

Bryant Servello
CSE 352 – Homework 1
9/21/10

Problem 1

(2)

Expert System 1: Wildlife Recognition

Description: Allow a user to identify an organism based on its traits and behavior

Use: This system would be useful for anyone who is curious about a plant or animal they have seen (consider bird watching books, botanical encyclopedia, etc). By making an expert system with this functionality, it would make searching for an animal much easier.

Knowledge Needed:

The expert rules for this system would encode facts about a plant or animal to narrow down the search. For instance:

$\text{hasFeathers}(x, \text{true}) \wedge \text{hasBeak}(x, \text{true}) \rightarrow \text{type}(x, \text{bird})$

$\text{type}(x, \text{mammal}) \wedge \text{size}(x, \text{large}) \wedge \text{hasTusks}(x, \text{true}) \rightarrow \text{species}(x, \text{elephant})$

The user input ‘facts’ would be the attributes of the organism. For instance:

Large

Brown

Furry

4 Legs

This might be enough data for the system to identify a bear (pictures could be used for verification).

Expert System 2: Vacation Scheduler

Description: Help the user select a perfect vacation

Use: This system would help the user pick the vacation best suited to his/her desires and budget

Knowledge Needed:

The expert rules for this system would process the user’s desires. For instance:

$\text{temperature}(x, \text{warm}) \wedge \text{swimming}(x, \text{true}) \rightarrow \text{possibleDestination}(x, \text{Hawaii})$

From the user, the system would need parameters to be given for what is desirable in a vacation to form the database of facts. For instance:

Warm

Sunny

SCUBA diving

Coral Reef

These ‘facts’ (gained by asking the user questions) would help the system produce a list of suitable locations to travel to.

(4)

If I were a judge in the Loebner Contest, I think I would use mostly questions involving “why.” Since “why” questions often have very complicated, in-depth answers, they present a hurdle for a machine. Questions such as these might be tricky:

“Why do people work every day?”

“Why did you decide to participate in this event?”

I would also ask things which are more conversational than data-related. Here are a few examples:

“How was your trip here?”

“What’d you think of the lecture today?”

Alternatively, instead of looking for answers that do not quite hold up to my expectations, I could ask questions which an AI entity would be *better* qualified to answer. Questions like:

“How far is it between the Earth and Mars?”

“What is 9736.5 divided by 122.6?”

These questions, barring an exceptionally smart human or an AI programmed carefully to avoid the appearance of being “too smart,” would reveal if the respondent was a human or machine.

Finally, questions that are interested in an emotional response would be particularly devious:

“How did you feel the last time you were lost?”

“Have you ever been in love?”

“What are you most afraid of? Why?”

Clever programming might provide answers for some of these question tropes.

“Why” questions can often be answered with “I don’t know” or something similar. Many humans will take this easy way out rather than pondering the question, particularly if the question is intimidating in its depth or its complexity. Also, responding with “I don’t know” subtly makes the machine seem more human since we expect computers to be knowledgeable. Of course, this response has a few pitfalls. Easy questions to which the robot answers “I don’t know” will almost certainly reveal it as AI.

Personal questions can be handled, though it has to be done carefully. Especially ones that are conversational, since those must be very fluid and natural in order to seem human (people are experienced at answering questions in a conversational capacity, so the answers are usually very pertinent to the question rather than digressing). Looking for operative words in the question can help, especially in determining what the response type should be. Also, non-committal or terse answers can make the AI seem more human while not directly answering the question. For instance:

“How was your trip here?”

“Alright”

The brevity of the answer helps keep the robot from stumbling as well.

Trick questions that test to see if the robot is “too smart” can be fooled by making the program hesitate before providing numerical answers, or use some rounding in the way a human might. For instance:

“What is 9736.5 divided by 122.6?”

“About 80, I think”

Emotional questions may have to have some associations in order to seem natural. These could come straight from the programmer's experience. For instance:

"What are you most afraid of?"

"Oh, I'm terrified of heights!"

Alternatively, these questions could be "dodged" by changing the subject slightly, or feigning reticence. Like this:

"Have you ever been in love?"

"I'd rather not talk about that..."

Since the question was meant to have an emotional significance, it's natural that a person might not wish to talk about it. Furthermore, the judge would be inclined not to push the matter in order to avoid offending the participant.

Problem4

(1)

A = Savings Adequate

B = Invest Savings

C = Income Adequate

D = Invest Stocks

E = Has Children

F = Has Partner

G = Partner Works

R1: $\neg A \rightarrow B$

R2: $A \wedge C \rightarrow D$

R3: $\neg E \rightarrow A$

R4: $F \wedge G \rightarrow C$

a) Goal: D

Facts: EFG

Backwards Chaining:

R2: Add A & C as sub-goals

To get A, use R3

R3 requires $\neg E$. E is in the DBF. Goal unreachable

b) Extend to include:

More children cost more

Partner's income is low

Big expenses require savings

A = Savings Adequate

B = Invest Savings

C = Income Adequate

D = Invest Stocks

E = Has Children

F = Has Partner

G = Partner Works

H = Has Many Children

I = Partner's Income is Low

J = Savings Good

K = Can Buy House

R1: $\neg A \rightarrow B$

R2: $A \wedge C \rightarrow D$

R3: $\neg E \vee J \rightarrow A$

R4: $F \wedge G \wedge \neg I \wedge \neg H \rightarrow C$

R5: $J \rightarrow K$

To make complete sense, more rules would need to be added. For example, the problem leaves it ambiguous about how savings can be achieved, so I have not added any rules which add J to the DBF. Unfortunately, adding such a rule would require the inclusion of additional atomic formulae, so I leave it out.

