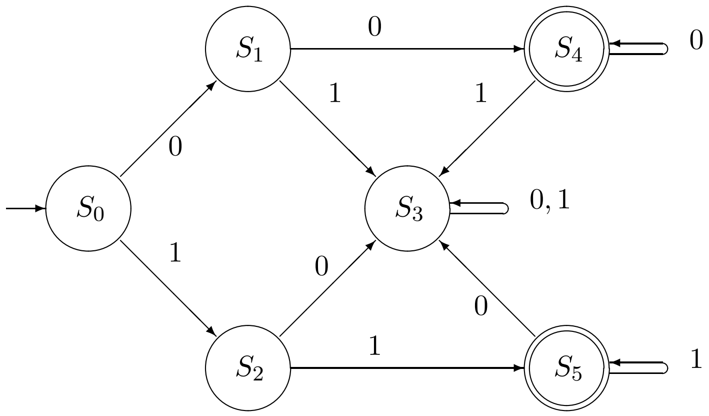
Department of Computer Science, Australian National University
COMP2600 — Formal Methods in Software Engineering
Semester 2, 2011

Week 6 Tutorial Solutions

Finite State Automata and their Languages

Question 1

The diagram for automaton A_1 is as follows:



1.1 Review of Notation

1. Easy.
2. S_3.
3. $L(A) \;=\; \{w \in \Sigma^* \mid (w = 0^n \;\vee\; w = 1^n) \;\wedge\; n \geq 2\}$

1.2 Proofs Involving Automata

1. The statement $\forall n \in \mathbb{N}.\ N^*(s_5, 1^n) = s_5$ says that any sequence of 1's leaves A_1 in the state s_5.

   It's the prime property of state s_5, one of the accepting states of the machine.

   Proof is by induction on n.

2. The statement $\forall w_1, w_2 \in \Sigma^*.\ N^*(s_0,\ w_1 01 w_2) \notin F$ says that any string containing an 0 followed immediately by a 1 will not be accepted.

   It is a sufficient condition for some strings to not be accepted. It may well be used in the proof of 1.1.3, above.
   The key lemmas in proving it are $\forall s \in S.\ N^*(s, 01) = s_3$ and $\forall w \in \Sigma^*.\ N^*(s_3, w) = s_3$.

3. The major proof obligations are, in English
   (a) If a word is in $L(A_1)$ then it is either a string of at least two 0's or a string of at least two 1's. *and*
   (b) Any string which consists of at least two 0's or consists of at least two 1's will be accepted by A_1.

   In mathematics these get written as:
   (1.2.3.1) $N^*(S_0, w) \in \{S_4, S_5\} \implies \exists n.\ n \geq 2 \land (w = 0^n \ \lor\ w = 1^n)$
   (1.2.3.2) $n \geq 2 \land (w = 0^n \ \lor\ w = 1^n) \implies N^*(S_0, w) \in \{S_4, S_5\}$

4. We can establish the proof obligations in 1.2.3 from the following simpler goals. These subgoals come from the above by case analysis.
   (1.2.4.1) $N^*(S_0, w) = S_4 \implies \exists n.\ n \geq 2 \land w = 0^n$
   (1.2.4.2) $N^*(S_0, w) = S_5 \implies \exists n.\ n \geq 2 \land w = 1^n$
   (1.2.4.3) $n \geq 2 \land w = 0^n \implies N^*(S_0, w) = S_4$
   (1.2.4.4) $n \geq 2 \land w = 1^n \implies N^*(S_0, w) = S_5$

Note: (1.2.4.4) is provable by induction on n and (1.2.4.3) is similarly easy.

