Sample solutions for the examination of Finite automata and formal languages (DIT322/TMV028) from 2021-08-18

Nils Anders Danielsson

Note that in some cases I have not explained "step by step" why a certain algorithm produces a certain result, even though students who took the exam were asked to do this.

1. (a) Yes, the right-hand side of every production contains either a single terminal or exactly two nonterminals.

   (b) The CYK table:

$\{ S \}$		
$\{ S \}$	$\{ B \}$	
$\{ A \}$	$\{ B \}$	$\{ B \}$
a	b	b

   (c) There is no such parse tree. Here is the CYK table for the string *aba*:

$\emptyset$		
$\{ S \}$	$\emptyset$	
$\{ A \}$	$\{ B \}$	$\{ A \}$
a	b	a

   If there were a parse tree for *aba* in $P(G, S)$, then the topmost cell in the table would have contained the nonterminal S. However, this cell is empty.

2. The following context-free grammar defines the given language:

$$G = (\{ S \}, \{ 0, 1 \}, (S \to 0S1 \mid \varepsilon), S)$$

   We can convert this grammar to a pushdown automaton A such that $N(A) = L(G)$. If we follow the technique presented in the course, then we get the following automaton:

$$A = (\{ q \}, \{ 0, 1 \}, \{ S, 0, 1 \}, \delta, q, S, \{ q \})$$

$$\begin{aligned}
&\delta \in \{ q \} \times \{ \varepsilon, 0, 1 \} \times \{ S, 0, 1 \} \to \wp(\{ q \} \times \{ S, 0, 1 \}^*) \\
&\delta(q, \varepsilon, S) \;\, = \{ (q, 0S1), (q, \varepsilon) \} \\
&\delta(q, 0, 0) \;\; = \{ (q, \varepsilon) \} \\
&\delta(q, 1, 1) \;\; = \{ (q, \varepsilon) \} \\
&\delta(q, _, _) = \emptyset
\end{aligned}$$

3. All three regular expressions denote the same language as 1^*:

   - $L(e_1) = L(1^*)$:

$$\begin{aligned}
&(1^*)^* = \{ (e^*)^* = e^* \} \\
&1^*
\end{aligned}$$

   - $L(e_2) = L(1^*)$:

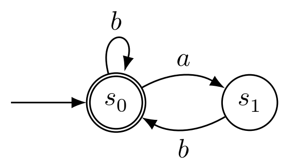
$$\begin{aligned}
&(1^*)^+ &&= \{ \text{By definition} \} \\
&1^*(1^*)^* &&= \{ (e^*)^* = e^* \} \\
&1^*1^* &&= \{ e^*e^* = e^* \} \\
&1^*
\end{aligned}$$

   - $L(e_3) = L(1^*)$:

$$\begin{aligned}
&(1^+)^* &&= \{ \text{By definition} \} \\
&(11^*)^* &&= \{ e^* = \varepsilon + ee^* \} \\
&\varepsilon + 11^*(11^*)^* &&= \{ \text{Shifting} \} \\
&\varepsilon + 1(1^*1)^*1^* &&= \{ \text{Denesting} \} \\
&\varepsilon + 1(1 + 1)^* &&= \{ \text{Idempotence} \} \\
&\varepsilon + 11^* &&= \{ e^* = \varepsilon + ee^* \} \\
&1^*
\end{aligned}$$

4. (a) First note that, for any regular expressions e_1 and e_2, $L((e_1^* + e_2)^*) = L((e_1 + e_2)^*)$. $L(e_1^*) \supseteq L(e_1)$, so $L((e_1^* + e_2)^*) \supseteq L((e_1 + e_2)^*)$. To see that $L((e_1^* + e_2)^*) \subseteq L((e_1 + e_2)^*)$, take a string $w \in L((e_1^* + e_2)^*)$. There must be a natural number n and strings $w_1, ..., w_n \in L(e_1^* + e_2)$ such that $w = w_1 \cdots w_n$. For each string w_i ($i \in \{ 1, ..., n \}$) it must either be the case that $w_i \in L(e_2)$, or $w_i \in L(e_1^*)$. In the latter case there must be some natural number k_i and strings $w_{i,1}, ..., w_{i,k_i} \in L(e_1)$ such that $w_i = w_{i,1} \cdots w_{i,k_i}$. Thus w can be expressed as a finite sequence of strings in $L(e_1)$ and $L(e_2)$, which means that $w \in L((e_1 + e_2)^*)$.

