THE UNIVERSITY OF AUCKLAND

FIRST SEMESTER, 2006
Campus: City

COMPSCI.366

The Foundations of Artificial Intelligence

(Time allowed: 45 minutes)

This test is out of **100** 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):	

Question	*Mark*	**Marks Available**
1		**8**
2		**12**
3		**9**
4		**6**
5		**8**
6		**2**
7		**5**
8		**8**
9		**6**
10		**4**
11		**10**
12		**4**
13		**18**
Total		**100**

Question 1

For each of the following English sentences, choose the first-order predicate calculus formula that best describes the sentence.
[8 marks]

There is a solution for every problem.

1. $\exists x \exists y$: $Problem(x) \wedge Solution(x,y)$
2. $\forall x \exists y$: $Problem(x) \wedge Solution(x,y)$
3. $\exists y \forall x$: $Problem(x) \wedge Solution(x,y)$
4. $\forall y \exists x$: $Problem(x) \wedge Solution(x,y)$
5. $\exists x \forall y$: $Problem(x) \wedge Solution(x,y)$
6. $\forall x \forall y$: $Problem(x) \wedge Solution(x,y)$

2. $\forall x \exists y$: $Problem(x) \wedge Solution(x,y)$

Only students who have answered all questions should leave the room.

1. $\forall x$: $Student(x) \wedge Questions\_answered(x) \rightarrow Leave\_room(x)$
2. $\forall x$: $Questions\_answered(x) \wedge Leave\_room(x) \rightarrow Student(x)$
3. $\forall x$: $Student(x) \wedge Leave\_room(x) \rightarrow Questions\_answered(x)$
4. $\forall x$: $Leave\_room(x) \rightarrow Student(x) \wedge Questions\_answered(x)$

3. $Student(x) \wedge Leave\_room(x) \rightarrow Questions\_answered(x)$

Students like either coffee or tea.

1. $\forall x$: $Student(x) \rightarrow Likes\_coffee(x) \vee Likes\_tea(x)$
2. $\forall x$: $Likes\_coffee(x) \vee Likes\_tea(x) \rightarrow Student(x)$
3. $\forall x$: $[Student(x) \rightarrow Likes\_coffee(x)] \vee [Student(x) \rightarrow Likes\_tea(x)]$
4. $\forall x$: $Student(x) \rightarrow [Likes\_coffee(x) \wedge \neg Likes\_tea(x)] \vee [\neg Likes\_coffee(x) \wedge Likes\_tea(x)]$

4. $\forall x$: $Student(x) \rightarrow [Likes\_coffee(x) \wedge \neg Likes\_tea(x)] \vee [\neg Likes\_coffee(x) \wedge Likes\_tea(x)]$

Question 2

Unify the following sets of literals or indicate if this is not possible. *P* and *Q* are predicates, *f* and *g* are functions, *a* and *b* are constants, and *x* and *y* are variables.[1]
[12 marks]

$\{P(x), Q(a)\}$

{FAIL}

$\{P(x,y), P(a,x)\}$

$\{a/x, a/y\}$

$\{Q(a,y), Q(x,f(y))\}$

{FAIL}

$\{P(f(g(y))), P(x)\}$

$f(g(y))/x$

{Q(a), Q(g(y))}

{FAIL}

{P(b,y), P(x,g(x))}

$\{b/x, g(b)/y\}$

[1] There was an error in the test, which states that *x* and *z* are variables, rather than *x* and *y*.

Question 3

Convert the following formulas into clause form.
[9 marks]

$\forall x \forall y \forall z$: $[P(x) \wedge Q(y)] \vee R(z)$

$P(x) \vee R(z_1)$, $Q(y) \vee R(z_2)$

$\forall x \exists y \forall z$: $P(x) \wedge Q(y) \rightarrow R(z)$

$\neg P(x) \vee \neg Q(f(x)) \vee R(z)$

$\exists x$: $P(x) \wedge [\exists y: Q(y)] \rightarrow [\exists y: R(y)]$

$\neg P(a) \vee \neg Q(y) \vee R(f(y))$

Question 4

Given the following set of propositional formulas, prove P by resolution.[2]
[6 marks]

$$Q$$
$$\neg T$$
$$P \lor R$$
$$\neg Q \lor S$$
$$\neg R \lor \neg S \lor T$$

$\neg P$ $\quad$ $P \lor R$
\ /
R $\quad$ $\neg R \lor \neg S \lor T$
\ /
$\neg S \lor T$ $\quad$ $\neg T$
\ /
$\neg S$ $\quad$ $\neg Q \lor S$
\ /
$\neg Q$ $\quad$ Q
\ /
$\square$

[2] There was an error in the test, which had T rather than $\neg T$ in the set of clauses.

