Sample solutions for the examination of Finite automata and formal languages (DIT321/DIT322/TMV027/TMV028) from 2021-03-18

Nils Anders Danielsson

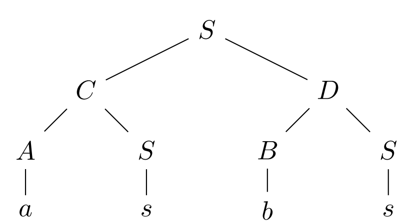
Note that in some cases I have not explained "step by step" why a certain algorithm produces a certain result, even though students who took the exam were asked to do this.

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

   (b) The CYK table:

$\{S\}$			
$\emptyset$	$\emptyset$		
$\{C\}$	$\emptyset$	$\{D\}$	
$\{A\}$	$\{S\}$	$\{B\}$	$\{S\}$
a	s	b	s

   (c) A parse tree:



2. We have that $L(e_1) \neq L(e_2)$, because $0 \in L(e_1)$, but $0 \notin L(e_2)$ (every string in $L(e_2)$ contains the symbol 1). Furthermore $L(e_1) = L(e_3)$ (and thus $L(e_2) \neq L(e_3)$):

$$
\begin{aligned}
& (01^*)^+ && = \{ \text{By definition} \} \\
& 01^*(01^*)^* && = \{ e^*e^* = e^* \} \\
& 01^*1^*(01^*)^* && = \{ \text{Shifting} \} \\
& 01^*(1^*0)^*1^* && = \{ \text{Denesting} \} \\
& 01^*(1^* + 0)^* && = \{ \text{Commutativity} \} \\
& 01^*(0 + 1^*)^*
\end{aligned}
$$

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:

$$\begin{aligned} e_0 &= \varepsilon + ae_1 \\ e_1 &= \varepsilon e_0 + be_2 \\ e_2 &= \varepsilon + \varepsilon e_0 + ae_3 \\ e_3 &= be_2 \end{aligned}$$

Let us find a solution for e_0. We can start by eliminating e_3:

$$\begin{aligned} e_0 &= \varepsilon + ae_1 \\ e_1 &= \varepsilon e_0 + be_2 \\ e_2 &= \varepsilon + \varepsilon e_0 + abe_2 \end{aligned}$$

Let us now eliminate e_2. Using Arden's lemma we get the unique solution $e_2 = (ab)^*(\varepsilon + \varepsilon e_0) = (ab)^*(\varepsilon + e_0)$:

$$\begin{aligned} e_0 &= \varepsilon + ae_1 \\ e_1 &= \varepsilon e_0 + b(ab)^*(\varepsilon + e_0) \end{aligned}$$

We can now eliminate e_1:

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

Using Arden's lemma we finally get the following unique solution:

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

The penultimate step follows because

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

and

$$\begin{aligned} &\big(a + (ab)^+\big)^*\big(\varepsilon + (ab)^+\big) &&\subseteq \\ &\big(a + (ab)^+\big)^*\big(\varepsilon + a + (ab)^+\big) &&\subseteq \\ &\big(a + (ab)^+\big)^*\Big(\varepsilon + \big(a + (ab)^+\big)^*\Big) &&= \\ &\big(a + (ab)^+\big)^*\big(a + (ab)^+\big)^* &&= \\ &\big(a + (ab)^+\big)^*. \end{aligned}$$

The last step follows because $ab \subseteq (ab)^+$ and

$$\begin{aligned} &\big(a + (ab)^+\big)^* &&\subseteq \\ &\big(a^* + (ab)^*\big)^* &&\subseteq \\ &\big((a + ab)^*\big)^* &&= \\ &(a + ab)^*. \end{aligned}$$

We get that the regular expression $e = (a+ab)^*$ 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
$\rightarrow * \{ s_0 \}$	$\{ s_0, s_1 \}$	$\emptyset$
$* \{ s_0, s_1 \}$	$\{ s_0, s_1 \}$	$\{ s_0, s_2 \}$
$\emptyset$	$\emptyset$	$\emptyset$
$* \{ s_0, s_2 \}$	$\{ s_0, s_1, s_3 \}$	$\emptyset$
$* \{ s_0, s_1, s_3 \}$	$\{ s_0, s_1 \}$	$\{ s_0, s_2 \}$

We can rename the states:

	a	b
$\rightarrow *A$	B	C
$*B$	B	D
C	C	C
$*D$	E	C
$*E$	B	D

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 get the following equivalence classes: $\{ A, D \}$, $\{ B, E \}$, $\{ C \}$. Thus the following DFA B is minimal and satisfies $L(B) = L(A)$:

	a	b
$\rightarrow *A$	B	C
$*B$	B	A
C	C	C

(c) Using the minimal DFA above, or the regular expression $(a+ab)^*$, we can see that the language $L(A)$ consists of exactly those sequences of a's and b's that do not start with a b and do not contain two successive b's.

