Menoufia University

Subject:

Knowledge Based System

Faculty of Computers & information

Department: CS & IT

1st semester Final Exam

3 pages

Year: 4th year

Time Allowed: Three Hours

Full Mark: 70

Please Answer All Questions;

Question 1 **12 Marks**

1. Illustrate the basic architecture of Knowledge based system.
2. Why is the concept of knowledge different from the concepts of data and information?
3. There are distinct phases when transforming human knowledge into some form of KBS. The knowledge engineer plays a prominent role in these phases. What are these phases?
4. What are the difficulties of developing the KBS ?
5. Draw a hierarchal diagram about the knowledge management roles and their responsibilities.

Question 2 **16 Marks**

1. Suppose we have a working memory consisting of the following assertions,
   a. Use the Forward-chaining technique to diagnose the following symptoms.
   b. Use the Backward-chining technique to ensure your previous diagnosis.

1: runny nose
2: temperature = 104°F
3: headache
4: cough
The rules in our rule base are:
R1: if (nasal congestion and viruses), then diagnose (influenza) exit
R2: if (runny nose), then assert (nasal congestion)
R3: if (body aches), then assert (itchiness)
R4: if (temp > 100), then assert (fever)
R5: if (headache), then assert (itchiness)
R6: if (fever and itchiness and cough), then assert (viruses)

We have rules 1 through 6, as with the previous example, and four assertions. The goal is diagnosis (influenza).

Initially, R1 fires: Since the goal, diagnosis (influenza), goes with the conclusion of the rule R1, new goals are created: (nasal congestion) and (virosis), and back chaining is recursively used with these recent goals.

R2 fires, matching the goal nasal congestion. A new goal is created: (runny nose). Back chaining is recursively used. Since (runny nose) is in working memory, it returns true.

R6 fires, matching the goal virosis. Back-chaining recursion takes place with new goals: (fever), (itchiness), and (cough)

R4 fires, adding the goal (temperature > 100). Since (temperature = 104) is in working memory, it returns true.

R3 fires, adding the goal (body aches).

On recursion, there is no information in working memory or rules that match this goal. Therefore, it returns false and the next matching rule is chosen.

Here, the rule is R5, which fires, so the goal (headache) is added. Since (headache) is in working memory, it returns true.

The goal (cough) is in working memory, so that returns true.

Finally, the entire recursive procedures have returned true. This proves the result.

1. Give one predicate calculus representation for each of the following English sentences.

a) Not all people have a cell phone
b) Students who took CS101 but did not take CS102 must take Math103.
c) All Germans speak the same languages

a. **Not all people have a cell phone.**

∃p person(p) ^ ~hasCellPhone(p)

b. **Students who took CS101 but did not take CS102 must take Math103.**

∀s student(s) ^took(s,CS101)^ ~took(s,CS102) → requiredToTake(s,Math103)

c. ∀ *x,y,l German(x)^German(y)^Speaks(x,l)=> Speaks(y,l)*

2. Use truth tables to determine whether each of the following is a *valid*, *satisfiable*, or *unsatisfiable* sentence in propositional logic.

a) (P → Q) ↔ (¬P ∨ Q) b) ((P → Q) ^ (Q → R)) ↔ (P → R)

(P → Q) ←→ (~P ∨Q)

P	Q	P ! Q	~P ∨ Q	WFF
F	F	T	T	T
F	T	T	T	T
T	F	F	F	T
T	T	T	T	T

Since the formula is always satisfied, we say that the formula is **valid.**

.

((P → Q) ^ (Q →R)) ←→ (P→ R)

P	Q	R	P→ Q	Q → R	(P→ Q) ^ (Q →R)	P→ R	WFF
F	F	F	T	T	T	T	T
F	F	T	T	T	T	T	T
F	T	F	T	F	F	T	F
F	T	T	T	T	T	T	T
T	F	F	F	T	F	F	T
T	F	T	F	T	F	T	F
T	T	F	T	F	F	F	T
T	T	T	T	T	T	T	T

The formula is satisfied by some but not all assignments, so we say it is **satisfiable**.

Question 4 16 Marks

1.

Dana has been murdered. Alfonso, Bonnie, and Clyde are suspects. Only one is guilty and the other two are innocent. The innocent ones told the truth to the police, but the guilty one may have lied. Alfonso said that Bonnie and Dana were friends and that Clyde did not like Dana. Bonnie said that she was not in town at the time of the murder, and moreover, she did not know Dana. Clyde said that he saw both Alfonso and Bonnie with Dana just before the crime was committed. Your job is to prove that Bonnie is the murderer (i.e., murderer (B)). You should do this via a proof by contradiction. I.e., assume ¬ murderer (B) and show that this leads to a something of the form P ∧ ¬P, which is a contradiction since P cannot not be both true and false. First convert all the wff's below of the form P → Q into the logically equivalent form ¬P ∨ Q; this will make it easier to use the inference rules involving resolution.

If Alfonso is innocent, Bonnie and Dana were friends and Clyde did not like Dana.

1a. innocent(A) → friends(B, D)
1b. innocent (A) → ¬likes(C, D)

If Bonnie is innocent, she was not in town, and she did not know Dana.