   Let us now convert the regular expression $e' = (ab + b)^*$ to an ε-NFA A. Instead of using the algorithm from the course text book I give A directly and prove that $L(A) = L(e')$ (by the argument above $L(e') = L(e)$). 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 s_0:

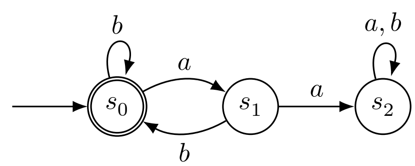
$$\begin{aligned} e_0 &= \varepsilon + ae_1 + be_0 \\ e_1 &= be_0 \end{aligned}$$

Let us find a solution for e_0. We can start by eliminating e_1:

$$e_0 = \varepsilon + (ab + b)e_0$$

Using Arden's lemma we then get the following unique solution: $e_0 = (ab + b)^*$. Because $e_0 = e'$ we have $L(A) = L(e') = L(e)$.

(b) The ε-NFA A does not make any use of non-determinism or ε-transitions, so it can be converted directly to a DFA B:



The DFA B is minimal: all states are accessible, the state s_0 is distinguishable from the other two (because it is the only accepting one), and s_1 is distinguishable from s_2 (they are distinguished by the b-transition, because s_0 and s_2 are distinguishable).

(c) The language $L(e) = L((ab + b)^*)$ consists of exactly those strings of zero or more a's and b's for which every a is immediately followed by a b.

5. (a) The language L is defined mutually with the language L' in the following way:

$$\frac{u \in L \quad v \in L'}{uav \in L} \qquad \frac{}{b \in L} \qquad \frac{v \in L}{va \in L'}$$

See parts (b) and (c) for a proof showing that $L = L(G)$.

(b) The property $L \subseteq L(G)$ follows from $\forall w \in L.\ w \in L(G, S)$, which is proved mutually with $\forall w \in L'.\ w \in L(G, A)$ by induction on the structure of L and L'. We have three cases to consider:

- $\dfrac{}{b \in L}$: In this case we should prove that $b \in L(G, S)$. We can construct the following derivation:

$$\dfrac{\dfrac{}{S \to b \in P} \qquad \dfrac{\dfrac{}{\varepsilon \in L_{\mathrm{L}}(G, \varepsilon)}}{b \in L_{\mathrm{L}}(G, b)}}{b \in L(G, S)}$$

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

- $\dfrac{u \in L \quad v \in L'}{uav \in L}$: In this case we should prove that $uav \in L(G, S)$, given the inductive hypotheses that $u \in L(G, S)$ and $v \in L(G, A)$. We can construct the following derivation:

$$\dfrac{\dfrac{}{S \to SaA \in P} \qquad \dfrac{u \in L(G, S) \qquad \dfrac{\dfrac{v \in L(G, A) \qquad \dfrac{}{\varepsilon \in L_{\mathrm{L}}(G, \varepsilon)}}{v \in L_{\mathrm{L}}(G, A)}}{av \in L_{\mathrm{L}}(G, aA)}}{uav \in L_{\mathrm{L}}(G, SaA)}}{uav \in L(G, S)}$$

- $\dfrac{v \in L}{va \in L'}$: In this case we should prove that $va \in L(G, A)$, given the inductive hypothesis that $v \in L(G, S)$. We can construct the following derivation:

$$\dfrac{\dfrac{}{A \to Sa \in P} \qquad \dfrac{v \in L(G, S) \qquad \dfrac{\dfrac{}{\varepsilon \in L_{\mathrm{L}}(G, \varepsilon)}}{a \in L_{\mathrm{L}}(G, a)}}{va \in L_{\mathrm{L}}(G, Sa)}}{va \in L(G, A)}$$

