

## Foundations of Computation

The practical contains a number of exercises designed for the students to practice the course content. During the practical session, the tutor will work through some of these exercises while students will be responsible for completing the remaining exercises in their own time. There is no expectation that all the exercises will be covered in the practical session.

Covers: Lecture Material Week 7

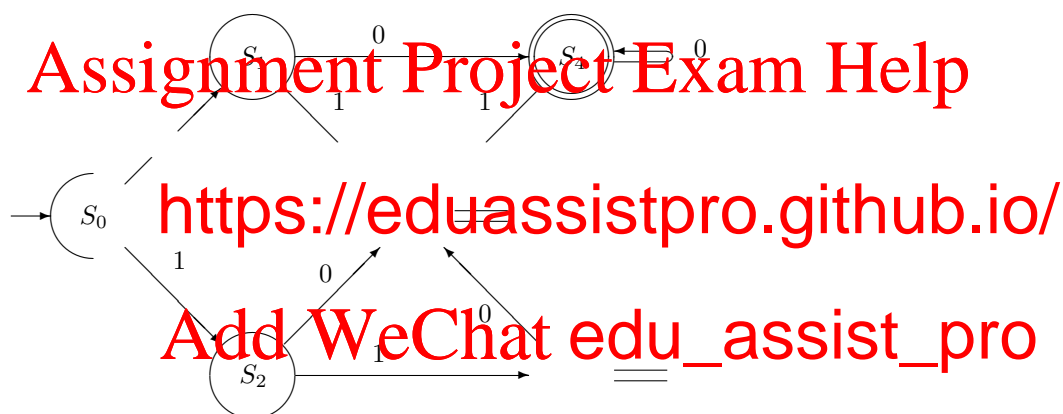
At the end of this tutorial, you will be able to

- design Deterministic Finite Automata given a language;
- determine whether a word is recognised by an automaton;
- describe the language that an automaton recognises.

### Exercise 1

### DFAs and Acceptance

The automaton  $A_1$  is specified by the following diagram.



1. Translate the graphical representation of  $A_1$  into the form of a 5-tuple,  $(\Sigma, S, s_0, F, N)$ .

**Solution.**

$$\Sigma = \{0, 1\};$$

$$S = \{S_0, S_1, S_2, S_3, S_4, S_5\};$$

$$s_0 = S_0;$$

$$F = \{S_4, S_5\};$$

$$N =$$

State	0	1
$S_0$	$S_1$	$S_2$
$S_1$	$S_4$	$S_3$
$S_2$	$S_3$	$S_5$
$S_3$	$S_3$	$S_3$
$S_4$	$S_4$	$S_3$
$S_5$	$S_3$	$S_5$

2. The next state function is  $N$  and so  $N^*$  is the eventual state function. What is  $N^*(S_2, 00010)$ ? Please show your solution step by step.

**Solution.**

$$\begin{aligned}
 N^*(S_2, 00010) &= N^*(N(S_2, 0), 0010) && \text{you can skip this step} \\
 &= N^*(S_3, 0010) \\
 &= N^*(S_3, 010) \\
 &= N^*(S_3, 10) \\
 &= N^*(S_3, 0) \\
 &= N^*(S_3, \epsilon) \\
 &= S_3
 \end{aligned}$$

3. What is the purpose of the state  $S_3$ ?

**Solution.**  $S_3$  is a ‘failure’ state; any string that enters  $S_3$  will stay in  $S_3$  and never be accepted.

4. What language does  $A_1$  accept? Phrase your answer in the form

$$L(A_1) = \{w \in \Sigma^* \mid P(w)\}$$

where  $P(w)$  is some predicate on words.

**Solution.**  $L(A) = \{w \in \Sigma^* \mid \exists n \in \mathbb{N}. (w = 0^n \vee w = 1^n) \wedge n \geq 2\}$ .

5. Consider the statement  $\forall n \in \mathbb{N}. N^*(S_5, 1^n) = S_5$ . Express it in English and give a proof. Which technique would you be using?

