Knowledge-based systems in bioinformatics – 1MB602

Exam 2006-12-21

Contact: Robin Andersson 018-471 66 81

Grading:

- 3: ≥ 20p
- 4: ≥ 27p
- 5: ≥ 36p

Max: 45p

General instruction: keep it short!

Good luck!

1) (2+2p)

(a) (2p) Define the procedure my-reverse using the, in the course discussed, accumulate procedure given below:

```
(define (accumulate op init lst)
     (if (null? lst)
         init
         (op (car lst)
             (accumulate op init (cdr lst)))))
```

my-reverse takes a list as an argument and returns the list reversed as a result, e.g.:

```
(my-reverse (list 1 2 3 4 5))
;Value 1: (5 4 3 2 1)
(my-reverse (list (list 1 2) 3 (list 4 5))
;Value 2: ((4 5) 3 (1 2))
```

(b) (2p) Modify my-reverse (call the new procedure my-reverse2) so that it also reverses all sub-lists of the list (still using the accumulate procedure), e.g.:

```
(my-reverse2 (list (list 1 2) 3 (list 4 5)))
;Value 3: ((5 4) 3 (2 1))
(my-reverse2 (list (list 1 (list 2 4 5)) 3 4 5))
;Value 4: (5 4 3 ((5 4 2) 1))
(my-reverse2 (list 1 2 3 4 5))
;Value 5: (5 4 3 2 1)
```

2) (5p)

Define a matrix procedural object, dispatch driven, that takes three arguments: the number of rows, the number of columns, and an initial value for the fields in the matrix, upon creation. Implement methods for accessing fields, columns, and rows and a method for setting values (of fields). Add a print method to the matrix object. Do not use vectors in your solution!

Example:

```
> (define m (matrix 5 5 0))
> (m 'print)
(0 0 0 0 0)
(0 0 0 0 0)
(0 0 0 0 0)
(0 0 0 0 0)
(0 0 0 0 0)
> ((m 'set) 1 1 3)
> ((m 'set) 2 4 5)
> ((m 'set) 4 2 7)
> ((m 'set) 4 5 6)
> (m 'print)
(3 0 0 0 0)
(0 0 0 5 0)
(0 0 0 0 0)
(0 7 0 0 6)
(0 0 0 0 0)
> ((m 'get-row) 4)
(0 7 0 0 6)
> ((m 'get-col) 2)
(0 0 0 7 0)
```

3) (3+2p)

(a) (3p) Decide, using truth tables, whether each of the following logical implications in propositional logic is satisfiable, contingent, valid (tautology), or contradictory.

1. $\models (A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$
2. $A \rightarrow \neg B,\ B \vee C,\ C \rightarrow A \models A \vee C$
3. $(A \rightarrow B) \wedge (C \rightarrow D) \models (A \vee C) \rightarrow (B \vee D)$

(b) (2p) Provide a proof of validity in natural deduction for one of the logical implications you find valid (inference rules are listed on the last page).

4) (1.5+1.5p)

(a) (1.5p) State three issues one needs to consider when choosing a knowledge representation language for the specification of an ontology.
(b) (1.5p) Why are these issues important?

5) (3p)

Depth first search (DFS) is neither optimal nor complete. How can DFS be used to guarantee optimality and completeness? Briefly describe the strategy.

6) (3p)

A* search is one of the most popular search algorithms to date. However, when using it for real life problems one often encounter memory problems. How can this be dealt with?

7) (2+1+1p)

(a) (2p) Describe the Bayes' optimal classifier.
(b) (1p) Describe the problems of using this method for classification.
(c) (1p) What improvements can be made? Name another strategy for classification that deals with the problems.

8) (2p)

In multi-objective genetic algorithms the fitness of a solution is decided in terms of dominance. What is dominance in this respect? Give an example situation where dominance ranking could be used.

9) (2p)

When would the genetic algorithm approach benefit from using a generational approach? When is a steady state approach more beneficial?

10) (6p)

Consider the following decision table describing factors affecting sunburn (hair color, height, weight, and whether lotion is used or not).

Hair	Height	Weight	Lotion	Decision
Blonde	Average	Light	No	Sunburned
Blonde	Tall	Average	Yes	None
Brown	Short	Average	Yes	None
Blond	Short	Average	No	Sunburned
Red	Average	Heavy	No	Sunburned
Brown	Tall	Heavy	No	None
Brown	Average	Heavy	No	None
Blond	Short	Light	Yes	None

Build a decision tree from the decision table using the gain criterion for feature selection.

Hint: log2(a/b) = log2(a) – log2(b), log2(1) = 0, log2(2) = 1, log2(3) ≈ 1.58, log2(4) = 2, log2(5) ≈ 2.32, log2(6) ≈ 2.58, log2(7) ≈ 2.81, log2(8) = 3

11) (4+4p)

The 8-queens problem is a classical problem in AI. The goal is to place eight queens on an 8*8 chessboard (matrix) in a safe way, i.e., such that no queen attacks any other. A queen attacks any piece in the same row, column, or diagonal.

(a) (4p) Formulate the 8-queens problem as a search problem. Specify representation, operators, and a strategy for searching the state space. (max 0.5 page)

(b) (4p) Provide a strategy for solving the problem using a genetic algorithm. Specify representation, selection operator(s), mutation operator(s), crossover, and a strategy for generating new generations of solutions. (max 0.5 page)

Inference rules:

$$\frac{\alpha, \neg\alpha}{\bot}\ (\bot I)$$

$$\frac{\boxed{\begin{array}{c}\alpha\\ \bot\end{array}}}{\neg\alpha}\ (\neg I) \qquad \frac{\boxed{\begin{array}{c}\neg\alpha\\ \bot\end{array}}}{\alpha}\ (\neg E) \qquad \frac{\alpha \rightarrow \beta, \beta \rightarrow \alpha}{\alpha \leftrightarrow \beta}\ (\leftrightarrow I)$$

$$\frac{\alpha_1, \alpha_2, \ldots, \alpha_n}{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n}\ (\wedge I) \qquad \frac{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n}{\alpha_i}\ (\wedge E) \qquad \frac{\alpha \leftrightarrow \beta}{\alpha \rightarrow \beta}\ (\leftrightarrow E)$$

$$\frac{\alpha_i}{\alpha_1 \vee \alpha_2 \vee \ldots \vee \alpha_n}\ (\vee I) \qquad \frac{\alpha \vee \beta, \alpha \rightarrow \chi, \beta \rightarrow \chi}{\chi}\ (\vee E)$$

$$\frac{\boxed{\begin{array}{c}\alpha\\ \beta\end{array}}}{\alpha \rightarrow \beta}\ (\rightarrow I) \qquad \frac{\alpha \rightarrow \beta,\ \alpha}{\beta}\ (\rightarrow E)$$