Proof of (1.2.4.2) can be proved via lemmas:
(1.2.5.1) $N^*(S_5, w) = S_5 \implies \exists n.\ n \geq 0 \land w = 1^n$
(1.2.5.2) $N^*(S_2, w) = S_5 \implies \exists n.\ n \geq 1 \land w = 1^n$

An alternative approach to (1.2.3.1) might be to argue that the complement of the set given is $\{\epsilon, 0, 1\} \cup \{\alpha 01 \beta \mid \alpha, \beta \in \Sigma^*\} \cup \{\alpha 10 \beta \mid \alpha, \beta \in \Sigma^*\}$

Then, to show that none of these are accepted:

- easy for $\{\epsilon, 0, 1\}$
- for $\{\alpha 01 \beta \mid \alpha, \beta \in \Sigma^*\}$
  - show $N^*(s, 01) = S_3$ for all states s, by cases
  - $N^*(S_3, \beta) = S_3$ by induction on β
  -

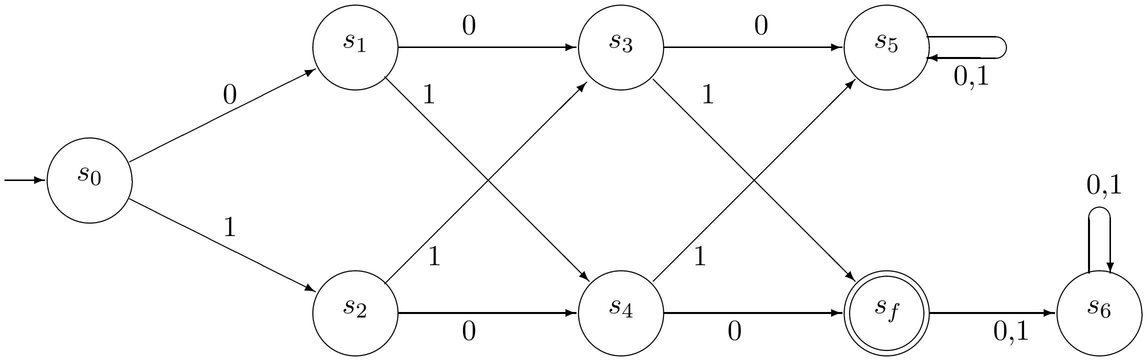
$$\begin{aligned} N^*(S_0, \alpha 01 \beta) &= N^*(N^*(S_0, \alpha), 01\beta) \\ &= N^*(N^*(N^*(S_0, \alpha), 01), \beta) \\ &= N^*(S_3, \beta) = S_3 \end{aligned}$$

- similarly $N^*(S_0, \alpha 10 \beta) = S_3$

That is, in words, from any state, 01 takes you to S_3, and once in S_3, you stay there

Question 2

a) This is the transition diagram for A_2:-



b) The language accepted by this automaton is the set of bitstrings of length three which have odd parity; it is the set $\{001, 010, 100, 111\}$.

2.1 Proofs of Properties

1. The main proof obligations are:
   (2.1.1.1) $N^*(S_0, w) = s_f \implies (w \in \{001, 010, 100, 111\}$
   (2.1.1.2) $(w \in \{001, 010, 100, 111\} \implies N^*(S_0, w) = s_f$

2. The subsubgoals for (2.1.1.2) are really easy:
   (2.1.2.1) $N^*(S_0, 001) = s_f$,
   (2.1.2.2) $N^*(S_0, 010) = s_f$,
   (2.1.2.3) $N^*(S_0, 100) = s_f$,
   (2.1.2.4) $N^*(S_0, 111) = s_f$.

3. A good subsubgoal for (2.1.1.1) are
   (2.1.3.1) $N^*(S_0, w) = s_f \implies$
   $((w = v0 \land N^*(S_0, v) = s_4) \lor (w = v1 \land N^*(S_0, v) = s_3))$
   This can be proved from
   (2.1.3.2) $N^*(S_0, w) = s_f \implies$
   $((w = v00 \land N^*(S_0, v) = s_2) \lor (w = v01 \land N^*(S_0, v) = s_1)$
   $\lor (w = v10 \land N^*(S_0, v) = s_1) \lor (w = v11 \land N^*(S_0, v) = s_2))$