**Solution.** In English, this reads that “the automaton remains in state  $S_5$  if it processes an arbitrary long sequence of 1’s”. We prove this statement by induction on  $n$ .

For  $n = 0$ , we have that  $N^*(S, 1^0) = N^*(S, \epsilon) = S$  (by def. of  $N^*$ ). Now let  $n > 0$ . Then

$$\begin{aligned}
 &= N^*(N(S_5, 1), 1^{n-1}) \text{ (by def.} \\
 &= N^*(S_5, 1^{n-1}) \text{ (by def. of the tra} \\
 &= S_5 \text{ (by IH)}
 \end{aligned}$$

## Exercise 2

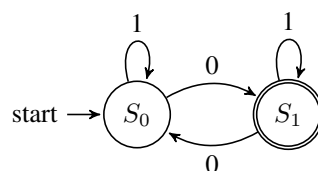
## Deterministic Finite Automata

Consider the alphabet  $\Sigma = \{0, 1\}$ .

1. Construct a DFA that recognises the language

$$\{w \in \Sigma^* \mid w \text{ has an odd number of 0's}\};$$

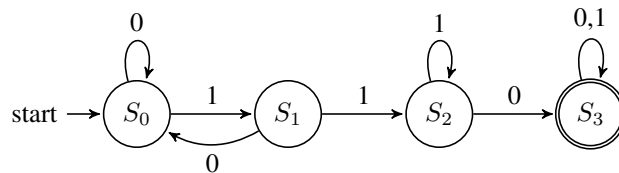
**Solution.**



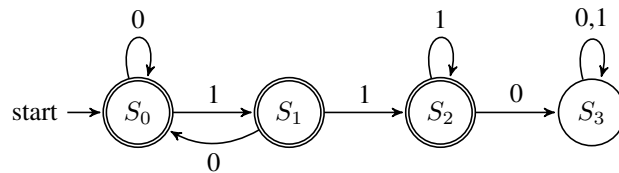
2. Construct a DFA that recognises the language

$$\{w \in \Sigma^* \mid w \text{ contains the substring } 110\};$$

**Solution.**



3. Construct a DFA that recognises the language  
 $\{w \in \Sigma^* \mid w \text{ does not contains the substring } 110\}$ ;  
**Solution.**



### Exercise 3

Let  $A_2$  be an automaton with next state function given in the following table. The initial state of the automaton is  $S_0$  and the only accepting state is  $S_f$ .

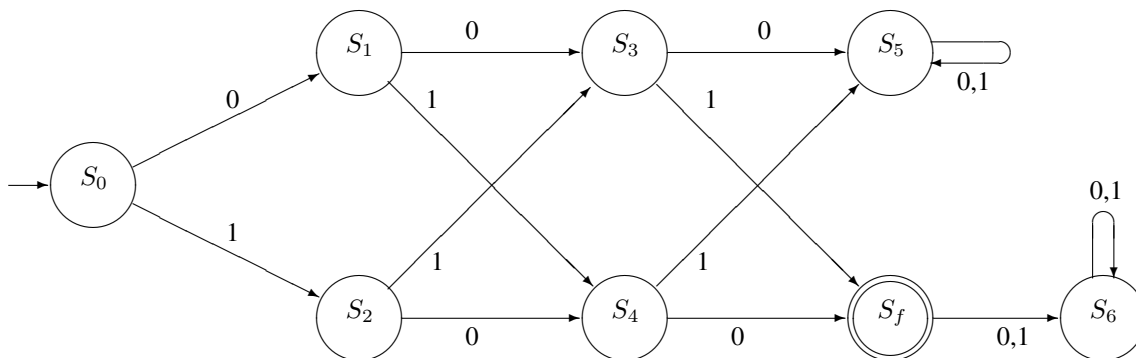
State	0	1
$S_0$	$S_1$	$S_2$
$S_1$	$S_3$	$S_4$
$S_2$	$S_4$	$S_3$
$f$	6	6
$S_6$	$S_6$	$S_6$

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

1. Draw the transition diagram for  $A_2$ .  
**Solution.**



2. Describe the language  $L(A_2)$  accepted by  $A_2$ .

**Solution.** The language accepted by this automaton is the finite set of strings of length equal to 3 over the alphabet  $\{0, 1\}$  with an odd number of 1's:  $\{001, 010, 100, 111\}$ .

