UNIVERSITY OF DUBLIN
TRINITY COLLEGE

Faculty of Engineering, Mathematics and Science

School of Computer Science & Statistics

Integrated Computer Science Programme **Trinity Term 2014**
B.A. (Mod.) CSLL
B.A. (Mod.) Business & Computing

Artificial Intelligence I

Wed 21/05/2014 **LUCE UPPER** **9:30-11:30**

Dr Tim Fernando

Instructions to Candidates:

Attempt **two** questions (out of the three given).
All questions carry equal marks. 50 marks per question.

You may not start this examination until you are instructed to do so by the Invigilator.

Materials permitted for this examination:

Non-programmable calculators are permitted for this examination — please indicate the make and model of your calculator on each answer book used.

1. (a) Consider the basic AI problem of

   (†) an agent acting intelligently in its environment.

   What is the *Symbol-system Hypothesis*, and how is it relevant to the problem (†)?

   [10 marks]

   (b) What is the *Church-Turing thesis*, and what does it add to the Symbol-system hypothesis in addressing (†)?

   [10 marks]

   (c) What do non-determinism and search have to do with (†)?

   [10 marks]

   (d) Define a search predicate `search(L)` that searches every finite string L over the alphabet $\{0, 1\}$ to see if it satisfies the goal predicate `goal(L)` that consults the user through the built-in predicate `read/1`.

   ```
   goal(L) :- write(L), write('?'), nl, read(yes).
   ```

   [15 marks]

   (e) What does Cantor's theorem (about the set of all sets of non-negative integers) say about the problem of searching every infinite string over the alphabet $\{0, 1\}$?

   [5 marks]

2. Recall that a finite list of propositional definite clauses (i.e. definite clauses in which all predicates have arity 0) can be encoded as a list of lists — e.g.

```
h :- c.
h :- f,g.
f :- g.
g.
```

as `[[h,c], [[h,f,g], [f,g], [g]]`. Now, consider the binary Prolog predicate `prove(Atom,KB)` defined below.

```
prove(Atom,KB) :- member([Atom|B],KB), proveAll(B,KB).
proveAll([],_).
proveAll([Head|Tail],KB) :- prove(Head,KB), proveAll(Tail,KB).
```

(a) To show that the predicate `prove` above does a depth-first search, define a 3-ary predicates `arc(Node1,Node2,KB)` such that `prove(Atom,KB)` is equivalent to the predicate `search([Atom],KB)` given by

```
search([],KB).
search(Node,KB) :- arc(Node,Next,KB),
                   search(Next,KB).
```

[5 marks]

(b) Assuming each arc in the graph defined in (a) has cost 1, describe a heuristic estimate of the distance of a node to a goal node that is an underestimate.

[5 marks]

(c) Oultine how to modify the predicate `search(Node,KB)` in part (a) above to implement A-star search.

[15 marks]

(d) Explain what it means for A-star to be admissible, and give 3 conditions that are together sufficient for admissibility.

[10 marks]

(e) Recall that a Constraint Satisfaction Problem (CSP) is a triple (*Var*, *Dom*, *Con*) consisting of

- a finite set *Var* of *variables*
- a function *Dom* assigning each variable $X \in Var$ a finite set $Dom(X)$, called the *domain* of X

- a finite set *Con* of *constraints* that may or may not be satisfied by instantiating variables $X \in \mathit{Var}$ with values in $\mathit{Dom}(X)$.

Can we formulate the problem of proving an atom `Atom` from a finite list `KB` of propositional definite clauses in terms of a CSP? Explain.

[15 marks]

3. (a) What is an *interpretation*, and how is it used to define the notion of *logical consequence*?

[10 marks]

(b) Let the knowledge base KB consist of the two clauses `dog(fido)` and

```
dog(X) :- barks(X).
```

For each the sentences (i) – (iii) below, say whether it is a logical consequence of KB. Be sure to justify your answer.

(i) `barks(fido).`

(ii) `dog(fido) :- smells(fido).`

(iii) `smells(fido) :- smells(fido).`

[15 marks]

(c) What is an *intended interpretation* and how is it used to guide the construction of a knowledge base, KB? Give two ways an atom may be true in the intended interpretation, but fail to be a logical consequence of KB.

[10 marks]

(d) What does it mean for a proof system to be *sound*? What does it mean for it to be *complete*? Give an example of a proof system that is sound but not necessarily complete. Give an example of a proof system that is complete but not necessarily sound.

[10 marks]

(e) What does it mean for a proof system to be *non-monotonic*, and what does the *Complete Knowledge Assumption* (CKA) have to do with it? Be sure to say what CKA is.

[5 marks]