UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFORMATICS 1 — COMPUTATION & LOGIC

Tuesday 1$^{\text{st}}$ April 2014

00:00 to 00:00

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.**
3. **CALCULATORS MAY NOT BE USED IN THIS EXAMINATION.**

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. (a) Use truth tables to determine which of the following entailments are valid. [*4 marks*]

   i. $A \to (B \vee C) \models A \to C$

   ii. $A \to (B \wedge C) \models A \to C$

   iii. $(A \vee B) \to C \models A \to C$

   iv. $(A \wedge B) \to C \models A \to C$

A	B	C	$A \to (B \vee C)$	$A \to (B \wedge C)$	$(A \vee B) \to C$	$(A \wedge B) \to C$	$A \to C$
$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$
$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
$\bot$	$\top$	$\bot$	$\top$	$\top$	$\bot$	$\top$	$\top$
$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
$\top$	$\bot$	$\top$	$\top$	$\bot$	$\top$	$\top$	$\top$
$\top$	$\top$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$
$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
			$\not\models A \to C$	$\models A \to C$	$\models A \to C$	$\not\models A \to C$	

   i and *iv* are invalid, because there exist valuations for which the premise turns out true, but the conclusion false; *ii* and *iii* are valid because there exists no such valuation.

   (b) This part concerns the 256 possible truth valuations of the following eight propositional letters A, B, C, D, E, F, G, H. For each of the following expressions, say how many of the 256 valuations satisfy the expression, and briefly explain your reasoning. For example, the expression D is satisfied by half of the valuations, that is 128 of the 256, since for each valuation that makes D true there is a matching valuation that make D false. [*6 marks*]

   i. $A \wedge B$ 64 (1/4 * 256)

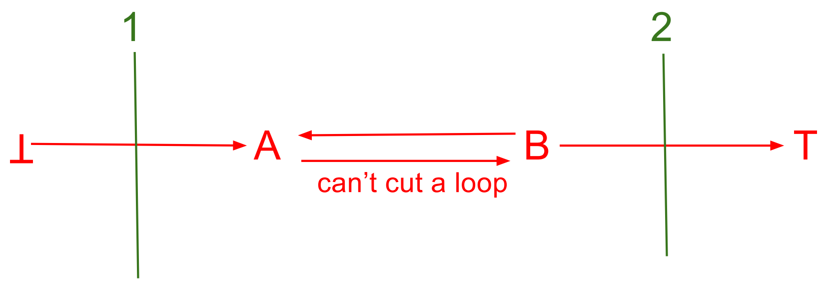
   ii. $(A \vee B) \wedge C$ 96 (3/4 * 1/2 * 256)

   iii. $(A \to B) \to C$ 160 (1/2 + 1/8 * 256)

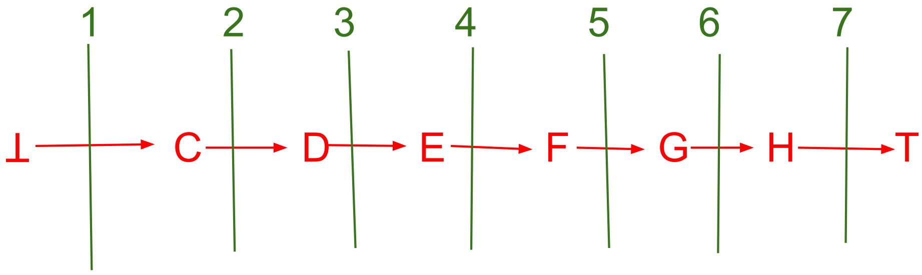
   iv.

$$(A \to B) \wedge (B \to A) \wedge (C \to D) \wedge (D \to E) \wedge (E \to F) \wedge (F \to G) \wedge (G \to H)$$

   We can use the arrow rule to solve this:



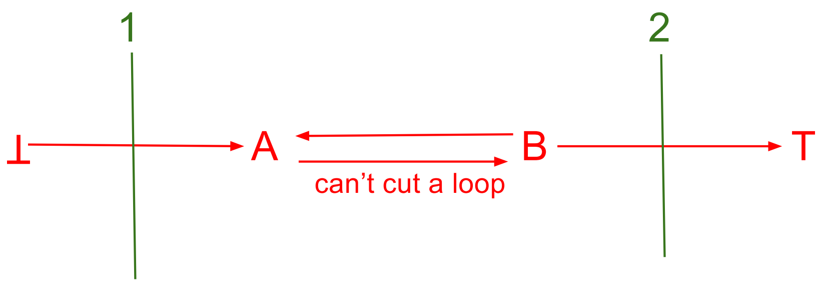
We find 2 valuations for A and B.



