III

SL.NO:CAA179462

SRI SIVASUBRAMANIYA NADAR COLLEGE OF ENGINEERING

(An Autonomous Institution, Affiliated to Anna University, Chennai)
Rajiv Gandhi Salai (OMR), Kalavakkam – 603 110

THEORY EXAMINATIONS

Register Number	205001085		
Name of the Student	V. Sabarivasan		
Degree and Branch	BE CSE	Semester	V
Subject code and Name	UCS1504 Artificial Intelligence		
Assessment Test No.	II	Date	18/10/2022

Details of Marks Obtained

Part A		Part B				Part C			
Question No.	Marks	Question No.	(a) Marks	(b) Marks	Total Marks	Question No.	(a) Marks	(b) Marks	Total Marks
1	2	7	½			10			
2	2					11	8		
3	1	8	6			12	10		
4	½								
5	0	9	6			13			
6	2								
Total (A)		Total (B)				Total (C)			42½

Grand Total (A+B+C)				Marks (in words)			
Signature of Faculty							

ssn

12

PART-C

(i) It's not sunny and it's colder than yesterday.

(ii) We will go swimming only if it's sunny

(iii) If we don't go swimming, then we will take canoe trip.

(iv) If we take a canoe trip, then we will ~~becom~~ be home by sunset.

Conclusion.

We will be home by sunset.

(a) Propositional logic:

$s \rightarrow$ It is sunny.

$c \rightarrow$ It is ~~co~~ colder

$sm \rightarrow$ go swimming

$ct \rightarrow$ take canoe trip.

$hm \rightarrow$ ~~be~~ be home by sunset.

(i) $\sim s \wedge c$

(ii) $s \rightarrow sm$

(iii) $\sim sm \rightarrow ct$

(iv) $ct \rightarrow hm$.

To derive: hm.

(b) Apply inference rules:

$\sim s \wedge c$ (Given)

$\sim s$ (Decomposition) $\rightarrow$ (1)

$s \rightarrow sm$ (Given) $\rightarrow$ (2)

$\sim sm$ (Modus Tollens (1), (2)) $\rightarrow$ (3)

$\sim sm \rightarrow ct$ (Given) $\rightarrow$ (4)

ct (Modus Ponens (3), (4)) $\rightarrow$ (5)

$ct \rightarrow hm$ (Given) $\rightarrow$ (6)

$\underline{\underline{hm}}$ (Modus ponens (5)(6))

$\therefore$ We arrived to the conclusion.

Inference rules

1) $\dfrac{p, q}{p \vee q}$ (Addition)

2) $\dfrac{p \wedge q}{p}$ (Simplification)

3) $\dfrac{p \vee q,\ \sim p \vee r}{q \vee r}$ (Resolution)

4) $\dfrac{p \rightarrow q,\ q \rightarrow r}{p \rightarrow r}$ (Hypothetical syllogism)

5) $\dfrac{p \vee q}{\sim p}$ (Disjunction)

6) $\dfrac{\sim q;\ p \rightarrow q}{\sim p}$ (Modus Tollens)

7) $\dfrac{p,\ p \rightarrow q}{q}$ (Modus Ponens)

(11) Forward chaining algorithm:

⇒ It is a type of reasoning algorithm in which the goal state is reached from initial state by applying inference rules to get data.

⇒ This algorithm follow bottom to top approach.

⇒ It is used in expert systems and databases.

Algorithm:

FOL-FC (KB, α) returns subset or false
{ repeat until new is empty.
new = { }
for each sentence 'r' in KB, do
$(p_1 \wedge p_2 \wedge \dots \wedge p_n) \rightarrow q$ }
for each θ & subset
$(p_1 \wedge p_2 \wedge \dots \wedge p_n')\theta = p'\theta$

~~if p~~
if not ϕ fail return ϕ
add new to KB
return subset. }

Backward chaining algorithm:

⇒ It is a type of reasoning algorithm in which the goal state is derived via backtracking from goal state towards the initial state.

⇒ This algorithm follows top to bottom approach.

⇒ This algorithm is used in game theory.

Algorithm:

FOL-BC (KB, goals, θ) returns instructions
{ input: KB, goals, θ
local variables: ~~KB, goals~~, answer, set of instructions.

if goals is ϕ return $\{\theta\}$

$q' \leftarrow$ subset (θ, First (goals))

for each 'q' in KB, do

$(p_1 \wedge p_2 \cdots \wedge p_n) \rightarrow q$ and

$\theta' \leftarrow$ UNIFY (q, q')