Question 5

Given the standard min/max operations for fuzzy logic, compute the following fuzzy sets.
[8 marks]

$\tilde{A} = \{(a, 0.5), (b, 0.7), (c, 0.4)\}$

> Complement of $\tilde{A}$:
>
> $\tilde{A}^c = \{(a, 0.5), (b, 0.3), (c, 0.6)\}$

$\tilde{A}_1 = \{(a, 0.4), (b, 0.6), (c, 0.3)\}$ $\tilde{A}_2 = \{(a, 0.8), (b, 0.2), (c, 0.5)\}$

> Intersection of $\tilde{A}_1$ and $\tilde{A}_2$:
>
> $\tilde{A}_1 \cap \tilde{A}_2 = \{(a, 0.4), (b, 0.2), (c, 0.3)\}$

$\tilde{A}_1 = \{(a, 0.1), (b, 0.9), (c, 0.7)\}$ $\tilde{A}_2 = \{(a, 0.4), (b, 0.5), (c, 0.8)\}$

> Union of $\tilde{A}_1$ and $\tilde{A}_2$:
>
> $\tilde{A}_1 \cup \tilde{A}_2 = \{(a, 0.4), (b, 0.9), (c, 0.8)\}$

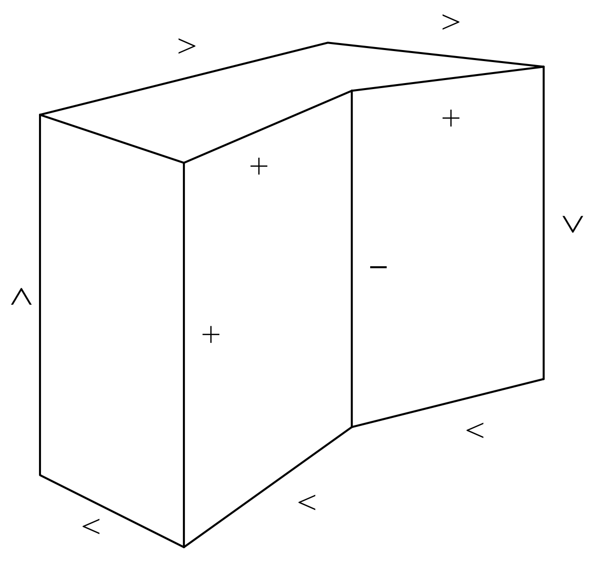
Question 6

Compute the 0.5-level set of the fuzzy set $\tilde{A} = \{(a, 0.3), (b, 0.7), (c, 0.4), (d, 0.8)\}$.
[2 marks]

$A_{0.5} = \{b, d\}$

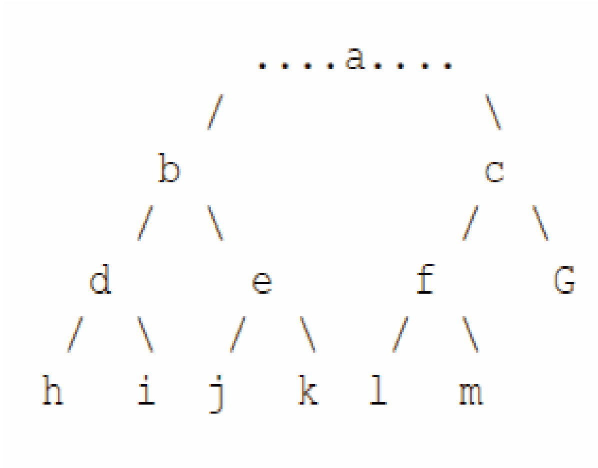
Question 7

Show one consistent labelling for the following polyhedron drawing (as it could have resulted from Waltz filtering). Use the label + for convex lines, – for concave line, and < for boundary lines.
[5 marks]



