Knowledge Representation continued

- A representational scheme should

    1- be adequate to express all of the necessary information.
    2- Support efficient execution of the resulting code.
    3- Provide a natural scheme for expressing the required knowledge.

❖ AI representation languages

to talk about facts , rules we need a “language”, A.I. using many languages for the purpose of representation. we will use predicate calculus as a representational language.

❖ The languages in A.I. must :-

a) Handle qualitative knowledge.

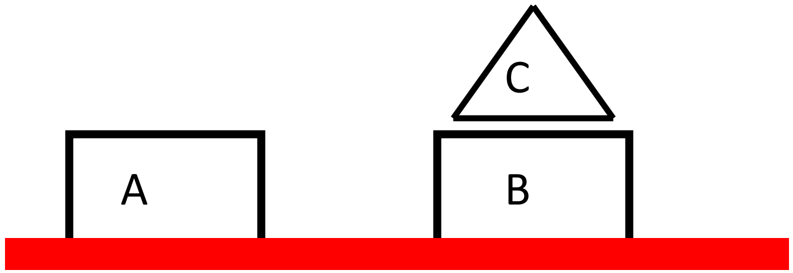
b) Allow new knowledge to be inferred from a basic set of facts.

c) Allow representation of general principles as well as specific situations.

d) Capture complex semantic meaning.

e) Allow for meta – level reasoning.

a) Handle qualitative knowledge:-



Using predicate calculus , the world could be described by the logical assertions

Clear (c)

Clear (a)

On table (a)

On table (b)

On (c,b)

Cube (a)

Pyramid (c)

•
•
•

Color

Left

Right

•
•
•

b) Allow new knowledge to be inferred from a basic set of facts:

To determine whether a block is clear

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

"For all X , X is clear if there does not exist a Y such that Y is on x"

c) Allow representation of general principles as well as specific situations:

- **Using variables**
- **Free from type & scope**

d) Capture complex semantic meaning:

Required large amounts of highly structured interrelated knowledge.

Describing the relationship between events occurring over time.

" a blue bird is a small blue – colored bird and a bird is a feathered flying animal"

Size (bluebird , small)
Has covering (bird , feather)
Has color (blue bird , blue)
Has property (bird , flies)
Is a (blue bird , bird)
Is a (bird , animal)

e) Allow for meta – level reasoning

1) An intelligent system not only should know things but also know what it knows.

2) It should be able not only to solve problems but also to explain how it should the problem and why it made certain decisions.

3) It should be able to describe its knowledge in both specific and general terms.

4) It should be able to recognize the limitations of its knowledge and learn from its interactions with the world

Knowledge about what you know

↓

Meta – knowledge
(higher level of knowledge)

The Propositional Calculus

- The propositional and predicate calculus are first of all languages , using their words, phrases and sentences we can represent and reason about properties and relationships in the world.

- Propositional calculus symbols
  P, Q, R , S, T (propositions) (statements)
- Truth Symbols
  true , false
- Connectives
- ^ = AND =Conjunction.
- ˅ = OR = disjunction.
- ¬ = NOT
- ==> = Implication.
- = = equivalence

- Propositional symbols denote propositions or statements about the world that may be either true or false.

- Propositions are denoted by upper case letters near the end of the English alphabet.

- Sentences in the propositional calculus are formed from those atomic symbols according to the following propositional calculus sentences.

Ex. True , P , Q and R senteces

Negation: ¬P , ¬False is a sentences
Conjunction : P ^ ¬Q is a sentence
Disjunction : P v ¬P is a sentence
implicant : P ==> Q is a sentence
Equavalence : P v Q = R is a sentence.
Legal sentence is called well- formed formulas or WFFs.

P ===> Q

Premise antecedent conclusion(consequent)

Semantics of propositional calculus

- Ex:- P="It is raining" Q= "I am at work "

P ==> ¬Q sentences

P ^ Q

- Note: propositions in this logic are only true or false

Propositional logic Assumes that the world consists of facts this is quite limited.

- An interpretation of a set of propositions is the assignment of truth value either T or F, to each propositional symbol.

P	¬P
T	F
F	T

P	Q	PvQ
F	F	F
F	T	T
T	F	T
T	T	T

P	Q	P^Q
F	F	F
F	T	F
T	F	F
T	T	T

P	Q	P==>Q
F	F	T
F	T	T
T	F	F
T	T	T

P	Q	P=Q
F	F	T
F	T	F
T	F	F
T	T	T

Ex : $(\neg P \vee Q) = (P \Longrightarrow Q)$

P	Q	¬P	¬P v Q	P==>Q	(¬P v Q) = (P ==> Q)
F	F	T	T	T	T
F	T	T	T	T	T
T	F	F	F	F	T
T	T	F	T	T	T

- We can also prove the following propositional calculus equivalences
- ¬(¬P) =P
- (P v Q) = (¬P ==> Q)
- Demorgan's Law :¬(P ^Q) = ¬P v ¬Q
- Demorgan's Law :¬(P v Q) = ¬P ^ ¬Q

- Distributive Law: P v (Q ^ R) = (P V Q) ^ (P v R)
  P ^ (Q v R) = (P ^ Q) v (P ^ R)
- Commutative Law: (Q ^ R) = (R ^ Q)
  (Q v R) = (R v Q)
- Associative Law: P v (Q v R) = (P v Q) v R
  P ^ (Q ^ R) = (P ^ Q) ^ R
- Contrapositive Law: P ==>Q = ¬Q ==> ¬P

The Predicate Calculus

- Propositional calculus assumes that the world consists of facts this is quite limited.

- Predicate logic (First order predicate logic) is more flexible because the world consists of objects, with properties among Objects relations hold.

Ex: weather (tuesday , rain).

The same sentence is much harder in propositional logic.

e .g. in propositional logic

P="Ahmed is father of Ali"
Q="Ali is father of kamal"

- If we need Grand father of kamal , because there is no functions and because there are no relations, both of which are awkward and don't allow for generalizations rel(X).

- Predicates are actions or states that relate terms.
  e.g. father (   ).
  Be clear that they are not the same as functions like.

- Literals are predicates applied to any term e.g.
  female (….), isage(….)

- Clauses are conjoined literals that imply another literal
- (Well Sort of) e.g. A ^ B ^ C ^D ==> H.

Connective and Quantifiers

- Not (¬), AND (^) conjunction, OR (v) disjunction

- If ==> implication: which is true if the antecedents are true
  P ==> Q means if P is true then Q is true.
- Iff <==> equality: the two terms are equivalent.

- Quantifiers

- ∃X: means there exists at least one X such that
- ∀X: means for all X

- There is a connection between ∀ and ∃ via negation
- ∀X ¬likes(X, computers)
  is equivalent to ¬∃X likes(X,computers)

- $\forall X$ likes(X, icecream)
  is equivalent to $\neg\exists X \; \neg$likes(X,icecream).

Predicates Calculus Symbols

- Predicate Calculus consists of

1- A ..Z, a .. Z
2- set of digits 0,1,…9
3- the underscore _

- Predicate calculus symbols may represent either variables, Constants, functions or predicates.

- Variables are represented by symbols beginning with an uppercase letter   e.g. Gorge.

- Constant symbols must begin with lowercase letter
  e.g.   ahmed , age

- Functions symbols begin in lowercase, 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).

e.g. f(x,y)