THE UNIVERSITY OF AUCKLAND

SECOND SEMESTER, 2007
Campus: City

COMPSCI.367
The Practice of Artificial Intelligence

(Time allowed: 45 minutes)

This test is out of **90** marks.
Attempt **ALL** questions.
Write your answers in the space provided in this booklet. There is space at the back for answers that overflow the allotted space.
The use of calculators is **NOT** permitted.

Surname (Family Name):	
First Name(s):	
UoA ID Number:	
Login Name (UPI):	

Section	Mark	Marks Available
A		**15**
B		**24**
C		**16**
D		**35**
Total		**90**

Print Name:

PART A: Symbolic Reasoning

Question 1

Translate the following into English. [6 marks].

1. $\neg$ A	NOT A
2. A $\vee$ B	A OR B
3. A $\Rightarrow$ B	A IMPLIES B
4. $\forall$ B	FOR ALL B
5. A $\wedge$ B	A AND B
6. $\exists$ A	THERE EXIST AN A

Question 2

What implication can be drawn from the following logical statement? [4 marks]

$(\forall X(human(X) \Rightarrow mortal(X)) \wedge human(socrates)) \Rightarrow mortal(socrates)$

Question 3

Given the following Prolog code, write a predicate for grandmother (note: your grandmother is your mother's mother). [5 marks]

```
mother(M, C) :- parent(M, C), female(M).

grandmother(G,C) :- mother(G,M), mother (M,C).
```

Continued…

Print Name:

_____

PART B: Knowledge Engineering

Question 4

What is the *knowledge elicitation bottleneck*? [4 marks]

The knowledge elicitation bottleneck refers to the observation that the task of eliciting or obtaining knowledge from domain experts is the most difficult and time consuming task in developing a knowledge based system. Once the knowledge has been elicited, modeling and coding are relatively easy tasks and hence knowledge elicitation is a bottleneck in the development process.

Question 5

What is *heuristic* knowledge? [2 marks]

Heuristic knowledge is knowledge that is usually correct or true though there may be exceptions. Heuristic knowledge is like a "rule of thumb" or general guide such as "if it Auckland and it's winter, then it's raining."

Question 6

Not all knowledge is suitable for inclusion in a knowledge based system. List four types of knowledge that are **not** suitable. [4 marks]

1. Aesthetic knowledge
2. Sensory knowledge
3. Subjective knowledge
4. Worldly knowledge

Also allow unstable or dynamic knowledge

Continued…

Print Name:

Question 7

Forward Chaining and *Backward Chaining* are two inferencing methods for rules, each is commonly used for different problem types. Describe a problem you would use each for. [4 marks]

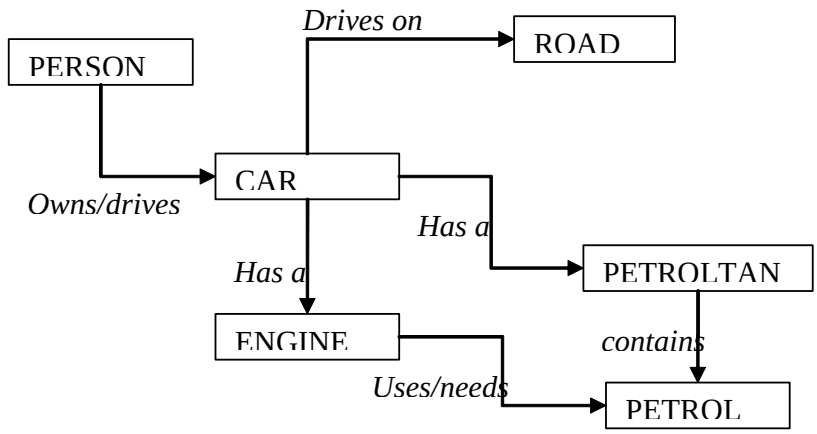
1. Forward Chaining is used for…process control problems like running a power plant or flying a plane

2. Backward Chaining is used for…diagnosis or fault finding or product selection

Question 8

Create a semantic network to describe a car. Your network should include the concepts: *car*, *driver*, *engine*, *petrol*, *petrol tank*, and *road.* [10 marks]

There is no single correct semantic network: but it should like similar to:



Print Name:

PART C: CLIPS

Question 9

Defining a tree data structure in CLIPS.

A *tree* fact will store the name of the root node of the tree. A *node* fact in the tree has a name and a list of that node's children (if any).

