Informatics 1A Computation and Logic
Guide to Standard Answers (and Marking Scheme)

Total marks for the assignment is 40. Points for each question are indicated after the solution is given.

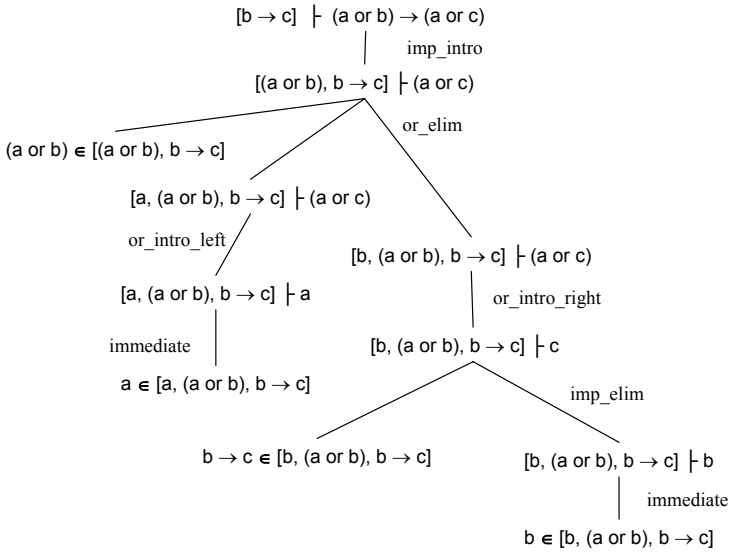
1. An appropriate truth table is:

a	b	$not(a)$	$not(b)$	$a \rightarrow b$	$a \rightarrow b \; and \; not(b)$	$(a \rightarrow b \; and \; not(b)) \rightarrow not(a)$
t	t	f	f	t	f	t
t	f	f	t	f	f	t
f	t	t	f	t	f	t
f	f	t	t	t	t	t

so the expression is a tautology.

(Marks: 2)

2. (a) An appropriate proof is:



(b) We are missing the counterpart, for conjunction, of the *and_elim* proof rule. In other words, we need a rule that can "unpack" conjunctive expressions in our set of axioms.

(c) A sound proof system permits no invalid proofs. A complete proof system permits all valid proofs.

(Marks: 6 - (a)4, (b)1, (c)1)

3. (a) Representing each circuit as a propositional expression gives:
   - Circuit 1: $not((not(a)\ or\ b)\ and\ not(b))\ or\ not(a)$
   - Circuit 2: $not(not(a\ and\ not(b))\ and\ not(b))\ or\ not(a)$

   (b) The truth table for both circuits is then as below:

a	b	not(a)	not(b)	not(a) or b	(not(a) or b) and not(b)	not((not(a) or b) and not(b))	not((not(a) or b) and not(b)) or not(a)	a and not(b)	not(a and not(b))	not(a and not(b)) and not(b)	not(not(a and not(b)) and not(b))	not(not(a and not(b)) and not(b)) or not(a)
t	**t**	**f**	**f**	**t**	**f**	**t**	**t**	**f**	**t**	**f**	**t**	**t**
t	**f**	**f**	**t**	**f**	**f**	**t**	**t**	**t**	**f**	**f**	**t**	**t**
f	**t**	**t**	**f**	**t**	**f**	**t**	**t**	**f**	**t**	**f**	**t**	**t**
f	**f**	**t**	**t**	**t**	**t**	**f**	**t**	**f**	**t**	**t**	**f**	**t**
							Circuit 1				Circuit 2	

   (c) The two expressions are equivalent. Both expressions are tautologous - they give the output "true" regardless of the inputs. So the engineer was right on both counts.

