UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFORMATICS 1A: LOGIC AND COMPUTATION

Monday 18 August 2008

14:30 to 16:30

Convener: M O'Boyle
External Examiner: R Irving

INSTRUCTIONS TO CANDIDATES

1. **Candidates in the third or later year of study for the degrees of MA(General), BA(Relig Stud), BD, BCom, BSc(Social Science), BSc(Science) and BEng should put a cross (X) in the box on the front cover of the script book.**
2. **Note that ALL QUESTIONS ARE COMPULSORY.**
3. **DIFFERENT QUESTIONS MAY HAVE DIFFERENT NUMBERS OF TOTAL MARKS. Take note of this in allocating time to questions.**

Write as legibly as possible.

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. Imagine that you have a system with three sensor units: Sensor 1, Sensor 2 and Sensor 3. Each sensor is monitoring some aspect of the environment. Sensors can become faulty so there is a system of self-monitoring between the sensors, allowing each one to provide a report on whether it believes any sensors to be faulty. The information provided by a sensor, however, may not be reliable.

   Each sensor has made the following report:

   Sensor 1 reports : "Sensor 2 is faulty and Sensor 3 is non-faulty."

   Sensor 2 reports : "If Sensor 1 is faulty then so is Sensor 3."

   Sensor 3 reports : "I am non-faulty but at least one of the other two sensors is faulty."

   Let $s1$, $s2$ and $s3$ mean "Sensor 1 is non-faulty", "Sensor 2 is non-faulty" and "Sensor 3 is non-faulty", respectively.

   (a) Rewrite the reports made by Sensor 1, Sensor 2 and Sensor 3, each as an expression in propositional logic. [*3 marks*]

   (b) If all the sensors are non-faulty, which sensors gave incorrect reports? Explain your answer using a truth table. [*3 marks*]

   (c) If the statements made by all the sensors are true, which sensors are faulty? Explain your answer using a truth table. [*6 marks*]

2. You are given the following proof rules:

Rule number	Sequent	Supporting proofs
1	$\mathcal{F} \vdash A$	$A \in \mathcal{F}$
2	$\mathcal{F} \vdash A \ and\ B$	$\mathcal{F} \vdash A,\ \mathcal{F} \vdash B$
3	$\mathcal{F} \vdash A \ or\ B$	$\mathcal{F} \vdash A$
4	$\mathcal{F} \vdash A \ or\ B$	$\mathcal{F} \vdash B$
5	$\mathcal{F} \vdash C$	$\mathcal{F} \vdash (A \ or\ B),\ [A \mid \mathcal{F}] \vdash C,\ [B \mid \mathcal{F}] \vdash C$
6	$\mathcal{F} \vdash B$	$A \rightarrow B \in \mathcal{F},\ \mathcal{F} \vdash A$
7	$\mathcal{F} \vdash A \rightarrow B$	$[A \mid \mathcal{F}] \vdash B$

   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 or B is true.

   Using the proof rules above, prove the following:

$$[p \rightarrow (a \ or\ b),\ a \rightarrow x,\ b \rightarrow x] \quad \vdash \quad p \rightarrow x$$

   Show precisely how the proof rules are applied. [*15 marks*]

3. The proposition d can be proved from the following set of axioms in clausal form:

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

(a) Explain what clausal form notation means, in terms of the conjunction and disjunction of propositional expressions. [*5 marks*]

(b) Give a proof, using resolution, of d from the axioms above in clausal form. Show each step of your proof in detail. [*10 marks*]

4. In this question we consider Finite State Machines (FSMs) for recognising natural numbers (the counting numbers, that is, $0, 1, 2, 3, 4, \ldots$) with certain properties. We will work with two different representations of natural numbers: In unary notation, the natural number n is represented by a string of n "1"s. The natural number 0 is a special case which we represent by the empty string ϵ. In binary notation, a number is represented by a string of "0"s and "1"s. If $b = b_k \ldots b_1 b_0$ is a binary string (of length $k + 1$) then the number represented by that binary string is the number $n = b_0 + 2b_1 + \ldots + 2^k b_k$.