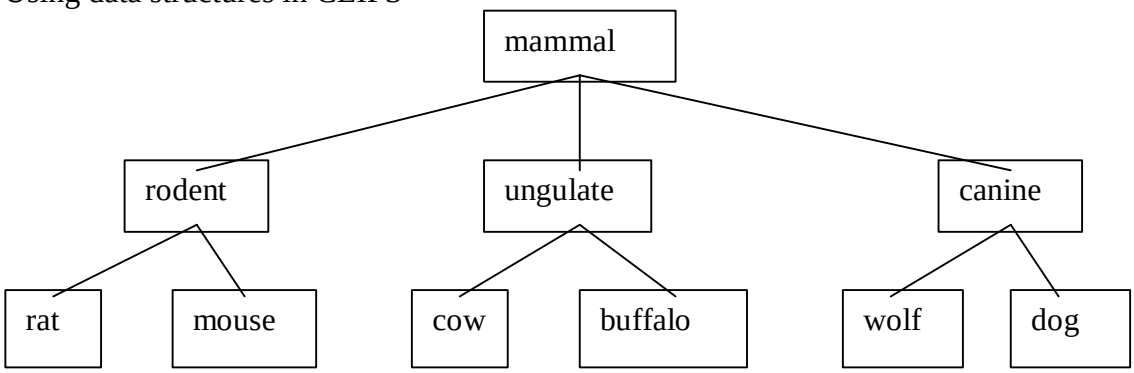
Write CLIPS templates for trees and nodes. [5 marks]

Should look similar to:

```
(deftemplate node (slot name) (multislot children))
    (deftemplate tree (slot root))
```

Question 10

Using data structures in CLIPS



Given the tree above, write the facts, using the templates you developed in (9) above, that encode this tree's information. [6 marks]

Depending on how they answered 9 it should look something like:

```
(deffacts initialTree
   (node (name mammal) (children rodent ungulate canine))
   (node (name rodent) (children rat mouse))
   (node (name rat))
   (node (name mouse))
   (node (name ungulate) (children cow buffalo))
   (node (name canine) (children wolf dog))
   (node (name cow))
   (node (name buffalo))
   (node (name wolf))
   (node (name dog))
   (tree (root mammal))
```

Continued…

Print Name:

Question 11

What happens if you define two rules in CLIPS both with the same name? [5 marks]

The second rule will overwrite the first. Rules must have unique names.

PART D: Search

Important Notes

For all questions assume the following:

1. We assume that all search spaces are finite (i.e., there are only finitely many states), remember this does not mean that the number of nodes expanded by an arbitrary search algorithm need to be finite!
2. A search algorithm's performance profile includes the following: completeness, optimality, time complexity, space complexity and optimal efficiency.
3. Assume that all heuristics are admissable.

Question 12

What does it mean for a heuristic to be *admissable*? [5 marks]

It means that the heuristic never overestimates the cost of the optimal path from that node to its cheapest goal.

Continued…

Print Name:

Question 13

What piece of information is stored in each search node that allows the search algorithm to return the solution path discovered? [5 marks]

A pointer to the predecessor of the node.

Question 14

Given two heuristics, *h1* and *h2*, under what conditions would A* using *h1* be guaranteed **not** to expand more nodes than using *h2*? [5 marks]

When *h1* is "more informed" than *h2*, i.e., for all non-goals *n*, $h1(n) > h2(n)$

Question 15

What *informed* search algorithm is not complete and why? [5 marks]

Greedy search is not complete in search spaces containing loops where the heuristic values of the nodes in the loop are lower than the heuristic values of any of the other nodes in the open list (this is assuming that no loop checking/duplicate state checking is being done).

Continued…

Print Name:

Question 16

What is the difference between *breadth-first* search and *uniform-cost* search and what difference does it make to their performance profile? [5 marks]

Breadth-first search uses the length of the search path from the start node to the given node to select which node to expand next, while uniform cost uses the sum of the node's path edge costs. This means that uniform-cost is guaranteed to find an optimal solution even when the edge costs are not uniform.

Question 17

What is the difference between *greedy search* (aka best-first search) and *A* search* and what difference does it make to their performance profile? [5 marks]

Greedy searches by h value and A* searches by $g + h$ values. This means that A* is complete, optimal and optimally efficient.

Question 18

What happens to *depth-first* search's performance profile (e.g., completeness, optimality, time complexity, space complexity, etc.) when we go from not checking for duplicate states to checking for duplicate states? [5 marks]

The space complexity goes from linear to exponential and the algorithm goes from not being complete to being complete.

Continued…

Print Name: _____

This page left blank for any questions that overflow.

Print Name: _____

This page left blank for any questions that overflow.