Bloom's Taxonomy → 6 levels

AI

Cognitive, Conitive, Psycho Sensory based

UNIT-2

KNOWLEDGE REPRESENTATION

~~Reasoning~~:-

Knowledge → It can be defined as a body of facts and principles accumulated by human kind or it can be understood as the fact, act or state of knowing.

Knowledge is having familiarity with language, concepts, procedures, rules, ideas, abstractions, places, customs, facts, associations coupled with an ability to use these notions effectively in modelling different aspects of the world.

The meaning of knowledge is closely related to the meaning of intelligence.

Intelligency requires the position of or access of knowledge.

A common way to represent knowledge external to a computer or human is in the form of written language.

For example → Ramu is tall → expresses a simple fact ie the attribute possessed by a person.

→ Ramu loves his mother → complex binary relation b/w 2 person

Lisp Prolog

Date ______

Page ______

There are 2 entities related to representation of knowledge in AI :-

1) <u>Fact</u> :- The truth in some relevant world. These are the things to be represented.

2) <u>Representation of Facts</u> :- It must be done in some chosen formalism.
↳ chosen structure

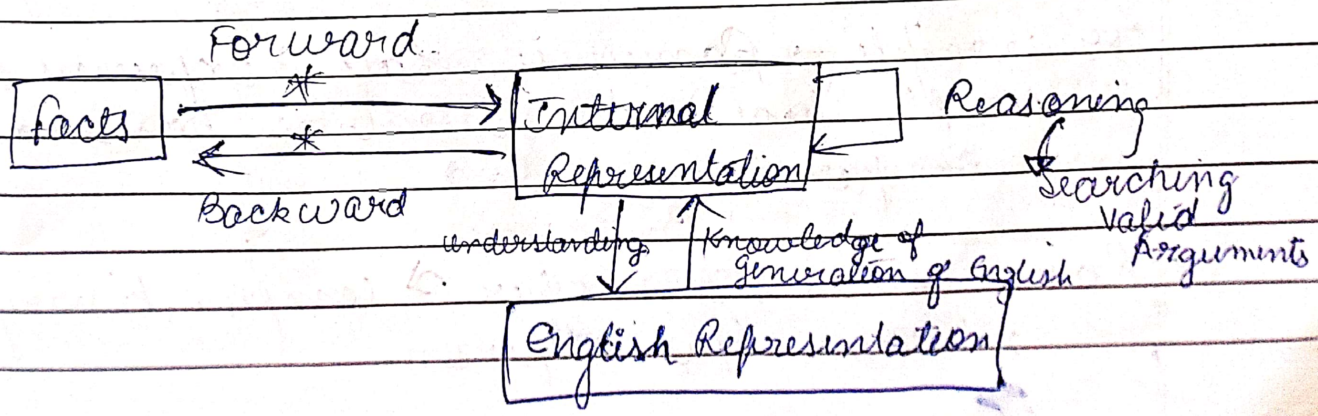
These are the things we will actually be able to manipulate

The structuring of these entities are done at 2 levels :-

(1) <u>Knowledge</u> → Facts including each agent's behaviours and current goals are described in this level

(2) <u>Symbol</u> → Representation of objects at knowledge level are defined in terms of symbols that can be manipulated by programs.

<u>Mapping b/w facts and Representations</u> :-



Consider the statement →
(i) "Markus is a man".
Using the mathematical logical representation
⇓
man(Markus)

Suppose we have a logical representation which says,
(2) "All men have legs"
⇓
$\forall x : man(x) \rightarrow has\ legs(x)$

Using (1) & (2) we have
(3) has legs(Markus)

Inductive Reasoning
↳ Sherlock example

⇒ This is a type of Deductive Representation.
or Deductive Reasoning

Using backward mapping we can say that "Markus has legs".

↳ 2 Statements

1st statement → "All men have legs"

2nd statement → "Every man has legs"

Conclusion → Every man have atleast 1 leg

Note :- We must first decide what facts the sentences represent and then convert those facts into new representation.

