UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFORMATICS 1A - LOGIC AND COMPUTATION

Wednesday 5 December 2007

14:30 to 16:30

Convener: M O'Boyle
External Examiner: R Irving

INSTRUCTIONS TO CANDIDATES

1. **Note that ALL QUESTIONS ARE COMPULSORY.**
2. **DIFFERENT QUESTIONS MAY HAVE DIFFERENT NUMBERS OF TOTAL MARKS. Take note of this in allocating time to questions.**

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. (a) Briefly explain what it means for an expression to be a tautology. [*2 marks*]

(b) Show, using a detailed truth table, that the expression

$$((a \ and \ b) \rightarrow c) \leftrightarrow (a \rightarrow (b \rightarrow c))$$

is a tautology. [*10 marks*]

(c) You are given the following proof rules:

Rule number	Sequent	Supporting proofs
1	$\mathcal{F} \vdash A$	$A \in \mathcal{F}$
2	$\mathcal{F} \vdash A \leftrightarrow B$	$\mathcal{F} \vdash A \rightarrow B,\ \mathcal{F} \vdash B \rightarrow A$
3	$\mathcal{F} \vdash A \rightarrow B$	$[A\|\mathcal{F}] \vdash B$
4	$\mathcal{F} \vdash A \ and \ B$	$\mathcal{F} \vdash A,\ \mathcal{F} \vdash B$
5	$\mathcal{F} \vdash C$	$(A \ and \ B) \in \mathcal{F},\ [A, B\|\mathcal{F}] \vdash C$
6	$\mathcal{F} \vdash C$	$A \rightarrow B \in \mathcal{F},\ \mathcal{F} \vdash A,\ [B\|\mathcal{F}] \vdash C$

where $\mathcal{F} \vdash A$ means that expression A can be proved from set of axioms $\mathcal{F}$; $A \in \mathcal{F}$ means that A is an element of set $\mathcal{F}$; $[A|\mathcal{F}]$ is the set constructed by adding A to set $\mathcal{F}$; $A \rightarrow B$ means that A implies B; $A \ and \ B$ means that A and B both are true; and $A \ or \ B$ means that at least one of A,B is true. Using the proof rules above, prove the following:

$$[] \quad \vdash \quad ((a \ and \ b) \rightarrow c) \leftrightarrow (a \rightarrow (b \rightarrow c))$$

Show precisely how the proof rules are applied. [*20 marks*]

2. The proposition c can be proved from the following set of axioms:

$$[a, \quad (a \ and \ b) \rightarrow c, \quad (d \ or \ e) \rightarrow b, \quad a \rightarrow e]$$

(a) Written in clausal form the set of axioms above is:

$$[[a],\ [not(a), not(b), c],\ [not(d), b],\ [not(e), b],\ [not(a), e]]$$

Explain what clausal form notation means. [*5 marks*]

(b) Explain how to convert the set of axioms above into clausal form. You may find the following equivalences helpful.

$A \rightarrow B$	is equivalent to	$not(A) \ or \ B$
$not(A \ or \ B)$	is equivalent to	$not(A) \ and \ not(B)$
$not(A \ and \ B)$	is equivalent to	$not(A) \ or \ not(B)$
$(A \ and \ B) \ or \ C$	is equivalent to	$(A \ or \ C) \ and \ (B \ or \ C)$
$not(not(A))$	is equivalent to	A

Show, in detail, each step in your conversion. [*10 marks*]

(c) Give a proof, using resolution, of c from the axioms in clausal form above. Show each step of your proof in detail. [*13 marks*]

3. In the design of digital circuits it often is necessary to synchronise signals arriving from different sub-circuits. One way to do this is by slowing signals coming from appropriate parts of the circuit using unit-delay transducers. A unit-delay transducer is one that receives as input a stream of symbols and reproduces this input one time unit later. Suppose that a time unit is the length of time, L, taken to move from one state to another in the finite state machine. If the transducer reads a symbol at time T then it will reproduce that symbol as output at time $T + L$.

   (a) Draw the deterministic finite state machine for the unit-delay transducer described above, operating on the input alphabet $\{a, b\}$. [*15 marks*]

   (b) Your transducer operates for a delay of a single time unit and for an alphabet consisting of two symbols, but you could build a similar transducer for longer delays and for more symbols. How would the number of states in your transducer increase as you increased the delay and the number of symbols? Briefly explain your answer. [*5 marks*]

4. Draw a non-deterministic finite state machine that accepts the language described by the regular expression:

$$(a|bb)^*(ba^*|c)$$

[*10 marks*]

5. Assuming that your alphabet is the set of characters $\{a, b\}$, write regular expressions for the following languages:

   (a) All strings containing exactly one a.

   (b) All strings with an even number of b's.

   (c) All strings containing at least one occurrence of a and at least one occurrence of b.

[*7 marks*]

6. You are given the following equalities between languages represented by regular expressions, where $L(R)$ is the language represented by regular expression R:

$$\begin{array}{rcl} L(RR^*|\epsilon) & = & L(R^*) \\ L((R|S)T) & = & L(RT|ST) \\ L(R|R) & = & L(R) \end{array}$$

   Use these rules to show that the regular expression $((aa^*|\epsilon)c)|((b|b)c)$ represents the same language as the regular expression $(a^*|b)c$. [*3 marks*]