NOTES ON RELATIONS AND FOPL

GODFREY MUGANDA

1. Cartesian Products and Relations

Let D be a set.

The set of all ordered pairs $\{(a, b) \mid a \in D \text{ and } b \in D\}$ is called the *Cartesian product* of D with itself, and is denoted by $D \times D$.

$$D \times D = \{(a, b) \mid a \in D \text{ and } b \in D\}$$

Ofcourse you can have the Cartesian product of two sets:

$$A \times B = \{(x, y) \mid x \in A \text{ and } y \in B\}$$

You can have the Cartesian product of any number of sets:

$$A_1 \times A_2 \times A_3 \ldots, \times A_n$$

Basically a Cartesian product of sets is the set of all ordered tuples of elements drawn from those sets.

A *relation* is a subset of a Cartesian product. A *binary relation* is a set a subset of the cartesian product of two sets. Similarly you can have $n - ary$ relations for any integer $n \geq 1$. A unary relation on a set S is just a subset of S.

Notice that you can think of a subset $S \subseteq D$ as a boolean-valued function defined on D:

$$S : D \to \{T, F\}$$

where the $S(x) = T$ if and only if $x \in S$. Here we have used the same symbol S for both the subset $S \subseteq D$ and the function $S : D \to \{T, F\}$.

Similarly, and n-ary relation $R \subseteq D_1 \times D_2 \times \cdots \times D_n$ can be viewed as a boolean function

$$S : D_1 \times D_2 \times \cdots \times D_n \to \{T, F\}.$$

A boolean-valued function is called a *predicate*.

2. First Order Predicate Logic

The idea behind First Order Predicate Logic (FOPL) is to have a system of framing valid arguments, and know that those arguments are valid quite apart from the meaning of the statements involved in those arguments.

For example, $P \implies P$ is valid regardless of what the meaning of P or what it stands for.

If you have predicate logic, you have the possibility of allowing a computer to determine the validity of an argument, because an argument will be valid if it is constructed according to certain rules.

FOPL is a symbolic language that allows you to construct statements and arguments about some domain of discourse, say a set D.

FOPL uses n-ary predicate symbols to represent possible relations on the set D, the domain of discourse.

If $S(x)$ is a unary predicate, it can be used to represent a subset of D.

Similarly, n-ary predicate symbols can be used to represent n-ary relations on D.

Predicate logic also allows n-ary *function symbols* for each $n \geq 0$. Whereas predicate symbols are used to represent relations on the domain, function symbols are used to represent functions on the domain.

So for example, a binary function symbol f would represent a function $f : D \times D \to D$.

For example, you might interpret a predicate $M(x, y)$ to represent the relation that x is married to y, where x and y are elements of the domain D.

You might also interpret a function symbol $f(x)$ to mean the "father of " function. In this case $M(f(a), b)$ would mean that the father of a is married to b.