RICH & KNIGHT

Date

Page

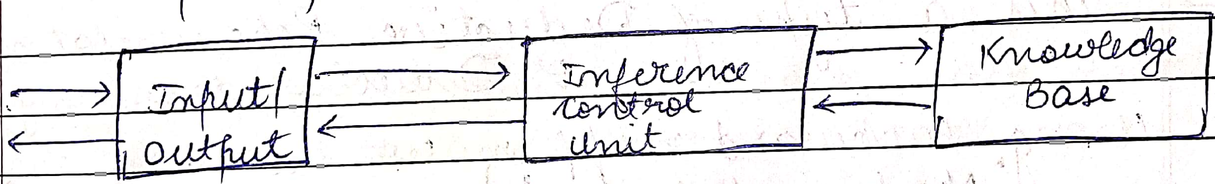
Knowledge-Based Systems

These are the systems that depend on a rich base of knowledge to perform difficult task.

It includes work in vision, learning, general problem solving and NLU.

The systems get their power from the expert knowledge that has been coded into facts, heuristics and procedures.

The knowledge is stored in a knowledge base that is separate from the control and inferencing components. It is possible to add new knowledge or refine existing knowledge without recompiling the control and inferencing program.



Representation of Knowledge :-

The objective of a knowledge representation is to express knowledge in a computer practical form so that it can be used to enable our AI agents to perform as per the expectation

A KR language is defined by 2 aspects :-

→ Syntax :- The syntax of a language defines which configuration of the components of the language constitute valid sentences

KR = Knowledge Representation

FREEMIND

Date

Page

→ Semantics :- It defines which facts in the world the sentences refer to and hence the statement about the world that each sentence makes.

Approaches to KR or Requirements of KR :-

A good system for R for K in a particular domain should possess following properties :-

1) **Representational ~~Add~~ Adequacy** :- The ability to represent all the diff kinds of knowledge that might be needed in that domain.

2) **Inferential Adequacy** :- ~~It is related to~~ The ability to manipulate the representation structures in such a way as to derive new structures corresponding to new knowledge infered from old ones.

3) **Inferential Efficiency** :- The ability to incorporate addⁿ info in to the K structure which can be used to focus the attention on the inference mechanisms in the most promising directions.

4) **Acquisitional Efficiency** :- The ability to acquire new info easily. Idealy the agents should be able to control its whole knowledge acquisitions but direct insertion of info by a K Engineer would be acceptable.

DB = Database

Fully OOP language

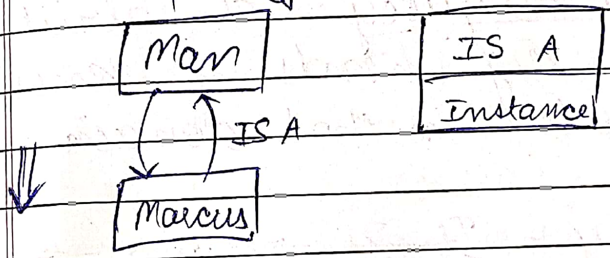
Page ??

Techniques of KR :-

1) <u>Simple Relational Knowledge</u> :- It is used in DB systems to represent declarative facts

Declarative K is passive K expressed as statements or facts about the world.

2) <u>Inheritable Knowledge</u> :- It uses the concept of property inheritence.



Marcus is a man.

3) <u>Inferential Knowledge</u> :- It uses the concept of standard logical rules of inference

4) <u>Procedural Knowledge</u> :- It is compiled knowledge related to the performance of some tasks

unclear

FUZZY LOGIC :-

1) <u>Fuzzy System</u> :- It includes fuzzy logic and fuzzy set theory

Knowledge exist in 2 distinct form

→ Objective Knowledge that exist in mathematical form and is used to represent the engineering problem

→ Subjective Knowledge that exist in linguistic form and is usually impossible to quantify.

Date
Page

The concept of fuzzy logic is used to co-ordinate these 2 forms of knowledge in a logical way.

Many real world problems have been modelled, simulated and replicated with the help of fuzzy systems.
So applicat^n of fuzzy systems are
Info. Retrieval System, Navigation system, Robot Vision etc.

Definition of Fuzzy logic:-
Fuzzy logic is derived from fuzzy set theory dealing with reasoning ie approximate rather than precisely deduced from classical to valued logic.

Note:- A declarative sentence ie either true or false is a proposition.
Exception → commands, questions and exclamations.

