Artificial Intelligence

CAT II

Answer Key

Part A

1. Propositional: Simple, connectives, atomic sentences, axioms, No. of inference rules
Predicate: Complex, connectives and quantifiers, properties and relations, inference rules

2. Tic tac toe, 8 puzzle, water jug problem, Backgammon, go, chess, Checkers

3. Everyone who loves all animals is loved by someone."
$\forall x[\forall y\ Person(x) \wedge Animal(y) \Rightarrow Loves(x, y)] \Rightarrow [\exists z\ Loves(z, x)]$

4. "Squares neighboring the wumpus are smelly".
Objects – Wumpus, Squares
Properties – smelly
Relationship - neighboring

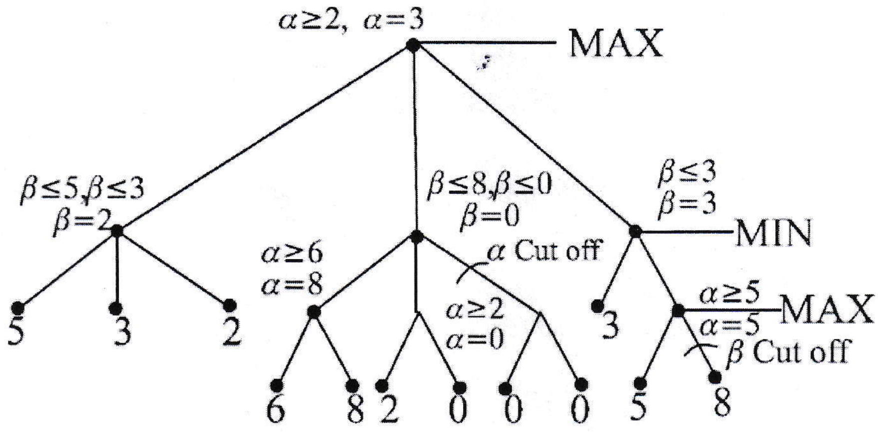
5. Removal of existential quantifier and its variable as a function $\forall x\ \exists y\ likes(x, y)$
$\forall x\ likes(x, F(y))$

.

6. Degree of belief ranges between 0 to 1.
Probability, fuzzy logic, certainty factor, Dempster Shafer theory

Part B

7. Eliminating a branch of a subtree, same answer as minimax with less complexity



9. P(A) = 0.10, P(B) = 0.05, P(B|A)=0.07
Bayes' theorem tells you:
P(A|B) = (0.07 * 0.1)/0.05 = 0.14
In other words, if the patient is an alcoholic, their chances of having liver disease is 0.14 (14%).

Part C

10. Unification: The process of finding all legal substitutions that make logical expressions look identical. This is a recursive algorithm.

Generalized Modus Ponen:

- This is a general inference rule for FOL that does not require instantiation
- Given:

$$\frac{p_1{}',\ p_2{}',\ \ldots,\ p_n{}',\ \ (p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q)}{q\theta}$$

  p1′, p2′ ... pn′ (p1 ∧ ... pn) ⇒ q
  Subst(θ, pi′) = subst(θ, pi) for all p
- Conclude:
  - Subst(θ, q)

function UNIFY(x, y, θ) **returns** a substitution to make x and y identical

inputs: x, a variable, constant, list, or compound

y, a variable, constant, list, or compound

θ, the substitution built up so far

if θ = failure **then return** failure

else if x = y **then return** θ

else if VARIABLE?(x) **then return** UNIFY-VAR (x, y, θ)

else if VARIABLE?(y) **then return** UNIFY-VAR (y, x, θ)

else if COMPOUND?(x) **and** COMPOUND?(y) **then**

return UNIFY(ARGS[x], ARGS[y], UNIFY(OP[x], OP[y],θ))

else if LIST?(x) **and** LIST?(y) **then**

return UNIFY(REST[x], REST[y], UNIFY(FIRST[x], FIRST[y],θ))

else return failure

function UNIFY-VAR(var, x, θ) **returns** a substitution

inputs: var, a variable

x, any expression

θ, the substitution built up so far.

if {var/val} ∈ θ **then return** UNIFY(val, x, θ)

else if{x/val} ∈ θ **then return** UNIFY(var, val, θ)

else if OCCUR-CHECK? (var, x) **then return** failure

else return {var/x} to θ

Assume the KB has:

Knows (*John*, *Jane*)

Knows (*y*, *Leo*)

Knows (*y*, *Mother*(*y*))

Knows (*x*, *Elizabeth*)

The results of unification are:

UNIFY (Knows (*John*, *x*), Knows (*John*, *Jane*)) = {*x*/*Jane*}

UNIFY (Knows (*John*, *x*), Knows (*y*, *Leo*)) = {*x*/*Leo*, *y*/*John*}

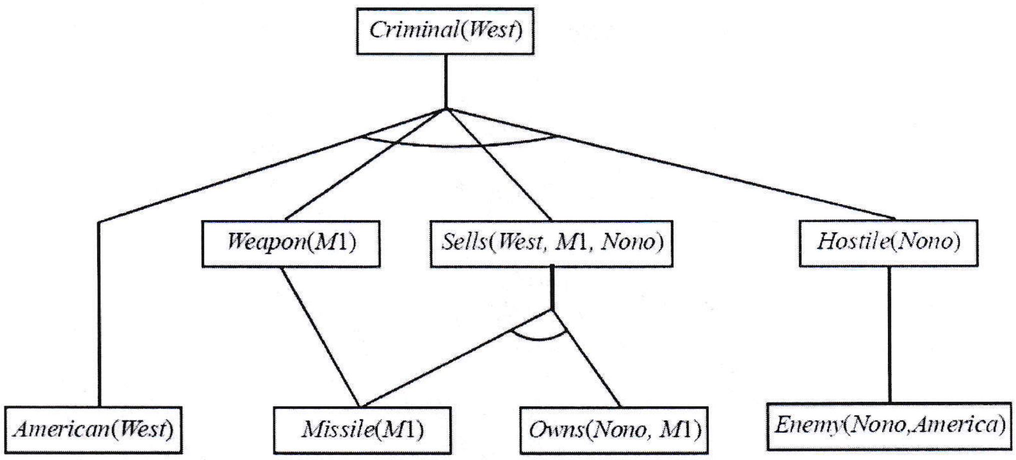
UNIFY (Knows (*John*, *x*), Knows (*y*, *mother*(*y*))={*y*/*John*, *x*/*Mother*(*John*)}

UNIFY (Knows (*John*, *x*), Knows (*x*, *Elizabeth*)) = fail

11. Forward chaining algorithm for first order logic

```
function FOL-FC-ASK(KB, α) returns a substitution or false
  repeat until new is empty
      new ← { }
      for each sentence r in KB do
          (p1 ∧ ... ∧ pn ⇒ q) ← STANDARDIZE-APART(r)
          for each θ such that (p1 ∧ ... ∧ pn)θ = (p'1 ∧ ... ∧ p'n)θ
                      for some p'1, ..., p'n in KB
              q' ← SUBST(θ, q)
              if q' is not a renaming of a sentence already in KB or new then do
                  add q' to new
                  φ ← UNIFY(q', α)
                  if φ is not fail then return φ
      add new to KB
  return false
```

Example



Forward Chaining

- From the sentences in the KB which in turn derive new conclusions.
- Forward chaining is preferred when new fact is added to the database, and we want to generate its consequences.

Backward Chaining

- Backward chaining is preferred when there is a goal to be proved.
- From the given conclusion find all the implications which attempts to establish their premises in turn

13. Sentence to FOL

1. It is a crime for an American to sell weapons to hostile nations.
   $\forall x, y, z\ American\ (x) \wedge Weapon\ (y) \wedge Nation\ (z) \wedge Hostile\ (z) \wedge Sells\ (x, y, z) \Rightarrow Criminal\ (x)$
2. Nono....has some missiles
   $\exists x\ Owns\ (Nono, x) \wedge Missile\ (x)$
3. All of its missiles were sold to it by Colonel west.
   $\forall x\ Owns\ (Nono, x) \wedge Missile\ (x) \Rightarrow Sells\ (West, x, Nono)$
4. We will also need to know that missiles are weapons.
   $\forall x\ Missile\ (x) \Rightarrow Weapon\ (x)$
5. An enemy of America counts as hostile.
   $\forall x\ Enemy\ (x, America) \Rightarrow Hostile\ (x)$
6. West, who is an American
   $American\ (West)$
7. Nono, is a nation
   $Nation\ (Nono)$
8. Nono, an enemy of America
   $Enemy\ (Nono, America)$
9. America is a nation
   $Nation\ (America)$

