Joseph Scarabino 9/20/11

HW 1

Problem 1

2. One expert system that would be useful would be for a gas powered grill. Many problems can arise in a grill, especially a large one with several burners. This knowledge may not be known by everyone, but will frequently be used. Equipment maintenance is a mundane task that can be automated. Information that will probably be needed by the system would include: if the propane tank is empty, if all the burners are lit, if you smell gas, if you can hear the igniter.

Another expert system that would be, and is, useful is a system to help people do their taxes. Everyone must do their taxes every year, and it is not easy for most. An expert system is very helpful for this task. Information that would be needed would include: income, children, dependents, write offs, etc.

4. I feel that in order to get a feel for whether you are talking to a computer or a human, unrelated questions will not do the trick. I would try to carry on a conversation with it. Humans seem to have an ability to keep a conversation flowing, where a computer could only provide automated responses to a specific question. In this situation, I believe the computer would answer the question and probably sound human, but would then attempt to change the subject.

Another strategy would be to draw upon emotions, as a human would be better suited to describe feelings than a machine. Ask about specific emotional events, such as the September 11 attacks, and how it made them feel. In this instance, the computer would probably provide some background information on the event and use adjectives to describe the event instead of reflecting on its own emotions.

Problem 2

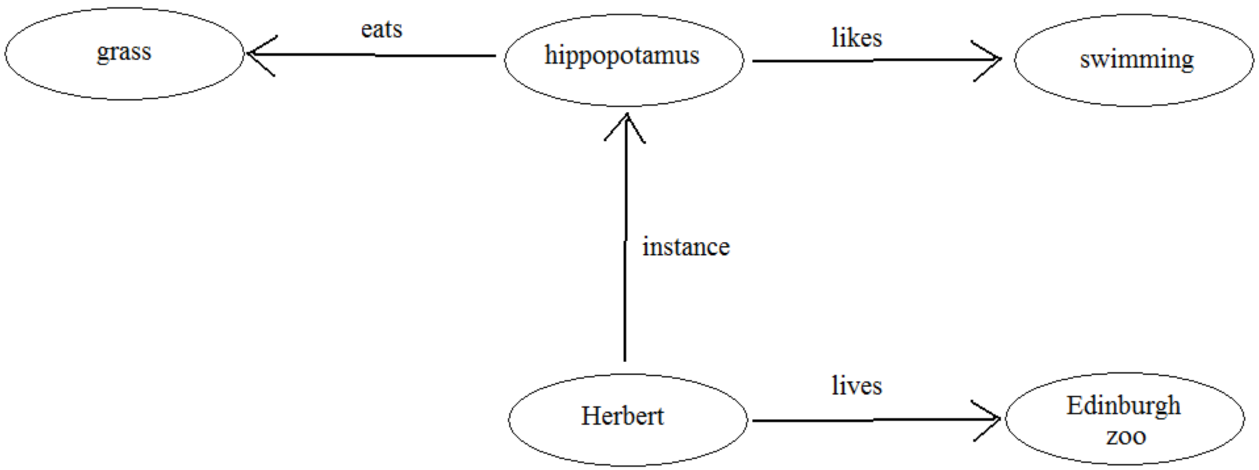
(should have more detailed work shown –like in Problem 3 solutions)

2. Every apple is either green or yellow: $\forall x\,(apple(x) \rightarrow green(x) \lor yellow(x))$

No apple is blue: ¬E x (apple(x)& blue(x))

If an apple is green then it is tasty: $\forall x\,(apple(x) \land green(x) \rightarrow tasty(x))$

Every man likes a tasty apple: $\forall x\,(man(x) \rightarrow \exists y\,(apple(y) \land tasty(y) \land likes(x,y)))$



3a.

3b. $\forall$ x (hippopotamus(x) → eats(x, grass) ∧ likes(x, swimming))
∃ x (hippopotamus(x) ∧ name(x, herbert) ∧ lives(x, edinburgh zoo))

3aa. Unlike most hippos, Herbert is brown.
Like all other hippos, Herbert being in the sun.

3bb. Herbert does not like being alone.
If people are around then Herbert does tricks.

Problem 3

1. Alison is friends with Richard.
Fred's father is friends with Joe's father.
Someone likes Richard.

If Alison is friends with Richard then Alison likes Richard.
Alison likes Richard or Alison likes chocolate.
Alison likes Richard or Alison likes chocolate, and Alison does not like chocolate. (Alison likes Richard.)