Rules of Inference:-

(1) Modus Ponens :-
$p \wedge (p \rightarrow q) \rightarrow q$

(2) Modus Tollens :-
$(\sim q \wedge (p \rightarrow q)) \rightarrow \sim p$

In fuzzy logic, the truth values are multi-valued such as absolute true, partially true, Absolute false etc

FREEMIND

Date ______

Page ______

Fuzzy Proposition :- It is a statement $\tilde{P}$ which acquires a fuzzy truth value $T(\tilde{P})$

$\tilde{P}$: Ram is Honest

$T(\tilde{P})$: 0.8 ⟶ Partially Honest $T(\tilde{P})$: 1 ⟶ Fully Honest

how it be calculated based upon experience

Fuzzy Connective :-	Definition
① Negative $T(\tilde{P})$	$1 - T(\tilde{P})$
② Disjunction $(\tilde{P}, \tilde{Q})$	$\max(T(\tilde{P}), T(\tilde{Q}))$
③ conjunction	$\min(T(\tilde{P}), T(\tilde{Q}))$
④ Conditional (Implication)	$\max(1 - T(\tilde{P}), T(\tilde{Q}))$

$p \to q$

$\sim p \vee q$

Eg $\tilde{P}$: Marry is efficient

$\tilde{Q}$: Ram is efficient

$T(\tilde{P}) = 0.8$ $T(\tilde{Q}) = 0.65$

Here value of Fuzzy Connectives definitions can be calculated easily.

Fuzzy Quantifiers :-

In crisp logic, the ~~predicates~~ propositions are quantified by some quantifier.

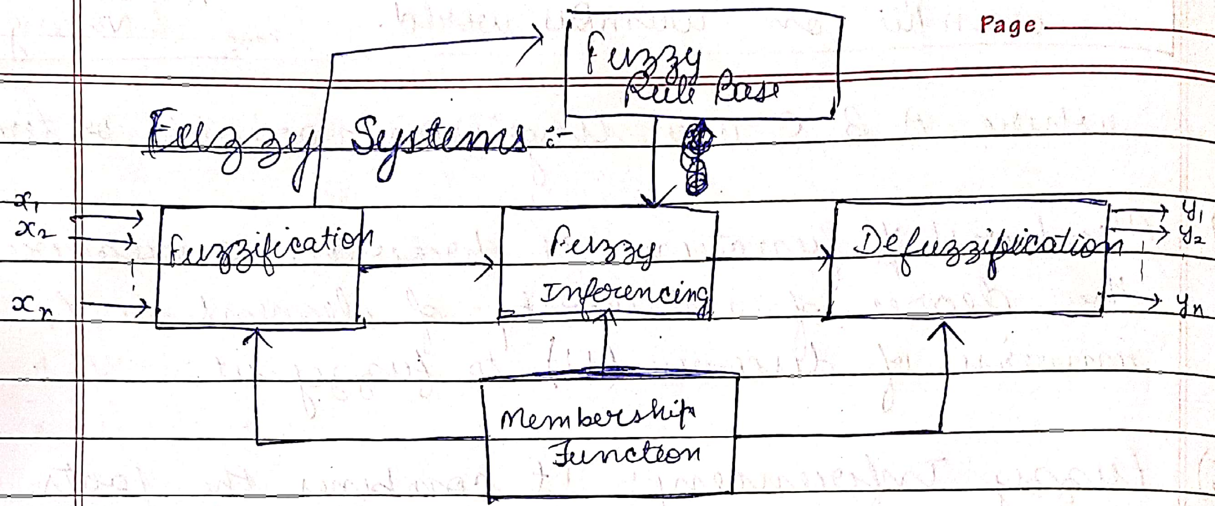
There are 2 classes :-

→ Absolute Quantifiers → Eg ~~[illegible]~~ much greater

→ Relative Quantifiers → Eg almost, around, atleast

Scanned with OKEN Scanner

Fuzzy Systems :-



Fuzzy System Elements :-

1) Input Vector :- These are the crisp values which are transformed into fuzzy set in the fuzzification block. It is represented by

$$X = [x_1, x_2 \_\_\_ x_n]^T$$

2) Output Vector :- It comes out from the defuzzification block which transforms an output fuzzy set to crisp values.

$$Y = [y_1, y_2 \_\_\_ y_n]^T$$

3) Fuzzification :- It is a process of transforming crisp values into grades of membership for linguistic terms such as far, small, near of fuzzy sets.

