UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2002

BEng Honours Degree in Computing Part III
MEng Honours Degree in Information Systems Engineering Part IV
BSc Honours Degree in Mathematics and Computer Science Part III
MSci Honours Degree in Mathematics and Computer Science Part III
MSc in Advanced Computing
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the City and Guilds of London Institute
This paper is also taken for the relevant examinations for the Associateship of the Royal College of Science

PAPER C394=I4.40

ADVANCES IN ARTIFICIAL INTELLIGENCE

Tuesday 30 April 2002, 10:00
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions
Calculators required

1a. State formally the property of soundness of the ASLD abductive proof procedure for abductive logic programming (with definite logic programs). Illustrate the property by means of a concrete example.

b. Give a concrete example of the lack of the completeness property for the ASLDIC abductive proof procedure for abductive logic programming (with definite logic programs and integrity constraints).

c. Use the ASLDICN abductive proof procedure to compute all possible explanations for the observation

O: p(10)

with respect to the following abductive logic program:

T: *p (X)* ***if*** *q(X)* ***and*** *r(X)*
q(X) ***if not*** *s(X)* ***and*** *a(X)*
s(X) ***if*** *b(X)*
r(X) ***if not*** *t(X)*
t(X) ***if not*** *p(X)*

H: *(all ground atoms in the predicates) a, b*

IC: ***if*** *a(X)* ***and*** *w(X)* ***then false***
if *a(X)* ***and*** *b(X)* ***then false***

Describe the working of the abductive proof procedure graphically. For the selection of literals in derivations, assume a left-most selection rule. Return any explanation explicitly.

The three parts carry, respectively, 30%, 20% and 50% of the marks.

2a. Consider the following planning problem, represented in STRIPS:

Operators: *buy-ticket(X,Y)* pre: *have-money* ***and*** *at(X)* ***and*** $X \neq Y$
post: *have-ticket(X,Y)* ***and not*** *have-money*

fly(X,Y) pre: *at(X)* ***and*** *have-ticket(X,Y)* ***and*** $X \neq Y$
post: *at(Y)* ***and not*** *have-ticket(X,Y)* ***and not*** *at(X)*

Initial state: *at(heathrow)* ***and*** *have-money* ***and*** *heathrow* $\neq$ *barbados.*

Goal: *at(barbados)*

Namely, you are at Heathrow airport and want to fly to Barbados, but have no ticket. You have, however, enough money with you to buy a ticket to Barbados.

i. Apply GRAPHPLAN to compute a suitable plan for the goal, ignoring all inequalities. Remember that GRAPHPLAN uses fictional no-op actions, in addition to the actions represented above.

- Return the full graph computed by GRAPHPLAN, with all appropriate arcs. Define any syntactic/graphical convention you adopt.
- Explicitly give the computed plan as a sequence of actions.
- Indicate, for the action level of time-step 1 and the proposition level of time-step 2, all mutual exclusions.

ii. Is the plan computed in part i. minimal? Justify your answer.

iii. Give an example of a non-minimal plan for the given domain, initial situation and goal.

b. Briefly describe a hierarchical planning extension to GRAPHPLAN.

Parts a.i, a.ii, a.iii and b carry, respectively, 50%, 10%, 10% and 30% of the marks.

3 a. Draw a diagram of an artificial neuron, showing its input vector, weight vector, input potential, activation state and output. Explain how the neuron works.

b. Use a multilayer perceptron with neurons with non-linear activation function *h(x)*, where

$h(x) = 1$, *if* $x > 0$; $h(x) = 0$, *otherwise*

to compute the XOR function, where

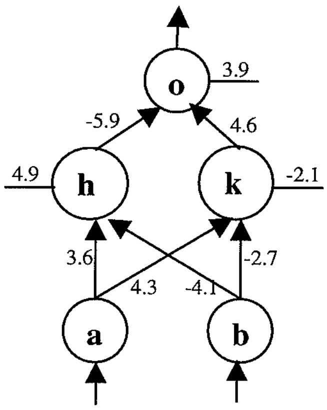
XOR(A,B) = true if and only if the truth-value of A is not the same as the truth-value of B.

Use 1 to represent truth-value *true*, and 0 to represent truth-value *false*. Draw a diagram of your network indicating its hidden neurons and set of weights (including its biases).

c. The network N below was trained using Backpropagation, with input neurons **a**,**b** $\in \{-1,1\}$, the identity function ($f(x)=x$) as activation function of **a** and **b**, and *tanh(x)* as activation function of both the hidden neurons **h**, **k**, and the output neuron **o**, where

$tanh(x)=(e^{2x} - 1)/(e^{2x} + 1)$, $e = 2.718$.

Network N



Note that, in the network N, 3.9 is the *bias* of **o**, 4.9 is the *bias* of **h**, and -2.1 is the *bias* of **k**.

i. Compute the activation of the neuron **o** for inputs **a** = 1, **b** = 1 and for inputs **a** = -1, **b** = 1. Show your working.

ii. Compute one weight change (Δ**W**) for each of the weights of network N, given the training example: **a** = -1, **b** = 1, t = -1, where t is the target output of neuron **o**. Use learning rate η = 0.1. Remember that the derivative *tanh'(x)* of *tanh(x)* is given by $tanh'(x) = 1 - tanh^2(x)$.

Parts a, b, c.i and c.ii carry, respectively, 20%, 30%, 20% and 30% of the marks.

4a Consider the following planning domain:

I am inviting some friends for dinner, and I want to offer them a pasta dish and a salad. In order to prepare the pasta dish I need pasta, and in order to prepare the salad I need lettuce. I have pasta but no lettuce. There are two shops in my neighbourhood that sell lettuce.

i. Represent the given domain in STRIPS, Situation Calculus and Event Calculus, identifying the actions/operators, the goal, and the initial state. For the Situation Calculus, use successor state axioms. For the Event Calculus, use the non-abductive formulation. Also, for the Event Calculus, you do not need to give the domain-independent axioms. Finally, for all three formalisms, assume that after using some ingredient to prepare a dish the ingredients are no longer available, but shops have unlimited supplies of them. Finally, assume that ingredients required to prepare dishes do not change over time.

ii. For each representation in part i. give *one* possible *minimal* plan for the given goal, given the initial situation.

b Given the clauses

C: p(a,X) ***or*** *r(X,X)*

C1: p(a,b) ***or*** *q(b)*

apply inverse resolution to compute three possible clauses C2 such that C is the resolvent of C1 and C2. For each returned C2, give the corresponding most general unifier.

Parts a.i, a.ii and b carry, respectively, 60%, 10% and 30% of the marks.