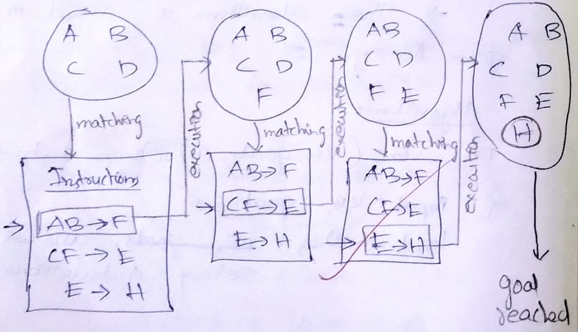
answer $\leftarrow$ FOL-BC (KB, $[p_1 \cdots p_n]$, θ')

return answer

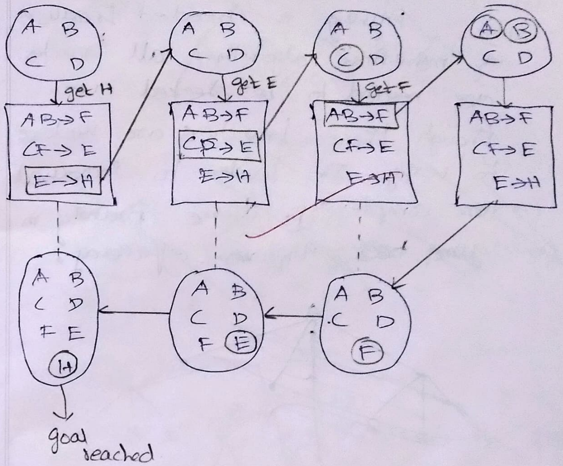
}

Example: (Get H)

→ Using forward chaining:



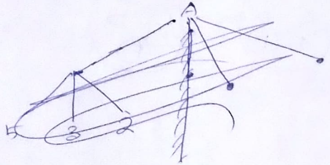
⇒ Using Backward chaining:

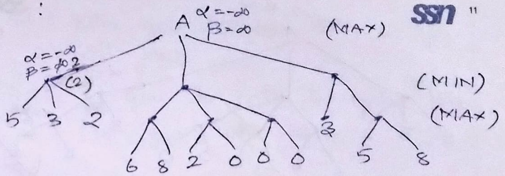


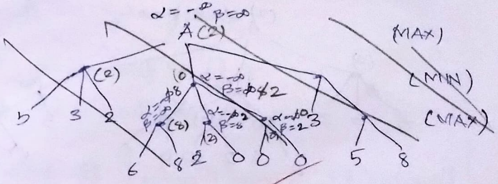
PART-B

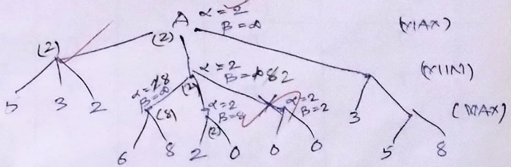
(7) Pruning in game trees:

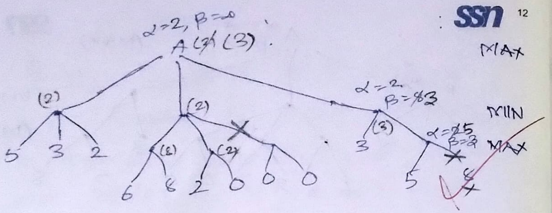
Pruning is needed because in min-max algorithm all branches are needed to be checked even though those branches are useless to verify. It leads to increased time complexity. Hence pruning in game trees improves efficiency.

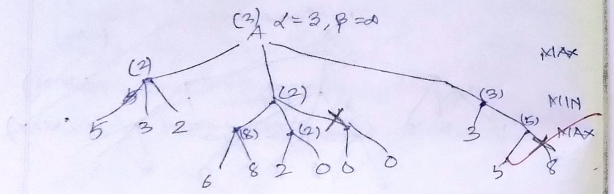












8 S1 : Either you do your HW, or you will flunk.

S2: If you do not do your HW, then you will flunk.

hw : do HW.

fl : get flunk.

⇒ S1: $hw \vee fl$

⇒ S2: $\sim hw \rightarrow fl$.

hw	fl	~hw	hw∨fl (1)	~hw→fl (2)
T	T	F	T	T
T	F	F	T	T
F	T	T	T	T
F	F	T	F	F

From the truthtable, we can see that (1) ≡ (2).

∴ Hence the given sentences are satefiable.

(9) Bay's theorem:

$$\boxed{P(A/B) = \frac{P(B/A)\,P(A)}{P(B)}}$$

⇒ Event A : $P(A) = 0.1$ //

⇒ Event B : $P(B) = 0.05$ //

⇒ Event A is patient has liver disease

⇒ Event B is patient is an alcoholic.

To find: probability of getting a liver disease in future, if patient is an alcoholic. $[P(A/B)]$.

⇒ $P(B/A) = 0.07$ //

Applying Bay's theorem,

$$P(A/B) = \frac{P(B/A)\,P(A)}{P(B)}$$

$$= \frac{(0.07)(0.1)}{0.05}$$

$= 0.14$ (or) 14 %

PART-A

(1) Propositional logic	Predicate logic
Does not involve quantifiers	Involves quantifiers.
Functions are not used.	Involves the usage of functions.
Provides any one of the truth values	Provides the logic behind the predicate.