4) Fuzzy Rule Base :- It is a collection of propositions containing linguistic variables

IF (x is A) (y is B) Then (z is C)

x, y, z are variables

write short notes on Knowledge Based Agents and read about the example on Wumpus world. Russel & Norwig

FREEMIND Date Page

where A, B, C are linguistic variable or terms.

5) Membership function: It provides a measure of the degree of similarity of element in the universe of discourse (U) to fuzzy set.

6) Fuzzy Inferencing:- It combines the facts obtained from fuzzification with the Rule base and conducts the fuzzy reasoning process.

7) Defuzzification :- It translates the results back to real world values.

In many situations for a system whose output is fuzzy it is easier to take a crisp decision if the output is represented as a single quantity.

The typical defuzzification methods are:-

- Centroid Method
- Centre of Sums
- Mean of Maxima

First Order Logic :- (or First order Predicate Logic)

It is another way of Knowledge Representation in AI. It is an extension to Propositional logic. It is a powerful language that develops info about the objects in a more easy way & can also express relationship b/w those objects.

Scanned with OKEN Scanner

FREEMIND

Date

Page

Syntax of FOL :-

Some basic elements of FOL are :-

- constants
- Variables
- Predicates
- Functions
- Connectives
- Equality (=)
- Quantifiers

Atomic Sentences :- most basic sentences of FOL and are formed from a predicate symbol followed by a parantheses ~~follow~~ and a seq. of terms and is represented as Predicate (term1, term2 ... term n)

For eg. • Hari & Raghu are brother

⇒ Brother (Hari, Raghu)

• Tommy is a dog

⇒ Dog (Tommy)

Complex sentences :- It is made by combining atomic sentences using connectives.

FOL statements divided into 2 parts

1.) Subject which is main part of statement

2) Predicate → It is a relationship which binds the atoms in a statement.

Eg $\underbrace{x}_{\text{Subject}}$ $\underbrace{\text{is an integer}}_{\text{Predicate}}$

Quantifiers :- It is a language which generates quantification. They are the symbols that permit to determine the identity of variable in terms of range & scope in a logical expression.

Quantifiers :-

Universal (range)
⇒ (For all, For every, for each)

Existential (scope)
For some, there exist
($\wedge$)

Universal Quantifier → It is a symbol of logical representation which specifies that the statement within its range is true. For everything and for every instance of a particular thing.
It is represented as "$\forall$"

Existential Quantifier ⇒ It is a type of quantifier which states that a statement within its scope is true. For atleast one instance of something. Represented as "$\exists$"

Eg. All men drink coffee ⇒ x_1 coffee $\wedge$ x_2 coffee ---
$\forall x$: man(x) → $\boxed{\text{drink}(x, \text{coffee})}$ ⤵ Atomic sentence

Eg All birds fly
$\forall x$: bird(x) → fly(x)

Read about | HORN CLAUSE

Date

Page

Eg Every men respect their parents.

x, y

respects(men, parent)

$\forall x: man(x) \rightarrow respects(x, parent)$

Eg some boys are intelligent

x_1 int. $\vee$ x_2 int. ---

$\exists x: boy(x) \wedge intelligent(x)$

Inferences in FOL :-

It is used to deduce new facts and sentences from existing ones.

Terminologies :-

1) Substitution → It is fundamental operation performed on terms & formulas. It occurs in all inference system in FOL.

$F[a/x] \rightarrow$ Substitute a for x

2) Equality :- Eg Brother of John = Smith

Brother(John) = Smith

$\sim x = y \Rightarrow x \neq y$

Inference Rules for quantifiers :-

1) Universal Generalisation :- It is a valid inference Rule which states that If P(c) is true for any arbitrary element

c in the universe of discourse, Then for all x, P(x) is the conclusion.

ie $$\frac{P(c)}{\therefore \forall x\, P(x)}$$

2) Universal ~~Existential~~ Instantiation :- It is also called Universal Elemination. It can be applied multiple times to add the new sentences. We can infer any sentence P(c) by substituting a ground term c from all x P(x), for any object in the Universe of Discourse

ie $$\frac{\forall x\, P(x)}{\therefore P(c)}$$

Rahul likes Ice-cream

⇑

Eg Every person likes ice-cream.

