Thursday
28-05-2020

AI Assignment-2

UNIT-2

N.V.S.K.kalyani
174N1A0584
3rd B.Tech CSE-C

2 marks

1. Rewrite PEAS for Wumpus World?

The Wumpus World is a cave consisting of rooms connected by passage ways.

i) performance measures:

* Gold - score : +1000
* Death - score : -1000
* Taking a step - score : -1
* use of arrow - score : -10

ii) Environment

* Square adjacent to wumpus are smelly.
* Square adjacent to pit are breezy.
* Glitter iff gold is in the same square.
* shooting kills wumpus if you are facing it.
* shooting uses up the only arrow.
* Grabbing picks up gold if in same square.
* Releasing drops the gold in same square.

iii) Actuators : Left turn, Right turn, Forward, Grab, Release, shoot.

iv) Sensors : Stench, Breeze, Glitter, Bump, Scream

2. Define

i) logic :- It is a study of methods that helps in distinguishing correct reasoning from incorrect reasoning.

ii) model :- when we need to precise, we will use the term model in place of "possible world".

iii) entailment :- Means, that one thing follows from another: $\alpha \models \beta$. And also, is a relationship blw sentences, syntax based on semantic

iv) Inference :- In the general sense means: Give some pieces of information (prior, observed variables, knowledge base).

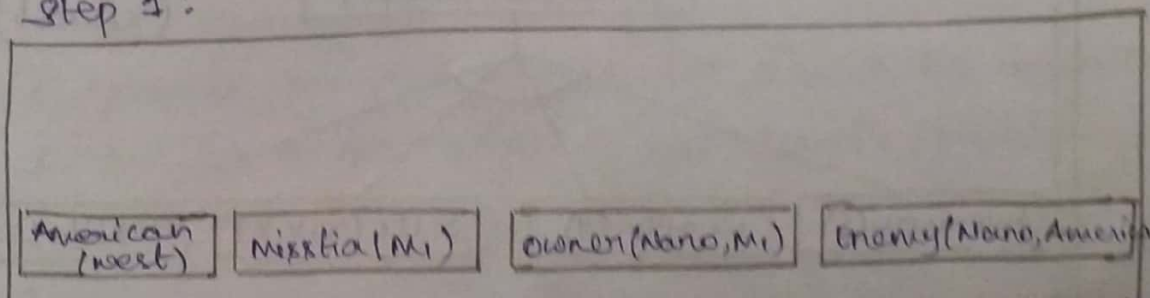
v) unification :-

* Generalized modus ponens is a lifted version of modus ponens - it raises modus ponens from propositional to first-order logic.
* This process is called unification and the key component of all first-order inference algorithm

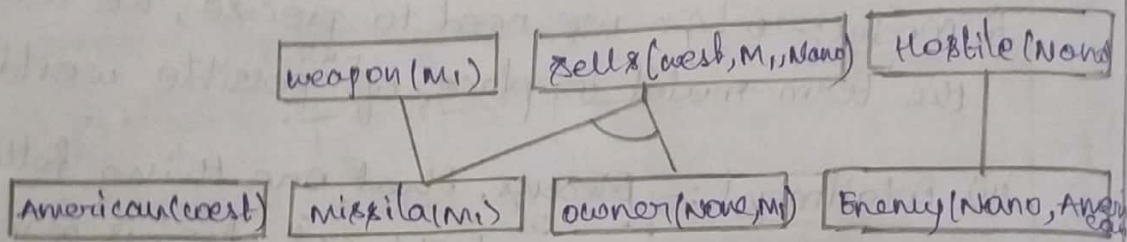
3. Draw the diagrams for forward & Backward chaining?

