❖ *English Language, Formal Language* ❖

3.2. Formal Language, Informally

The proposed formal test of validity requires that for any argument we do the following.

1. Get the form of the argument.
2. Test that form for validity.

Here we begin mastering the first task: isolating an argument's logical skeleton, beneath the irrelevant flesh of subject matter.

1. Logical Form: Four Examples. Our 'logical x-ray' for isolating form is a special-purpose language which speaks *only* about logical form – call it "the formal language," or "Formalese." If an English argument with both logical form and subject matter is translated into this formal language, only the form of the argument will survive the translation. Since the formal language has no words for anything but form, the subject matter will be 'lost in translation'.

So we've taken our first task – "Get the form" – and broken it down into two smaller tasks.

1. Get the form of the argument.
    1a. Build a language of pure logical form.
    1b. Translate from English into the formal language.

(And then:
2. Test the form.)

We've already seen examples of the sort of logical form such a language would discuss. For instance, we stripped the subject matter words from the following argument.

1. Either the test is on Monday, or the test is on Wednesday.
2. The test is not on Monday.

(So,) The test is on Wednesday.

And that left us with this rough logical skeleton.

Either ● or ▲ .
Not ● .

(So,) ▲ .

(Once again, ● and ▲ are just blanks marking the spots where the subject matter sentences used to be.) Though all the subject matter words are gone here, the words "either... or" and "not" remain – suggesting that they are the rare sort of English words discussing logical form, not subject matter.

We likewise strip away the subject matter words from this argument.

1. It's warm and it's sunny.

(So,) It's warm.

But the word "and" remains – providing another example of logical form.

● and ▲ .

(So,) ● .

We build a list of examples of logical form – the sorts of things a language of form should have words for.

Four Examples of Logical Form:

1.
2. Not
3. And
4. Either… or

A blank space remains in the list, for one additional example of logical form: marking when different spots in the argument are occupied by the **same subject matter sentence**, and when by **different** subject matter sentences.

To see why that's essential to logical form, compare two different arguments. The first is a valid English argument, with a sketch of its familiar logical form.

Either we're having tacos for dinner, or <u>we're having chicken for dinner</u>.	Either ● or ▲ .
We aren't having tacos for dinner.	Not ● .
(So,) <u>We're having chicken for dinner</u>.	(So,) ▲ .

As the underlining emphasizes, the same subject matter sentence "We're having chicken for dinner" appears as both the right half of the first premise, and as the conclusion. We recognized this in the logical form by putting the same symbol, "▲," in both spots.

The next argument, by contrast, is clearly *invalid* – its conclusion does *not* follow from its premises.

Either we're having tacos for dinner, or <u>we're having chicken for dinner</u>.	Either ● or ▲ .
We aren't having tacos for dinner.	Not ● .
(So,) <u>It's snowing</u>.	(So,) ★ .

But the only difference between this argument and its earlier, valid cousin is that here the same subject matter sentence does *not* appear as both the right half of the first premise and the conclusion. (We recognize this in the

logical form by putting one symbol, "▲," in the right half of the first premise, and another, "★," as the conclusion.)

Now since these arguments differ in validity, and we assume the only factor affecting an argument's validity is its logical form, these arguments must differ in logical form. But the only difference between them is whether or not the same subject matter sentence appears in both spots. So: **when it's the same subject matter sentence in both spots, and when it's not, makes a difference to the logical form**.

Valid	**Invalid**
Either ● or ▲ . Not ● . ————	Either ● or ▲ . Not ● . ————
(So,) ▲ .	(So,) ★ .

(Note: the subject matter itself doesn't affect the validity of the argument – only when it's the *same* subject matter sentence in different spots, whatever that subject matter may be. So in the valid form on the left, having "▲" in two different spots showed that the same subject matter sentence appeared in both places. But "▲" says nothing about chicken, since what the subject matter is in those two spots is *not* a matter of logical form.)

Adding this missing example of form completes our list.

Four Examples of Logical Form:

1. When it's the same subject matter sentence as before, when not.
2. Not
3. And
4. Either… or

These are the sorts of things which our language of logical form should talk about.

2. Outline of the Formal Language. Running through the list in order, our formal language *first* needs a way of marking when different spots have the same subject matter sentence, when not. Using geometrical shapes for this task, as we’ve done so far, proves impractical, since for arguments with a large number of subject matter sentences we will soon run out of distinct, easily-drawn shapes to mark those different sentences.

Instead we use capital letters “P” through “Z”. And though this provides only 11 markers, we can add numerical subscripts to them (for example: “P_1,” “P_2,” “Z_{256}”) to get infinitely many different markers. For obvious reasons we call these **sentence letters**.

> **Sentence letters**: capital letters “P” through “Z,” with or without numerical subscripts.

As we’ll see, linking sentence letters with subject matter sentences provides the translation bridge from English to Formalese. So an essential first step in such a translation will be establishing a ‘translation dictionary’ linking sentence letters with subject matter sentences – as in the following example.

> **P**: We’re having tacos for dinner
> **Q**: We’re having chicken for dinner

We call such a dictionary a **translation table**.

Second on our list: a way to say “not” in the formal language. For this we introduce the following symbol, called the **tilde**.

> **~**

While in English proper positioning of “not” can be tricky, Formalese keeps matters simple: whenever we want to deny a formal sentence, we attach a tilde to the *left edge* of that sentence. So using the above translation table,

we can translate the denial "We're *not* having tacos for dinner."

P: We're having tacos for dinner

We're **not** having tacos for dinner

~P

Third on our list: a way of saying "and" in the formal language. For this purpose we introduce the **wedge**.

$\wedge$

Just as the English "and" comes between two English sentences, linking them together, so the wedge comes between and links two formal sentences – as the following example illustrates.

P: We're having tacos for dinner
Q: We're having chicken for dinner

We're having tacos for dinner **and** we're having chicken for dinner

$(P \wedge Q)$

Note that with the wedge we wrap parentheses outside the formal sentence: left parenthesis outside the left part, "P," right parenthesis outside the right part, "Q". Parentheses will later prove essential for understanding more complicated formal sentences.

We only add parentheses when a sentence has *both a left and a right part*. (By contrast, we didn't add parentheses to the tilde sentence "~P" – since a tilde doesn't link together left and right parts.) Here is a simple memory aid.

Left and right *parts*: left and right *parentheses*

Finally, we need a way of expressing “either… or” in formal language. We do this with the **vel**.

$\vee$

“Either… or” links together left and right sentences. The vel does the same, as in this example.

P: We’re having tacos for dinner
Q: We’re having chicken for dinner

Either We’re having tacos for dinner, **or** we’re having chicken for dinner

$(P \vee Q)$

Because a vel joins together left and right parts, the vel takes left and right parentheses outside these parts.

In logic jargon the symbols “~,” “$\wedge$,” and “$\vee$” are called **connectives**. Connectives are the formal language counterpart to the form phrases of English. (Parentheses don’t count as connectives; they’re just punctuation.)

Though this much understanding of the formal language might seem sufficient to capture the logical form of English sentences, in fact it will serve only for the simplest cases – more complex cases posing an obstacle for this casual grasp of the formal language. In the sections that follow we improve on this situation through a two-pronged strategy: cataloguing the stylistic complications of English, and developing a better understanding of the mechanics of Formalese.