UNIVERSITY OF LONDON

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2002

BEng Honours Degree in Computing Part I
MEng Honours Degrees in Computing Part I
BSc Honours Degree in Mathematics and Computer Science Part I
MSci Honours Degree in Mathematics and Computer Science Part I
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 C140=MC140

LOGIC

Wednesday 24 April 2002, 14:00
Duration: 90 minutes
(Reading time 5 minutes)

Answer THREE questions

Paper contains 4 questions
Calculators not required

1a List the boolean connectives $\to, \neg, \wedge, \leftrightarrow, \vee$ in decreasing order of binding strength. Then draw the formation trees of the formulas

i) $\neg p \to q \vee r \wedge s$

ii) $\neg p \to \neg q \wedge \neg r \leftrightarrow \neg s \vee t$

b Given that p is true and q, r are false, calculate the truth value of each of:

i) $(p \wedge r) \vee (p \wedge \neg q \to \neg p \vee q)$

ii) $\neg(\neg p \leftrightarrow \neg(q \to \neg p) \vee r)$

c Using equivalences, show that $p \to (q \to \neg(p \wedge \neg q))$ is valid.

d Using natural deduction, show that

$$\vdash\ p \vee (q \to \neg(p \vee \neg q)).$$

You may use the lemma $A \vee \neg A$, for any suitable formula A. Do not rewrite any formula using equivalences.

The four parts carry, respectively, 20%, 20%, 25%, 35% of the marks.

2a Let A, B be first-order formulas. Explain what it means to say that

i) A is valid,

ii) A is satisfiable,

iii) A and B are logically equivalent.

b Use equivalences to show that the following pairs of formulas are logically equivalent:

i) $\exists x(x{=}y \vee \text{green}(x))$ and $\exists u(u{=}u \wedge y{=}u) \vee \exists v\, \text{green}(v)$

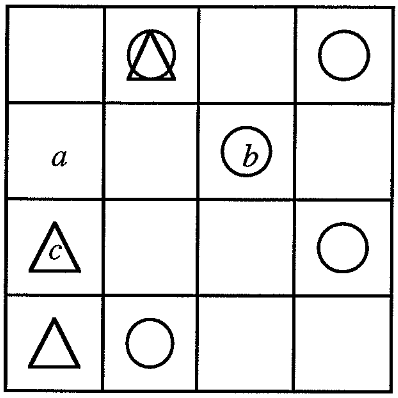
ii) $\exists x \forall y\, (\text{friend}(x,y) \to \text{happy}(x))$ and $\forall x \exists y\, \text{friend}(x,y) \to \exists v\, \text{happy}(v)$

c Using natural deduction, show that

$\forall x(\text{cold}(x) \vee \text{hot}(x))$
$\neg\exists x(\text{red}(x) \wedge \text{blue}(x))$
$\forall x(\text{hot}(x) \to \text{red}(x))$
$\exists x\, \text{blue}(x) \qquad \vdash \qquad \exists x\, \text{cold}(x)$

3 In this question, L is a signature consisting of three constants *a, b, c,* two unary relation symbols *triangle, circle,* and two binary relation symbols *above, left_of.* M is the L-structure whose domain consists of 16 objects, represented by the 16 squares in the diagram.

M



The interpretations in M of the symbols of L are as suggested by the diagram. So there is an object (in the top row) for which both *triangle* and *circle* are true; *above*(x,y) means x is above y (not necessarily in the same column); *left_of*(x,y) means x is to the left of y (not necessarily in the same row), etc.
E.g., *circle*(*b*), $\neg$ *circle*(*a*), *left_of*(*c,b*), and $\neg$ *above*(*a,b*) are all true in M.

a Translate each of the following statements into an L-formula that has the same meaning in any L-structure with the same domain and interpretations of *left_of* and *above* as M. (E.g., 'there is a circle' could be translated as $\exists$x *circle*(x).)

i) no object is both a circle and a triangle

ii) every circle has a circle somewhere below it

iii) every object in the top row is a triangle

iv) x is in the same row as y (here, x and y are variables)

v) some row has no circles

b Translate the following L-sentences into *natural* English.