2.2 Automaton Simplification

1. Same as above with s_6 merged into s_5.

2. The minimisation algorithm goes through the steps:

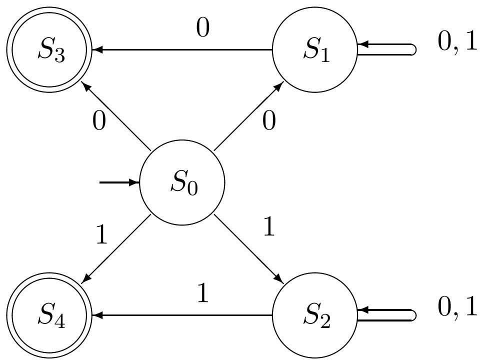
```
[[0,1,2,3,4,5,6],[f]]          initial split
[[4],[0,1,2,3,5,6],[f]]        testing with 0
```

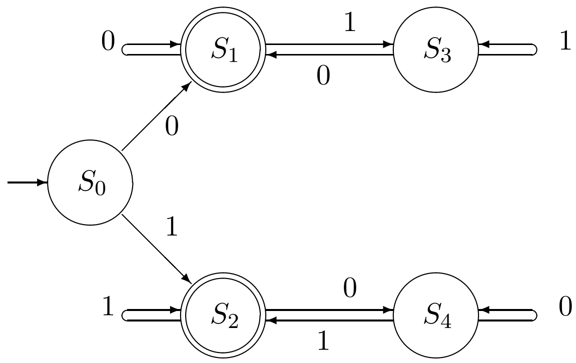
```
[[4],[1],[3],[0,2,5,6],[f]        testing with 1
[[4],[1],[3],[0],[2],[5,6],[f]]   testing with 0 again
subsequent tests with 0 or 1 produce no further split.
```

and so it shows s_5 and s_6 are euivalent.

Question 3

The following diagrams show the simplest solutions for the nondeterministic and deterministic cases, respectively.





Some students will probably discount the case where the input string has length one. Fair enough.

There is no advantage in terms of number of states or number of ordered pairs making up the relation/function. It may be argued that the states in the first have simpler descriptions.

Question 4

The problem is to show that the language $\{0^i1^j \mid gcd(i,j) = 1\}$ is not a regular language.

As with the standard proof that $\{a^nb^n\}$ is not regular, we rely on the fact that a finite number of states cannot be used to remember exactly how many zeroes occur before the first one.

A crucial thing to note is that if 0^i1^j is in the language then i and j are relatively prime. In what follows, we'll work with primes and eliminate any problems of factoring.

- Assume the language can be recognized by an automaton $(\{0,1\}, S, s_0, F, N)$.
- Consider the sequence $N^*(s_0, 00)$, $N^*(s_0, 000)$, $N^*(s_0, 0^5)$, $N^*(s_0, 0^7)$, $N^*(s_0, 0^{11})$ ..... All of these terms name states in S and by the pigeon hole principle they cannot be all distinct. There must be primes j and k such that $k > j$ and

$$N^*(s_0, 0^j) = N^*(s_0, 0^k) \tag{1}$$

- Since j and k are distinct primes $0^j 1^k$ is in the language; that is:

$$N^*(s_0, 0^j 1^k) \in F \tag{2}$$

- The basic property of N^* gives

$$N^*(s_0, 0^j 1^k) \;=\; N^*(N^*(s_0, 0^j), 1^k) \tag{3}$$

  which is a formula in which we can substitute equation 1 to get:

$$N^*(s_0, 0^j 1^k) \;=\; N^*(N^*(s_0, 0^k), 1^k) \;=\; N^*(s_0, 0^k 1^k) \tag{4}$$

- Plugging this into equation 2 we get

$$N^*(s_0, 0^k 1^k) \in F \tag{5}$$

  indicating that $0^k 1^k$ is in the language, which is a contradiction since $gcd(k, k) = k$.