CPE/CSC 481 - W06

Knowledge-Based Systems

Final Exam

Instructor: Franz J. Kurfess

This is the final exam for CPE/CSC 481-W06 *Knowledge-Based Systems*. It is a take-home exam, and you may use textbooks, course notes, or other material, but you must develop and formulate the text and code for your answers yourself. You are allowed to and should use a computer for the programming questions. Copy and paste the code to the exam or attach it on a separate sheet, and also submit an electronic version via `handin`. Please provide references if you use material other than the textbooks or course notes. You are not allowed to discuss the questions and answers with other students or anybody else.

If you need clarifications about questions, you can contact me via email at `fkurfess@calpoly.edu`, or see me during my office hours on Mon, 2:10 - 4:00 p.m.; Fri, 10:10a.m. - 1:00 p.m.. The deadline for the exam is Thursday, March 16, 2004, at 4:00 p.m.. You can submit a printed or an electronic copy of the exam. Leave the printed copy in the drop box in front of the CSC department office (room 14-254), or give it to me during my office hours. Submit the electronic copy and any code via handin to `grade480` on `hornet/falcon`.

Please note that I will be out of town on Tue, Wed, and possibly Thursday for a CSU meeting on e-Portfolios. I assume that I'll have email access, but I'm not sure how reliable it will be, and how often I'll have time to check my email.

The number of points is indicated on the margin, with subtotals for the different tasks.

Student Name: **Date:**

Signature:

Task I – Multiple Choice Questions
Mark the correct answers (only one per question).

1. What does it mean that a logical sentence is *valid*? 3

   ☐ the sentence is true under all possible interpretations in some possible worlds
   ☐ the sentence is true under all possible interpretations in all possible worlds
   ☐ the sentence is true if there exists a true interpretation in some possible world
   ☐ the sentence is syntactically correct

2. Which of the following statements is the best characterization of *backward reasoning*? 3

   ☐ Available evidence is combined step by step until an acceptable solution is found
   ☐ A possible solution is formulated as a hypothesis, and evidence for the hypothesis is systematically assembled.
   ☐ All types of reasoning where backtracking is not allowed.
   ☐ It is essentially the equivalent of breadth-first search.

3. What is refraction? 3

   ☐ recently used facts are not used for a certain time ("refractory period")
   ☐ used for recursive rules by allowing certain types of rules to fire repeatedly
   ☐ a reasoning method based on fractitioning, i.e. splitting the rule base repeatedly into smaller groups of rules
   ☐ rules fire only once for a specific set of facts

4. Which of the following statements is the best characterization of *joint probability* for two events A and B in probability theory? 3

   ☐ the probability that one of two independent events occurs
   ☐ the probability that both of two independent events occur
   ☐ for two *dependent* events A and B, the probability that event A occurs given that B has already occurred
   ☐ for two *independent* events A and B, the probability that event A occurs given that B has already occurred

5. What is the historical background for *Bayesian reasoning*? 3

   ☐ The initial development was an ad-hoc attempt at dealing with uncertainty in expert system.
   ☐ It is a relatively recent development that subsumes some of the other approaches to uncertainty as special cases.
   ☐ It was developed by Thomas Bayes, an eighteenth-century British clergyman.
   ☐ It was initially developed in the 1960s by Lotfi Zadeh as an attempt to formalize common-sense reasoning

6. For which of the following approaches to uncertainty and approximate reasoning does *inverse* (or a posteriori) probability play the most central role? 3

   ☐ certainty factors
   ☐ fuzzy logic
   ☐ Bayesian reasoning
   ☐ Dempster-Shafer theory

7. Which of the following terms are examples of *fuzzy qualifiers*? 3

   ☐ somewhat, fairly, quite, very, extremely
   ☐ most, many, usually, often, rarely
   ☐ tall, large, high,
   ☐ size, height, width, depth

8. What is the role of the *membership function* for fuzzy sets? 3

   ☐ it returns true or false to indicate if an element is a member of the set
   ☐ it indicates the probability that an element is a member of the set
   ☐ its value indicates how strongly a particular element is affiliated with the set
   ☐ it is one of several inference rules for fuzzy logic

Subtotal Task 1: 24

Task II – Short Questions

1. *Bayesian Reasoning* and *Dempster-Shafer Theory* are two approaches that can be used to deal with uncertainty in knowledge-based systems. What are the most important aspects of these two approaches? Describe the underlying principles and concepts, their advantages and drawbacks, and the main differences.

