IFL: Logicbite 9

PL languages introduced

Peter Smith

The final sections of *IFL* Chapter 8 are pivotal. §8.6 gestures towards why logicians are so concerned with formalized languages. §8.7 gives the design brief for what I'll be calling PL languages, apt for exploring the logic of the (tidied-up) connectives 'and', 'or' and 'not'. §8.8 takes a first look at one PL language.

I should highlight one point of contrast with many elementary introductions to propositional logic. Initially, our formal languages will have just the three built-in connectives, symbolized $\wedge, \vee, \neg$. Almost always, a fourth basic connective is introduced at the same time, the conditional 'if ..., then ...', symbolized $\rightarrow$ or $\supset$. However, as we will see, the question of how to deal with the ordinary-language conditional raises troublesome questions I just don't want to tangle with at this early stage. So I will avoid complications and keep things simple by concentrating at the outset on just those three core connectives, leaving the conditional until later. (Of the textbook authors I have mentioned so far, only Paul Teller also takes this sensible line!)

❦

Following up the design brief in §8.7 we now have a chapter on the syntax or grammar of PL languages followed by a chapter on semantics. In other words, we first have Chapter 9 which tells us officially which strings of symbols make sentences of a PL language as opposed to mixed symbol-salads. Then Chapter 10 tells us about how to give meaning to such sentences and how to assign them truth-values.

Unlike the previous Logicbites, I'm *not* going to pepper this one with introductory quotations from other authors. That's because everyone's presentation of the material is slightly different, with some divergences being trivial, and others a little more significant. So I think it will be less confusing if you *first* read the presentation in *IFL* Chapters 9 and 10. Then, in the next Logicbite, when you are in a better position to make comparisons, I'll comment on the ways that a few other authors do things, highlighting where we have made a couple of more-than-stylistic choices.

❦

A very few reading notes on Chapter 9:

(1) Remember the announcement that I made at the beginning of §8.7. Earlier, in the pre-formal chapters of *IFL*, we used italic letters like 'A', 'B', 'C' in contexts where we want to generalize about sentences in arguments. Now, when we want to generalize about sentences of artificial languages such as PL languages, we are going to be using letters like 'α', 'β', 'γ'. I explain why we go Greek in Chapter 11.

(2) Example. In §9.1, one rule for building up more complex PL sentences from simpler ones is this: putting '$\wedge$' between two PL sentences and enclosing the result in brackets gives us another PL sentence. But we can say the same more snappily like this (where 'wff' abbreviates 'well-formed formula'):

> If α and β are wffs, so is $(\alpha \wedge \beta)$.

Note, the Greek letters here do *not* belong to our PL languages. The Greek letters are symbols we are using for talking *about* PL languages in a perspicuous way.

(3) At the foot of p. 73, I very briefly say what it is for a wff to 'have the form $(\alpha \wedge \beta)$'. I should have made a bit more of a fanfare in introducing that idea. The idea, I hope, is quite clear from what I do say. A wff has that form if it is the result of writing a left

bracket followed by some wff α followed by '$\wedge$' followed by some wff β followed by a right bracket. Similarly, a wff has the form $\neg\alpha$ if it is the result of writing a negation sign followed by some wff α. I'm just saying I should have really highlighted this usage of 'form'!

(4) Perhaps §9.2 is unnecessarily longwinded, and makes things look messier than they need be. For I could probably have equally clearly gone straight from the 'constructional history' $\mathbf{A}$ to the 'parse tree' $\mathbf{A}^P$ (missing out the intermediate $\mathbf{A}^T$). [On the other hand, as things stand, I do make use of trees of the intermediate kind in the following chapter; but maybe I should have done everything in terms of parse trees.]

(5) Again, maybe I should have highlighted (perhaps literally, in one of those grey panels) the important though obvious remarks about 'form' and 'main connectives' that are indented near the bottom of p. 77.

❦

Chapter 10 can usefully be carved into four parts.

(1) In §§10.1 and 10.2 we talk about interpreting the basic sentences of a PL language with a glossary, assigning meaning to the 'P's and 'Q's, the building blocks of the language. Sentences built up from the basic blocks using the connectives then get derived meanings, depending on the stipulated meaning of the connectives.

(2) In §§10.3 and 10.4 we stipulate that basic sentences of a PL language get sufficiently sharp meanings for them to be determinately true or false as the case may be in any situation. (So in translating an everyday argument involving everyday propositions into a suitable PL language, we must e.g. artificially ignore questions of vagueness. It is a nice question, as they say, whether we can adequately deal with vagueness in a formal logical way: but we certainly can't in PL languages.)

(3) Given our stipulations, in any possible circumstance, the basic sentences of a PL language get determinate truth-values. In §§10.5, 10.6 and 10.7 we explain how the truth values of basic sentences will then fix the truth values of more complex PL sentences built up using our three connectives – every sentence of the PL language gets a unique value, true or false.

(4) I'd suggest reading the sections up to this point twice, to fix the key ideas in your mind. §10.8 is then an unexciting and inevitably fiddly afterword explaining one conventional way of 'setting out your working' when determining the truth value of a complex sentence given the truth values of its basic constituents.

❦

Now read *IFL* Chapters 9 and 10.