(c) The property $L(G) \subseteq L$ follows from $\forall w \in L(G, S).\ w \in L$, which is proved mutually with $\forall w \in L(G, A).\ w \in L'$ by complete induction on the lengths of the strings. A derivation of $w \in L(G, S)$ must end in the following way:

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

There are two possibilities for α:

- $\alpha = b$: In this case we have the following derivation, and w is equal to b:

$$\dfrac{\dfrac{}{S \to b \in P} \qquad \dfrac{\dfrac{}{\varepsilon \in L_{\mathrm{L}}(G, \varepsilon)}}{b \in L_{\mathrm{L}}(G, b)}}{b \in L(G, S)}$$

We can easily construct a derivation showing that $w = b \in L$:

$$\frac{}{b \in L}$$

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

$$\dfrac{\dfrac{}{S \to SaA \in P} \qquad \dfrac{u \in L(G,S) \qquad \dfrac{\dfrac{v \in L(G,A) \quad \dfrac{}{\varepsilon \in L_{\mathrm{L}}(G,\varepsilon)}}{v \in L_{\mathrm{L}}(G,A)}}{av \in L_{\mathrm{L}}(G,aA)}}{uav \in L_{\mathrm{L}}(G,SaA)}}{uav \in L(G,S)}$$

  Note that $|u| < |w|$ and $|v| < |w|$. The inductive hypotheses thus imply that $u \in L$ and $v \in L'$. We can now construct a derivation showing that $w = uav \in L$:

$$\frac{u \in L \quad v \in L'}{uav \in L}$$

A derivation of $w \in L(G, A)$ must end in the following way:

$$\frac{A \to \alpha \in P \quad w \in L_{\mathrm{L}}(G,\alpha)}{w \in L(G,A)}$$

Here α must be equal to Sa, the derivation must end in the following way, and w must be equal to ua for some $u \in L(G, S)$:

$$\dfrac{\dfrac{}{A \to Sa \in P} \qquad \dfrac{u \in L(G,S) \qquad \dfrac{\varepsilon \in L_{\mathrm{L}}(G,\varepsilon)}{a \in L_{\mathrm{L}}(G,a)}}{ua \in L_{\mathrm{L}}(G,Sa)}}{ua \in L(G,A)}$$

Note that $|u| < |w|$. One of the inductive hypotheses thus implies that $u \in L$. We can now construct a derivation showing that $w = ua \in L'$:

$$\frac{u \in L}{ua \in L'}$$

6. M is not context-free (and thus also not regular). If M were context-free, then, by the results taken from "Quotients of Context-Free Languages" by Ginsburg and Spanier, $M' = \left\{\, wcw \;\middle|\; w \in \{\, a, b \,\}^* \,\right\}$ would also be context-free. Now consider the following function:

$$\begin{aligned}
&h \in \{\, a, b, c \,\} \to \{\, a, b \,\}^* \\
&h(a) = a \\
&h(b) = b \\
&h(c) = \varepsilon
\end{aligned}$$

The set of context-free languages is closed under string homomorphisms, so if M' were context-free, then $h(M') = \left\{ ww \mid w \in \{ a, b \}^* \right\}$ would also be context-free. However, this language is not context-free.

N is regular (and thus also context-free). In fact, $N = \{ a, b, c \}^*$ (which is regular). First note that $N \subseteq \{ a, b, c \}^*$, because no string in N contains any symbol other than a, b or c. Let us now prove that $\{ a, b, c \}^* \subseteq N$. Take a string $w \in \{ a, b, c \}^*$. We should prove that $w \in N$. We can do this by induction on the number of occurrences of c in w. We have two cases:

- 0: In this case $w \in \{ a, b \}^*$, so $w \in N$ by the second rule defining N.
- $1 + n$: In this case $w = ucv$, where $u \in \{ a, b \}^*$, $v \in \{ a, b, c \}^*$, and the number of occurrences of c in v is n. By the second rule we get that $u \in N$, and by the inductive hypothesis we get that $v \in N$. Thus, by the third rule, $w = ucv \in N$.