LECTURE 3:USING PREDICATE LOGIC TO REPRESENT FACTS

Symbols used to represent simple facts in logic are:

→ material implication

¬(not) - complement ┐ P

V (OR) or PVQ

∩ (AND) - and P∩Q

∀ (for all) All girls like ice cream

Ǝ (there exists)

Predicate logic symbols

≡ Equivalence

Truth symbols: true, false

PROPOSITIONAL CALCULUS

Is a language:

Using the symbols, words and sentences we can represent and reason about properties and relationships in the world

Language is described by pieces that make it, i.e. the set of symbols

Propositional symbols

Propositional symbols are:

P, Q, R, S

Truth symbols are: true, false

Connectives:∩(and), V(or), ¬(not), →(material implication), ≡ (equivalence)

Propositional symbols denote propositions or statements about the world that can either be true of false, e.g. the car is red, water is wet

Propositional Calculus sentences

The negation of a sentence is a sentence e.g.

¬P and ¬false are sentences

The conjunction and, and of sentences is a sentence e.g. P and ¬P is a sentence

The disjunction or , or of two sentences is a sentence e.g. P V ¬P

The implication of one sentence from another is a sentence e.g. P→ Q is a sentence

Propositional Calculus sentences

The equivalence of two sentences is a sentence e.g. P V Q ≡ R

Legal sentences are also called well formed formulas or WFFs.

P∩Q (p and q) are called conjuncts

PVQ (p or q) are called disjuncts

P→Q p implies q,

P is the premise or antecedent and Q is the consequent or conclusion

Propositional Calculus sentences

The symbols ( ) and [ ] are used to group symbols into subexpressions and so control their order of evaluation and meaning e.g.

$(P \vee Q) \equiv R$ is not the same as $P \vee (Q \equiv R)$

Semantics of Propositional calculus

Semantics → meaning of sentences

Because AI programs must reason with their representational structures, it is important to demonstrate that the truth of their conclusions depends only on the truth of their initial knowledge or premises, so that logical errors are not introduced by the inference procedures

Semantics of Propositional calculus

A propositional symbol corresponds to a statement about the world e.g. P may denote the statement " It is raining" or Q the statement "I live in a brown house"

A proposition may be true or false given some state of the world.

Truth value assignment to a proposition is called an interpretation, which is an assertion about their truth in some possible world

Predicate Calculus

- Differs from propositional calculus
- Propositional calculus operates with atomic symbols e.g. P, Q, R, S etc.
- We have learnt that a symbol say P or Q denotes a single proposition, so there is no way to access the components of an individual assertion

Predicate Calculus

Predicate calculus provides the ability to access components of an individual assertion /proposition e.g. instead of a single propositional symbol, P denote an entire sentence ‘It rained on Tuesday”, we can create predicate weather that creates a relationship between a date and the weather: e.g. weather(tuesday, rain). Through inference rules we can manipulate calculus expressions assessing their individual components and inference new sentences

Predicate Calculus symbols

1. Set of letters, both upper and lower cases of the English alphabet
2. The set of digits 0-9
3. The underscore (__)

Symbols must begin with a letter followed by any sequence of these legal characters

The symbols are used to denote objects, properties or relations in a world of discourse

Predicate Calculus symbols

Just like in most programming languages, the symbols meaning or intended meaning help in understanding the program code for example:

l(g, k) and likes(george, kate) are formally equivalent (i.e. they have the same structure), the second can be of great help for (human readers) in indicating the relationship the expression represents.

Predicate Calculus symbols

Descriptive names(symbols) are solely to improve readability of expressions

Parenthesis ( ), commas(,) and periods(.) are solely used to construct well formed expressions and they do not denote objects or relationships in the world.

They are called improper symbols

Predicate Calculus symbols

Predicate calculus symbols may represent either”

- Variables
- Constants
- Functions or predicates

Constant symbols must begin with a lower case letter e.g. george, tree, tall, are examples of well formed constant symbols

Predicate Calculus symbols

The constants true and false are reserved as truth symbols

Variables are represented by symbols beginning with an upper case letter, e.g. George, BILL, KATE,

BILL AND KATE are legal variables but George and bill are not legal variables

Predicate Calculus symbols

Predicate calculus also allows functions on objects in the world of discourse.

Function symbols begins a lower case letter

Functions denotes a mapping of one or more elements in a set (called domain of the function) into a unique element of a second set (the range of the function)

Predicate Calculus symbols

Functions:

- Arithmetic
- Addition
- Multiplication
- Division

Functions may define mappings between non numeric domains

Every function symbol is associated with an arity indicating the number of elements in the domain mapped onto each element of the range

Predicate Calculus symbols

Father could denote a function of arity 1 that maps people onto their unique male parent

Plus could be a function of arity 2 that maps two numbers onto their arithmetic sum.

