answers

January 30, 2003

CSC/CPE 481 **Midterm Exam** Winter 2001

Prof.: Franz J. Kurfess

Task I – Multiple Choice Questions

Mark the correct answers (only one per question).

a) One of the fundamental concepts in knowledge-based systems is

- □ the availability of common-sense knowledge in the shell
- □ the fact that they automatically recognize the limits of their knowledge
- □ the tight integration of knowledge and its usage
- □ the separation of knowledge and its usage 3

b) A characterization of expert systems derived from N. Wirth's *algorithms + data structures = program* is

- □ rules + facts = expert system
- □ knowledge + inference = expert system
- □ shell + knowledge = expert system
- □ expert knowledge + automated system = expert system 3

c) Why is an efficient method for pattern matching so important for the performance of rule-based systems?

- □ Each pattern matching operation itself can be very complex and time-consuming.
- □ Pattern matching is a very elementary and freqent operation, similar to variable assignment in procedural languages.
- □ The number of pattern matching operations can be extremely high, since all rules are compared against all the facts in every cycle.
- □ Conventional computers are not very well suited for pattern matching, and special hardware is required for an efficient implementation. 3

d) Which statement characterizes an unsound inference mechanism best?

- □ there may be a contradiction between the syntax and the semantics of logical statements
- □ it may generate incorrect conclusions
- □ it may not generate all possible solutions for a problem
- □ the addition of new knowledge to a knowledge base may lead to contradictions 3

e) What is an important obstacle for the use of network-based representation schemes (e.g. semantic networks) for reasoning?

- □ the syntax of the networks is not precisely defined
- □ since networks are graphs, cycles and infinite loops may occur
- □ it is difficult to formulate specific inference methods for the different types of links in a network
- □ the addition of new knowledge to a network may lead to contradictions 3

f) The programming language Prolog is based on which of the following concepts?

- □ Horn clauses, resolution, and unification
- □ pattern matching, modus ponens, and rules
- □ frames, slots and fillers
- □ nodes, links, and spreading activation 3

g) What is the A* search algorithm?

- □ a search algorithm whose name is inspired by the way astronomers search for new stars in the universe
- □ a combination of depth- and breadth-first search algorithms that combines the advantages of both while overcoming their most severe limitations
- □ a search algorithm that explores the search space by constructing a search graph instead of a search tree
- □ a combination of greedy and lowest path-cost search algorithms that is guaranteed to find the optimal solution under certain conditions 3

h) The reasoning method used by Clips is based on

- □ resolution and unification
- □ forward-chaining and pattern matching
- □ higher-order logic and metaknowledge
- □ Boolean Algebra for rules 3

i) What is the purpose of the *agenda* in Clips?

- □ contains all currently available facts
- □ restricts the facts that can be utilized at a certain point
- □ contains all activated rules
- □ contains a trace of fired rules 3

j) The implementation of recursive and iterative evaluation schemes in Clips is frequently achieved through the use of which constructs?

- □ `while` and `for` loops
- □ recursive functions based on the Lambda calculus
- □ `assert` and `retract`
- □ invocation of external functions, e.g. in C/C++ 3

30

Task II – Short Questions

a) Explain the difference between search in knowledge-based systems, and search in data structures. 6

- knowledge-based systems:

- data structures:

b) What are the main differences between the *software lifecycle* and the *knowledge-based system lifecycle*? 6

- software lifecycle

- knowledge-based system lifecycle

c) Why are *incremental development* and *rapid prototyping* so popular for the development of knowledge-based systems? 6

d) The CLIPS expert system shell uses `assert` and `retract` for the modification of its knowledge at runtime. 6

- From a logical point of view, these operations are not without problems; what is the problem they may cause?

- Why are they useful in practice?

e) Why are variables and functions so important for the representation of knowledge, both for predicate logic and for rule-based systems? 6

Task III – Problem

In this task, you need to trace the evaluation of a short CLIPS program. This program is a variation of the "blocks world" program discussed in class. Instead of putting blocks on the floor to get them out of the way, the idea behind this program is to move those blocks onto a third stack.

a) Please fill out the form on the next page, based on the program printed below. 30

```
;;;   Blocks World Program 3
;;;  (Modified version, no blocks on the floor)

(deftemplate goal (slot move) (slot on-top-of))

(deffacts initial-state
   (stack A B C D)
   (stack E F G H)
   (stack I J K L)
   (goal (move K) (on-top-of C))
)

(defrule move-directly
   ?goal <- (goal (move ?block1) (on-top-of ?block2))
   ?stack-1 <- (stack ?block1 $?rest1)
   ?stack-2 <- (stack ?block2 $?rest2)
   =>
   (retract ?goal ?stack-1 ?stack-2)
   (assert (stack $?rest1))
   (assert (stack ?block1 ?block2 $?rest2))
   (printout t ?block1 " moved on top of " ?block2 "." crlf))

(defrule move-to-third-stack
   ?goal <- (goal (move ?block1) (on-top-of stack))
   ?stack-1 <- (stack ?block1 $?rest1)
   ?stack-2 <- (stack $? ?block2 $?)
   ?stack-3 <- (stack ?block3&~?block2&~?block1 $?rest3)
   =>
   (retract ?goal ?stack-1)
   (assert (stack ?block1 ?block3 $?rest3))
   (assert (stack $?rest1))
   (printout t ?block1 " moved on top of " ?block3 crlf))

(defrule clear-upper-block
   (goal (move ?block1))
   (stack ?top $? ?block1 $?)
   =>
   (assert (goal (move ?top) (on-top-of stack))))

(defrule clear-lower-block
   (goal (on-top-of ?block1))
   (stack ?top $? ?block1 $?)
   =>
   (assert (goal (move ?top) (on-top-of stack))))
```

b) Unfortunately, the program does not always work as intended. Changing the sequence in which the three stacks are initially defined, e.g. to

```
(deffacts initial-state
   (stack E F G H)
   (stack A B C D)
   (stack I J K L)
   (goal (move K) (on-top-of C))
)
```

leads to an endless loop. Can you determine a possible cause for this? How can it be remedied? 10

	40
Total Points:	100