(Marks: 7 - (a)2, (b)4 (2 per truth table), (c)1)

4. Apply the equivalences in the following order:

	Start with $((p \; and \; q) \rightarrow r) \; and \; (p \rightarrow q)$	
$(p \; and \; q) \rightarrow r$	is equivalent to giving the formula	$not(p \; and \; q) \; or \; r$
	$(not(p \; and \; q) \; or \; r) \; and \; (p \rightarrow q)$	
	then	
$p \rightarrow q$	is equivalent to giving the formula	$not(p) \; or \; q$
	$(not(p \; and \; q) \; or \; r) \; and \; (not(p) \; or \; q)$	
	then	
$not(p \; and \; q)$	is equivalent to giving the formula	$not(p) \; or \; not(q)$
	$(not(p) \; or \; not(q) \; or \; r) \; and \; (not(p) \; or \; q)$	

Other orders of application are permitted as long as they are applied correctly and the result is correct.

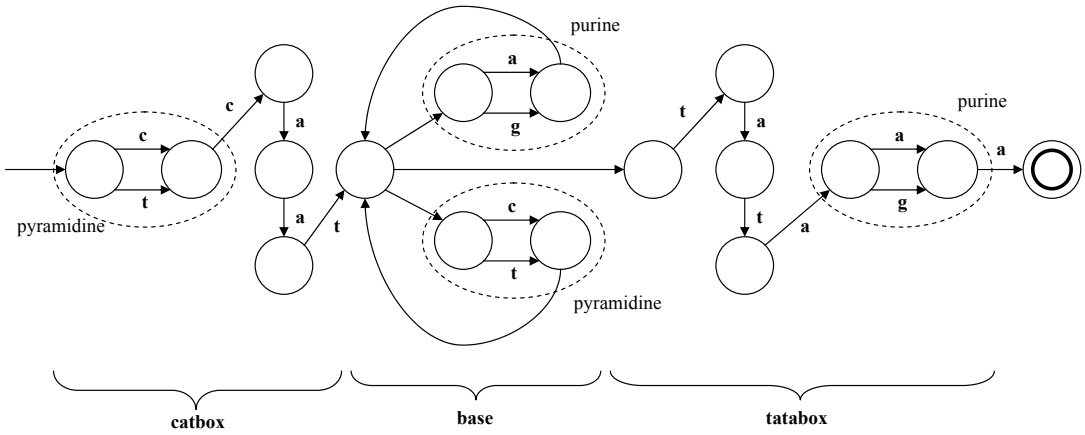
(Marks: 3 - 1 for each step)

5. The steps of inference are:

   - First negate b to give $not(b)$
   - Add this to the set to give [$[a]$, $[not(a), b]$ $[not(b)]$]
   - Resolve $[not(b)]$ with $[not(a), b]$ to give $[not(a)]$
   - Resolve $[not(a)]$ with $[a]$ to give []
   - [] means we have a contradiction so $not(b)$ is false
   - So b is true.

   (Marks: 4 - 1 for first two steps, 1 each for next two steps, and 1 for last two)

6. (a) An appropriate FSM is as follows:



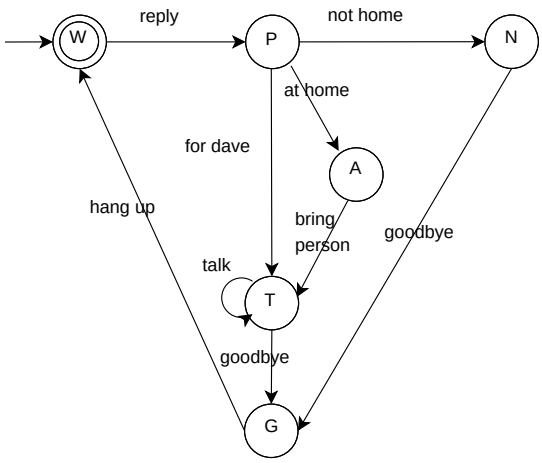
(b) This FSM can't be built because the base sequences can be of arbitrary length but are required to be maintained with equal numbers of the character "a". To enforce this constraint we would need some means of counting.