10

Aspect	*Bayesian Reasoning*	*Dempster-Shafer Theory*
Principles and Concepts	• • • • •	• • • • •
Differences	• • • • •	• • • • •
Advantages	• • • • •	• • • • •
Problems	• • • • •	• • • • •

2. The rule given below is an attempt at capturing the relationships between clothing items. It was written byt a summer intern, and is probably not formulated very efficiently. Your task is to translate the rule into something close to plain English, identify problematic aspects, and rewrite the rule so that it is more efficient. 10

```
;; This rule is an attempt at specifying relationships of clothing items.
;; It is a first attempt at restricting possible combinations of items.
;; There is probably room for improvement!

(deftemplate items
    (slot color)
    (slot pattern)
    (slot brightness))

(deffacts initial-state
    (items (color green) (pattern dots) (brightness 3))
    (items (color blue) (pattern stripes) (brightness 2))
    (items (color red) (brightness 1))
    (items (color green)(pattern dots) (brightness 3))
    (items (color red) (pattern stripes) (brightness 2))
    (items (color blue))
    (items (pattern stripes))
    (items (brightness 4))
    )


(defrule clothing-rule
  (items (color ?x1) (pattern ?y1) (brightness ?z1))
  (items (color ?x2) (pattern ?y2) (brightness ?z2))
  (items (color ?x3) (pattern ?y3) (brightness ?z3))
  (test (or (and  (eq ?x3 green)
                  (eq ?x2 red))
            (and  (neq ?x1 ?x2)
                  (neq ?x1 ?x3)
                  (neq ?x2 ?x3))
     (neq ?y1 ?y2 ?y3)
     (< ?z1 ?z2)
     (> ?z3 ?z2)))
  =>
   (printout t "Shirt:   "  ?x1 "  " ?y1 "  " ?z1 crlf)
   (printout t "Pants:   "  ?x2 "  " ?y2 "  " ?z2 crlf)
   (printout t "Jacket:  "  ?x3 "  " ?y3 "  " ?z3 crlf crlf)
)
```

Problematic aspects:

More efficient formulation:

3. While it is clear that the Semantic Web can benefit significantly from knowledge representation and reasoning methods developed for knowledge-based systems, that community has developed and adopted methods other than CLIPS or JESS, such as RDF, OWL, and description logics. In your opinion, what are the main reasons for this? 10

Subtotal Task 2: 30

Task III – CLIPS Program: Towers of Hanoi

The "Towers of Hanoi" program is a very popular exercise in Computer Science, especially for the exploration of recursion. In case you don't remember the details, here is a brief description (excerpted from http://www.cs.princeton.edu/introcs/23recursion, by Robert Sedgewick and Kevin Wayne):

> We have three pegs and N disks that fit onto the pegs. The disks differ in size and are initially arranged on one of the pegs, in order from largest (disk N) at the bottom to smallest (disk 1) at the top. The task is to move the stack of disks to the right one position (peg), while obeying the following rules: (i) only one disk may be shifted at a time; and (ii) no disk may be placed on top of a smaller one.
> One legend says that the world will end when a certain group of monks accomplishes this task in a temple with 64 golden disks on three diamond needles.

Your task here is to design, implement, and analyze a CLIPS (or JESS) program that solves the Towers of Hanoi problem. It requests the number of disks N as input, and prints out the sequence of moves to rearrange the disks. Label the discs in increasing order of size from 1 (smallest) to N, and the pegs as A, B, C. Here is an example output for 3 disks:

```
CLIPS> (run)
  How many disks in the tower? 3
  Move disc 1 from A to C
  Move disc 2 from A to B
  Move disc 1 from C to B
  Move disc 3 from A to C
  Move disc 1 from B to A
  Move disc 2 from B to C
  Move disc 1 from A to C
CLIPS>
```

The purpose of this exam question is to examine the capabilities and limitations of CLIPS and JESS, especially with respect to the formulation and execution of recursive programs. In order to do this, you are required to perform the following subtasks:

- Recursion in CLIPS or JESS: Identify and describe ways of implementing recursive programs in rule-based languages such as CLIPS and JESS.
- General solution strategy: Describe your general approach to solve the Towers of Hanoi problem. You can use English, or pseude-code to do this.
- Rule-based approach: Based on your strategy identified in the previous part, now describe how you translate it into rules.
- CLIPS (or JESS) program: List your program here, or append a printout. Make sure to add comments to your code, and add a `README` file with instructions for running your program.
- Performance and Program analysis: Examine your program with respect to its space and time requirements. Give an indication of the memory usage (e.g. in terms of number of facts asserted) and the computation time (e.g. in seconds, or in terms of number of rules fired, iterations through loops, recursive calls) for N ranging from 1 through at least 12. Identify advantages and problems of your solution, e.g. in comparison with an implementation in a conventional language, or a different strategy.

a) *Recursion in* CLIPS *or* JESS: 8

b) *General Strategy*: 4

c) *Rule-based approach*: 4

d) CLIPS *(or* JESS*) Code*, including instructions to run your program. 20

Instructions

Code

e) *Program analysis*: Memory space, computation time, advantages, problems. List the indicators for memory space consumption and computation time in the table below. Please note if you use any special settings of CLIPS or JESS (e.g. constraint checking, strategy).
It is not essential to obtain precise numbers, the changes as N increases are more important. For example, measuring the running time of your program in seconds with a clock is good enough. 10

N	*Space*	*Time*	*Comment*
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			

CLIPS or JESS settings

Space

Time

Advantages

Problems

Subtotal Task 3: 46

Total Points: 100