3. State what you need to prove to establish that every word of the language you have described is accepted by  $A_2$ . (You do not need to give proofs.)

**Solution.** We have to show that  
 $N^*(S_0, 001) = S_f$ ,

$$\begin{aligned} N^*(S_0, 010) &= S_f, \\ N^*(S_0, 100) &= S_f, \\ N^*(S_0, 111) &= S_f. \end{aligned}$$

Give a proof of the fact that every word accepted by  $A_2$  is in the language you have described.

**Solution.** We first show that every word accepted by  $A_2$  has length exactly three. So consider that  $N^*(S_0, w) = S_f$ . Looking at the transition diagram, this can only happen if

- $w$  is of the form  $xd$  where  $d \in \{0, 1\}$  and  $x$  is a string,
- $N^*(S_0, x) \in \{S_3, S_4\}$ .

Again, if  $N^*(S_0, x) \in \{S_3, S_4\}$ , this can only happen if

- $x$  is of the form  $ye$  where  $e \in \{0, 1\}$  and  $y$  is a string,
- and  $N^*(S_0, y) \in \{S_1, S_2\}$

Again, the latter is only possible if  $y$  is a single letter  $y \in \{0, 1\}$  so that in total  $w = yed$  where all of  $y, e, d \in \{0, 1\}$  are single letters, so that  $w$  is of length three.

There are eight words of length three, and by tracing the transition diagram, we have that  $A_2$  accepts *exactly* the words in the language described above.

In summary: if  $w$  is accepted by  $A_2$ , then  $w$  is of length 3 and one of the words  $\{001, 010, 100, 111\}$  as the other length-3 words are *not* accepted.

#### Exercise 4

#### Non-Existence of Automata

Let  $L$  be the language of strings over  $\{0, 1\}$  that contain more 0's than 1's.

Prove that there is *no* finite state automaton that can recognise  $L$ .

**Solution.**

Suppose there exists a DFA  $A$

There are only finitely many different states,  $n, m$ , with  $n > m$ , such that

$$N^*(S_0, 0^n) = N^*(S_0, 0^m)$$

The string  $0^n 1^m$  contains more 0's than 1's, so it is in  $L$  and is accepted.

$$N^*(N^*(S_0, 0^n), 1^m) = N^*(S_0, 0^n 1^m) \in F$$

And therefore

$$N^*(S_0, 0^m 1^m) = N^*(N^*(S_0, 0^m), 1^m) = N^*(N^*(S_0, 0^n), 1^m) \in F$$

so  $0^m 1^m$  is accepted. But  $0^m 1^m \notin L$ , giving us a contradiction.

#### Exercise 5

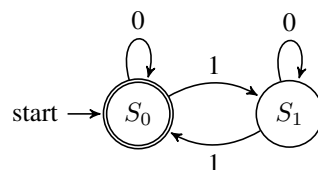
#### Deterministic Finite Automata

Consider the alphabet  $\Sigma = \{0, 1\}$ .