Ans Forward chaining :-

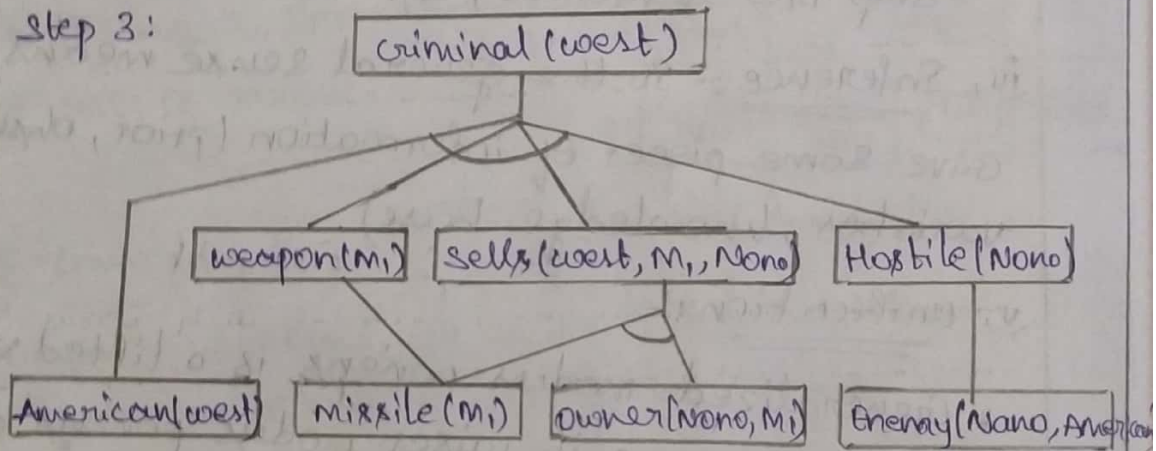
Step 1 :



step 2:



step 3:



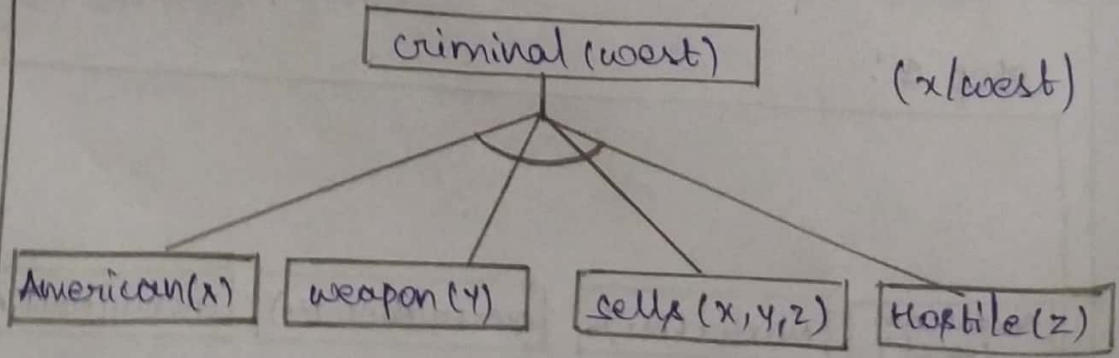
Backward chaining

step 1:

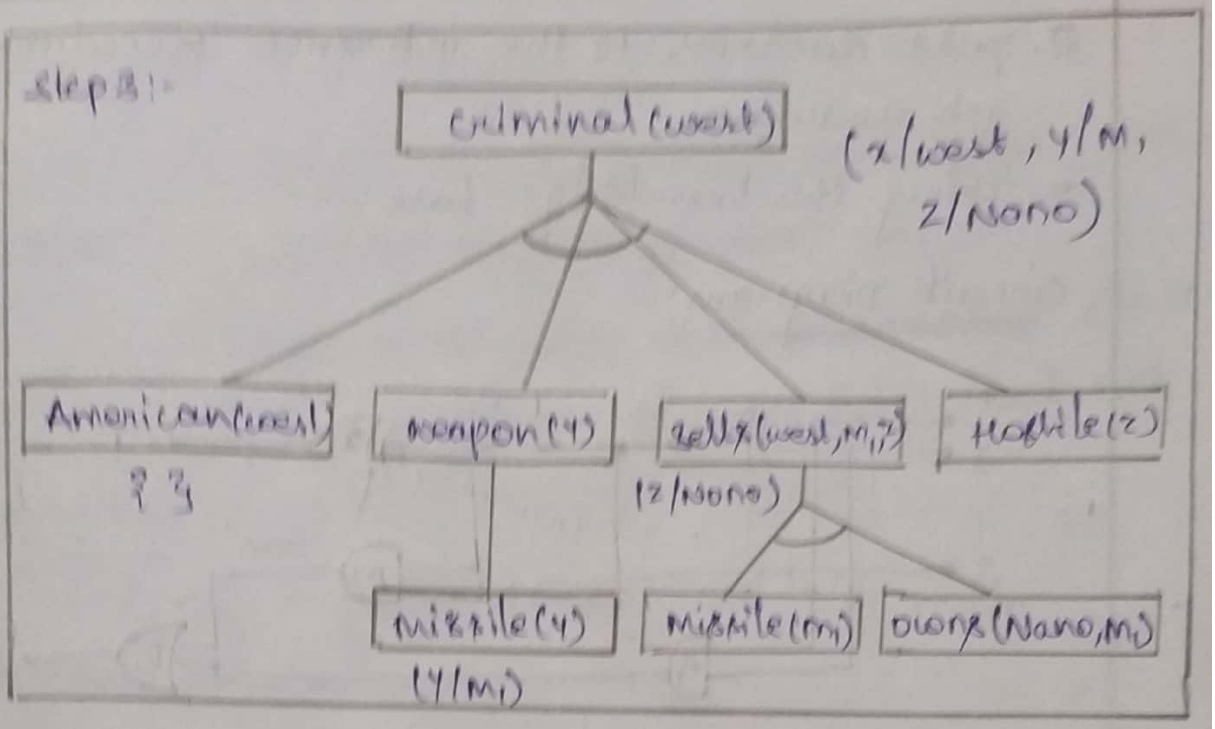
Backward chaining/ Example

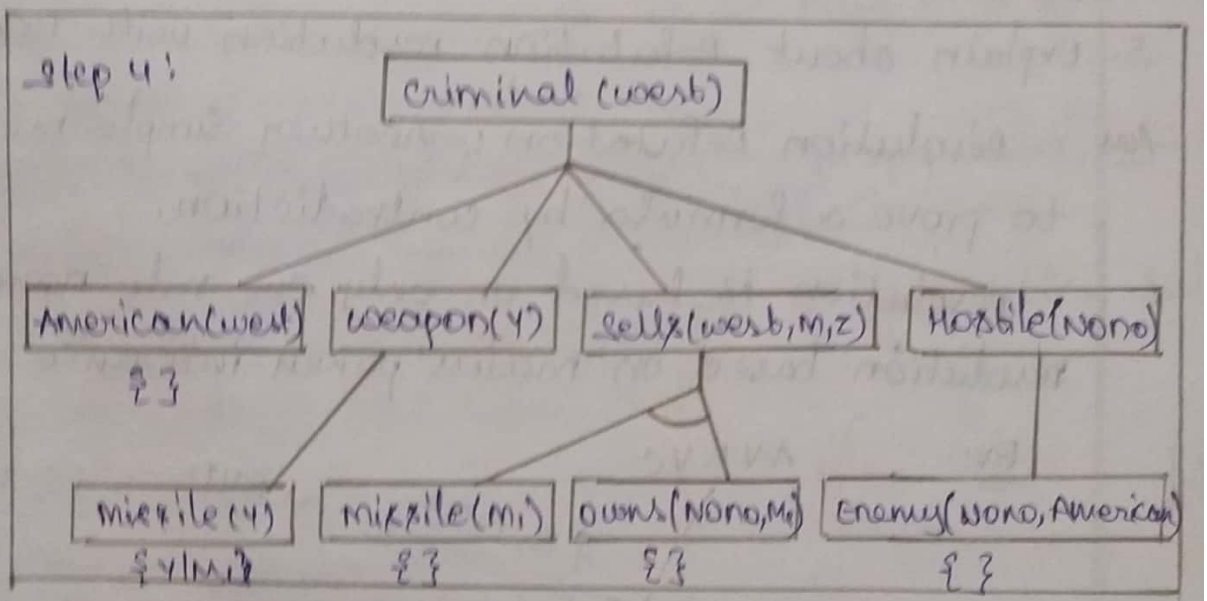
criminal(west)

step 2:



1TH21A0584



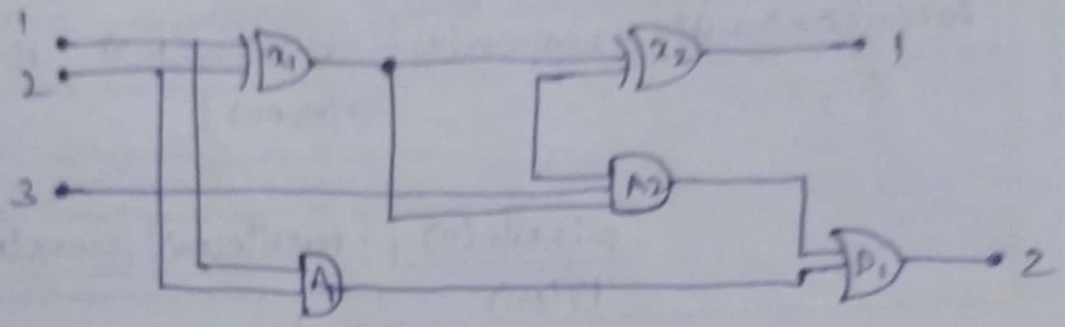


4. List the knowledge Engineering process step & also sketch the circuit Diagram.

Ans
1. Identify the task.
2. Assemble the relevant knowledge - knowledge acquisition.
3. Decide on a vocabulary of predicates, functions and constants.
4. Encode general knowledge about the domain.
5. Create a description of the specific problem instance.

6. pose Querries to the inference procedure and get answers.
7. Debug the knowledge base.

circuit Diagram:-

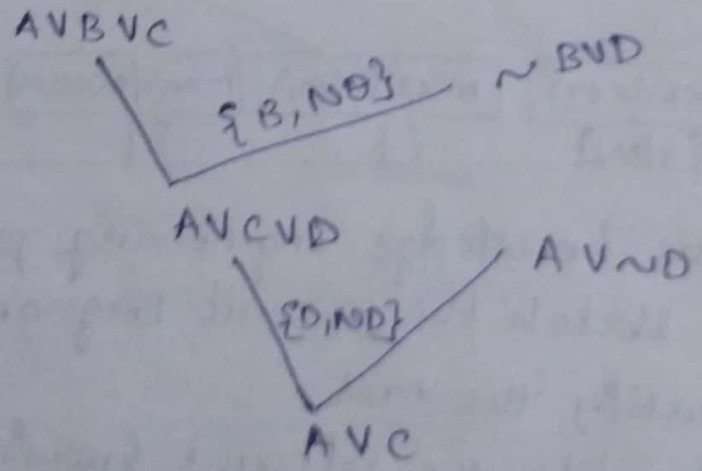


5. Explain about Refutation Resolution with Example

Ans Resolution Refutation :- Another simple method to prove a formula by contradiction.

Resolution is based on only one rule named as resolution based on modus ponen inference rule

Ex:-



10 marks

1. Explain in detail about propositional logic with full Explaination and diagrams?

Ans A proposition in logic is a declarative statements which are either true or false (but not both) in a given context. for Example,

- "Jack is a male",
- "Jack loves Mary" etc.

* Given some propositions to be true in a given context,
- logic helps in Inferencing new proposition, which is also true in the same context.
* Support we are given a set of proposition such as "It is hot today" and
"If it is hot it will rain", then
we can infer that
"It will rain today".

