4.C: the ***PhilosophyIdeas*** overview of

Predicate Calculus

The development of Propositional Logic (PL) resulted in a very successful system for reasoning about complete propositions (those capable of being true or false). There were four main connectives – 'and' [**∧**], 'not' [**¬**], 'or' [**v**], and 'if…then' [→], and a set of rules specifying when each connective can be introduced into a proof, or eliminated from the proof. For example, if you have used p to prove q, you can introduce '→', by writing 'p→q'. If you have 'p' and 'p→q', you can eliminate '→' by just writing 'q'. The new system, however, left out the subjects and predicates which syllogistic logic successfully dealt with, it couldn't handle relations, and it wasn't suited to reasoning about science and mathematics, which wanted to refer to entities with properties, and then generalise about them.

The revolutionary move was the introduction of mathematical '**variables**' into descriptions of sentences. Instead of 'the sun is yellow', you can write 'x is yellow', or 'the sun is F', or even 'Fx'. A sharp distinction is then made between objects and predicates, and 'quantifiers' are introduced to specify the range of values assigned to the variables (rather as the old logic talked of 'all' and 'some'). The two quantifiers applied to the variable 'x' are the 'existential' quantifier (written '∃**x**'), and the 'universal' quantifier (written '∀**x**'). The first one reads as 'there is an entity 'x' which….', and the second one reads as 'for all the entities assignable to 'x'…..'. At first the universal quantifier seemed to cover every possible entity, but this raised problems, and the settled version of Predicate Calculus (PC) has a '**domain**' of objects available as values for the variables. Thus you can reason about the domain of natural numbers, or electrons, or elks, without worrying about the metaphysical status of things outside that domain. It was accepted that a domain could not be empty, because that would make the variables meaningless. The language described here is 'first-order' PC, because the quantified variables (x, y, …) range over objects, but we could also range over the predicates (F, G, …), which would make it 'second-order' PC.

Just as we have introduction and elimination rules for the connectives of Propositional Logic, so we can now add similar rules for the two **quantifiers**. '∃x(Fx)' means 'there exists an entity which is F', so to introduce '∃' we just need to find something in the domain which is F. If we know 'Fa' (the object 'a' is F), then we have Existential Quantifier Introduction. To eliminate the quantifier, we can say that the quantifier implies the existence of some object, so we can prove something new from the object alone, and drop the quantifier.

If we say 'no thing fails to be F', that is the same as saying 'everything is F', and if we say 'not everything fails to be F', that means 'something is F'. Hence you can define the two quantifiers in terms of one another. Nevertheless, we also have rules to introduce and eliminate the universal quantifier '∀'. If you are able to prove a proposition P(x) for one value of x, and in that proof no assumption at all was made about the entity assigned to x, then we can introduce '∀xP(x)'. We can then return from 'P for all values of x' to 'so obviously P for one value of x', and eliminate the universal claim. This way of describing the logic (as 'natural deduction') gives secure rules for the use of all the symbols in the language.

A common mode of proof is as before, to assume that what we want to prove is *not* true, and then show that such an assumption leads to contradiction, so it must have been true after all. Proofs are usually laid out one step at a time, with an indication of which rules and assumptions are being applied at each step. There is also 'conditional proof', which establishes that if one thing is assumed to be true, then some other thing follows from it. It is also helpful to construct branching tree structures for the proofs (known as 'tableaux'), since some parts of proofs (involving 'or') need parallel lines of proof.

Once this system of logic was established, it was first proved to be 'consistent' (meaning its theorems were never contradictory), and then that it was 'complete', in the sense that every valid sentence in the language can be proved. This made the system sufficiently secure, and it has all of expressive power of syllogistic and propositional logic, and more, so that it is often now referred to as '**classical logic**'. It can even deal with relations, as special forms of predicate, so that 'Rxy' means 'x has the R relation to y'. Fans say that this system is the best logic, and maybe even 'the one true logic', but critics develop non-standard logics which challenge some of the basic assumptions.

Predicate Calculus is designed in order to prove things, and is particularly successful in that regard for the logic of mathematics, but its more common use in philosophy is to express the 'logical form' of natural language sentences. For example, if we refer to 'The Queen of Sheba', there is only one such queen, and we want to say so. We can just introduce a symbol meaning 'unique', but we can also write '∃x(Qx ∧ ∀y(Qy → y=x))', which says there is an entity x which is Q ('queen'), and if any other entity y is Q, then it's the same as x. That is, there is only one queen here. This is a bit unfriendly and long-winded, but it expresses the idea of uniqueness very precisely, using just the simple vocabulary of the logical language. The symbolic sentence we just used included the '=' sign, which is a standard addition to the language. The sign enhances the expressive power of the language, such as expressing precise numbers of entities, rather than the two basic quantifiers. The rules for '=' allows an assumption-free introduction of a=a (self-identity), and a=b can be eliminated by making the substitution of 'b' for 'a' in some formula.

The universal and existential quantifiers differ in one important respect: the existential quantifier assumes that there is an object in the domain to which it refers, while the universal quantifier may offer a generalisation which actually has no instances. Thus if we say 'the President is short' we can write ∃x(Px ∧ Sx), because the quantifier means there is such a person, but if we say 'all hobbits are short' we must write ∀x(Hx → Sx), using the conditional, which only says that *if* there are hobbits (and there may not be), then they are all short.

If you wish to study philosophy in the analytic tradition, you must study predicate logic. This is not because you are expected to continually prove things, but because it is the language to which philosophers turn when they try to increase the precision of their claims. The language can sort out ambiguities, pin down precisely what you are talking about (the 'scope' of your claims), and show exactly which elements of our talk are used in our reasoning.