Module Title: Informatics 1 - Computation and Logic
Exam Diet (Dec/April/Aug): August 2014
Brief notes on answers:

1. Consider the inhabitants A and C, who each claim that they are Smiley. At least one of these inhabitants must be lying, since only one of the 3 is named Smiley. No more than one can be lying, since knights never lie and the spy would not be lying if he claimed to be named Smiley. The one who is lying must be a knave, since the spy is actually named Smiley. Thus one of A and C is a knave, and the other is a spy. Thus the knight is the one who testifies to the truth of A's statement. Since knights always tell the truth, this means A is telling the truth, and hence that A is named Smiley, which means he is the spy.

   10 marks for a completely correct answer. 5 marks if there is partially correct reasoning, but the final answer is incorrect.

2. (a) Answer to first part of question 2

p	q	$p \to q$	$p \to (p \to q)$	$p \to (p \to (p \to q))$	$(p \to q) \leftrightarrow (p \to (p \to (p \to q)))$
T	T	T	T	T	T
T	F	F	F	F	T
F	T	T	T	T	T
F	F	T	T	T	T

   (b) Answer to second part of question 2

p	q	r	$p \vee q$	$\neg p \vee r$	$Expr1$	$p \wedge r$	$q \wedge \neg p$	$q \wedge r$	$Expr2$	$Expr1 \leftrightarrow Expr2$
T	T	T	T	T	T	T	F	T	T	T
T	T	F	T	F	F	F	F	F	F	T
T	F	T	T	T	T	T	F	F	T	T
T	F	F	T	F	F	F	F	F	F	T
F	T	T	T	T	T	F	T	T	T	T
F	T	F	T	T	T	F	T	F	T	T
F	F	T	F	T	F	F	F	F	F	T
F	F	F	F	T	F	F	F	F	F	T

   Some notes:

   Expr1 refers to $(p \vee q) \wedge (\neg p \vee r)$

   Expr2 refers to $(p \wedge r) \vee (q \wedge \neg p) \vee (q \wedge r)$

   (c) Answer to third part of question 2

p	q	r	$p \to q$	$q \to r$	$Expr1$	$p \wedge q \wedge r$	$\neg p \wedge \neg q$	$\neg p \wedge \neg q \wedge \neg r$	$Expr2$	$Expr$
T	T	T	T	T	T	T	F	F	T	T
T	T	F	T	F	F	F	F	F	F	T
T	F	T	F	F	F	F	F	F	F	T
T	F	F	F	T	F	F	F	F	F	T
F	T	T	F	T	F	F	F	F	F	T
F	T	F	F	F	F	F	F	F	F	T
F	F	T	T	F	F	F	T	F	F	T
F	F	F	T	T	T	F	T	T	T	T

   Some notes:

Expr refers to $Expr1 \leftrightarrow Expr2$
Expr1 refers to $(p \rightarrow q) \wedge (q \rightarrow r)$
Expr2 refers to $(p \wedge q \wedge r) \vee (\neg p \wedge \neg q \wedge \neg r)$

3. (a) A formula is in conjunctive normal form if it is the conjunction of clauses, each clause being a disjunction of literals, where each literal is either a variable or its negation. 3 marks for a correct answer, partial credit if some aspect of the definition of CNFs is missed out.

   (b) (i). $not(q)$ *or* r and $(p$ *or* $not(q)$ *or* $r)$ *and* $(not(q)$ *or* $r)$ are both correct answers. 5 marks for a correct answer; 0 or 2 marks for an incorrect answer.

   (ii). The answer is $(p$ *and* $not(q))$ *or* $(q$ *and* $not(p))$. 5 marks for a correct answer; 0,2 or 3 marks for an incorrect one.

   (iii). The answer is p *and* $not(q)$. 3 marks for a correct answer; 0 or 2 for an incorrect one.

   (c) The following is a correct application of the Davis-Putnam algorithm. Other choices of variable ordering are possible too. 9 marks for a fully correct answer, marks deducted for each error.

$[[a, \neg a, \underline{\mathbf{b}}], [a, \neg c, d], [\underline{\mathbf{b}}, c, \neg d], [c, \neg d], [\underline{\neg\mathbf{b}}.\neg d], [a, c]]$
$[[a, \neg a, \neg d], [a, \underline{\neg\mathbf{c}}, d], [\underline{\mathbf{c}}, \neg d], [\underline{\mathbf{c}}, \neg d], [a, \underline{\mathbf{c}}]]$
$[[a, \neg a, \underline{\neg\mathbf{d}}], [a, \underline{\mathbf{d}}, \underline{\neg\mathbf{d}}], [a, \underline{\mathbf{d}}, \underline{\neg\mathbf{d}}], [a, \underline{\mathbf{d}}]]$
$[[\underline{\mathbf{a}}, \underline{\neg\mathbf{a}}], [\underline{\mathbf{a}}]]$
$[a]$

4. (a) Finite-state transducers output a symbol at each transition, in addition to reading a symbol. They do not have final states. Finite-state acceptors have final states, and do not output symbols. Finite-state acceptors accept languages, while finite-state transducers transform the input string to an output string.

   2 marks for each correct difference between a transducer and an acceptor.

   (b) A drawing of the machine is given in Figure 1. 9 marks for a fully correct answer.

5. Proof by contradiction. Suppose there were a finite-state machine M which accepted only those strings whose length is a power of 2. Let K be the number of states of the machine. Let n be the first power of 2 larger than K. By assumption on M, M accepts the string of n 1's. Consider the trace of M on 1^n. This must begin with the initial state and end with a final state, and by the pigeon-hole principle, there must be some state q which is repeated in the trace. Now the portion of the trace between the two closest occurrences of q can be repeated to give another valid trace ending in an accepting state, but since the length of this portion is less than n, the corresponding string accepted has length greater than n but less than $2n$, which contradicts the assumption that the machine only accepts strings whose length is a power of 2.