(a) First consider the question of designing FSMs over the binary alphabet $\Sigma = \{0, 1\}^*$. Assume that a binary string $b \in \{0, 1\}^*$ is input to an FSM with its least significant digit first (b_0, then b_1, then b_2, $\ldots$).

i. Draw a deterministic FSM to recognise the set of binary strings which represent the *even* natural numbers. [*3 marks*]

ii. Is it possible to design an FSM to recognise the set of binary strings which represent numbers which are powers of 2 (the numbers $1, 2, 4, 8, \ldots$ and so on)? Justify your answer, either by drawing the machine or by explaining why you think no such machine exists. [*3 marks*]

(b) Now consider the question of designing FSMs over the unary alphabet $\Sigma = \{1\}^*$.

i. Draw a deterministic FSM to recognise the set of unary strings which represent the *even* natural numbers. [*3 marks*]

ii. Is it possible to design an FSM to recognise the set of unary strings which represent numbers which are powers of 2 (the numbers $1, 2, 4, 8, \ldots$ and so on)? Justify your answer, either by drawing the machine or by explaining why you think no such machine exists. [*3 marks*]

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

$$((ab^*a)|(aa^*b))ba^*$$

[*12 marks*]

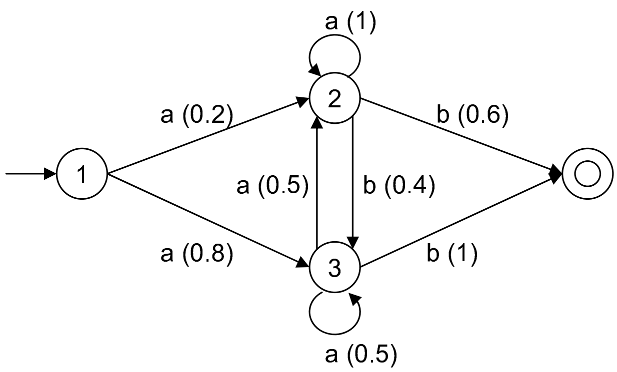
6. In this question we are concerned with modelling the behaviour of a gas cooker with two burners, a small burner and a large burner. The cooker has a gas tap for each burner, to turn on and turn off the gas for that burner. The cooker also has an ignite button: when the ignite button is pressed, it lights any burner whose gas is turned on (if both taps are on it lights both burners; if both taps are off it does nothing). When the tap for a burner is closed while the tap is on and the burner is lit, it switches off that burner. We say that the cooker is in a safe state when both taps are switched off. The starting state is the safe state.

   (a) Draw a Finite State Machine (FSM) to model the behaviour of this cooker. The input alphabet for the FSM is $\{\mathrm{s}o, \mathrm{s}f, \mathrm{l}o, \mathrm{l}f, \mathrm{i}g\}$. The input s$o$ indicates that the tap for the small burner has been turned on; the input sf indicates that the tap for the small burner has been turned off; the input lo indicates that the tap for the large burner has been turned on; the input lf indicates that the tap for the large burner has been turned off; the input ig indicates that the ignite button has been pressed. *[20 marks]*

   (b) Use the FSM you have constructed for the gas cooker to show whether or not it meets the following safety requirements:

   - Pressing the ignite button will light all burners with open gas taps, regardless of the order in which the taps were opened.
   - Turning off a tap will always put the appropriate burner into an unlit state.
   - An unlit burner can never light without a sequence of turning on the appropriate tap followed by ignition.
   - The cooker is in its safe (off) state only when both burners are turned off. *[10 marks]*

7. The diagram below describes a Probabilistic Finite State Machine that accepts strings over the alphabet $\{a, b\}$. Each transition between states has a probability of occurrence (for example the transition between states 1 and 2 accepting character "a" has probability 0.2).



Explain how you would calculate the probability that the string *aab* would be accepted by this FSM. [*4 marks*]