4. Note that the language $L((0 + 01)^*)$ consists of exactly those sequences of zeros and ones that do not start with a one and do not contain two successive ones. Let $M = (Q, \{ 0, 1 \}, \Gamma, \delta, \mathit{zero}, \sqcup, \{ \mathit{accept} \})$, where Q, Γ and δ are defined in the following way:

$$Q = \{ \mathit{zero}, \mathit{zero\text{-}or\text{-}one}, \mathit{accept} \}$$
$$\Gamma = \{ 0, 1, \sqcup \}$$

$$\begin{array}{l} \delta \in Q \times \Gamma \rightharpoonup Q \times \Gamma \times \{\,\mathsf{L}, \mathsf{R}\,\} \\ \delta(\mathit{zero}, 0) = (\mathit{zero\text{-}or\text{-}one}, \sqcup, \mathsf{R}) \\ \delta(\mathit{zero}, \sqcup) = (\mathit{accept}, \sqcup, \mathsf{R}) \\ \delta(\mathit{zero\text{-}or\text{-}one}, 0) = (\mathit{zero\text{-}or\text{-}one}, \sqcup, \mathsf{R}) \\ \delta(\mathit{zero\text{-}or\text{-}one}, 1) = (\mathit{zero}, \sqcup, \mathsf{R}) \\ \delta(\mathit{zero\text{-}or\text{-}one}, \sqcup) = (\mathit{accept}, \sqcup, \mathsf{R}) \end{array}$$

The machine always moves to the right. It halts and rejects if the first symbol (if any) is a one, or if two successive ones are encountered. Otherwise it halts and accepts once it has reached the end of the input.

5. (a) The language is defined in the following way:

$$\frac{u, v \in L}{aubv \in L} \qquad \frac{}{b \in L}$$

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

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

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

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

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

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

$$\frac{\frac{}{S \to aSbS \in P} \quad \frac{\frac{u \in L(G, S) \quad \frac{\frac{v \in L(G, S) \quad \frac{}{\varepsilon \in L_{\mathrm{L}}(G, \varepsilon)}}{v \in L_{\mathrm{L}}(G, S)}}{bv \in L_{\mathrm{L}}(G, bS)}}{ubv \in L_{\mathrm{L}}(G, SbS)}}{aubv \in L_{\mathrm{L}}(G, aSbS)}}{aubv \in L(G, S)}$$

(c) The property $L(G) \subseteq L$ follows from $\forall w \in L(G, S).\ w \in L$. 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:

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

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 $w = b \in L$:

$$\dfrac{}{b \in L}$$

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

$$\dfrac{\dfrac{}{S \to aSbS \in P} \qquad \dfrac{\dfrac{u \in L(G, S) \qquad \dfrac{\dfrac{v \in L(G, S) \qquad \dfrac{}{\varepsilon \in L_{\mathrm{L}}(G, \varepsilon)}}{v \in L_{\mathrm{L}}(G, S)}}{bv \in L_{\mathrm{L}}(G, bS)}}{ubv \in L_{\mathrm{L}}(G, SbS)}}{aubv \in L_{\mathrm{L}}(G, aSbS)}}{aubv \in L(G, S)}$$

  Note that $|u| < |w|$ and $|v| < |w|$. The inductive hypothesis thus implies that $u \in L$ and $v \in L$. We can now construct a derivation showing that $w = aubv \in L$:

$$\dfrac{u, v \in L}{aubv \in L}$$

6. M is equal to $L(b(cb)^*)$, and thus regular (and context-free). Proof:

- $\forall w \in M. w \in L(b(cb)^*)$: This can be proved by induction on the structure of M. We have two cases:
  - $\dfrac{}{b \in M}$: It is clear that $w = b \in L(b(cb)^*)$.
  - $\dfrac{u, v \in M}{ucv \in M}$: The inductive hypotheses tell us that $u \in L(b(cb)^*)$ and $v \in L(b(cb)^*)$. Thus there are natural numbers $m, n \in \mathbb{N}$ such that $u = b(cb)^m$ and $v = b(cb)^n$. We get that $w = ucv = b(cb)^m cb(cb)^n = b(cb)^{1+m+n} \in L(b(cb)^*)$.
- $\forall w \in L(b(cb)^*). w \in M$: If $w \in L(b(cb)^*)$, then $w = b(cb)^n$ for some $n \in \mathbb{N}$. For each $n \in \mathbb{N}$, let the proposition $P(n)$ be $b(cb)^n \in M$. We can prove $\forall n \in \mathbb{N}. P(n)$ by mathematical induction. There are two cases:

– 0: A derivation can be constructed in the following way:

$$\frac{}{b \in M}$$

– $1 + n$: The inductive hypothesis is $P(n)$, i.e. $b(cb)^n \in M$. A derivation can be constructed in the following way (note that $b(cb)^{1+n} = b(cb)^n cb$):

$$\frac{b(cb)^n \in M \quad \frac{}{b \in M}}{b(cb)^n cb \in M}$$

N is not context-free (and thus also not regular). If we assume that N is context-free, then we can derive a contradiction. If N is context-free, then $N \cap \{ a, b \}^*$ is also context-free (because $\{ a, b \}^*$ is regular). Note also that every string obtained using the final rule contains the symbol c, and no string obtained using only the other two rules contain this symbol, so $N \cap \{ a, b \}^* = N'$, the language that we obtain by removing the final rule:

$$\frac{}{b \in N'} \qquad\qquad \frac{u \in \{ a, b \}^*}{uu \in N'}$$

If N' is context-free, then the language $N' \setminus \{ b \} = \{ uu \mid u \in \{ a, b \}^* \}$ is also context-free (because $\{ b \}$ is regular). However, this is a contradiction: $\{ uu \mid u \in \{ a, b \}^* \}$ is not context-free.