(Marks: 5 - (a)4 - 1pt for each part of gene (catbox, base, tatabox) and 1 for composition, (b)1)

7. (a) A basic set of appropriate states might be:

W	dave is waiting for a call
P	Dave has found out who the caller requests
N	The requested person is not at home
A	The requested person is at home
T	Someone is talking to the caller
G	Dave has said goodbye

(b) An example of the right sort of machine is as follows:



(c) It is deterministic if every arc leaving a state leads to a unique successor state. The answer should show whether or not this is true for the machine given.

(Marks: 4 - (a)1, (b)2, (c)1)

8. The basic method given in the lectures is this:

   - First find all traces through the FSM that would accept the target string, noting the probability on each transition. For the example, there are two accepting traces for `aabb`:

     ```
     1 a(0.6) 1 a(0.4) 2 b(0.5) 2 b(0.5) 3
     1 a(0.6) 1 a(0.6) 1 b(0.2) 1 b(0.8) 3
     ```

   - Then take the product of the probabilities on each trace:

     ```
     0.6 * 0.4 * 0.5 * 0.5 = 0.06
     0.6 * 0.6 * 0.2 * 0.8 = 0.0576
     ```

   - Then take the sum of the probabilities per trace: .6 + 0.576 = 0.1176*

   Variants on this calculation are acceptable as long as they make sense as calculations of probability. Students do not have to do the arithmetic - the equations suffice.

   (Marks: 5 - 2 for traces, 2 for 'and' eqs, 1 for 'or' eq)

9. (Marks: 4 - 2 for each sub-question)

(a)

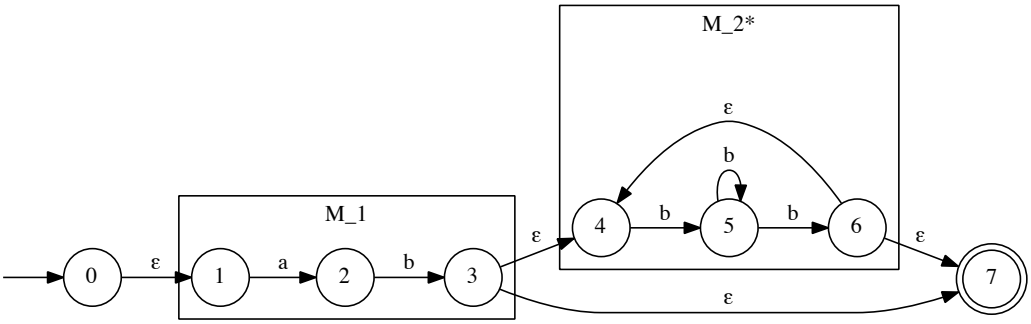


Figure 1: $M_1 M_2^*$

(b)

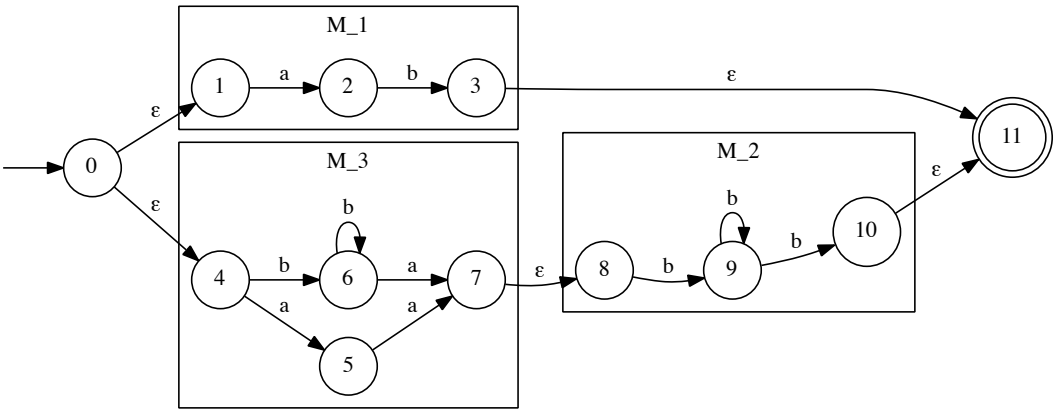


Figure 2: $M_1|(M_3 M_2)$