1.2 The predicate Calculus

In propositional calculus, each atomic symbol (P, 0, etc.) denotes a proposition of some complexity. There is no way to access the components of an individual assertion. Predicate calculus provides this ability. For example, instead of letting a single propositional symbol, P denotes the entire sentence "it rained on Tuesday," we can create a predicate weather that describes a relationship between a date and the weather:

weather (tuesday, rain). Through inference rules we can manipulate predicate calculus expressions, accessing their individual components and inferring new sentences.

Predicate calculus also allows expressions to contain variables. Variables let us create general assertions about classes of entities. For example, we could state that for all values of X, where X is a day of the week, the statement weather(X, rain) is true; i.e., it rains every day. As with propositional calculus, we will first define the syntax of the language and then discuss its semantics.

DEFINITION

PREDICATE CALCULUS SYMBOLS

The alphabet that makes up the symbols of the predicate calculus consists of:

1. The set of letters, both upper- and lowercase, of the English alphabet.
2. The set of digits, 0, 1, ...,9.
3. The underscore, _.

Symbols in the predicate calculus begin with a tetter and are followed by any sequence of these legal characters.

Legitimate characters in the alphabet of predicate calculus symbols include

a R 6 9 p_z

Examples of characters not in the alphabet include

% @ / & ""

Legitimate predicate calculus symbols include

tom_and_jerry bill XXXX friends_of Ali George fire3

Examples of strings that are not legal symbols are

3jack "no blanks allowed" ab%cd ***71 duck!!!

Predicate calculus symbols may represent either variables, constants, functions, or predicate. Constants name specific objects or properties in the world. Constant symbols must begin with a lowercase letter. Thus george, tree, tall, and blue are examples of well-formed constant symbols. The constants *true* and *false* are reserved as truth symbols.

Variable symbols are used to designate general classes of objects or properties in the world. Variables are represented by symbols beginning with an uppercase letter. Thus George, BILL, and KAte are legal variables, whereas geORGE and bill are not.

Predicate calculus also allows functions on objects in the world of discourse. Function symbols (like constants) begin with a lowercase letter. Functions denote a mapping of one or more elements in a set (called the *domain* of the function) into a unique element of another set (the *range* of the function). Elements of the domain and range are objects in the world of discourse. In addition to common arithmetic functions such as addition and multiplication, functions may define mappings between nonnumeric domains.

DEFINITION

SYMBOLS and TERMS

Predicate calculus symbols include:

1. *Truth symbols* true and false (these are reserved symbols).

2. *Constant symbols* are symbol expressions having the first character lowercase.
3. *Variable symbols* are symbol expressions beginning with an uppercase character.
1. *Function symbols* are symbol expressions having the first character lowercase.

Functions have an attached arity indicating the number of elements of the domain mapped onto each element of the range.

A function expression consists of a function constant of atiry n, followed by n terms, t1,t2 tn, enclosed in parentheses and separated by commas.

A predicate calculus *term* is either a constant, variable, or function expression. Thus, a predicate calculus *term* may be used to denote objects and properties in a problem domain. Examples of terms are:

cat
times(2,3)
X
blue
mother(jane)
kate

Symbols in predicate calculus may also represent predicates. Predicate symbols, like constants and function names, begin with a lowercase letter. A predicate names a relationship between zero or more objects in the world. The number of objects so related is the arity of the predicate, Examples of predicates are

likes equals on near part_of

An *atomic sentence*, the most primitive unit of the predicate calculus language, is a predicate of arity n followed by n terms enclosed in parentheses and separated by commas.

Examples of atomic sentences are:-

likes(george, kate)	Iikes(X,george)
likes(george,susie)	Iikes(X,X)
likes(george, sarah,tuesday)	friends(bill, richard)
friends(bill,george)	friends(father_of(david),father_of(andrew))
helps(bill ,george)	helps(richard,bill)

The predicate symbols in these expressions are likes, friends, and helps. A predicate symbols may be used with different numbers of arguments. In this example there are two different likes, one with two and the other with three arguments. When a predicate symbol is used in sentences with different arities, it is considered to represent to different relations. Thus, a predicate relation is defined by its name and its arity.

