IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2014-15

EEE/EIE PART III/IV: MEng, BEng and ACGI

ARTIFICIAL INTELLIGENCE

Thursday, 11 December 9:00 am

Time allowed: 3:00 hours

Corrected copy

Corrected Copy

There are FIVE questions on this paper.

Answer FOUR questions.

All questions carry equal marks

Any special instructions for invigilators and information for candidates are on page 1.

Examiners responsible First Marker(s) : J.V. Pitt

Second Marker(s) : T-K. Kim

The Questions

1 The Wishful Thinking Power Company supplies its subscribers from a central generating station. Electricity is supplied from the central station to the various subscribers through a network of transformers, for example as shown Figure 1.1. Note that nodes T1-T6 are the transformers and nodes S1-S10 are the subscribers.

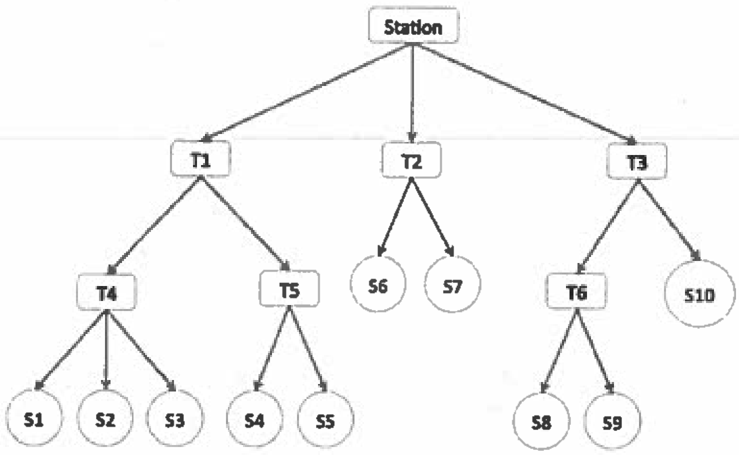

Figure 1.1: Electricity supply through a network of transformers

a) Specify a Prolog data structure to represent a network of this kind.

Illustrate your answer by encoding the network of Figure 1.1 in this data structure.

[6]

b) Write a Prolog program for `supplies/3`, where `supplies(X, Y, Grid)` is true, if node `X` supplies node `Y` (directly or indirectly) in the network `Grid`.

[10]

c) Sometimes a transformer may malfunction or need to be taken out of operation for servicing. In these cases, all the nodes and subscribers currently supplied by that transformer are to be re-connected to the parent transformer. Write a Prolog program for `reconnect/3`, where `reconnect(X, Grid, NewGrid)` is true if all the nodes supplied by `X` in `Grid` are now supplied by `X`'s parent transformer in `NewGrid`.

[4]

Note that the examiners are not necessarily expecting syntactically perfect, bug-free, executable Prolog code. However, in the context of the question, evidence of correctly specifying and manipulating Prolog data structures is required for full marks. It is also advisable to annotate your answers with declarative comments to indicate what relation the clauses are intended to compute.

2 a) Describe the behaviour of the Depth First, Breadth First, and Iterative Deepening Depth First search algorithms, and compare and contrast their performance with respect to appropriate criteria.

[6]

b) The general form of the pebble motion problem can be specified as follows: Let $G = \langle N, E, R \rangle$ be a graph. Let $P = \{1, \ldots, k\}$ be a set of k pebbles with $k < |N|$. An *arrangement* of pebbles is a mapping $S : P \rightarrow N$ such that $S(i) \neq S(j)$ for $i \neq j$. A move $m = (p, u, v)$ consists of moving the pth pebble currently occupying node u to an unoccupied node v.

The Pebble Motion on Graphs problem is to decide, given an arrangement S_0 and an arrangement S_+, if there is a valid sequence of moves that transforms S_0 into S_+.

(i) Specify a relation between a node, and the nodes to which it is connected, i.e. for each node n, the set $\{n' \mid (n, e, n') \in R\}$.

(ii) Define a Prolog data structure for representing states (i.e. arrangements), and illustrate this with an example.

(iii) Specify, in Prolog, the state transformer for the move operation.

(iv) Given that there is only one state transformer, briefly explain how your state transformer generates all the possible state transformations using findall in the predicate one_step_extensions of the General Graph Search (GGS) engine.

[10]

c) In a variation of the problem, nodes are allowed to have multiple pebbles, up to a certain limit for each node. An arrangement S is now a mapping from nodes to integers.

Specify this data structure in Prolog and define the state transformer for the move operation in this variation.

[4]

3 a) Describe the behaviour of the A* search algorithm, and specify its performance with respect to appropriate criteria.

[3]

b) In the context of A* search, define what is meant by the terms *heuristic*, and *admissible heuristic*. Explain why admissibility is important in A* search.

[3]

c) Explain why A* search is complete, and prove that it is optimal.

[5]

d) In the context of A* search, explain why, for a given search problem, one heuristic might be 'better' (more efficient) than another. Even if it were 'better', suggest why it might still be outperformed by the other heuristic.

[5]

e) Explain the differences between the minimax algorithm and the alpha-beta algorithm.

In the context of a 2-player game between two programs, which would win: the program using the minimax algorithm to select its move, or the program using the alpha-beta algorithm to select its move? Justify your answer.

[4]

4 a) In the context of logic and automated reasoning, explain what is meant by the terms *resolution* and *unification.*

[3]

b) In the context of translating English into clausal form, explain what is meant by *skolemisation.*

[3]

c) Express these six statements as formulas of First Order Predicate Logic, and translate them into clausal form:

For any coyote and any roadrunner, the coyote chases the roadrunner.

Any roadrunner that says beep-beep is smart.

For anything smart and any coyote, the smart thing avoids the coyote.

If one thing chases another thing, and that other thing avoids the first thing, then the first thing is frustrated.

Wile E. is a coyote.

Speedipus Rex is a roadrunner.

[4]

d) Prove, using resolution and showing the unifiers, that:

If Speedipus Rex says beep-beep, then Wile E. is frustrated.

[6]

e) Explain the relationship between clausal form and Prolog goals, clauses and queries. What algorithm does the Prolog inference engine use to search for an answer to a query? Briefly comment on this choice.

[4]

5 In answering this question, it is essential that the KE-tree (proof) is properly annotated to show the inference steps (i.e. labelled with the inference rule, and the major and minor premises).

a) Using the KE calculus, prove that:

$$((p \vee q) \rightarrow (p \vee r)) \rightarrow (p \vee (q \rightarrow r))$$

[4]

b) Using the KE calculus, prove that:

$$\{q \rightarrow r, r \rightarrow (p \wedge q), p \rightarrow (q \vee r)\} \models p \leftrightarrow q$$

[4]

c) You want to throw a party, respecting people's incompatibilities. You know that:

John comes, if Mary or Anne comes.
Anne comes, if Mary does not come.
If Ann comes, then John does not come.

(i) Formalise these statements in propositional logic.

(ii) Using the proof procedure KE for model building, show who is coming to the party (and who is not).

It is essential that you annotate the KE-tree(s) to justify your reasoning.

[6]

d) Using the KE calculus for propositional modal logic, show that:

$$(\Box p \rightarrow \Diamond p) \leftrightarrow (\neg \Box p \vee \neg \Box \neg p)$$

is a theorem of modal logic S5. Annotate your proof to show which rules have been used.

Similarly, show that this formula is also a theorem of modal logic K.

[6]

