Sample solutions for the examination of Finite automata theory and formal languages (DIT321/TMV027) from 2019-08-21

Nils Anders Danielsson

1. (a) None.
   (b) S and A.
   (c) $L(G) = \{ a \}$.

2. The Turing machine is $(Q, \{ 0 \}, \Gamma, \delta, \mathit{even}, \sqcup, \{ \mathit{accept} \})$, where Q, Γ and δ are defined in the following way:

$$\begin{aligned} Q &= \{ \mathit{even}, \mathit{odd}, \mathit{accept} \} \\ \Gamma &= \{ 0, \sqcup \} \end{aligned}$$

$$\begin{aligned} &\delta \in Q \times \Gamma \rightharpoonup Q \times \Gamma \times \{ \mathsf{L}, \mathsf{R} \} \\ &\delta(\mathit{even}, \sqcup) = (\mathit{accept}, \sqcup, \mathsf{R}) \\ &\delta(\mathit{even}, 0) = (\mathit{odd}, 0, \mathsf{R}) \\ &\delta(\mathit{odd}, 0) \ \ = (\mathit{even}, 0, \mathsf{R}) \end{aligned}$$

   The machine always moves to the right. If it reads an even number of zeros and then a blank, then it accepts. If it reads an odd number of zeros and then a blank, then it rejects. The input alphabet is $\{ 0 \}$, so these two cases are exhaustive.

3. (a) The ε-NFA A corresponds to the following system of equations between languages, where e_0 corresponds to the start state s_0:

$$\begin{aligned} e_0 &= a(e_0 + e_2) + be_1 + e_1 \\ e_1 &= ae_2 \\ e_2 &= \varepsilon + be_1 \end{aligned}$$

   Let us solve for e_0. We can start by eliminating e_2:

$$\begin{aligned} e_0 &= a(e_0 + \varepsilon + be_1) + be_1 + e_1 \\ &= a + ae_0 + (ab + b + \varepsilon)e_1 \\ e_1 &= a(\varepsilon + be_1) = a + abe_1 \end{aligned}$$

Using Arden's lemma we get the unique solution $e_1 = (ab)^*a$. Let us now eliminate e_1:

$$\begin{aligned} e_0 &= a + ae_0 + (ab + b + \varepsilon)(ab)^*a \\ &= (\varepsilon + (ab + b + \varepsilon)(ab)^*)\, a + ae_0 \\ &= (ab + b + \varepsilon)(ab)^*a + ae_0 \end{aligned}$$

(The last step follows because ε is a member of the language generated by $(ab + b + \varepsilon)(ab)^*$.) Using Arden's lemma we get the unique solution

$$\begin{aligned} e_0 &= a^*(ab + b + \varepsilon)(ab)^*a \\ &= a^*(b + \varepsilon)(ab)^*a \end{aligned}$$

(where the last step follows because the language generated by a^*ab is contained in the language generated by a^*b).

Thus the regular expression $e = a^*(b+\varepsilon)(ab)^*a$ satisfies $L(e) = L(A)$.

(b) If the ε-NFA A is converted to a DFA using the subset construction (with inaccessible states omitted), then we obtain the following DFA (possibly with different names for the states):

	a	b
$\rightarrow \{ s_0, s_1 \}$	$\{ s_0, s_1, s_2 \}$	$\{ s_1 \}$
$* \{ s_0, s_1, s_2 \}$	$\{ s_0, s_1, s_2 \}$	$\{ s_1 \}$
$\{ s_1 \}$	$\{ s_2 \}$	$\emptyset$
$* \{ s_2 \}$	$\emptyset$	$\{ s_1 \}$
$\emptyset$	$\emptyset$	$\emptyset$

Let us now minimise this DFA. Note first that all of its states are accessible. If the algorithm from the course is used to find equivalent states, then we see that every state of this DFA is distinguishable from every other state. Thus the DFA is already minimal.

(c) The language $L(A)$ is equal to $L(a^*(b + \varepsilon)(ab)^*a)$. Thus the strings generated by A consist of zero or more a's, followed by zero or one b's, followed by zero or more occurrences of ab, and finally one a.