FOL to CNF

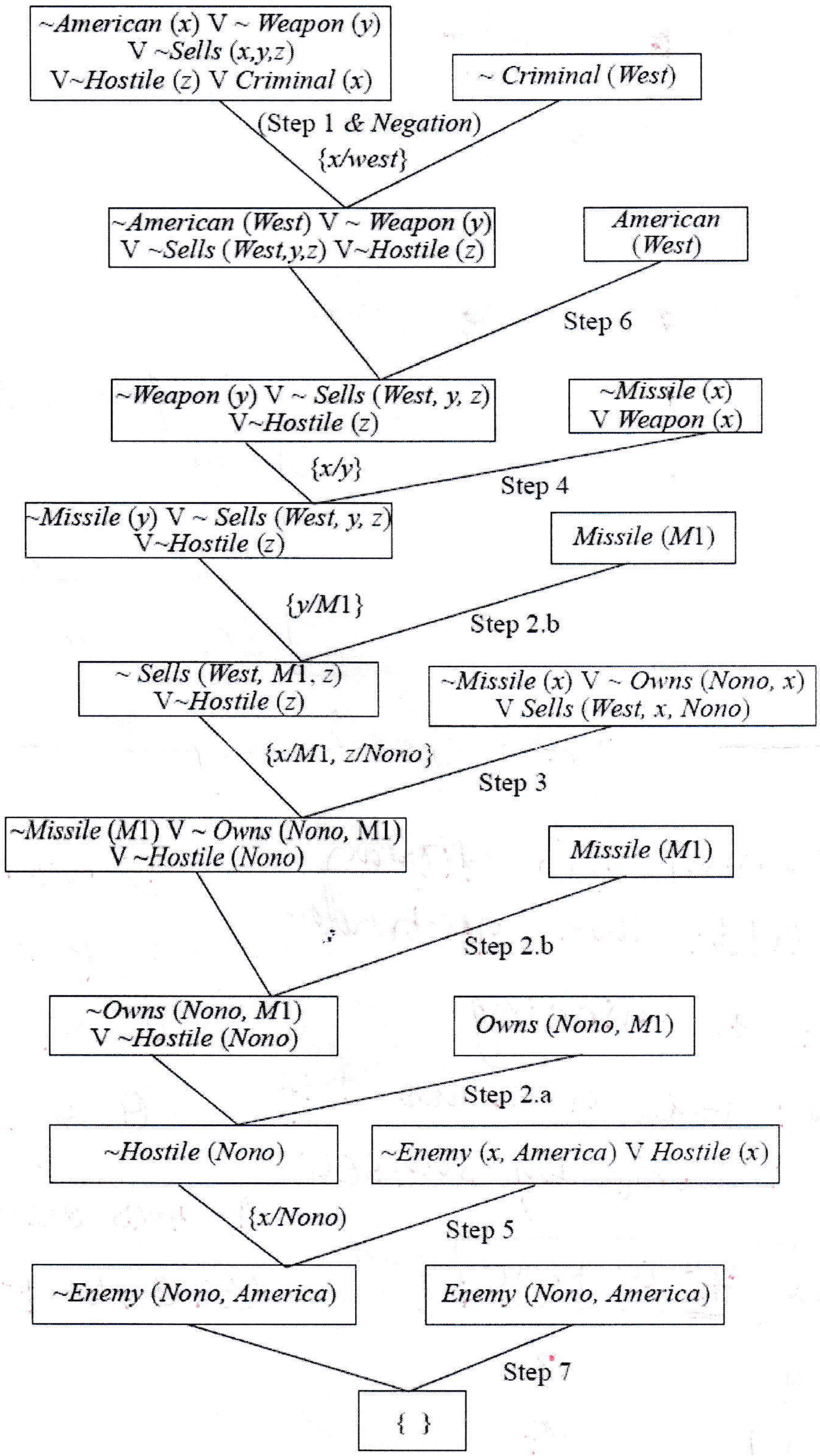
1. $\sim American(x) \vee \sim Weapon\ (y) \vee \sim Sells\ (x, y, z) \vee \sim Hostile(z) \vee Criminal(x)$

2 a. $Owns(Nono, M1)$

   b. $Missle(M1)$

3. $\sim Missile(x) \vee \sim Owns(Nono, x) \vee Sells\ (West, x, Nono)$
4. $\sim Missile(x) \vee Weapon\ (x)$
5. $\sim Enemy\ (x, America) \vee Hostile\ (x)$
6. $American\ (West)$
7. $Enemy\ (Nono, America)$

CNF to Resolution



Ans:

8. Homework → H

Flunk → F

S1: $H \vee F$

S2: $\sim H \rightarrow F$

S2:

H	F	$\sim H$	$\sim H \rightarrow F$
0	0	1	0
0	1	1	1
1	0	0	1
1	1	0	1

S1:

H	F	$H \vee F$
0	0	0
0	1	1
1	0	1
1	1	1

As per truth table both the stmts (S1, S2) are same.

$p \rightarrow q \approx \sim p \vee q$

$(\sim(\sim H)) \vee F$

$\equiv H \vee F$

12. S : Sunny this afternoon

C : colder than yesterday

(p) Swim : go swimming

(q) Canoe : take a canoe trip

(t) Sun : home by sunset

$\boxed{\text{Prove 't' is true}}$

① $\sim S \wedge C$	$\sim S$, C
② $p \rightarrow S$	$\sim p \vee S$
③ $\sim p \rightarrow q$	$p \vee q$
④ $q \rightarrow t$	$\sim q \vee t$

① AND elimination $\sim S$, C

②, ③, ④ $p \rightarrow q \approx \sim p \vee q$

Proof:

(i) $p \vee q$ — ③

(ii) $\sim q \vee t$ — ④

(iii) $p \vee t$ — resolution 3 & 4

(iv) $\sim p \vee s$ — ②

(v) $t \vee s$ — resolution (iii) & 2

(vi) $\sim s$ — ①

(vii) t — resolution using (v) &

Rules: ⑥