2. Alison eats everything that she likes.
Predicates: likes(x,y) - x likes y
eats(x,y) - x eats y
Restricted: $\forall_{likes(alison,x)}$ eats(alison,x)
Logic: $\forall$ x (likes(alison,x) $\rightarrow$ eats(alison,x))

There exists some bird that doesn't fly.
Predicates: bird(x) - x is a bird
flies(x) - x can fly
Restricted: $\exists_{bird(x)} \neg$ flies(x)
Logic: $\exists$ x (bird(x) $\wedge$ $\neg$ flies(x))

Every person has something that they love.
Predicates: person(x) - x is a person
loves(x,y) - x loves y
Restricted: $\forall_{person(x)} \exists_y$ loves(x,y)
Logic: $\forall$ x (person(x) $\rightarrow$ $\exists$ y (loves(x,y)))

In predicate logic, unlike propositional logic, relations may be descriptive. This gives an insight into the relationship, as opposed to arbitrary symbols to denote the relation. As long as the relation is specifically defined and used consistently, this makes predicate logic a great tool.

Problem 4

1. S : savings are adequate

I : income is adequate
A : invest savings
B : invest stocks
C : has children
P : has a partner
J : partner has a job

Rules R1: If not S, then add A
(NO need to write it in words form, NO need to write add!- this is a part of the conflict resolution)

$\neg S \rightarrow A$

R2: If S and I, then add B

$S \wedge I \rightarrow B$

R3: If not C, then add S

$\neg C \rightarrow S$

R4: If P and J, then add I

$P \wedge J \rightarrow I$

Conflict Resolution

1. Rules are numbered according to their names.
2. During each scanning of the list, applicable rules are pushed into a stack successively.
3. After each scan, one rule from the top of the stack is fired.
4. During reach session, any rule may be fired once.

a) IDB { C, P, J }
Goal is A.
A is not one of the initial facts, so we must use Rule R1 since it is the only rule that fires A.

Subgoal is ¬S.
S is not one of the initial facts, so we must use rule R3 since it is the only rule that fires S.
Subgoal is ¬C.
C is an initial fact. Since C and ¬C cannot both be true, ¬C must be false.
Therefore, A cannot be proven.

b) S : savings are adequate
I : income is adequate
A : invest savings
B : invest stocks
C : has children
P : has a partner
J : partner has a job
E : expenses are/will be high
L : job is low-paying
H : job is high-paying
M : you have a job

Rules
R1: If not S, then add A
R2: If S and I, then add B
R3: If not C and not E, then add S
R4: If C and E, then add not S
R5: If P and J and H, then add I
R6: If P and J and L, then add not I
R7: If M and H and not E, then add S

Problem 5

A: the alarm is ringing

B: there is a burglar
C: you are coughing
F: there is a fire
H: it is hot
P: there is a burst pipe
R: it is raining
S: it is smoky
W: you are wet

Rules

R1: If C, then add S
$C \rightarrow S$

R2: If W and not R, then add P
$W \wedge \neg R \rightarrow P$

R3: If not C and A, then add B
$\neg C \wedge A \rightarrow B$

R4: If S and H, then add F
$S \wedge H \rightarrow F$

Conflict Resolution

1. Rules are numbered according to their names.
2. During each scanning of the list, applicable rules are pushed into a stack successively.
3. After each scan, one rule from the top of the stack is fired.
4. During reach session, any rule may be fired once.

The questions in the system arise when information is needed that cannot be deduced from a rule in the system.

Goal is F.
Use rule R4, since it is the only one that proves F.

Subgoals are S and H.
Use rule R1, since it is the only one that proves S.
Subgoal is C.
Since there is no information in the initial database, we must ask the user.
Ask "Are you coughing?"

- If yes, add C to DBF
  Add S to DBF using rule R1
  Return to subgoal H.
  Since there is no information regarding H, we must ask.
  Ask "Are you hot?"

  - If yes, add H to DBF.
    Add F to DBF using rule R4.
    Conclude that there is a fire.

  - If no, add ¬ H to DBF
    Subgoal F cannot be proved.
- If no, add ¬ C to DBF.

If no conclusion yet, continue to Goal P.
Use rule R2, since it is the only rule that proves P.
Subgoals W and ¬ R.
Since there is no rule that proves W, we must ask the user.
Ask "Are you getting wet?"

- If yes, add W to DBF
  Return to subgoal ¬ R.
  Since there is no rule to prove ¬ R, we must ask the user.
  Ask "Is it raining?"

- If no, add ¬ R to DBF
  Using rule R2, add P to DBF
  Conclude that there is a burst pipe.

- If yes, add R to DBF
  No conclusions can be drawn from this.

- If no, add ¬ W to DBF.
  No conclusions can be drawn from this.

If no conclusion yet, continue to Goal B.
Use rule R3, since it is the only rule that proves B.
Subgoals ¬ C and A.
We already asked the user if he is coughing.

