $(P \wedge R)$
4. $\sim Q$	3, $\wedge$–
5. $\sim S$	3, $\wedge$–

Lines (1) through (5) offer no further occasions for $\wedge$–. Turning to $\vee$–, we find two opportunities. First we use $\vee$– on lines (1) and (4).

1. $(P \vee Q)$	Premise
2. $(R \vee S)$	Premise
3. $(\sim Q \wedge \sim S)$	Premise
	Get: $(P \wedge R)$
4. $\sim Q$	3, $\wedge$–
5. $\sim S$	3, $\wedge$–
6. P	1, 4, $\vee$–

Next we use $\vee$– on lines (2) and (5).

1. $(P \vee Q)$	Premise
2. $(R \vee S)$	Premise
3. $(\sim Q \wedge \sim S)$	Premise
	Get: $(P \wedge R)$
4. $\sim Q$	3, $\wedge$–
5. $\sim S$	3, $\wedge$–
6. P	1, 4, $\vee$–
7. R	2, 5, $\vee$–

At this point there are no further Elim opportunities, yet we still don’t have the sentence on the “Get” line, “$(P \wedge R)$”. What now?

While we don't have "(P ∧ R)," we do have two close relatives: "P" on line (6) and "R" on (7). Since these are the left and right halves of "(P ∧ R)," we have all the ingredients to **build the needed sentence**, using the Intro rule ∧+. While we refrain from indiscriminately applying ∧+, use of ∧+ here would be far from indiscriminate; for it yields the very sentence we're seeking. Hence we permit ourselves use of ∧+ here.

1. (P ∨ Q)	Premise
2. (R ∨ S)	Premise
3. (~Q ∧ ~S)	Premise
	Get: (P ∧ R)
4. ~Q	3, ∧–
5. ~S	3, ∧–
6. P	1, 4, ∨–
7. R	2, 5, ∨–
8. (P ∧ R)	6, 7, ∧+

That completes the deduction.

This illustrates the basic use we have for an Intro rule like ∧+: to 'set up' the desired sentence when we have the necessary parts – here, the very sentence on the "Get" line.

Deduction Strategy: if there are no further opportunities to use the Elim rules, and the sentence on the "Get" line hasn't been obtained, try to build that sentence from available sentences using an Intro rule.

A second use for Intro rules is **setting up Elim rules**. For sometimes we lack the sentences needed for an Elim rule, but do have a close relative of the sentence(s) needed. In that case an Intro rule can close the gap.

Here’s a trivial example.

1. (P ∨ Q)	Premise
2. (~Q ∨ R)	Premise
3. ~P	Premise
————————	Get: R

We first scan for cases of Elim rules. Only one can be found.

1. (P ∨ Q)	Premise
2. (~Q ∨ R)	Premise
3. ~P	Premise
————————	Get: R
4. Q	1, 3, ∨–

Now we’re out of openings for Elim rules, yet haven’t obtained our desired sentence, “R”. And since “R” is a sentence letter, it’s not the sort of sentence an Intro rule could build up for us.

But we can at least use an Intro rule to supply the **missing ingredient for an Elim rule**. Note that we have a still-unused disjunction on line (2), “(~Q ∨ R)”. To use ∨– on this disjunction, we need the negation of one of its parts – either the negation of the left part, “~ ~Q”, or the negation of the right part, “~R”.

Now while line (4), “Q,” isn’t itself the negation of the left part “~Q”, “Q” can take us to that negation through a simple application of the Intro rule **~✚**. As a matter of general strategy we don’t *randomly* add pairs of tildes. But here applying **~✚** is for a good cause: setting up an Elim rule.

1. (P ∨ Q)	Premise
2. (~Q ∨ R)	Premise
3. ~P	Premise
————————	Get: R
4. Q	1, 3, ∨–
5. ~ ~Q	4, ~✚

With “~ ~Q” in hand we have all the ingredients for a new case of ∨–, from lines (2) and (5).

1. (P ∨ Q)	Premise
2. (~Q ∨ R)	Premise
3. ~P	Premise
	~~Get~~: R
4. Q	1, 3, ∨–
5. ~ ~Q	4, ~✚
6. R	2, 5, ∨–

The deduction is then complete, and we cross off the “Get” line.

This is the second use we have for Intro rules: setting up an Elim rule.

Deduction Strategy: if there are no opportunities to use Elim rules, and the sentence on the “Get” line still hasn’t been deduced, try building the missing ingredient for an Elim rule through use of an Intro rule.

Note that we **only** use this strategy to set up cases of ∨–.

The reason is simple: ∧– takes only one sentence as input, a conjunction. But to build a conjunction through the Intro rule ∧✚, we already need both halves of the conjunction as inputs. Since those two sentences are all that ∧– would give us anyway, the whole ‘set up’ procedure would be pointless.

Adding a pair of tildes with ~✚, in order to then take them away with ~–, would be equally pointless.

Keep in mind: we fall back on Intro rules **only** when we’ve exhausted the Elim rules and still fall short of our goal in the deduction. That, together with our preference for using Elim rules, summarizes our deductive strategy.

General Deduction Strategy:

- Start by using Elim rules as many times as possible (checking after each use to see if you’ve backed into the the sentence on the “Get” line). Among these Elim rules, use ∧+ first.

- If the Elim rules are exhausted without obtaining the sentence on the “Get” line, try using Intro rules to either (i) build that sentence out of available lines, or (ii) set up a new use of ∨–.