i) *triangle*(*a*) $\wedge$ *above*(*c,a*)

ii) $\exists$x$\exists$y(*circle*(x) $\wedge$ *circle*(y) $\wedge$ *left_of*(x,y))

iii) $\forall$x(*circle*(x) $\to$ $\exists$y[*triangle*(y) $\wedge$ (*above*(x,y) $\vee$ *above*(y,x))])

iv) $\forall$x(*circle*(x) $\leftrightarrow$ *left_of*(*c*,x))

v) $\forall$x($\neg$ $\exists$y *left_of*(x,y) $\to$ $\neg$ *left_of*(*b*,x))

c Which of the sentences in part b are true in the structure M shown above? Briefly justify your answers.

d Draw a diagram of an L-structure N with the same domain and interpretations (meanings) of *left_of* and *above* as M (but with different interpretations of the symbols *circle, triangle, a, b, c*), such that all the sentences in part b are true in N.

4 In this question, L is the 2-sorted signature with sorts `Nat` and [`Nat`], containing constants $\underline{\mathbf{0}}, \underline{\mathbf{1}}, \underline{\mathbf{2}}, \ldots$: `Nat` and [ ]:[`Nat`], function symbols +, –, ×, :, ++, !!, #, and relation symbols <, ≤, and `merge`, of appropriate sorts.
The intended semantics is a structure whose domain consists of the natural numbers 0, 1, 2, ... (sort `Nat`), and all lists of natural numbers (sort `[Nat]`).
The symbols have the usual meanings: for example, `merge`(ys, zs, xs) holds iff xs is a permutation of ys++zs and the relative order of entries in ys and in zs is retained in xs.

xs, ys, zs are variables of sort [`Nat`], and i, j, k, n are variables of sort `Nat`.
in(n,xs) is an L-formula expressing that n occurs as an entry in xs.
x≠y abbreviates ¬(x=y).

a Write down in plain English (*not* just a literal translation) what each of the following formulas says about xs and (if relevant) n, k.

In each case, give an example of a list xs of length 4 and (if relevant) numbers n, k for which the formula is true.

i) $\forall k\, \forall ys\, \forall zs\, (xs = ys \mathbin{++} k{:}zs \;\rightarrow\; k \neq n)$

ii) $\exists n\, \exists ys\, \exists zs\, (xs = ys \mathbin{++} zs \;\wedge\; in(n,ys) \wedge in(n,zs))$

iii) $xs!!\underline{\mathbf{0}} = \underline{\mathbf{0}} \;\wedge\; \forall i(i+\underline{\mathbf{1}} < \#xs \rightarrow xs!!(i+\underline{\mathbf{1}}) = (xs!!i) + (\underline{\mathbf{2}} \times i) + \underline{\mathbf{1}})$

iv) $\exists ys\, \exists zs(\mathtt{merge}(ys,zs,xs) \wedge \forall i(in(i,ys) \rightarrow i \leq n) \wedge \forall j(in(j,zs) \rightarrow n<j) \wedge k = \#ys)$

v) $\exists ys(\#xs < \#ys \;\wedge\; ys!!\underline{\mathbf{0}} = \underline{\mathbf{0}} \;\wedge\; ys!!\#xs = k \;\wedge$
$\forall i(i < \#xs \rightarrow [xs!!i = n \wedge ys!!(i+\underline{\mathbf{1}}) = \underline{\mathbf{1}}+ys!!i] \;\vee\; [xs!!i \neq n \wedge ys!!(i+\underline{\mathbf{1}}) = ys!!i]))$

b Write down L-formulas expressing suitable pre- and post-conditions for each of the following Haskell functions. You may use the formula 'in', but any other formulas you use must be written out in full. Remember that `merge` is available.

i) tail xs -- returns the list obtained from xs by removing its first (0th) entry.
Example: tail [1,2,3,1] = [2,3,1].

ii) shrink xs -- returns a list obtained from xs by removing all but one occurrence of each entry. There is no restriction on the order of the entries in the returned list.
Example: shrink [4,3,3,4,1,3] = [3,4,1].

iii) split xs -- returns a pair (ys, zs) where ys is the list of all even entries in xs and zs is the list of all odd entries in xs. There is no restriction on the order of the entries in the returned lists.
Example: split [8,2,5,0,5] = ([8,2,0], [5,5]).

iv) min xs -- returns the list of all minimal elements of xs.
Example: min [3,5,2,7,2] = [2,2].

v) sum xs -- returns the sum of the entries in a non-empty list xs.
Example: sum [2,3,1] = 6.