Question 8

In the search tree below, the G node is a goal node, the rest of the nodes are not goal nodes.
(1) How many nodes would be created by the breadth-first search algorithm?
(2) How many nodes would be created by the iterative-deepening search algorithm?
(3) List the nodes created by the breadth-first algorithm in their order of creation.
(4) List the nodes created by the iterative-deepening algorithm in their order of creation.
[8 marks]



1. 13 nodes (or 7 if search stops at G)

2. 11 nodes.

3. a b c d e f G (h i j k l m)

4. a | a b c | a b d e c f G.

Question 9

1) What is the difference between a *genetic algorithm* and a *random search algorithm*?
[2 marks]

A GA is only partly random, mutation is usually low <5% and selection of the fittest has a random component. A random search algorithm is 100% random.

2) What is the difference between a *genetic algorithm* and a *greedy hill-climbing algorithm*?
[2 marks]

A hill climbing algorithm can get trapped in local maxima, whereas random mutations can allow a GA to escape being trapped.

3) Genetic algorithms can be seen as a combination of *local* and *global* search. If so, which of cross-over and mutations provides the local search and which the global one?
[2 marks]

Cross over = local search

Mutation = global search

Question 10

What is the common characteristic of all *stochastic search* algorithms? Describe in a short sentence.
[4 marks]

They all use some (usually a small %) random search element.

Question 11

1) List 2 behaviours that multi-agent systems should exhibit?
[2 marks]

Any of:
cooperation, coordination, communication, negotiation, independence, autonomous action……etc

2) For the following games decide if they are mostly *deterministic* or *non-deterministic*.
[4 marks]

Checkers………deterministic

Rugby…………non- deterministic

Chess…………. Deterministic

Quake…………non-deterministic (definitely in multi-player and probably vs. computer as well)

3) For the following games decided if the are mostly *discrete* or *continuous*.
[4 marks]

Checkers………discrete

Rugby…………continuous

Chess…………discrete

Quake…………continuous

Question 12

Briefly describe the difference between a *first-order intentional system* and a *second-order intentional system*.
[2 marks]

A 1st order system has beliefs and intentions a 2nd order system has beliefs about beliefs and intentiions

2) Why is it useful to describe multi-agent systems as having intentional notions?
[2 marks]

Intentional notions provide abstraction – we can reason/program at a more abstract level.

Question 13

Define the *facts* for the following STRIPS *actions.*

1) stack(x,y)
[3 marks]

name *Stack(x, y)*

pre ***Clear(y)*** $\wedge$ ***Holding(x)***
del ***Clear(y)*** $\wedge$ ***Holding(x)***
add ***ArmEmpty*** $\wedge$ ***On(x, y)***

2) unstack(x,y)
[3 marks]

name *UnStack(x, y)*

pre $On(x, y) \wedge Clear(x) \wedge ArmEmpty$
del $On(x, y) \wedge ArmEmpty$
add $Holding(x) \wedge Clear(y)$

3) pickup(x)
[3 marks]

name *PickUp(x)*

pre $Clear(x) \wedge OnTable(x) \wedge ArmEmpty$
del $OnTable(x) \wedge ArmEmpty$
add $Holding(x)$

4) putdown(x)
[3 marks]

name *PutDown(x)*

pre $Holding(x)$
del $Holding(x)$
add $Clear(x) \wedge OnTable(x) \wedge ArmEmpty$

5) If the current world state can be described by the following STRIPS *facts*.

clear(A)
on(A,B)
onTable(B)
onTable(C)
armEmpty

Describe the world state after the following sequence of STRIPS *actions*

unStack(A,B)
Stack(B,C)
pickUp(B)
Stack(A,C)

[6 marks]

clear(A)
on(A,C)
onTable(B)
onTable(C)
armEmpty

This assumes that in order to complete Stack(A,C) B must be put down, STRIPS will do this if the precondition of an action is not met it will see if any action will enable the preconditions to be met.