1. Construct a DFA that recognises the language

$$\{w \in \Sigma^* \mid w \text{ has an even number of 1's}\};$$

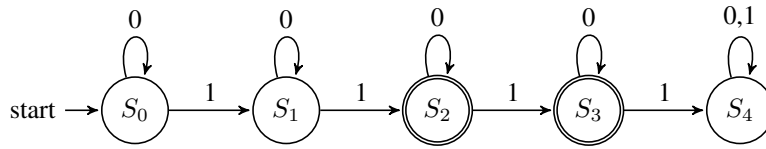
**Solution.**



2. Construct a DFA that recognises the language

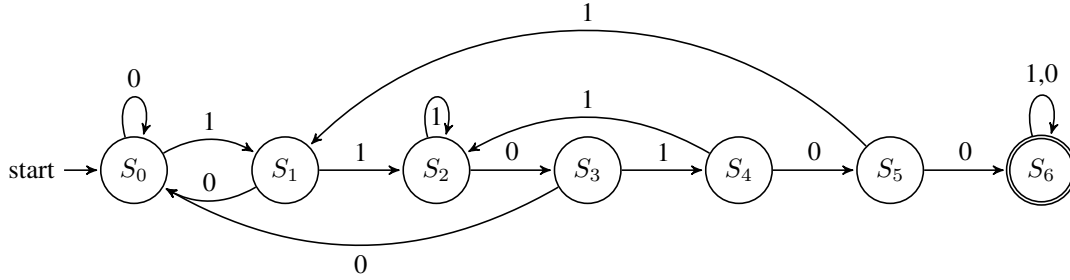
$$\{w \in \Sigma^* \mid w \text{ has two or three 1's}\};$$

**Solution.**



3. Construct a DFA that recognises the language  
 $\{w \in \Sigma^* \mid w \text{ contains the substring } 110100\}$ .

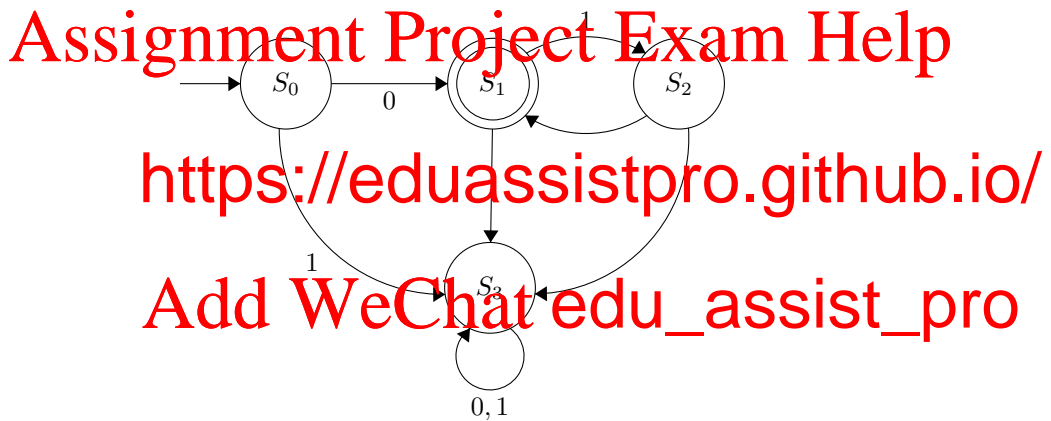
**Solution.**



### Exercise 6

### Formal Proofs of Automaton Languages I

Consider the following finite automaton  $B$ :



We would like to prove that the language  $L(B)$  that is accepted by  $B$  is equal to

$$L = \{0(11)^n : n \geq 0\}$$

Prove that  $L = L(B)$ . Ensure that you clearly state your two main proof obligations, and that the proof is given in full rigorous detail.

**Solution.** We first prove that  $B$  accepts all strings in  $L$ , that is,

$$x \in L \Rightarrow N^*(S_0, x) = S_1$$

Since all strings in  $L$  are of the form  $0(11)^n$  for some  $n \geq 0$ , we do a proof by induction on  $n$ .

Base case,  $n = 0$ .

$$N^*(S_0, 0(11)^0) = N^*(S_0, 0\epsilon) = N^*(S_0, 0) = S_1$$

Step case. Assume that  $N^*(S_0, 0(11)^n) = S_1$  and prove that  $N^*(S_0, 0(11)^{n+1}) = S_1$ .

$$N^*(S_0, 0(11)^{n+1}) = N^*(S_0, 011(11)^n) = N^*(S_1, 11(11)^n) = N^*(S_2, 1(11)^n) = N^*(S_1, (11)^n) = S_1$$

Where the last step is justified by since  $N^*(S_0, 0(11)^n) = S_1$  by assumption, and  $N^*(S_0, 0(11)^n) = N^*(S_1, (11)^n)$ , we have that  $N^*(S_1, (11)^n) = S_1$ .