2a. innocent (B) → ¬inTown(B)
2b. innocent (B) → ¬knows(B, D)

If Clyde is innocent, Alfonso and Bonnie were both with Dana just before Dana was murdered.
3a. innocent (C) → with(A, D)
3b. innocent (C) → with(B, D)

Everyone who was with Dana was in town. (Some slightly contrived background knowledge.)
4. ∀x with(x,D) → inTown(x)

Everyone knows his/her friends. (Some common-sense, background knowledge.)
5. ∀x ∀ y friends (x, y) → knows(x, y)

Everyone knows the people he/she likes. (Some common-sense, background knowledge.)
6. ∀ x ∀ y likes(x, y) → knows(x, y)

Among Alfonso, Bonnie, and Clyde, two are innocent.
7a. innocent (A) ∨ innocent (B)
7b. innocent (A) ∨ innocent (C)
7c. innocent (B) ∨ innocent (C)
If someone is not innocent, then he/she is the murderer.

6. ∀ x ¬innocent (x) → murderer(x)
7. We will use proof by contradiction to prove that Bonnie is the murderer. We will assume that :murderer(B) and then using inference show that a contradiction arises as a result of this assumption. First, we convert the givens to forms that don't involve implications.
8.
9. Number WFF Justification
10.
11. 1a. ~innocent(A) ∨ friends(B,D) Given
12. 1b. ~innocent(A) ∨ ~likes(C,D) Given
13. 2a. ~innocent(B) ∨~inTown(B) Given
14. 2b. ~innocent(B) ∨ ~knows(B,D) Given
15. 3a. ~innocent(C) ∨ with(A,D) Given
16. 3b. ~innocent(C) ∨ with(B,D) Given
17. 4. ∀x~with(x;D) ∨ inTown(x) Given
18. 5. ∀x ∀y ~friends(x,y) ∨ knows(x, y) Given
19. 6. ∀x ∀y ~likes(x, y) ∨ knows(x, y) Given
20. 7a . innocent(A) ∨ innocent(B) Given
21. 7b. innocent(A) ∨ innocent(C) Given
22. 7c. innocent(B) ∨ innocent(C) Given
23. 8. ∀x innocent(x) ∨ murderer(x) Given
24. Now we can perform inference to complete our proof.
25.
26. Number WFF Justification
27.
28. 9. ~murderer(B) Assumption
29. 10. innocent(B) ∨ murderer(B) Universal Instantiation of 8 with {x/B}
30. 11. innocent(B) Resolution of 9 and 10
31. 12. ~inTown(B) Resolution of 2a and 11
32. 13. ~knows(B,D) Resolution of 2b and 11
33. 14. ~friends(B,D) ∨knows(B;D) Universal Instantiation of 5 with {x/B, y/D}
34. 15. ~friends(B,D) Resolution of 13 and 14
35. 16. ~innocent(A) Resolution of 1a and 15
36. 17. innocent(C) Resolution of 7b and 16
37. 18. ~with(B,D) ∨ inTown(B) Universal Instantiation of 4 with {x/B}
38. 19. ~with(B,D) Resolution of 12 and 18
39. 20 .~innocent(C) Resolution of 3b and 19
40. 21. innocent(C) ^ ~innocent(C) AND Introduction on 17 and 20
41. Clearly, WFF number 21 is a contradiction. Thus, our assumption cannot be correct and we can conclude that Bonnie is the murderer. You could also have shown a similar contradiction involving A if you performed your steps in a different order.

Question 5 16 Marks

2. i. Write down logical representations for the following sentences, suitable for use with Generalized Modus Ponens:

a. Horses, cows, and pigs are mammals.
b. An offspring of a horse is a horse.
c. Bluebeard is a horse.
d. Bluebeard is Charlie's parent.
e. Offspring and parent are inverse relations.
f. Every mammal has a parent.

ii. Draw the proof tree generated by an exhaustive backward-chaining algorithm for the query ∃ h Horse *(h)* where clauses are matched in the order given.

a. Horse (x) => Mummal (x)
   Cow (x) => Mummal (x)
   Pig (x) => Mummal (x)
b. Offspring(x,y) ^ Horse (y)=> Horse (x)
c. Horse(Bluebeard)
d. Parent(Bluebeard, Charlie)
e. Offspring(x,y) => Parent(y,x)
   Parent(x,y) => Offspring (y.x)
   (Note we couldn't do Offspring (x,y) <=> Parent (y,x) because that is not in the form expected by generalized Modus Ponens)
f. Mummal(x) => Parent (G(x),x) (here G is a Skolem function)

3. Draw the proof tree generated by an exhaustive backward-chaining algorithm for the query
∃ h Horse *(h)* where clauses are matched in the order given.

Horse (h)

Offspring(h,y) Horse (y)

Parent(y,h) Horse(Bluebeard)

Parent(Bluebeard, Charlie) {y /Bluebeard, h /Charlie}

3. Put these propositional-logic sentences into *clausal form*.
   a) (P ^ Q) → (A ∨ B)
   b) (P ∨ Q) → (A ^ B)

a. (P ^ Q) →(A ∨ B)
  (P ^ Q) → (A ∨ B)
~(P ^ Q) ∨ (A ∨ B) definition of implication
~P ∨ ~Q ∨ A ∨ B DeMorgan's Law

b. (P ∨Q) → (A ^ B)

$(P \vee Q) \rightarrow (A \wedge B)$

$\sim(P \vee Q) \vee (A \wedge B)$ definition of implication

$(\sim P \wedge \sim Q) \vee (A \wedge B)$ DeMorgan’s Law

$(\sim P \vee (A \wedge B)) \wedge (\sim Q \vee (A \wedge B))$ distribution of _ over ^

$(\sim P \vee A) \wedge (\sim P \vee B) \wedge (\sim Q \vee A) \wedge (\sim Q \vee B)$ distribution of $\vee$ over ^

With my best wishes
Dr. Heba Elbeh