4. The two languages are equal.

- *One proof:* Let us first convert the DFA to a regular expression. The DFA corresponds to the following system of equations between languages, where e_a corresponds to the start state:

$$\begin{aligned} e_a &= 0e_b + (1 + 2)e_d \\ e_b &= \varepsilon + 0e_c + (1 + 2)e_d \\ e_c &= 0e_b + (1 + 2)e_c \\ e_d &= (0 + 1 + 2)e_d \end{aligned}$$

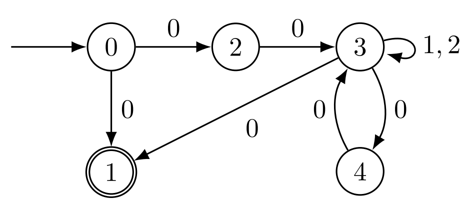
Using Arden's lemma we get the following (unique) solutions:

$$\begin{aligned}
e_d &= (0+1+2)^*\emptyset = \emptyset \\
e_c &= (1+2)^*0e_b \\
e_b &= (0(1+2)^*0)^*(\varepsilon + (1+2)\emptyset) = (0(1+2)^*0)^* \\
e_a &= 0(0(1+2)^*0)^* + (1+2)\emptyset = 0(0(1+2)^*0)^*
\end{aligned}$$

Let us now prove that e_a is equal to the language generated by e:

$$\begin{array}{ll}
0(0(1+2)^*0)^* & = \{\text{The shifting rule}\} \\
(00(1+2)^*)^*0 & = \{E^* = \varepsilon + EE^*\} \\
(\varepsilon + 00(1+2)^*(00(1+2)^*)^*)0 & = \{\text{The denesting rule}\} \\
(\varepsilon + 00(1+2+00)^*)0 & = \\
0 + 00(1+2+00)^*0 & = \\
0 + 00(00+1+2)^*0 &
\end{array}$$

- *An alternative proof (longer, but perhaps easier to come up with):* Let us start by converting the regular expression e to an ε-NFA B. Instead of using the algorithm from the course text book (which can yield rather large automata) I give B directly and prove that $L(B) = L(e)$. Here is B (its alphabet is $\{ 0, 1, 2 \}$):

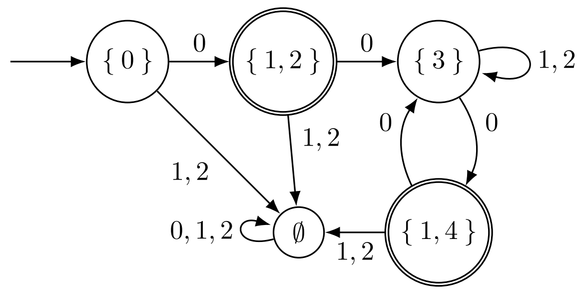
  

  This ε-NFA (without ε-transitions) corresponds to the following system of equations between languages, where e_0 corresponds to the start state:

  $$\begin{aligned}
  e_0 &= 0e_1 + 0e_2 \\
  e_1 &= \varepsilon \\
  e_2 &= 0e_3 \\
  e_3 &= 0e_1 + (1+2)e_3 + 0e_4 \\
  e_4 &= 0e_3
  \end{aligned}$$

  Using Arden's lemma we get the following (unique) solutions: $e_1 = \varepsilon$, $e_3 = (00+1+2)^*0$, $e_2 = 0(00+1+2)^*0$, and $e_0 = 0+00(00+1+2)^*0$. Because $e_0 = e$ we have $L(B) = L(e)$.

  If the ε-NFA B is converted to a DFA using the subset construction (with inaccessible states omitted), then we obtain the following DFA (possibly with different names for the states):



Let us denote this DFA by C. The algorithm from the course tells us that $L(C) = L(A)$, because the start states a and $\{ 0 \}$ are equivalent (✓ stands for "distinguishable"):

	a	b	c	d
$\{ 0 \}$		✓	✓	✓
$\{ 1,2 \}$	✓		✓	✓
$\{ 3 \}$	✓	✓		✓
$\{ 1,4 \}$	✓		✓	✓
$\emptyset$	✓	✓	✓	

Thus we get $L(e) = L(B) = L(C) = L(A)$.

5. (a) The grammar is $G = (\{ S \}, \{ 0,1 \}, P, S)$, where the set of productions P is defined by $S \to \varepsilon \mid 01S11$. See parts (b) and (c) for a proof showing that $L(G) = X$.