⇒ The following are propositions
- the reactor is on;
- the wing-flaps are up;
- Marvin K Mooney is president.

→ Definition:- Logics are formal language for representing information such that conclusions can be drawings.

* Syntax defines the sentences in the language.
* Semantics defines the "meaning" of sentence.
  - Define truth of a sentence in a "possible world".
* Ex:- $x + 2 \geq y$ is a sentence;
  $x2 + y >$ is not a sentence;

→ Syntax:- is a propositional logic defines the allowable sentences.

* The atomic sentence - the indivisible syntactic elements - consist of a single proposition symbol.
* complex sentence - are constructed from simpler sentence using logical connectives.

✓ Five types of connectives are available:

i) Not ($\sim$): A sentence such as $\sim W_{1,3}$ is called negation $W_{1,3}$.

ii) And ($\wedge$): A sentence whose main connective is A, such as $W_{1,3} \wedge P_{1,3}$ is called as conjunction

iii) Or ($\vee$): A sentence using $\vee$, such as $(W_{1,3} \wedge P_{1,3}) \vee W_{2,2}$ is a disjunction of the disjuncts $(W_{1,3} \wedge P_{1,3})$ and $W_{2,2}$.

iv) Implies ($\Rightarrow$): A sentence such as $(W_{1,3} \wedge P_{1,3}) \Rightarrow \sim W_{2,2}$ is called an implication or (conditional)

v) Equivalence ($\Leftrightarrow$): (if and only if).
The sentence $W_{1,3} \Leftrightarrow \sim W_{2,2}$ is a biconditional

Backus Normal form:

Sentence → AtomicSentence | complexsentence

AtomicSentence → True | true | Symbol

Symbol → P | Q | R | ....

ComplexSentence → ¬ sentence
(sentence ∧ sentence)
(sentence ∨ sentence)
(sentence ⇒ sentence)
(sentence ⇔ sentence)

→ Semantics:

* The semantic defines the rules for determining the truth of a sentence with respect to a particular mode

Ex: $\neg P_{1,2} \wedge (P_{1,2} \vee P_{3,1})$ = true ∧ (false ∨ true)
= true ∧ true
= true

→ Truth Table:-

* Truth table gives us operational definitions of important logical operators.

P	Q	~P	P∧Q	P∨Q	P ⇒ Q	P ⇔ Q
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	f	T	T

→ Validity and Satisfiability:-

* If we consider all possible models, there are different properties that may hold for a sentence
* A sentence is valid if it is true in all models.

  True: $A \vee \neg A$, $A \Rightarrow A$, $(A \wedge (A \Rightarrow B)) \Rightarrow B$.
* Validity is connected to inference via the deduction
  Theorem: $KB \models \alpha$ if and only if $(KB \Rightarrow \alpha)$ is valid.
* A sentence that is valid is said to be tautology.

→ Inference:-

* The aim of logical inference is to decide whether $KB \models a$ for some sentence a.

Ex:- $P_{2,2}$ entailed?

* Our first algorithm for Inference will be a direct implementation of the definition of entailment: enumerate the models, and checks that a is true in every model in which KB is true.

→ Logical Equivalence:

* Two sentence are logically Equivalent iff they are true in same models.
* $\alpha \equiv \beta$ if and only if $\alpha \models \beta$ and $\beta \models \alpha$.

$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$ commutativity of $\wedge$

$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$ commutativity of $\vee$

$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$ associativity of $\wedge$

$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$ associativity of $\vee$

$\neg(\neg\alpha) \equiv \alpha$ double-negation elimination

2. Explain about first order logic in detail with diagram

Ans Representation revisited...

* propositional logic as our representation language is sufficed to illustrate the basic concepts of logic and knowledge-based agents.

Syntax and semantics for FOL:

* It specifies more precisely the way in which the possible worlds of first-order logic reflect the ontological commitment to objects & relations.

