PY1003: Introduction to Logic
Lecture 9
Predicate Logic I

1. The limitations of sentential logic

Our primary aim in this module is to find a way of distinguishing between valid arguments and invalid ones. Sentential logic, with its truth-tables and truth-trees, has provided us with many resources for doing this. Unfortunately, these resources have their limits. Consider the following argument:

<u>David is rich</u>
Therefore, someone is rich

It is intuitively clear that this argument is valid. There is no way that it can be true that David is rich and yet false that someone is rich. Yet sentential logic does not enable us to capture this validity. To see this, consider how the argument would be represented in the language of sentential logic. We adopt the following translation scheme:

A: David is rich
B: Someone is rich

This enables us to formalise the argument as follows.

A |- B

This argument form, however, is invalid. This can easily be seen be considering the following interpretation I(A)=T, I(B)=F. This interpretation provides a counterexample to the argument. The moral is that sentential logic is not a language rich enough to capture all valid argument forms. It is time to learn a more complicated language: the language of *predicate logic.*

2. Names and predicates

Consider the sentence:

(1) David is rich

This sentence consists of two components. The first is a *name*, "David" which refers to an object. The second is a *predicate*, "is rich", which ascribes to the object a certain property. In predicate logic, we use lowercase letters from the beginning of the alphabet to symbolise names. We use uppercase letters to symbolise predicates. Using the letter 'a' for "David' and the letter 'R' for 'is rich', we express (1) as follows:

(1') Ra

In predicate logic, we use the same logical constants as we use in sentential logic. Suppose, for instance, that wanted to formalise the following sentence:

(2) If David is not rich then Victoria is not happy

Writing 'b' for 'Victoria' and 'H' for 'is happy', we offer the following translation:

(2') $\neg Ra \rightarrow \neg Hb$

The lowercase letters that we use to express names are called *individual constants.* The uppercase letters that we use to express predicates are called *predicate letters.*

3. Individual variables and the existential quantifier

We now need to work out how to symbolise the sentence:

(3) Someone is rich

To do this, note that a more complicated way of expressing (3) is as follows:

(3') There exists at least one person who is such that he/she is rich.

This sentence can, in turn, be expressed by the following sentence:

(3'') For some x, x is rich

As we saw above, we can use the uppercase letter 'R' to express the predicate 'is rich'. This enables us to rewrite (3'') as follows:

(3''') For some x, Rx.

To complete the translation, we introduce a new symbol: '∃'. This is called the existential quantifier symbol, and has the meaning 'For some...' Using this new symbol we can finish our translation of (3) into the language of predicate logic:

(3'''') ∃xRx

Here is some more terminology. The letter 'x' is the sentence above is called an *individual variable.* The letters 'y' and 'z' can also be used to express individual variables. So (3) could also be expressed, without loss of meaning, as follows: '∃yRy' and '∃zRz'. Sentences such as these, which begin with the existential quantifier, are called *existentially quantified sentences.*

4. Some more existentially quantified sentences

In English, the existential quantifier can be expressed in a number of ways. Here are some examples of some existentially quantified sentences in English, with an explanation of how we translate them to into predicate:

Something is green: ∃xGx
There exist tables: ∃xTx
At least one person is wise: ∃xWx

Notice that in all of these cases, I've used the initial letter of the English predicate to use as a predicate letter. For instance, "x is <u>W</u>ise" is symbolised as <u>W</u>x. This is common practice in predicate logic.

5. Existential quantifiers and compound sentences

We can use logical constants to combine sentences containing existential quantifiers. However, we need to be careful. Consider the sentence:

(4) Someone is happy and someone is rich.

We express this as follows:

(4') ∃xHx ∧ ∃xRx

However, (4) should be carefully distinguished from

(5) Someone is happy and rich

Notice that (5) and (4) clearly say different things, because (4) could be true when (5) is false: suppose that David is rich but sad, that Victoria is happy but poor, and that no-one else exists. We express (5) as follows:

(5’) $\exists x\ (Hx \wedge Rx)$

One way of capturing the difference between (4’) and (5’) is as follows. In (4’) the main connective is the conjunction symbol, which has the whole sentence in its scope. In (5’), however, the main connective is the existential quantifier, which ranges over the whole sentence.

Other compound sentences involving existential quantifiers are as follows:

Someone is happy or rich: $\exists x\ (Hx \vee Rx)$
If there is a happy person, then there is a rich person: $\exists xHx \rightarrow \exists xRx$
There is something that is neither green nor purple: $\exists x\ (\neg Gx \wedge \neg Px)$

(6) Negated existential claims

Suppose that we want to make a negated existential claim. We want to claim, for instance, that

(6) Nobody is happy

To assert (6) is simply to deny that there exists someone who is happy. So we translate (6) by negating the relevant existentially quantified claim:

(6’) $\neg\exists xHx$

Notice the difference between (6) and the distinct claim

(7) Somebody is not happy

which we symbolise as

(7’) $\exists x \neg Hx$

Clearly (6’) and (7’) are not logically equivalent, since (7’) could be true when (6’) is false: suppose that Victoria is not happy, but that David is.

5. Arguments in predicate logic

Recall the argument with which we began this lecture:

<u>David is rich</u>
Therefore, someone is rich

We are now at a stage where we can write out this argument in the language of sentential logic:

$Ra \vdash \exists xRx$

This formalisation reflects the arguments logical form. Since the validity of an argument depend upon its logical form, we know that any argument that shares this form will also be valid. For instance, we can be sure of the validity of the following argument:

<u>Grass is green</u>
Therefore, something is green

What we don’t have yet is the means to *demonstrate* that either of these arguments are valid. We will be developing such a method in forthcoming lectures.

6. Summary and further reading

In this lecture, we've been introduced to some of the basic components of the language of predicate logic. These are as follows:

The individual letters: 'a', 'b', 'c''….
The individual variables: 'x', 'y', 'z'
The predicate letters: P, Q, R…
The existential quantifier: ∃

I'd strongly recommend that you supplement the information provided in the lecture with some further reading. Useful texts include:

Colin Howson, *Logic With Trees*, ch.5, sec 1-2
Graeme Forbes, *Modern Logic,* ch.5, sec. 1-2