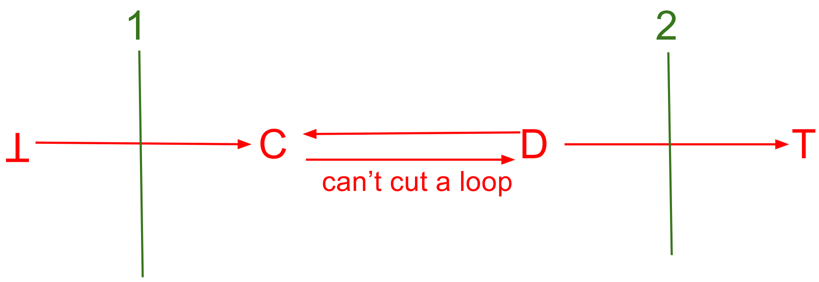
-and 7 for C, D, E, F, G, and H, giving $2 \times 7 = 14$ valuations in total

v.

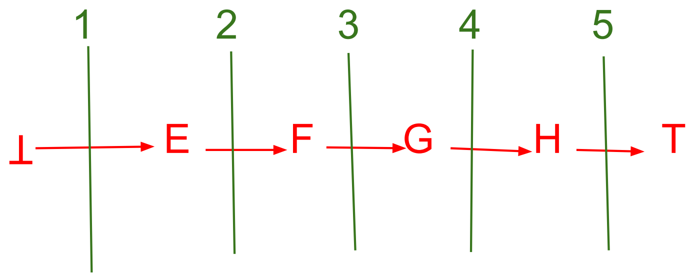
$$(A \to B) \land (B \to A) \land (C \to D) \land (D \to C)$$
$$\land (E \to F) \land (F \to G) \land (G \to H)$$



We find 2 valuations for A and B.



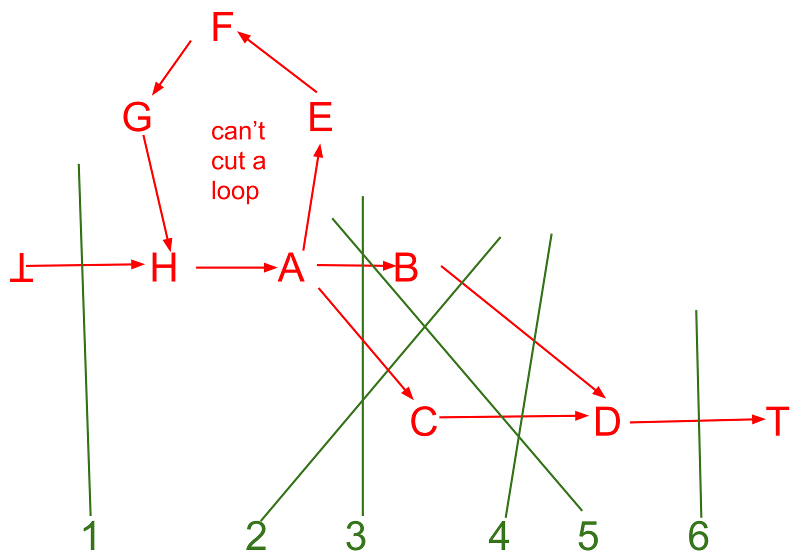
-2 valuations for C and D.



