Module Title: Informatics 1 - Computation and Logic
Exam Diet (Dec/April/Aug): December 2013
Brief notes on answers:

1. (a) For each of the following answers, a correct answer (with or without working) gets 3 marks. An incorrect answer without working gets zero. An incorrect answer with partially correct working gets 1 or 2 marks.

    (i). Tautology
    (ii). Contingency
    (iii). Contradiction
    (iv). Contingency
    (v). Tautology

   (b) (i). Suppose Sentence A were true. Then Sentence B would be true. But Sentence B states that Sentence A is false, therefore since Sentence B is true, Sentence A would be false.
   Suppose Sentence A is false. Then Sentence B would be false, which would mean Sentence A would be true.
   Thus neither case is possible, which creates a logical paradox.
   1 mark for the argument based on assumption that Sentence A is true, 1 mark for the argument based on it being false, and 1 mark for explaining why this creates a paradox.

   (ii). The paradox occurs because sentences are permitted to talk about the truth value of sentences.
   2 marks for this or for any mention of self-referentiality.

2. 15 marks for a fully correct solution, with either 10 or 5 marks for a partially correct one. See Figure 2 for the figure.

3. (a) The answer is $((not(a)\ or\ b) \rightarrow c)\ and\ (c \rightarrow not(d))\ and\ not((b\ and\ not(a)) \rightarrow (c\ and\ not(d)))$. 2 marks for a correct answer, zero for a wrong one.

   (b) The answer is $(c\ or\ a)\ and\ (c\ or\ not(b))\ and\ (not(c)\ or\ not(d))\ and\ b\ and\ not(a)\ and\ (not(c)$
   7 marks for a completely correct answer using the logical equivalences; partial credit if the logical equivalences are correctly stated but there is an error in applying them.

   (c) The answer is $[[c, \underline{\mathbf{a}}], [c, \neg b], [\neg c, \neg d], [b], [\underline{\neg\mathbf{a}}], [\neg c, d]]$. 1 mark for a correct answer, 0 for an incorrect one.

   (d) 10 marks for a correct proof as below, with each resolution step applied correctly, including elimination of duplicates at each stage. 6 marks for a correct resolution proof applied to the wrong formula, i.e., in case of an error in the earlier parts.
   $[[c, \underline{\mathbf{a}}], [c, \neg b], [\neg c, \neg d], [b], [\underline{\neg\mathbf{a}}], [\neg c, d]]$
   $[[c], [c, \underline{\neg\mathbf{b}}], [\neg c, \neg d], [\underline{\mathbf{b}}], [\neg c, d]]$
   $[[c], [c], [\neg c, \underline{\neg\mathbf{d}}], [\neg c, \underline{\mathbf{d}}]]$
   $[[\underline{\mathbf{c}}], [\underline{\neg\mathbf{c}}]]$
   $[\,]$

4. (a) See Figure 4a for a drawing of the finite-state machine. The input alphabet is $\{film1, film2, purchase, 5poundnote, cancel\}$ and the output alphabet is $\{notenoughmoney, printticketfilm1, printticketfilm2\}$. 15 marks for a completely correct answer with correct input and output alphabets and transition function; marks deducted for every error.

   (b) The trace is $[0, film1/, 1, cancel/, 0, film2/, 4, 5poundnote/, 5, 5poundnote/, 6, 5poundnote/, 7, 5poundnote/, 7, purchase/printticketfilm2, 0]$. 5 marks for a correct answer; 2 or 3 marks for a partially correct one.

5. The important thing to note is that a number is odd if and only if its representation in base 4 starts with a 1 (since low order digit is read first), and its representation in base 3 has an odd number of 1s. See Figure 5 for a drawing of the machine. State 2 is the only final state. 5 marks for the characterization of inputs, 5 marks for the finite state machine corresponding to this characterization.

6. (a) One correct answer is $(a|b)^*ab(a|b)^*a(a|b)^*|(a|b)^*a(a|b)^*ab(a|b)^*$. 5 marks for this or any other correct answer, 2 or 3 marks if there are minor errors.

   (b) Use equivalences 3,1,2,4 in that order to transform the first expression to the second one. 10 marks for doing this correctly; 2,5 or 8 marks for a partially correct answer.

