Sample solutions for the examination of Finite automata theory and formal languages (DIT321/TMV027) from 2019-03-21

Nils Anders Danielsson

1. The DFA corresponds to the following system of equations between languages, where e_a corresponds to the start state:

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

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

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

   Thus one possible answer is $(0+1)(1(1+2)^*0)^*$.

2. The Turing machine is $(Q, \{ 0, 1 \}, \Gamma, \delta, \mathit{start}, \sqcup, \{ \mathit{accept} \})$, where Q, Γ and δ are defined in the following way:

$$\begin{aligned} &Q = \{ \mathit{start}, \mathit{skip\text{-}ones}, \mathit{check\text{-}space}, \mathit{accept} \} \\ &\Gamma\ = \{ 0, 1, \sqcup \} \\ \\ &\delta \in Q \times \Gamma \rightharpoonup Q \times \Gamma \times \{ \mathsf{L}, \mathsf{R} \} \\ &\delta(\mathit{start}, 0) &&= (\mathit{skip\text{-}ones}, 0, \mathsf{R}) \\ &\delta(\mathit{skip\text{-}ones}, 1) &&= (\mathit{skip\text{-}ones}, 1, \mathsf{R}) \\ &\delta(\mathit{skip\text{-}ones}, 0) &&= (\mathit{check\text{-}space}, 0, \mathsf{R}) \\ &\delta(\mathit{check\text{-}space}, \sqcup) &&= (\mathit{accept}, \sqcup, \mathsf{R}) \end{aligned}$$

   The machine always moves to the right, and it checks that the symbols that are encountered are a zero, an arbitrary number of ones, another zero, and a blank. In that case it accepts, and in any other case it rejects.

3. (a) If the BIN transformation is applied to G, then one of the possible results is the grammar

$$G_1 = (\{ S, S_1, A, B, B_1 \}, \{ a, b \}, P_1, S),$$

where the set of productions P_1 is defined in the following way:

$$\begin{aligned} S &\to AS_1 \\ S_1 &\to BA \\ A &\to \varepsilon \mid a \\ B &\to BB_1 \mid bb \\ B_1 &\to bB \end{aligned}$$

The grammar G_1 contains a single nullable nonterminal, A. If the DEL transformation is applied to G_1, then we obtain the grammar

$$G_2 = (\{ S, S_1, A, B, B_1 \}, \{ a, b \}, P_2, S),$$

where the set of productions P_2 is defined in the following way:

$$\begin{aligned} S &\to AS_1 \mid S_1 \\ S_1 &\to BA \mid B \\ A &\to a \\ B &\to BB_1 \mid bb \\ B_1 &\to bB \end{aligned}$$

The grammar G_2 contains three nontrivial unit pairs, (S, S_1), (S_1, B) and (S, B). If the UNIT transformation is applied to G_2, then we obtain the grammar

$$G_3 = (\{ S, S_1, A, B, B_1 \}, \{ a, b \}, P_3, S),$$

where the set of productions P_3 is defined in the following way:

$$\begin{aligned} S &\to AS_1 \mid BA \mid BB_1 \mid bb \\ S_1 &\to BA \mid BB_1 \mid bb \\ A &\to a \\ B &\to BB_1 \mid bb \\ B_1 &\to bB \end{aligned}$$

If the TERM transformation is applied to G_3, then one of the possible results is the grammar

$$G' = (\{ S, S_1, A, B, B_1, B_2 \}, \{ a, b \}, P', S),$$

where the set of productions P' is defined in the following way:

$$\begin{aligned} S &\to AS_1 \mid BA \mid BB_1 \mid B_2B_2 \\ S_1 &\to BA \mid BB_1 \mid B_2B_2 \\ A &\to a \\ B &\to BB_1 \mid B_2B_2 \\ B_1 &\to B_2B \\ B_2 &\to b \end{aligned}$$

The use of these four transformations, in this order, is guaranteed to produce a grammar G' in Chomsky normal form satisfying $L(G') = L(G) \setminus \{ \varepsilon \}$. In this case we have $\varepsilon \notin L(G)$ (note that S is not nullable), so $L(G') = L(G)$.

(b) The CYK table:

$\emptyset$			
$\{ S \}$	$\{ B_1 \}$		
$\emptyset$	$\{ S, S_1, B \}$	$\{ S, S_1, B \}$	
$\{ A \}$	$\{ B_2 \}$	$\{ B_2 \}$	$\{ B_2 \}$
a	b	b	b

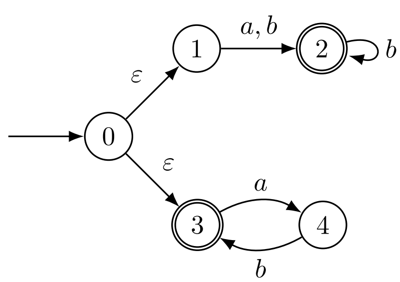
(c) We have $abbb \in L(G)$ if the start symbol S is a member of the CYK table's "topmost" cell. In this case it is not, so $abbb \notin L(G)$.

4. (a) Either an a followed by zero or more b's, or zero or more repetitions of ab, or a b followed by zero or more b's.