3) Existential Instantiation Elimination :- It is a valid inference rule in FOL and can only be applied only once to replace the existential sentences. One can infer P(c) ~~from~~ the formula in the form

$$\frac{\exists x\, P(x)}{\therefore P(c)}$$

4) Existential Generalisation Introduction :- If there is some element c in Universe of Discourse which has a property P ~~totion~~ then we can infer that there exist something in Universe that has a property P

Date
Page

ie $P(c)$ — Arman got good marks in AI

$\therefore \exists x\ P(x)$ — Someone got good marks in AI

Unification:- It is all about making the expressions look identical.

$L1: P(x, F(g)) \quad P(y, F(z)) : L2$

Identical if $x = y$ & $g = z$
Substitution → y/x & z/g

Conditions of Unification:-

→ Predicate symbol must be same, atoms or expression with different predicate symbols can never be unified.

→ No of arguments in both expressions must be identical.

→ Unification will fail if there are 2 similar variables present in the same expression.

Unification Algorithm:-

Unify (L_1, L_2)

1) If L_1 & L_2 ~~are~~ is a variable or constant, then

a) If L_1 & L_2 are identical, return NILL.
b) Else if L_1 is a variable then
i) If L_1 occurs in L_2 then return FAIL
else return (L_2 / L_1)

c) Else if L_2 is a variable then
i) If L_2 occurs in L_1 then return FAIL else return (L_1/L_2).

d) Else return FAIL.

2) If initial predicate symbol in L_1 & L_2 are not identical then return FAIL.

3) If L_1 & L_2 have diff. no of arguments then return FAIL.

4) Set Substitution to NIL. (ie Subst = NIL)

5) Loop, For i ← 1 to no of arg in L_1

a) call Unify with i^{th} argument of L_1 and i^{th} argument of L_2 in S.

b) If S = FAIL then return FAIL.

c) If S ≠ NULL then
i) Apply S to the remainder of both L_1 & L_2.
ii) SUBST = APPEND (S, SUBST)

6) return SUBST.

Implementation of Unification Algorithm:-

1) Initialise the substitution set to be empty.
2) Recursively ~~are~~ unify atomic sentences.
   a) Check for identical expression match.
   b) If one expression is a variable V_i and other is a term T_i which does not contain the variable V_i, then
      i) Substitute T_i/V_i in existing substitutions
      ii) Add T_i/V_i to the substitution set list.
      iii) If both the expressions are functions, then function names must be similar and no of arguments must be same in both the expressions.

Eg $P(x, g(x))$

① $P(z, y)$ If unification possible?

Ans Yes, Unification is possible here

ie z/x, $y/g(x)$

② $P(z, g(z))$ Unification ✓ z/x, $g(z)/g(x)$

③ $P(prime, f(prime))$ X because function name must be different.

Resolution is a single inference rule which can efficiently operate on CNF or Clausal form

(CNF = conjunctive Normal form)

↳ It is a sentence represented as a conjunction of clauses.

Clause → disjunction of literals is called clause.

Steps for Resolution:-

1) conversion of facts into FOL
2) Convert FOL statements to CNF.
3) Negate the statement which needs to be proven by contradiction
4) Draw Resolution graph (unification)

Scanned with OKEN Scanner

Eg :-

1) If it is sunny and warm day, you will enjoy.
2) If it is rainy, you will get wet.
3) It is a warm day.
4) It is raining
5) It is sunny

To prove :- You will enjoy

Step 1 :-
1) $\text{Sunny} \land \text{warm} \rightarrow \text{enjoy}$
2) $\text{Raining} \Rightarrow \text{wet}$
3) Warm
4) Raining
5) Sunny

Step 2 :-
1) $\sim(\text{Sunny} \land \text{warm}) \lor \text{enjoy}$
2) $\sim \text{Raining} \lor \text{wet}$
3) Warm
4) Raining
5) Sunny