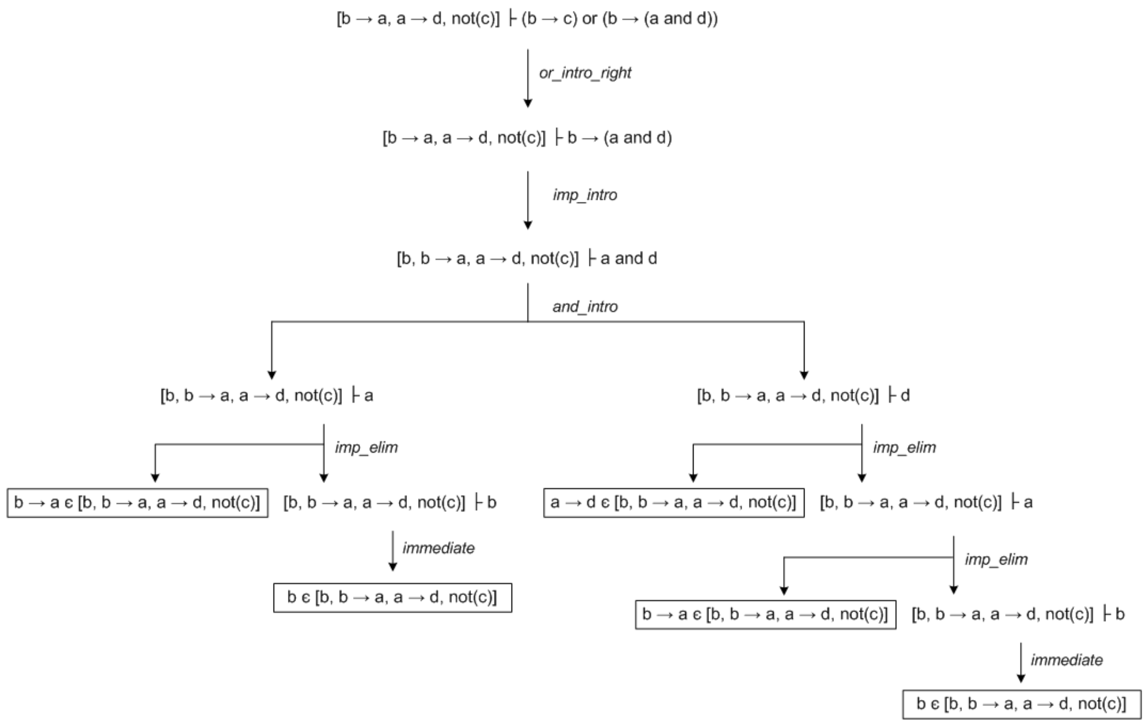

Figure 1: The Proof Tree in Sequent Calculus

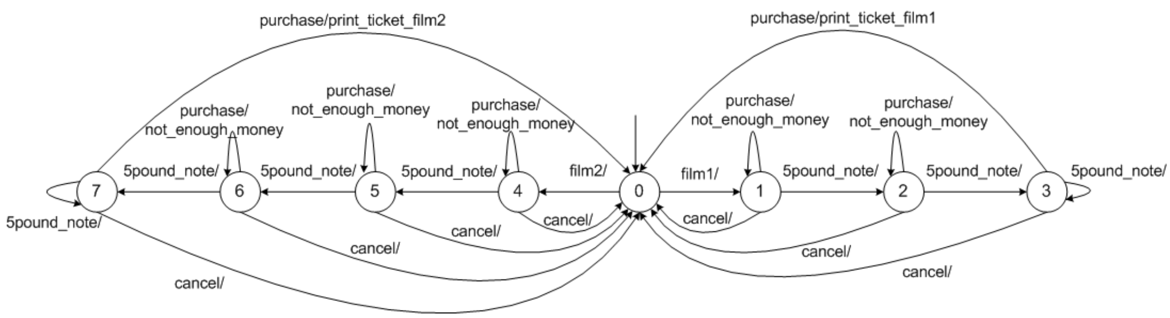

Figure 2: A Finite State Machine for the Ticket Vending Problem

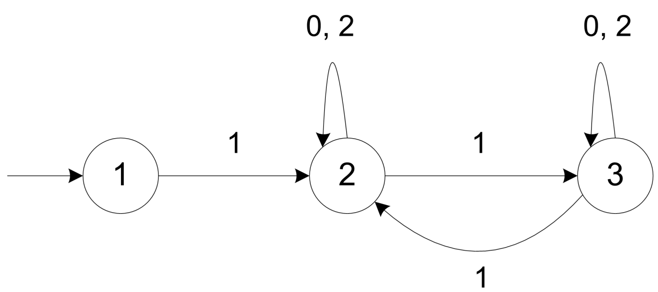

Figure 3: The Finite State Machine for the Odd Number Problem