(b) First note that

$$\begin{aligned} e &= ab^* + (ab)^* + bb^* \\ &= (a+b)b^* + (ab)^*. \end{aligned}$$

Let us convert the regular expression $e' = (a+b)b^* + (ab)^*$ to an ε-NFA A. Instead of using the algorithm from the course text book (which can yield rather large automata) I give A directly and prove that $L(A) = L(e')$. Here is A (its alphabet is $\{ a, b \}$):

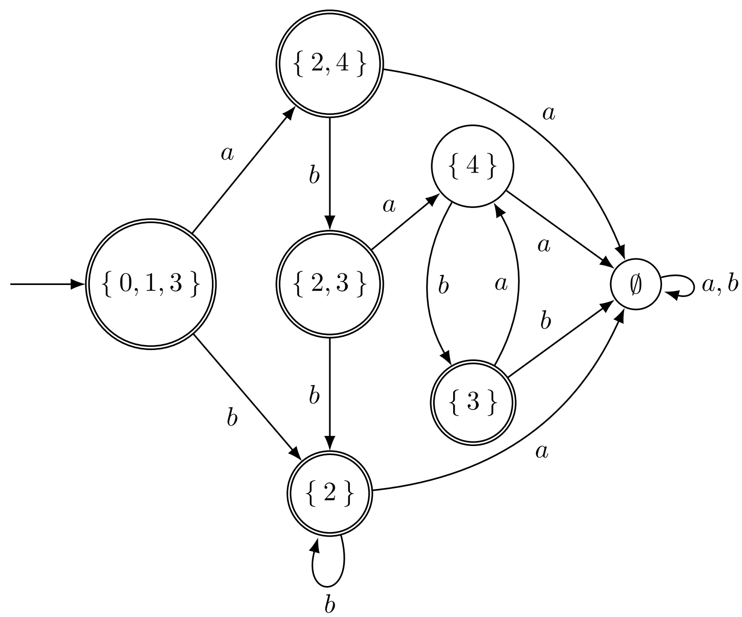


This ε-NFA corresponds to the following system of equations between languages, where e_0 corresponds to the start state:

$$\begin{aligned} e_0 &= e_1 + e_3 \\ e_1 &= (a+b)e_2 \\ e_2 &= \varepsilon + be_2 \\ e_3 &= \varepsilon + ae_4 \\ e_4 &= be_3 \end{aligned}$$

Using Arden's lemma we get the following (unique) solutions: $e_3 = (ab)^*$, $e_2 = b^*$, $e_1 = (a+b)b^*$ and $e_0 = (a+b)b^* + (ab)^*$. Because $e_0 = e'$ we have $L(A) = L(e')$.

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):



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.

5. (a) The grammar is $G = (\{ S \}, \{ 0, 1 \}, P, S)$, where the set of productions P is defined by $S \to 0 \mid S1S$. 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 0. In this case we can construct the following derivation showing that $w \in L(G, S)$:[1]

$$\frac{S \to 0 \in P \qquad \dfrac{\dfrac{}{\varepsilon \in L^*(G, \varepsilon)}}{0 \in L^*(G, 0)}}{0 \in L(G, S)}$$

[1] Antecedents of the form "a is a terminal" or "A is a nonterminal" are omitted from this and subsequent derivations.

- The string w is $u1v$ where $u, v \in X$. The inductive hypothesis for u tells us that $u \in L(G,S)$, and similarly $v \in L(G,S)$. Thus we can construct a derivation of $w \in L(G,S)$ in the following way:

$$\dfrac{S \to S1S \in P \qquad \dfrac{u \in L(G,S) \qquad \dfrac{\dfrac{v \in L(G,S) \qquad \dfrac{}{\varepsilon \in L^*(G,\varepsilon)}}{v \in L^*(G,S)}}{1v \in L^*(G,1S)}}{u1v \in L^*(G,S1S)}}{u1v \in L(G,S)}$$

(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:

- $\alpha = 0$: In this case the derivation must end in the following way, and w must be equal to 0:

$$\dfrac{S \to 0 \in P \quad \dfrac{\dfrac{}{\varepsilon \in L^*(G,\varepsilon)}}{0 \in L^*(G,0)}}{0 \in L(G,S)}$$

  We have $w = 0 \in X$.
- $\alpha = S1S$: In this case the derivation must end in the following way, and w must be equal to $u1v$:

$$\dfrac{S \to S1S \in P \qquad \dfrac{u \in L(G,S) \qquad \dfrac{\dfrac{v \in L(G,S) \qquad \dfrac{}{\varepsilon \in L^*(G,\varepsilon)}}{v \in L^*(G,S)}}{1v \in L^*(G,1S)}}{u1v \in L^*(G,S1S)}}{u1v \in L(G,S)}$$

  Note that $u, v \in L(G,S)$, and furthermore $|u| < |w|$ and $|v| < |w|$. The inductive hypothesis thus implies that $u \in X$ and $v \in X$. We get that $w = u1v \in X$.

6. (a) The language is equal to $\overline{M} \cap \overline{N}$, where

$$\begin{aligned} M &= \left\{\, w \in \{\,0,1\,\}^* \;\middle|\; |w| < 7 \,\right\}, \\ N &= \left\{\, w \in \{\,0,1\,\}^* \;\middle|\; \exists u, v \in \{\,0,1\,\}^*.\ w = u11v \,\right\} \\ &= \{\,0,1\,\}^* \{\,11\,\} \{\,0,1\,\}^*, \end{aligned}$$

and the complements are taken with respect to the language $\{ 0, 1 \}^*$. M is regular, because it is finite (and a finite language $\{ w_1, ..., w_n \}$, where $n \in \mathbb{N}$, is regular because it is generated by the regular expression $w_1 + \cdots + w_n$). N is regular, because it is generated by the regular expression $(0+1)^*11(0+1)^*$. Finally the set of regular languages is closed under complementation and intersection, so $\overline{M} \cap \overline{N}$ is regular.

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

(b) The language is finite, so it is regular, and thus also context-free.