-and 5 for E, F, G, and H, giving $2 \times 2 \times 5 = 20$ valuations in total vi.

$$(H \to A) \wedge (A \to B \wedge C) \wedge (B \vee C \to D) \wedge (A \to E) \wedge (E \to F) \wedge (F \to G) \wedge (G \to H)$$

Noting that $A \to B \wedge C$ is equivalent to $(A \to B) \wedge (A \to C)$ and $(B \vee C \to D)$ is equivalent to $(B \to D) \wedge (C \to D)$, we derive the following graph, giving 6 satisfying valuations:



2. You are given the following inference rules: (Γ, Δ vary over finite sets of expressions; A, B vary over expressions):

$$\frac{}{\Gamma, A \vdash \Delta, A}\ (I)$$

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\ (\wedge L) \qquad \frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \vee B, \Delta}\ (\vee R)$$

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta}\ (\vee L) \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}\ (\wedge R)$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \to B \vdash \Delta}\ (\to L) \qquad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \to B, \Delta}\ (\to R)$$

$$\frac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta}\ (\neg L) \qquad \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta}\ (\neg R)$$

(Where A and B are propositional expressions, Γ, Δ are sets of expressions, and Γ, A refers to $\Gamma \cup \{A\}$.)

(a) Explain what it means to claim that these rules are: [*4 marks*]

i. **sound**,

ii. and **complete**.

Sound: if the premises of any rule are valid, then so are the conclusions; so, any provable sequent is valid. Complete: any valid entailment is provable. Where a sequent is valid if any valuation that makes all of its premises true will make at least one of its conclusions true.

(b) Use these rules to show that [*10 marks*]

$$(A \to B) \to A \vdash A$$

$$\dfrac{\dfrac{}{A \vdash A}\ (I) \qquad \dfrac{\dfrac{}{A \vdash A, B}\ (I)}{\vdash A, A \to B}\ (\to R)}{(A \to B) \to A \vdash A}\ (\to L)$$

(c) Use these rules to build an attempted proof of

$$P \to (Q \vee R), (Q \wedge R) \to S \vdash P \to S$$

A few steps produce the goal $P, P \to (Q \vee R) \vdash S, Q \wedge R$. [*7 marks*]

$$\dfrac{P, P \to (Q \vee R) \vdash S, Q \wedge R \qquad \dfrac{}{P, P \to (Q \vee R), S \vdash S}\,(I)}{\dfrac{P, P \to (Q \vee R), (Q \wedge R) \to S \vdash S}{P \to (Q \vee R), (Q \wedge R) \to S \vdash P \to S}\,(\to R)}\,(\to L)$$

... continuing we get the intermediate goal $P, P \to (Q \vee R) \vdash S, R$

$$\dfrac{\dfrac{\dfrac{}{P, \vdash P, S, Q}\,(I) \qquad \dfrac{\dfrac{}{P, Q \vdash S, Q}\,(I) \qquad P, R \vdash S, Q}{P, Q \vee R \vdash S, Q}\,(\vee L)}{P, P \to (Q \vee R) \vdash S, Q}\,(\to L) \qquad P, P \to (Q \vee R) \vdash S, R}{P, P \to (Q \vee R) \vdash S, Q \wedge R}\,(\wedge R)$$

... continuing we get

$$\dfrac{\dfrac{}{P \vdash P, S, R}\,(I) \qquad \dfrac{P, Q \vdash S, R \qquad \dfrac{}{P, R \vdash S, R}\,(I)}{P, Q \vee R \vdash S, R}\,(\vee L)}{P, P \to (Q \vee R) \vdash S, R}\,(\to L)$$

(d) Explain how you can derive a counter-example from your attempted proof. Since the rules have the special property that any valuation contradicting any of the assumptions contradicts the conclusion, we can falsify the conclusion by falsifying any undischarged assumption. So taking either $P, R, \bar{S}, \bar{Q}$ or $P, Q, \bar{S}, \bar{R}$ suffices to falsify the conclusion. *[2 marks]*

(e) Is this a complete set of rules? Briefly justify your answer. Since we can trivially falsify any entailment containing only atoms, unless some atom occurs on each side — in which case we could discharge the assumtion using (I) — this observation, together with the special property the special property that any valuation contradicting any of the assumptions of a rule contradicts its conclusion, is sufficient to show that this set of rules is complete. *[2 marks]*

3. It is claimed that

$$P \to (Q \vee R), Q \to \neg S, S \vee R, R \to Q, (Q \wedge R) \to S \vdash P \to S$$

This question concerns the resolution of this claim.

