UNIVERSITY OF DUBLIN

TRINITY COLLEGE

Faculty of Engineering, Mathematics & Science

School of Computer Science & Statistics

B.A. (Mod.) Computer Science — Trinity Term 2012

Junior Sophister Exam

CS3061: Artificial Intelligence I

Wed, 09-05-2012 — GOLDHALL — 9:30 – 11:30

Dr Tim Fernando

Instructions to Candidates:

Answer two questions.

All questions carry equal marks.

Materials permitted for this examination:

Non-programmable calculators are allowed – please indicate the make and model of your calculator on each answer book used.

1. Recall that the *Halting Problem* (HP) has the following input/output specification. Given a program P and data D, return either 0 or 1 (as output), with 1 indicating that P halts on input D

$$\text{HP}(P, D) \;:=\; \begin{cases} 1 & \text{if } P \text{ halts on } D \\ 0 & \text{otherwise} \end{cases}$$

Answer the following concisely.

(a) What does HP have to do with Artificial Intelligence — in particular, the *Church-Turing Thesis*?

(10 marks).

(b) Outline an algorithm that outputs 1 precisely on those inputs (P, D) which require the output 1 according to the specification above for HP.

(10 marks).

(c) Does your algorithm in part (b) solve HP? If not, does it show it is unsolvable?

(10 marks).

(d) What does HP have to do with deriving an answer in Prolog?

(10 marks).

(e) Suppose we were to require that the programs P in HP were finite automata, and the data D were finite strings. Is the resulting problem HP still interesting? Why or why not?

(10 marks).

2. Recall our generic frontier search procedure that takes a list of candidate nodes as an argument, and assumes a unary predicate `isGoal/1` for goal nodes, and a binary predicate `arc/2` for arcs

```
search([Node|FrontierRest]) :-  isGoal(Node).
search([Node|FrontierRest]) :-
      findall(X,arc(Node,X),Children),
      addToFrontier(Children,FrontierRest,FrontierNew),
      search(FrontierNew).
```

(a) How can we get depth-first search from the code above? Show how we can avoid the use of a list as an argument to `search` (in the case of depth-first). So why have the frontier argument in the first place?

(10 marks).

(b) What other predicates (in addition to `isGoal` and `arc`) do we need for A* search?

(10 marks).

(c) What does it mean for A* to be *admissible*. List the three conditions sufficient to guarantee that A* is admissible.

(10 marks).

(d) Show that each of the three conditions in (c) is not necessary for A* to be admissible. What is the difference between a top-down and a bottom-up approach to logical consequence? Which approach is implemented in part (a) above?

(10 marks).

(e) What is a *Constraint Satisfaction Problem*, and outline a search strategy that might be suitable for it. Is A* suitable?

(10 marks).

3. (a) Consider the following knowledge base over the assumables `w,x,y,z`.

```
false :- a.
false :- b.
a :- b,c.
a :- d.
b :- w.
d :- y,z.
```

What are the *conflicts* and which are/is minimal?

(10 marks).

(b) What is the *complete knowledge assumption* and can we apply it to the knowledge base in part (a)? If so, what do we get?

(10 marks).

(c) What is the difference between "don't care" and "don't know" non-determinism. Give an example of each.

(10 marks).

(d) Translate the following knowledge base

```
p(a,b,c,d).
q(c,a,f(b)).
```

to

(i) a set of Object-Attribute-Value triples with predicate symbol $prop$

(7 marks).

(ii) a semantic network

(7 marks).

(iii) frames

(6 marks).