DEFINITION

PREDICATES and ATOMIC SENTENCES

Predicate symbols are symbols beginning with a lowercase letter.
Predicates have an associated positive integer referred to as the *arity* or "argument number" for the predicate. Predicates with the same name but different arities are considered distinct.
An atomic sentence is a predicate constant of arity n, followed by n terms, t1, t2 ..., tn, enclosed in parentheses and separated by commas.

The truth values, true and false, are also atomic sentences.
Atomic sentences are also called *atomic expressions, atoms,* or *propositions.* We may combine atomic sentences using logical operators to form *sentences* in the predicate calculus. These are the same logical connectives used in propositional calculus:

$\wedge$, $\vee$, $\neg$, $\rightarrow$ and = .

When a variable appears as an argument in a sentence, it refers to unspecified objects in the domain. First order predicate calculus includes two symbols, the *variable quantifiers* $\forall$ and $\exists$, that constrain the meaning of a sentence containing a variable. A quantifier is followed by a variable and a sentence, such as

$\exists$ Y friends(Y, peter)

$\forall$ X likes(X, ice_cream)

The *universal quantifier,* $\forall$*,* indicates that the sentence is true for all values of the variable. In the example, $\forall$ X likes(X, ice_cream) is true for all values in the domain of the definition of X. The *existential quantifier,* $\exists$, indicates that the sentence is true for at least one value in the domain. $\exists$ Y friends(Y, peter) is true if there is at least one object, indicated by Y that is a friend of peter.

DEFINITION

PREDICATE CALCULUS SENTENCES

Every atomic sentence is a sentence.

1. If s is a sentence, then so is its negation, $\neg$s.
2. If s1, and s2 are sentences, then so is their conjunction, s1 $\wedge$ s1
3. If s1 and s2 are sentences, then so is their disjunction, s1 $\vee$ s2'
4. If s1 and s2 are sentences, then so is their implication, s1 $\rightarrow$ s2
5. **If s1 and s2 are sentences, then so is their equivalence, s1 = s2·**

6. If X is a variable and s a sentence, then $\forall X$ is a sentence.

7. If X is a variable and s a sentence, then $\exists X$ is a sentence.

Several relationships between negation and the universal and existential quantifiers are given below. These relationships are used in resolution systems. The notion of a variable name as a dummy symbol that stands for a set of constants is also noted. For predicates p and q and variables X and Y:

$\neg \exists X\ P(X) \equiv \forall X \neg P(X)$

$\neg \forall X\ P(X) \equiv \exists X \neg P(X)$

$\exists X\ P(X) \equiv \exists Y\ P(Y)$

$\forall X\ q(X) \equiv \forall Y\ q(Y)$

$\forall X(P(X) \wedge q(X)) \equiv \forall X\ p(x) \wedge \forall Y\ q(Y)$

$\exists X\ (p(X) \vee q(X)) \equiv \exists X\ p(X) \vee \exists Y\ q(Y)$

Examples of English sentences represented in predicate calculus:

1- If it doesn't rain tomorrow, Tom will go to the mountains.

¬ weather (rain, tomorrow)→ go(tom, mountains).

2- Emma is a doberman pinscher and a good dog. Good dog

(emma) ∧ isa (emma, doberman)

3. All basketball players are tall.

∀X(basketball _ player(X) → tall (X))

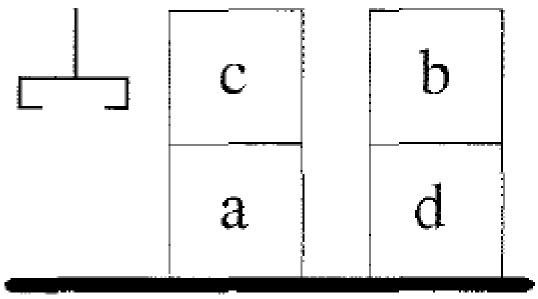
4- Some people like anchovies.

∃X (person(X) ∧ likes(X, anchovies)),

5- Nobody likes taxes

¬∃ X likes(X,taxes).

Example : Block world example



on(c,a)
on(b,d)
ontable(a)
ontable(d)
clear(b)
clear(c)
hand_empty

The following rule describes when a block is clear:

$$\forall X\ (\neg \exists Y\ on(Y,X) \rightarrow clear(X))$$