Hence  $x \in L \Rightarrow N^*(S_0, x) = S_1$ .

We now prove the converse, that if a string is accepted by  $B$ , then it must be in  $L$ .

$$N^*(S_0, x) = S_1 \Rightarrow x \in L$$

It is easier instead to prove the contrapositive: If a string is not in  $L$ , then  $B$  rejects it.

$$x \notin L \Rightarrow N^*(S_0, x) \neq S_1$$

Now, what kinds of strings are not in  $L$ ? Well,  $L$  is the set of all strings that start with a zero, and are followed by an even number of ones. So strings that don't satisfy this condition must be either

- The empty string.
- Any string that starts with a one.
- Any string that starts with a zero, and is followed by another string that contains at least one zero.
- Any string that starts with a zero, and is followed by a string of odd many ones.

The first case is trivial, as  $N^*(S_0, \epsilon) = S_0 \neq S_1$ .

The second case is also trivial, as by observing the DFA we can see that  $S_3$  is a trap state, so  $N^*(S_3, \delta) = S_3$  for all strings  $\delta$ . Then,  $N^*(S_0, 1\delta) = N^*(S_3, \delta) = S_3 \neq S_1$

The third case requires some work, but we can use the append theorem to help us. A string of this form looks like  $0\alpha 0\beta$  for some strings  $\alpha, \beta$ . Then,

$$N^*(S_0, 0\alpha 0\beta) = N^*(S_1, \alpha 0\beta) = N^*(N^*(S_1, \alpha), 0\beta)$$

Now, we are not sure where  $N^*(S_1, \alpha)$  is, but we know that  $N^*(S_1, \alpha) \neq S_0$ , as there is no way to enter  $S_0$  from any other state. So we check all the combinations.

- If  $N^*(S_1, \alpha) = S_1$ , then  $N^*(N^*(S_1, \alpha), 0\beta) = N^*(S_1, 0\beta) = N^*(S_3, \beta) = S_3$ .
- If  $N^*(S_1, \alpha) = S_2$ , then
- If  $N^*(S_1, \alpha) = S_3$ , then

So in all cases, we end up in the non-final state  $S_3$

The fourth case is a string of the form  $0(11)^n 1$ , which is a zero followed by odd number of ones. We have already proven before, that  $N^*(S_0, 0(11)^n) = S_1$  together with the append theorem to obtain

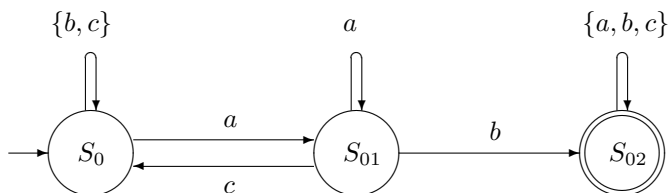
$$N^*(S_0, 0(11)^n 1) = N^*(N^*(S_0, 0(11)^n), 1) = N^*(S_1, 1) = S_3$$

All the cases are shown to be rejected by the machine, as required.

## Exercise 7

## Formal Proofs of Automaton Languages II

(Difficult.) Consider the following finite automaton



and prove that the DFA  $C$  recognises the language  $L = \{\gamma abd \mid \gamma, \delta \in \{a, b, c\}^*\}$ .

Ensure that you clearly state your two main proof obligations. Make sure that you give your proof in full rigorous detail. For example, be explicit about any use of the append theorem.

### Solution.

We need to show two things:

1.  $C$  accepts all strings  $\gamma ab\delta$ , for any  $\gamma, \delta \in \{a, b, c\}^*$ , i.e.,

$$N^*(S_0, \gamma ab\delta) = S_{02}$$

2. if  $C$  accepts a string  $w$ , i.e.,  $N^*(S_0, w) = S_{02}$ , then there exists  $\gamma, \delta \in \{a, b, c\}^*$  such that  $w = \gamma ab\delta$ .