   (b) The property $X \subseteq L(G)$ follows from $\forall w \in X.\ w \in L(G,S)$. Let us prove the latter statement by induction on the structure of the string, seen as a member of X. We have two cases to consider:

   - The string w is ε. In this case we can construct the following derivation showing that $w \in L(G,S)$:

$$\dfrac{S \to \varepsilon \in P \qquad \dfrac{}{\varepsilon \in L^*(G,\varepsilon)}}{\varepsilon \in L(G,S)}$$

   - The string w is $01u11$ for some $u \in X$. The inductive hypothesis for u tells us that $u \in L(G,S)$. Thus we can construct a derivation of $w \in L(G,S)$ in the following way:

$$\dfrac{S \to 01S11 \in P \qquad \dfrac{\dfrac{\dfrac{u \in L(G,S) \qquad \dfrac{\dfrac{\dfrac{}{\varepsilon \in L^*(G,\varepsilon)}}{1 \in L^*(G,1)}}{11 \in L^*(G,11)}}{u11 \in L^*(G,S11)}}{1u11 \in L^*(G,1S11)}}{01u11 \in L^*(G,01S11)}}{01u11 \in L(G,S)}$$

(Antecedents of the form "a is a terminal" or "A is a nonterminal" are omitted from this and subsequent derivations.)

(c) The property $L(G) \subseteq X$ follows from $\forall w \in L(G, S).\ w \in X$. Let us prove this by complete induction on the length of the string. The derivation of $w \in L(G, S)$ must end in the following way:

$$\dfrac{S \to \alpha \in P \quad w \in L^*(G, \alpha)}{w \in L(G, S)}$$

There are two possibilities for α:

- $\alpha = \varepsilon$: In this case the derivation must end in the following way, and w must be equal to ε:

$$\dfrac{S \to \varepsilon \in P \quad \overline{\varepsilon \in L^*(G, \varepsilon)}}{\varepsilon \in L(G, S)}$$

  We have $w = \varepsilon \in X$.

- $\alpha = 01S11$: In this case the derivation must end in the following way, and w must be equal to $01u11$ for some $u \in L(G, S)$:

$$\dfrac{S \to 01S11 \in P \qquad \dfrac{\dfrac{\dfrac{u \in L(G,S) \quad \dfrac{\dfrac{\overline{\varepsilon \in L^*(G,\varepsilon)}}{1 \in L^*(G,1)}}{11 \in L^*(G,11)}}{u11 \in L^*(G,S11)}}{1u11 \in L^*(G,1S11)}}{01u11 \in L^*(G,01S11)}}{01u11 \in L(G, S)}$$

  Note that $|u| < |w|$. The inductive hypothesis thus implies that $u \in X$. We get that $w = 01u11 \in X$.

6. (a) Let us denote the language by L. It is equal to $M \cap \overline{N}$, where

$$M = \left\{\, ww^{\mathrm{R}} \;\middle|\; w \in \{\, 0, 1 \,\}^* \,\right\},$$
$$N = \left\{\, w \in \{\, 0, 1 \,\}^* \;\middle|\; |w| < 14 \,\right\},$$

and the complement is taken with respect to the language $\{\, 0, 1 \,\}^*$. The language M is context-free. N is regular, because it is finite (and a finite language $\{\, w_1, \dots, w_n \,\}$, where $n \in \mathbb{N}$, is regular because it is generated by the regular expression $w_1 + \cdots + w_n$). Furthermore $\overline{N}$ is regular, because the set of regular languages is closed under complementation. Thus we get that $L = M \cap \overline{N}$ is context-free, because the intersection of a context-free language and a regular language is context-free.

However, the language L is not regular. If it were regular, then the union of L and the finite language $M \cap N$ would be regular, but

$$L \cup (M \cap N) = (M \cap \overline{N}) \cup (M \cap N) = M,$$

and M is not regular.

(b) Let us denote the language by L. It is equal to $M \cap \overline{N}$, where

$$\begin{aligned} M &= \left\{ ww^{\mathrm{R}} \mid w \in \{ 0 \}^* \right\} \\ &= \left\{ ww \mid w \in \{ 0 \}^* \right\} \\ &= \{ 00 \}^*, \\ N &= \left\{ w \in \{ 0 \}^* \mid |w| < 14 \right\}, \end{aligned}$$

and the complement is taken with respect to the language $\{ 0 \}^*$. The language M is regular because it is generated by the regular expression $(00)^*$, and $\overline{N}$ is regular because it is the complement of a finite language. Thus we get that $L = M \cap \overline{N}$ is regular, because the intersection of two regular languages is regular.

Every regular language is context-free, so the language is also context-free.