$\sim \text{Sunny} \lor \sim \text{warm} \lor \text{enjoy}$

Step 3 :- $\sim \text{enjoy}$

Step 4 :- $\sim \text{enjoy}$ $\quad \sim \text{Sunny} \lor \sim \text{warm} \lor \text{enjoy}$ — (1)

$\sim \text{Sunny} \lor \sim \cancel{\text{warm}}$ $\quad \cancel{\text{warm}}$ — (3)

contradiction value $\quad \sim \cancel{\text{Sunny}}$ $\quad \cancel{\text{Sunny}}$ — (5)

{ }

Hence Proved

Date
Page

Eg 1) John likes all kind of food.
2) Apple and vegetable are food.
3) Anything anyone eats and not killed is food.
4) Anil eats peanuts and still alive.
5) Harry eats everything that Anil eats
Goal or To prove:- 8) John likes peanuts.

6) If not killed, then alive
7) If alive, then not killed

Step 1 :- 1) $\forall x : food(x) \rightarrow likes(John, x)$
2) $Food(Apple) \wedge Food(Vegetable)$
3) $\forall x \forall y : (Eats(x,y) \wedge \sim Killed(x)) \rightarrow Food(y)$
4) $Eats(Anil, peanut) \wedge Alive(Anil)$
5) $\forall x \; Eats(Anil, x) \rightarrow Eats(Harry, x)$
6) $\forall x : \sim Killed(x) \rightarrow Alive(x)$
7) $\forall x : Alive(x) \rightarrow \sim Killed(x)$

To prove :- $Likes(John, Peanuts)$

Step 2 :- 1) $\forall x : \sim Food(x) \vee Likes(John, x)$

Rules for CNF → a) Eliminate all implications & re-write

2) $Food(Apple) \wedge Food(Vegetable)$
3) $\forall x \forall y : \sim(Eats(x,y) \wedge \sim Killed(x)) \vee Food(y)$
4) $Eats(Anil, peanut) \wedge Alive(Anil)$
5) $\forall x : \sim Eats(Anil, x) \vee Eats(Harry, x)$
6) $\forall x : Killed(x) \vee Alive(x)$
7) $\forall x : \sim Alive(x) \vee \sim Killed(x)$

Scanned with OKEN Scanner

FREEMIND

Date

Page

b) Move negation inwards and rewrite

1) Same
2) Same
3) $\forall x \forall y : \sim Eats(x, y) \lor Killed(x) \lor Food(y)$
4) Same
5) Same
6) Same
7) Same

c) Rename variables or Standardise variables

1) $\forall x : \sim Food(x) \lor Likes(John, x)$
2) $Food(Apple) \land Food(Vegetables)$
3) $\forall y \forall z : \sim Eats(y, z) \lor Killed(y) \lor Food(z)$
4) $Eats(Anil, peanut) \land Alive(Anil)$
5) $\forall w : \sim Eats(Anil, w) \lor Eats(Harry, w)$
6) $\forall g : Killed(g) \lor Alive(g)$
7) $\forall k : \sim Alive(k) \lor \sim Killed(k)$

d) Eleminate existential instantial quantifiers

e) Drop Universal Quantifiers

1) $\sim Food(x) \lor Likes(John, x)$
2) $Food(Apple) \land Food(Vegetable)$
3) $\sim Eats(y, z) \lor Killed(y) \lor Food(z)$
4) $Eats(Anil, peanut) \land Alive(Anil)$
5) $\sim Eats(Anil, w) \lor Eats(Harry, w)$
6) $Killed(g) \lor Alive(g)$
7) $\sim Alive(k) \lor \sim Killed(k)$

2.1) $Food(Apple)$

2.2) $Food(Vegetable)$

4.1) $Eats(Anil, peanut)$

4.2) $Alive(Anil)$

FREEMIND

Date

Page

means intersection vali statements ko Alag kardo

↓

f) Distribute conjunction over disjunction

Step 3:- ~Likes(John, Peanuts)

~Food(x) ~Likes(John, x) → ①

Peanut/x

~Food(Peanut)

~Eats(y, z) V Killed(y) V Food(z) → ③

Peanuts/z

~Eats(y, peanut) V Killed(y)

Eats(Anil, peanut)

Anil/y

Killed(Anil)

~Alive(K) V ~Killed(K)

Anil/K

~Alive(Anil)

Alive(Anil)

{ }