(a) Express each of the assumptions in clausal form. [*6 marks*]

i. $P \to (Q \vee R)$

ii. $Q \to \neg S$

iii. $S \vee R$

iv. $R \to Q$

v. $(Q \wedge R) \to S$

(b) Explain how you would use resolution to determine whether the claim is correct. [*4 marks*]

Add $\Big\{\{P\}, \{\neg S\}\Big\}$ — the negation of the conclusion — to our set of clauses and then apply resolution. The claim is true iff the empty clause is derivable.

(c) Use resolution to determine whether the claim is correct. (Show your working.) Start from the clauses [*5 marks*]

$$\Big\{\{\neg P, Q, R\}, \{\neg Q, \neg S\}, \{S, R\}, \{\neg R, Q\}, \{\neg Q, \neg R, S\}, \{P\}, \{\neg S\}\Big\}$$

Any order for resolution is OK. E.g. Starting with S we add the following clauses

$$\Big\{\{R\}, \{\neg Q, \neg R\}, \{\neg Q, R\}\Big\}$$

then taking R we get

$$\Big\{\{\neg Q\}, \{Q\}\Big\}$$

and hence,

$$\Big\{\{\}\Big\}$$

4. Consider the following clauses:

- $\{\neg A, \neg B, \neg C, \neg D\}$
- $\{\neg B, \neg C, D\}$
- $\{\neg A, \neg B, C\}$
- $\{\neg A, B, \neg C\}$

(a) Use the table overleaf to record the search for a satisfying valuation for these clauses, using the one watched literal algorithm.

Start with the empty valuation, watching the first literal in each clause. At each step of the search use one column of the table provided to record in the first four squares the truth values assigned to the atoms, A, B, C, D and use one of remaining rows for each clause, to record the new position of the watched literals at the conclusion of each step. When you need to find a new literal to watch always choose the first one available (reading each expression left-to-right). Place an $\times$ in the watched literal square when no suitable literal is available. [*6 marks*]

(b) What is the invariant for the one watched literal algorithm? [*2 marks*]

Every watched literal is either unassigned, or true.

(c) When must the search backtrack? [*2 marks*]

When, for at least one clause, there is no position for the pointer consistent with the invariant.

(d) Use table below to record the search for a satisfying valuation for these clauses, using the one watched literal algorithm.

A		$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
B			$\top$	$\top$	$\top$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$
C				$\top$	$\top$	$\top$	$\bot$		$\top$	$\bot$	$\bot$
D					$\top$	$\bot$					$\top$
$\neg A, \neg B, \neg C, \neg D$	1	2	3	4	x						2
$\neg B, \neg C, D$	1		2	3		x					
$\neg A, \neg B, C$	1	2	3				x			2	
$\neg A, B, \neg C$	1	2						3	x		

5. An ATM requires the user to enter a four-digit PIN.

When the user inserts her card, the machine requests the PIN. The user enters a string of four decimal digits, one digit at a time.

If the user enters the correct four digits of the password, the ATM displays a welcome string, the user can then request a balance or withdraw £100. Their card is then returned.

When the user enters an incorrect string of four digits, the ATM displays a screen that informs the user that an incorrect password was entered.

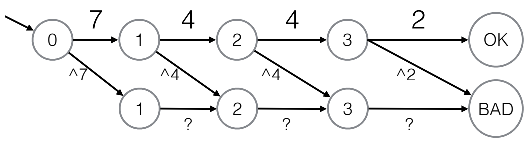
If the user enters the incorrect password three times, the account is locked, and a message is displayed.

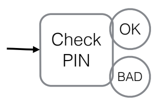
Here, assume that the correct password is 7442.