(2)

savings(x, not adequate) → invest(x, savings)
 savings(x, adequate) ∧ income(x, adequate) → invest(x, stocks)
 children(x, 0) → savings(x, adequate)
 partner(x, true) ∧ income(partner(x), adequate) → income(x, adequate)

Note: a few things here have changed from the original phrasing:

-“has no children” became “has 0 children”

-“partner has a job” became “partner has an adequate income”

The meaning remains largely unchanged, and this way of expressing it is more elegant

(3)

ID	Income	Savings	Invest	Children	Partner
----	--------	---------	--------	----------	---------

Income & Savings will contain values such as “adequate”

Invest will contain values such as “savings” or “stocks”

Children will contain an integer

Partner will contain the partner’s ID

Problem 5

A = Coughing

G = Burst Pipe

B = Smoky

H = Burglar

C = Wet

I = Fire

D = Raining

E = Alarm Rings

F = Hot

R1: $A \rightarrow B$

R2: $C \wedge \neg D \rightarrow G$

R3: $\neg A \wedge E \rightarrow H$

R4: $B \wedge F \rightarrow I$

The questions asked by the system determine the state of the surrounding world. By filling in its database of facts with true and false information about the situation, it can narrow down the number of possible scenarios.

Backwards Chaining:

S: = System Asks ___ U: = If User response is ___

Check for Fire:	Goal(I)
1) S: Are you coughing?	
2) U: Yes	Add(A,B)
3) S: Is it hot?	
4) U: Yes	Add(F)
5) S: There is a Fire!	Add(I)
4) U: No	Add(\neg F, \neg I)
5) Check for Burst Pipe	
2) U: No	Add(\neg A, \neg I)
3) Check for Burst Pipe	
Check for Burst Pipe:	Goal(G)
1) S: Are you wet?	
2) U: Yes	Add(C)
3) S: Is it raining?	
4) U: Yes	Add(D, \neg G)
5) Check for Burglar	
4) U: No	Add(\neg D,G)
5) You have a Burst Pipe!	
2) U: No	Add(\neg C, \neg G)
3) Check for Burglar	
Check for Burglar:	Goal(H)
1) S: Is there an alarm ringing?	
2) U: Yes	Add(E)
3) If(A)	Add(\neg H)
S: Everything is normal	
4) If(\neg A)	Add(H)
S: There is a Burglar!	

The system will check each of the 3 end conditions using backward chaining. By asking questions about the cases that are symptomatic of an end condition, it can determine which state the world is in. This is characteristic of backward chaining because it starts with the possible end states, and then tries to determine the facts by seeing what could lead to each goal end state.

Problem 6

(1)(2a)

Cycle	DBF contents	Rule Queue
0	A,B, \neg D, \neg H,I	R1, R2
1	A,B, \neg D, \neg H,I,C	R2,R3
2	A,B, \neg D, \neg H,I,C,E	R3,R7
3	A,B, \neg D, \neg H,I,C,E	R7
4	A,B, \neg D, \neg H,I,C,E,G	

(2b)

Cycle	DBF contents	Rule Queue
0	A,B,D,E,I	R1,R8
1	A,B,D,E,I,C	R8,R4
2	A,B,D,E,I,C,J	R4,R9
3	A,B,D,E,I,CJ,F	R9,R5
4	A,B,D,E,I,CJ,F,K	R5
5	A,B,D,E,I,CJ,F,K,L	

(2c)

Initial DBF: A,B, \neg D,E

Cycle	Goal Stack	Comment
0	L	Start
1	F	Rule 5 & E is in DBF
2	C,D	Rule 4

L is NOT supported.

We are given \neg D

F can only be achieved with C & D, so F cannot be reached.

L can only be reached with E&F, so L cannot be reached.

(2d)

Initial DBF: A, \neg D, \neg H,I

Cycle	Goal Stack	Rule	Comment
0	K,L		Start
1	J,L	9	
2	L	8	I is in the DBF
3	E,F	5	
4	F	2	A and \neg D are in the DBF
5	C,D	4	Cannot use D

{K,L} is not supported.

(4a)

DBF: $\neg A, B, C$

Cycle	Goal Stack	Rule	Comment
0	G	-	Start
1	D, ($\neg E$)	5,9	Conflict resolution: 5 fires first
2	A, B, ($\neg E$)	1	Cannot use A ($\neg A$ in DBF)
3	$\neg E$	-	Alternative path
4	$\neg A, B$	3	Done

G is supported

(4b)

DBF: $\neg A, B, C$

Cycle	Goal Stack	Rule	Comment
0	H	-	Start
1	D, $\neg E, (E, F)$	7,8	Conflict Resolution: 7 fires first
2	A, B, $\neg E, (E, F)$	1	Cannot use A ($\neg A$ in DBF)
3	E, F	-	Alternative path
4	B, C, F	4	B and C in DBF
5	$\neg A$	2	Done

H is supported

(4c)

DBF: $\neg A, B, C$

Cycle	Goal Stack	Rule	Comment
0	I	-	Start
1	D, E	6	
2	A, B, E	1	Cannot use A ($\neg A$ in DBF)

I is not supported