5.5. Quantified Predicate Logic: Three Formal Languages

Having set out the parts of the Chapter Five formal language, complete with quantifiers and variables, experience from past chapters suggests we now survey translation variations for this expanded language, then semantic (and later deductive) methods for the language.

But quantifiers and variables bring a level of complexity to the formal language well beyond anything we've encountered before – both to the construction rules and translation techniques, and to the semantics that (as before) rides along the rails of those construction rules. With greater expressive power comes greater opportunity for confusion and error; and with quantifiers and variables added the risk of such confusion looms large.

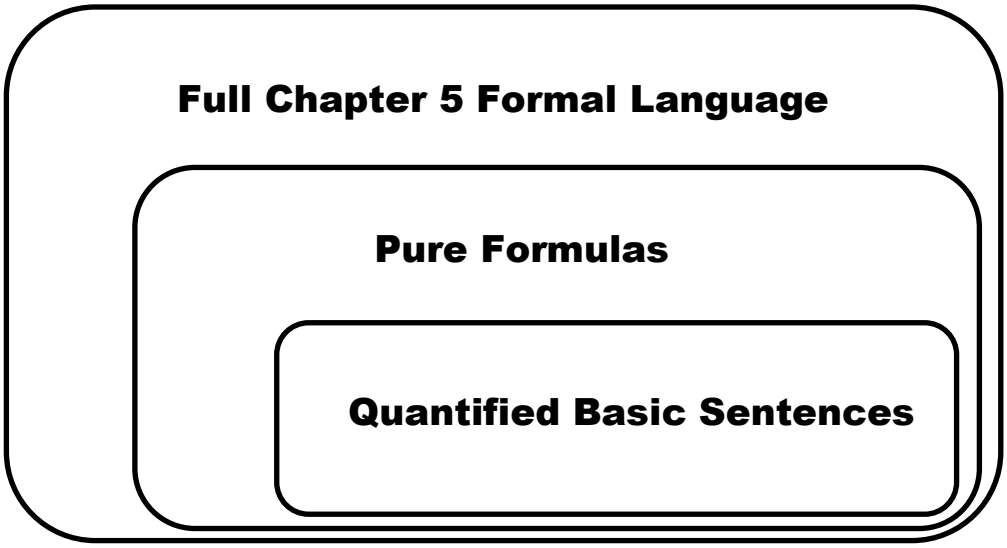
That's why we'll ease our way into this formal language, reaching the full language of Chapter Five only after passing through two smaller 'sub-languages' where sentence construction, and the quantifiers it brings, are more (and more) limited.

We begin with the smallest of the three, the language of **Quantified Basic Sentences**. In this little sub-language, quantification is so tightly constrained that there's little room for error or confusion.

From there we proceed to a more inclusive sub-language allowing quantifiers much larger scope – but still reining them in enough to avoid combinations of quantifiers that are trickier to work with (and to understand). This is the language of **Pure Formulas**.

Finally we throw out all the stops, developing construction rules and semantic techniques which apply to the **full formal language** of Chapter Five, no matter how complex the sentences involved – and, in particular, no matter how many quantifiers cluster or interact together.

Our roadmap is as follows.



It turns out that approaching quantified logic in this stepwise fashion brings further advantages, beyond a paced introduction to formal complexity.

For one thing, in previous chapters we've mastered English-to-formal translation before tackling semantics. But there's so much material to cover on formal translation here that following that order would mean an especially long delay before applying semantic tests (and an especially long trudge through the Valley of the Shadow of Formal Translation before even beginning to consider formally what those sentences mean). The complexity of mapping English sentences onto their proper formal translation will often require thinking carefully about which formal sentence means the same as the English original; so it's useful to get a precise semantic grasp of formal sentences as soon as possible.

Moreover, we'll end up using one common, quite simple, semantic framework which applies to all three formal languages. That's because we use a special family of sentences, called **instances**, to link quantified sentences to models. And as we graduate from one language to the next, all that needs to change is our account of instances for that (sub-)language, in order to apply the same semantic rules as before.

For instance, we'll use one set of **truth tree** rules, the same for all three formal languages, without needing to change those rules as we move from one language to the next. For the truth tree rules make generic appeal to

instances; and our account of instances is what changes, to correspond with our change of formal language.

The result is an ability to apply semantic methods (including truth trees) to quantified sentences early on, and to continue using those methods unchanged as we trade simpler (sub-)languages for more complex ones. We can take the formal language in doses, and semantically understand what we're doing at each step.