(a) Design a transducer-style finite state machine to model this. Describe the machine and explain clearly what the input and output alphabets mean, and what the set of states and transition function are. (You do not need to give a diagram including all the states and transitions, but your answer should make it clear what states there are and how the transition relation is constructed. It may be helpful to sketch parts of the FSM.) [*15 marks*]

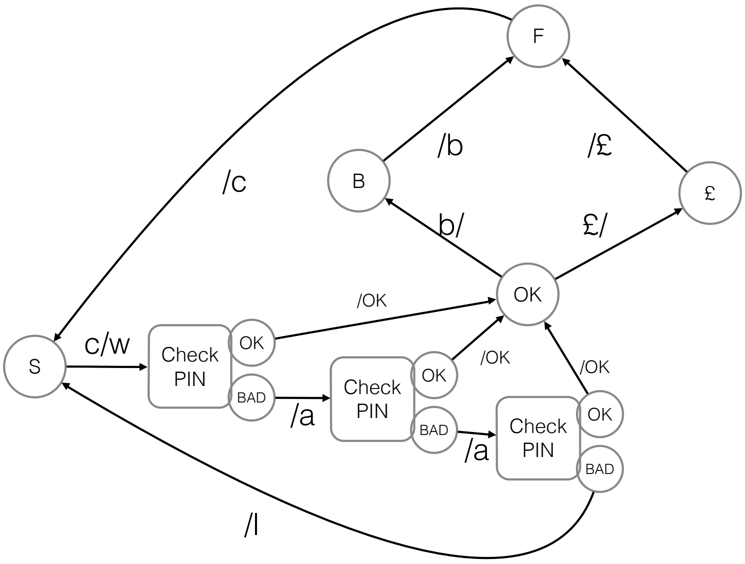
Your design should not let the user know whether the pin is correct until four digits have been entered. The user's card should be returned before any cash is dispensed.

First we give a machine for checking the 4-digit pin (we will use three of these)





Then we build ATM



- Alphabets:

input c – card; b – balance request; £– cash request; [0-9] – pin digits.

output w – welcome message; a – try again; OK – pin ok, present choice; b – print balance; £– issue cash; c – return card; l – lock account

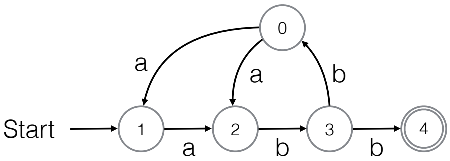
- States: Three copies of states in PIN recogniser. Other states as shown.

Note that the pin recogniser only uses input alphabet; $\hat{}d$ signifies any digit other than d.

(b) Give the trace of the machine corresponding to an interaction with a user who fails to enter the correct PIN on the first attempt, but then succeeds and makes a withdrawal. [*5 marks*]

c/w; 7/; 4/; 3/; 2/; /a; 7/; 4/; 4/; 2/; /ok; £/; /£; /c

6. (a) Which of the following strings are accepted by the NFA in the diagram? (The start state is indicated by an arrow and the accepting state by a double



border.)

i. abb yes

ii. abbabbabbaaabb no

iii. abbabbaabbabbabb yes

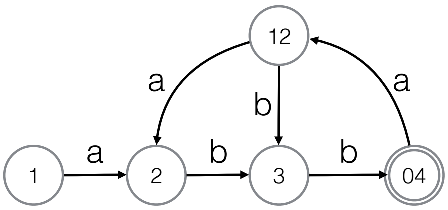
iv. abbabaabbabbabb no [*3 marks*]

(b) Write a regular expression for the language accepted by this NFA. [*3 marks*]

$a(bb(a \mid aa))^{*}bb$ or equivalent

(c) Draw a DFA that accepts the same language. Label the states of your DFA to make clear their relationship to the states of the original NFA. [*10 marks*]

The states of the DFA correspond to sets of states of the NFA. Black hole state (corresponding to the empty set) omitted by convention.



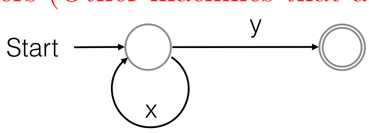
(d) For each of the following regular expressions, draw a non-deterministic finite state machine that accepts the language described by the regular expression.

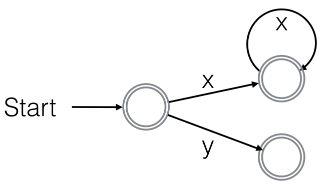
i. $x^{*}y$

ii. $(x^{*}|y)$

iii. $(x^{*}y)^{*}$ [*9 marks*]

Answers (Other machines that accept the same languages are OK):

(i) 

(ii) 

(iii)