Function expression- is a function symbol followed by its arguments.

Arguments are elements from the domain of functions f(X,Y), e.g.

Function expression

price(bananas)

father(David) is a function expression whose value is george

plus(2,3) is a function whose value is integer 5

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

Function expression

A predicate calculus term is either a constant, variable or function expression

A predicate calculus term may be used to denote objects and properties in a problem domain

Examples of terms cat, times(2,3), X, blue, mother(sarah), kate

A predicate names a relationship between zero or more objects in the world

Examples predicates

Examples of predicates are: likes, equals, on, near, part-of etc.

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

Examples of atomic sentences

likes(george, kate), likes(X, george)

likes(george, sue), likes(X,X)

likes(george, sarah tuesdays)

friends(bill, george), friends(father of (david) father of (andrew))

Helps(bill,george)

The predicate symbols in these examples are:

likes, friends, helps

Examples of atomic sentences

A predicate can be used with different numbers of arguments e.g. in the above expressions likes has been used with two (2) and three(3) arguments. When a predicate is used with different arities, it is considered to represent different relations. So a predicate is defined by its name and its arity

Atomic sentences(expressions or prepositions)

They use the same logical connections used in prepositional calculus:

$\cap$,V, $\neg$, $\rightarrow$, and $\equiv$

Predicate calculus include two symbols called variable quantifiers $\forall$ and $\exists$. These constrain the meaning of a sentence.

$\forall$ is the universal quantifier and $\exists$ is the existential quantifier

∀ and ∃ quantifiers

A quantifier is always followed by a variable and a sentence e.g.

∃Y friends (Y, peter)

∀X likes (X, ice-cream)

The universal quantifier(∀) indicates that the sentence is true for all values of the variables

In the example ∀X likes(X, ice-cream)is true for all values in the domain of the definition of X

∀ and ∃ quantifiers

- The existential quantifier (∃) indicates that the sentence is true for at least one value in the domain e.g.

∃ Y friends (Y, peter) is true if there is at least one object indicated by Y that is a friend of peter.

Examples

Universal quantifier ∀ (is similar to ∩)
AND operator

∀X P(X) is read as " For all X, P(X) or "For every X, P(X)"

∀X P(X) is same as P(x1)∩P(x2)…∩P(xn)∩… for all x_i in U

Examples

1)If P(X) denotes “X is an undergraduate student” and U {Enrolled students in ICT 471} then $\forall X P(X)$ is TRUE

2) If P(X) denotes “$X>0$” and U is the integers then $\forall X P(X)$ is FALSE

3)If P(X) denotes “X is mortal and U represents all human beings, then $\forall X P(X)$ is TRUE

Existential quantifier

Existential quantifier ∃ is similar to V (OR)

∃XP(X) is read as "there exists X, P(X)" or For some X, P(X"or "there exists one X such that P(X)"

∃XP(X) same as P(x1)VP(x2)…V P(xn)V…. For all x_i

In U

1) If P(X) denotes " x is a KMU student" and U is the set of all students

Existential quantifier

2)If P(X) denotes “x = x+1” and U is the integers then ∃XP(X) is FALSE

3) If P(X) denotes “X is a friend of Mickey Mouse and U is the carton characters, then ∃XP(X) is TRUE.

Examples of well formed sentences

plus(two, three) is a function and not an atomic sentence

equal(plus(two, three)five)) is an atomic sentence

equal(plus(2,3)),seven is an atomic sentence

- (false though given is an atomic sentence – (false though given the standard interpretation of plus and equal

Example of the use of predicate

Example of the use predicate calculus to describe a simple world

The domain of discourse is a set of family relationships in a biblical genealogy

mother(eve, abel)

mother(eve, cain) father(adam,abel)

Father(adam, cain

Example

$\forall X\ \forall Y$ father(X,Y) V mother(X,Y)→ parent(X,Y)

$\forall X\ \forall Y\ \forall Z$ parent(X,Y)∩parent(XZ)→sibling(Y,Z)

The predicate mother and father define a set of parent-child relationships. The implications give general definitions of other relationships such as parent and sibling in terms of these predicates. It

$\forall X \, \forall Y \, \forall Z \; parent(X,Y) \cap parent(XZ) \rightarrow sibling(Y,Z)$

To formalize this process so that it can be performed on a computer, care must be taken to define inference algorithms and to ensure that such algorithms indeed draw correct conclusions from a set of predicate calculus assertions.