- If C is already in the DBF, no conclusion can be drawn.
- If ¬ C is already in the DBF, we must ask the user about the alarm.
  Ask "Is there an alarm ringing?"
  - If yes, add A to DBF.
    Using rule R4, add B to DBF.
    We can conclude that there is a burglary.

  - If no, add ¬ A to DBF.
    No conclusions can be drawn from this.

Problem 6

2a. DBF = { A, B, ¬ D, ¬ H, I }
During the first scan, rule R1 is fired.
C is added to the DBF

DBF = { A, B, C, ¬ D, ¬ H, I }

During the second scan, rule R2 is fired.

E is added to the DBF.

DBF = { A, B, C, ¬ D, E, ¬ H, I }

During the third scan, rule R3 is fired.

E is already in the DBF

During the fourth scan, rule R7 is fired.

G is added to the DBF

DBF = { A, B, C, ¬ D, E, G, ¬ H, I }

During the fifth scan, no applicable rules.

Final DBF = { A, B, C, ¬ D, E, G, ¬ H, I }

2b. DBF = { A, B, D, E, I }

During the first scan, rule R1 is fired.

C is added to the DBF.

DBF = { A, B, C, D, E, I }

During the second scan, rule R4 is fired.

F is added to the DBF.

DBF = { A, B, C, D, E, F, I }

During the third scan, rule R5 is fired.

L is added to the DBF.

DBF = { A, B, C, D, E, F, I, L }

During the fourth scan, rule R8 is fired.

J is added to the DBF

DBF = { A, B, C, D, E, F, I, J, L }

During the fifth scan, rule R9 is fired.

K is added to the DBF

DBF = { A, B, C, D, E, F, I, J, K, L }

During the sixth scan, no applicable rules.
Final DBF = { A, B, C, D, E, F, I, J, K, L }

2c. DBF = { A, B, ¬ D, E }
Goal is L.
Use rule R5, since it is the only rule that proves L.
Subgoals E and F.
E is already in the DBF so it is satisfied.
Use rule R4, since it is the only rule that proves F.
Subgoals C and D.
¬ D is in the DBF, therefore D cannot be true.
L is not supported.

2d. DBF = { A, ¬ D, ¬ H, I }
Goal is K and L.
Use rule R9, since it is the only rule that proves K.
Subgoal J.
Use rule R8, since it is the only rule that proves J.
Subgoal I.
I is in the DBF already, so it is satisfied.
Add J to DBF due to rule R8. DBF = { A, ¬ D, ¬ H, I, J }
Add K to DBF due to rule R9. DBF = { A, ¬ D, ¬ H, I, J, K }
K is satisfied. Return to Goal L.
Use rule R5, since it is the only rule that proves L.
Subgoals E and F.
Use rule R4, since it is the only rule that proves F.
Subgoals C and D.
Since ¬ D is in the DBF, D cannot be true.

Therefore F cannot be proven using R4, and there is no other rule to prove it.
L cannot be proven, so goal { K, L } cannot be proven.

4a. DBF = { ¬ A, B, C }
Goal G.
Try to use rule R9 to prove G, though there is another that can be used.
Subgoal ¬ E.
Use rule R3, since it is the only rule that proves ¬ E.
Subgoals ¬ A and B.
Both ¬ A and B are in the DBF.
Fire rule R3 and add ¬ E to the DBF. DBF = { ¬ A, B, C, ¬ E }
Fire rule R9 and add G to the DBF. DBF = { ¬ A, B, C, ¬ E, G }
Goal { G } is supported by the DBF.

4b. DBF = { ¬ A, B, C }
Using the same steps as in problem (4a), we can start with DBF = { ¬ A, B, C, ¬ E, G }.
Goal H.
Try to use rule R7 to prove H, though there is another that can be used.
Subgoals D and ¬ E.
¬ E is already in the DBF so it is satisfied.
Use rule R1, since it is the only one that can prove D.
Subgoals A and B.
Since ¬ A is in the DBF, A cannot be true.
Therefore D cannot be proven.
Rule R7 cannot be used to prove H.
Use rule R8 to prove H, since it is the only applicable rule left.
Subgoals E and F.
Since ¬ E is already in the DBF, E cannot also be true.

Therefore H cannot be proven using rule R8.
Since there are no more applicable rules to prove H, { H } is not supported by the DBF.

4c. DBF = { ¬ A, B, C }
Using the same steps as in problem (4a), we can start with DBF = { ¬ A, B, C, ¬ E, G }.
Goal I.
Use rule R6, since it is the only one that can be used to prove I.
Subgoals D and E.
Since ¬ E is already in the DBF, E cannot also be true.
Therefore I cannot be proven using rule R6.
Since there are no other applicable rules to prove I, { I } is not supported by the DBF.