We tackle both in turn.

1. We establish the following facts that we will use later.

**Lemma 1**

$$N^*(S_0, \gamma ab) = S_{02}$$

Proof: the left hand side equals  $N^*(N^*(S_0, \gamma), ab)$  by the append theorem. But we have no way of knowing what  $N^*(S_0, \gamma)$  is, so we consider every possibility:

$$N^*(S_0, ab) = S_{02}$$

$$N^*(S_{01}, ab) = S_{02}$$

$$N^*(S_{02}, ab) = S_{02}$$

$$\text{Hence } N^*(S_0, \gamma ab) = N^*(N^*(S_0, \gamma), ab) = N^*(S_{02}, ab) = S_{02}.$$

We next prove

**Lemma 2**

$$N^*(S_{02}, \delta) = S_{02}$$

Proof. This is an easy induction on the length of  $\delta$ . For the base case ( $\delta = \epsilon$ ) we have  $N^*(S_{02}, \epsilon) = S_{02}$  by definition of  $N^*$ . For the step case, we calculate

$$\begin{aligned} N^*(S_{02}, x\delta) &= N^*(N^*(S_{02}, x), \delta) && \text{(Def. of } N^*) \\ &= N^*(S_{02}, \delta) && \text{(Holds for all } x \in \{a, b, c\}) \end{aligned}$$

(IH)

We can now show that  $C$  is a function.

$$\begin{aligned} N^*(S_0, \gamma ab\delta) &= N^*(N^*(S_0, \gamma ab), \delta) && \text{(Append Theorem)} \\ &= N^*(S_{02}, \delta) && \text{(Lemma 1)} \\ &= S_{02} && \text{(Lemma 2)} \end{aligned}$$

2. As in the first part, we isolate some observations that we will use later.

**Lemma 3**

$$N^*(S_0, w) = S_{02} \Rightarrow \exists \gamma', \delta. (w = \gamma' b\delta \wedge N^*(S_0, \gamma') = S_{01})$$

Proof: by induction on the length of  $w$ . Base case,  $w = \epsilon$ , which follows because  $N^*(S_0, \epsilon) \neq S_{02}$ , so the LHS of the implication is false, and the implication is vacuously true. Inductive case, suppose  $N^*(S_0, wx) = S_{02}$ . Now  $N^*(S_0, wx) = N(N^*(S_0, w), x)$  by the corollary to the append theorem. We cannot know what  $N^*(S_0, w)$  is, but can eliminate one possibility: there is no  $x \in \{a, b, c\}$  such that  $N(S_0, x) = S_{02}$ , so  $N^*(S_0, w) \neq S_0$ . If  $N^*(S_0, w)$  were  $S_{01}$  then  $x$  can only be  $b$ , and so Lemma 3 holds by setting  $\gamma' = w$  and  $\delta = \epsilon$ . If  $N^*(S_0, w)$  were  $S_{02}$ , then by the IH  $w = \gamma' b\delta$  and  $N^*(S_0, \gamma') = S_{01}$ . Therefore  $w = \gamma' b(\delta x)$ , whatever  $x$  is. So Lemma 3 holds in this case too.

We next observe that

**Lemma 4**

$$N^*(S_0, w) = S_{01} \Rightarrow \exists \gamma. w = \gamma a$$

Proof: simply because all arcs into  $S_{01}$  are labelled by  $a$ .

We now show that any string accepted by  $C$  has necessarily the form  $w = \gamma ab\delta$  for some strings  $\gamma$  and  $\delta$ . So assume that  $C$  accepts  $w$ , that is,  $N^*(S_0, w) = S_{02}$ . By Lemma 3 there are strings  $\gamma'$  and  $\delta$  such that

$$w = \gamma' b\delta \text{ and } N^*(S_0, \gamma') = S_{01}.$$

Using Lemma 4 we now obtain  $\gamma$  such that  $\gamma' = \gamma a$ . Putting everything together, we have that  $w = \gamma ab\delta$  as required.