→ Models For First order logic:

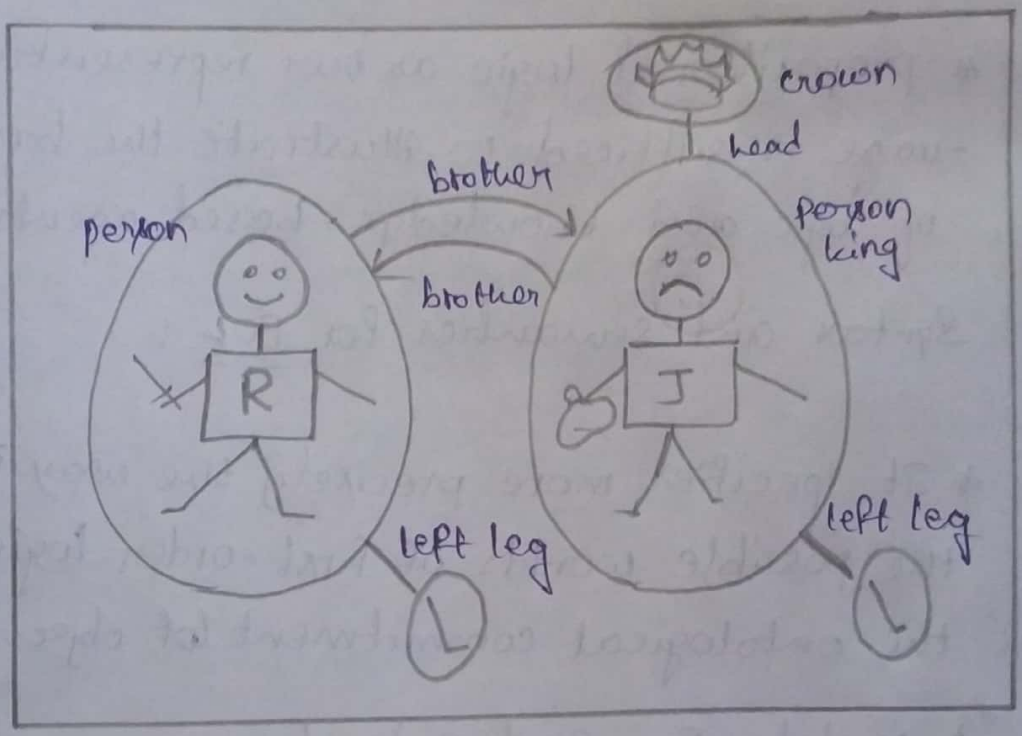
* Models of a logical language are the formal structures that constitute the possible worlds under consideration.
* models for propositional logic are just sets of truth values for the proposition symbols.

But for the model for FOL defines:

i, Domain
ii, Domain Elements

* Domain: The Domain of a model is the set of objects it contains.
* These objects are sometimes called domain elements.
* The domain is required to be nonempty-every possible world must contain at least one object

* The objects in the model are related in various ways as shown below: Brothers.



→ objects

i, Richard the Lionheart
ii, king of England from 1189 to 1199
iii, His younger brother - the evil king John, who ruled from 1199 to 1215;
iv, The left legs of Richard and John;
v, crown

✓ Relation is just the set of tuples of objects that are related.
✓ A tuple is a collection of objects arranged in a fixed order and is written with angle brackets.

Syntax for FOL using Backus Normal Form

Sentence → AtomicSentence | complexsentence

AtomicSentence → Predicate | predicate(Term...)

ComplexSentence → (Sentence) | [Sentence]

→ ¬ Sentence

Sentence ∧ Sentence

Sentence ∨ Sentence

Sentence ⇒ Sentence

Sentence ⇔ Sentence

Term → Function (Term....)

| constant

| variable

Quantifier → ∀ | ∃

constant → A | X1 | John

variable → a | x | ...

predicate → True | False | After | ....

Function → Mother | LeftLeg | ...

operator precedence: ¬, =, ∧, ∨, ⇒, ⇔