   15 marks for a fully correct proof, with partial credit for the various ideas involved: proof by contradiction, choice of the string whose trace to consider, application of the pigeon-hole principle to "pump" the trace, realization that pumping leads to acceptance of a string not in the language.

6. (a) The machine is drawn in Figure 2. 2 marks for a correct drawing, 0 for an incorrect one.

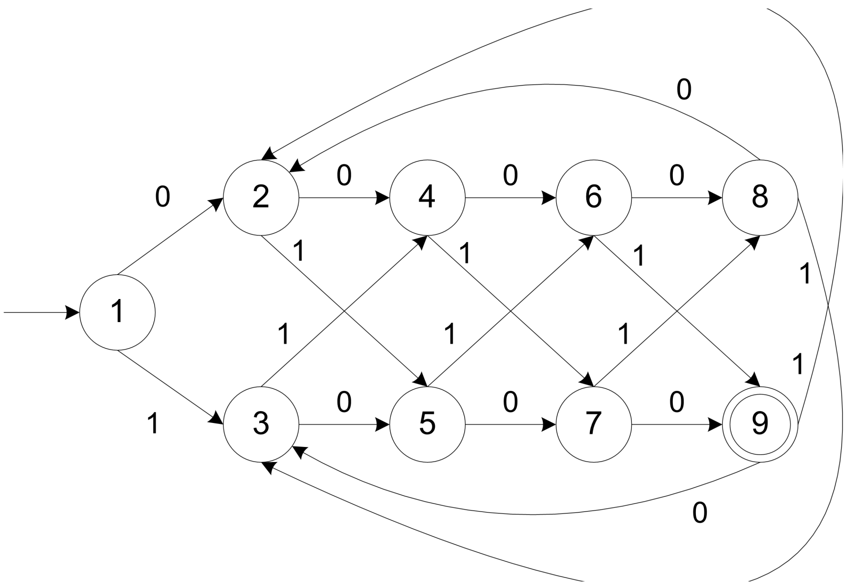

Figure 1: A Finite State Machine for Q 4(b)

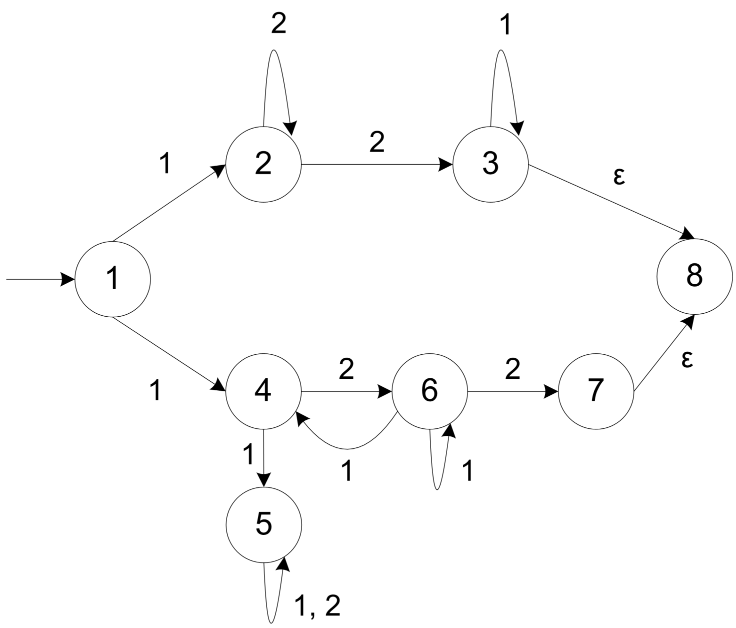

Figure 2: A Drawing of the Machine M in Q 6(a)

(b) Here's a correct regular expression: $(122^*1^*)|(1(211^*)^*)2)$. 10 marks for this or any other correct answer; 2,5 or 7 marks for partially correct solutions.

(c) An equivalent deterministic finite-state machine is drawn in Figure 3. 10 marks for this or any other correct machine.

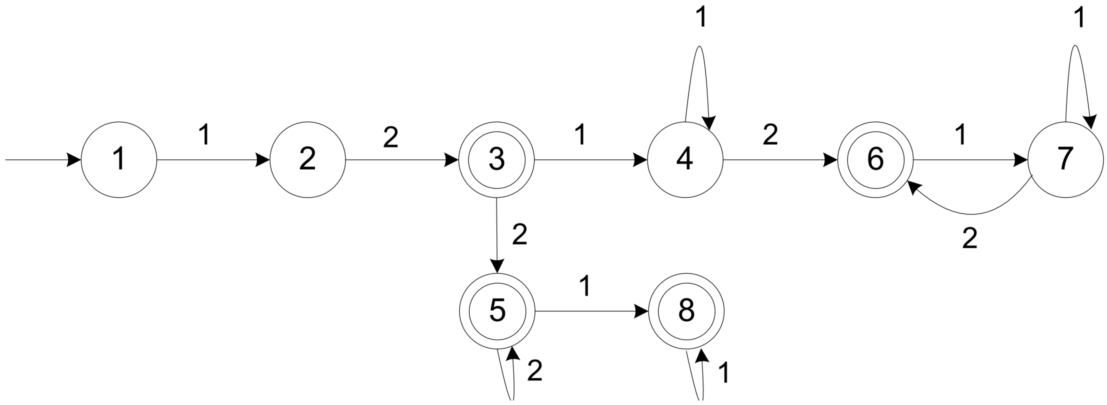

Figure 3: A Drawing of a Deterministic Finite-State Machine Equivalent to M