(2) Game playing problems in AI:

(i) Chess

(ii) TicTacToe.

(iii) Sudoku

(iv) Checkers.

---

(3) "Everyone who loves all animals is loved by someone."

⇒

~~$\exists x \, Love(x)$~~ $\vee$

~~$\exists x \exists y$~~

$\exists x \forall y \, Love(x,y) \vee \forall z \, Love(y,z)$

---

(4) "Squares neighboring the wumpus are smelly"

Objects: Wumpus

Property: Neighboring squares

Relation: S_{ab} smelly in room (a,b)

W_{ab} wumpus in room (a,b)

~~R~~ ~~$R = S_a$~~

~~$R = S_a \wedge W$~~

$R = S_{a,b} \wedge W_{a+1,b} \wedge W_{a-1,b} \wedge W_{a,b+1} \wedge W_{a,b-1} \wedge W_{a,b}$

---

(6) Degree of belief:

⇒ Degree of belief is defined as the states in which the knowledge base can find out whether the logic provided for that particular state is true or not. It requires a lot of inference rules and knowledge about the environment.

⇒ It can be handled by training the knowledge based model with suitable logics and inference rules to get faster decisions in some situations.

Part - A

① ~~Propositional~~ Predicate logic

• ~~Propositional~~ Predicate Logic use logical functions to represent meaningful sentences and uses Predicate functions

* More complex and requires consistent proofs.

* ~~Exact~~ Membership operators / quantifiers can be used to represent certain ~~member~~ of the group (ex $\forall$ - Universal quantifier, $\exists$ - Existential quantifier)

*

* Example if Ram is guy, he is brother of x.

Guy(Ram) $\rightarrow$ Brother(x, Ram)

Brother(a, b) $\rightarrow$ b is brother of a

~~Predicate~~ Propositional logic

* ~~Predicate~~ Propositional logic is a primitive way of representing sentences using logical function uses function like ($\vee, \wedge, \rightarrow, \leftrightarrow$)

* Less complex and can be easily proved.

* No such membership quantifiers are present

Ex: ~~Ram~~ Ram is a guy, he is ~~ba~~ tall

P $\rightarrow$ Ram is a guy

Q $\rightarrow$ Ram is tall

F: P $\rightarrow$ Q

2 Game playing Problems in AI

- Wumpus World
- Chess
- Tic Tac Toe
- FiFa (Foot ball game, where the AI plays a opponent and takes decision based on users input)

③ Everyone who loves all animals is loved by Someone

let ~~fun~~ function be Loves(x,y) mean x loves y

Animal(x) → x is an animal

$\forall x \forall y$ Loves (x, Animal (y)) → $\exists_z$ Loves (z, x)

④ "Squares neighbouring the Wumpus are smelly."

Objects : Squares present in the grid which are adjacent to the Wumpus

Property : The squares adjacent to the Wumpus are smelly. The grid will have a pungent smell.

Relation : If an agent enters a square and encounter a pungent smell . it implies there's a Wumpus in any of the room adjacent to the current square

neighbour

⑤ When resoluting a set of clauses to arrive at a final clause, we are ought to remove all the existential quantifiers

- Presence of existential quantifier will weaken the proof in a lot of ways.
- In order to remove the quantifier we use a function called Skolem function

Skolem function is used to remove an existential quantifier by adding a constant to the function called skolem variable thereby removing the quantifier

Ex: $\exists x$ tall (x)

we can add a skolem const and make it $\boxed{\text{tall (a)}}$

~~a can be~~ substituted

This 'a' can be converted to required object when we use unifiers

6

⑥ Degree of Belief: When the probability of an event is uncertain and the agent doesn't have sufficient percepts to take an appropriate decision, It makes use of the percepts collected before.

(.) Based on the previous percepts-action values, the agent believes that the current state is some state which it already encountered and take a decision

(.) The prior probability which the agent uses to take a ~~seat~~ rational decision is called Degree of Belief

(.) In order to handle it
(.) The agent can try to collect as much as percepts at current situation
(.) when comparing it with prior instances, the degree of comparison should neither be too small or too large.

for ex: In order to reach an airport within 120 mins, and previously have arrived in 500 mins because of too much traffic.

7

(.) The agent cannot compare that with current scenario and say we cannot reach in 120 mins

(.) The agent should ~~us~~ ~~appro~~ appropriate ~~sense~~ of comparison

Part - B

3 (.) When 2 players are playing a game in order for each player to win, each player has to take the best decision in each of their turn.

(.) In order to win one player maximizes and other minimizes

(.) When we use a pruning tree, based on the children nodes, we can assign the best value needed in each step

(.) Algos like α-β pruning and min-max helps to do so.

(.) When we prune using α-β pruning, for certain (α-β) combination, we can say that's the best value and no need to even explore the rest of the ~~two~~ nodes for that parent

(.) Pruning ~~lead~~ has less time complexity and more efficiency in game trees.