Formal Reasoning 2017
Solutions Test Block 2: Languages & Automata
(25/10/17)

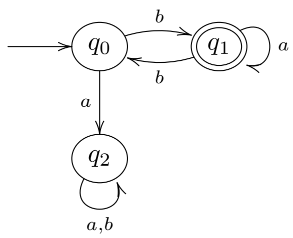
1. We define a context-free grammar G_1:

$$S \to bA$$
$$A \to aA \mid bS \mid \lambda$$

We call the language produced by this grammar L_1:

$$L_1 := \mathcal{L}(G_1)$$

(a) Give a deterministic finite automaton M_1 with $L(M_1) = L_1$.



(b) Give a regular expression r_1 with $\mathcal{L}(r_1) = L_1$.

$$b(a \cup bb)^* \quad \text{or} \quad (ba^*b)^*ba^* \quad \text{or} \quad b(a^*(bb)^*)^* \quad \text{or} \quad b(a^*(bb)^*a^*)^*$$

(c) Is the context-free grammar G_1 right-linear? Explain your answer.
Yes, it is. All non-terminals on the right side of the arrows are always completely at the right.

(d) We want to show that $bab \notin \mathcal{L}(G_1)$. For this someone proposes the following property as an invariant:

$$P(w) := \begin{array}{l} w \text{ starts with a symbol from the set } \{b, S\} \textit{ and} \\ w \text{ contains an odd number of symbols from } \{b, S\} \end{array}$$

Does this work? Explain your answer.
Yes, it works. It is obvious that $P(bab)$ does not hold, hence if P is an invariant, this implies that $bab \notin \mathcal{L}(G_1)$.

So now we have to prove that P is indeed an invariant. First we introduce a short notation:

$$|w|_{bS} := \text{the amount of symbols from } \{b, S\} \text{ in word } w$$

- $P(S)$ holds because S starts with an S and $|S|_{bS} = 1$ which is odd.
- Let v be a word such that $P(v)$ holds. Hence v starts with a b or an S and $|v|_{bS}$ is odd. Assume that $v \to v'$. We consider the following cases, where $u \in \{b, S\}$ and where x and y are arbitrary words over the terminals and non-terminals:

- $v = uxSy \to v' = uxbAy$. Obviously v' starts with a b or an S, because the first symbol didn't change. And $|v'|_{bS} = |v|_{bS}$ since we have one S less, but one b more. Hence $|v'|_{bS}$ is odd, so $P(v')$ holds.
- $v = uxAy \to v' = uxaAy$. Obviously v' starts with a b or an S. And $|v'|_{bS} = |v|_{bS}$ since the amount of b's and S's didn't change. Hence $|v'|_{bS}$ is odd, so $P(v')$ holds.
- $v = uxAy \to v' = uxbSy$. Obviously v' starts with a b or an S. And $|v'|_{bS} = |v|_{bS} + 2$ since we get one b and one S more. Hence $|v'|_{bS}$ is odd, so $P(v')$ holds.
- $v = uxAy \to v' = uxy$. Obviously v' starts with a b or an S. And $|v'|_{bS} = |v|_{bS}$ since the amount of b's and S's didn't change. Hence $|v'|_{bS}$ is odd, so $P(v')$ holds.
- $v = Sx \to v' = bAx$. Obviously v' starts with a b or an S. And $|v'|_{bS} = |v|_{bS}$ since we have one S less, but one b more. Hence $|v'|_{bS}$ is odd, so $P(v')$ holds.

So in all cases we have seen that $P(v')$ holds. Hence P is indeed an invariant.

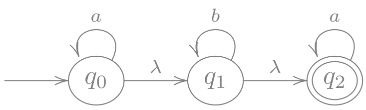
(e) Does the following equality hold?

$$L_1 = \{w \in \{a,b\}^* \mid P(w) \text{ holds}\}$$

Explain your answer.

No, it does not hold. The word $bbab$ is a counterexample. It is easy to see that $P(bbab)$ holds, but $bbab \notin L_1$. In the grammar it is easy to see that every second b, must be immediately followed by another b. This is caused by the production $A \to bS \to bbA$.

2. We define a non-deterministic finite automaton M_2:



We call the language recognized by this automaton L_2:

$$L_2 := L(M_2)$$

(a) Write M_2 as a quintuple $\langle \Sigma, Q, q_0, F, \delta \rangle$. Define δ by giving equations of the form $\delta(q_i, x) = \ldots$ for all possible inputs q_i and x.

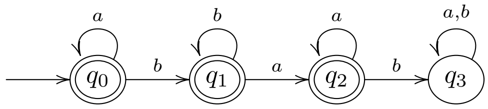
$M_2 = \langle \{a,b\}, \{q_0, q_1, q_2\}, q_0, \{q_2\}, \delta \rangle$, where δ is given as follows:

$$\begin{array}{lllllllll}
\delta(q_0,a) & = & \{q_0\} & \delta(q_0,b) & = & \emptyset & \delta(q_0,\lambda) & = & \{q_1\} \\
\delta(q_1,a) & = & \emptyset & \delta(q_1,b) & = & \{q_1\} & \delta(q_1,\lambda) & = & \{q_2\} \\
\delta(q_2,a) & = & \{q_2\} & \delta(q_2,b) & = & \emptyset & \delta(q_2,\lambda) & = & \emptyset
\end{array}$$

(b) Give a regular expression r_2 with $\mathcal{L}(r_2) = L_2$.

$$r_2 := a^* b^* a^*$$

(c) Give a *deterministic* finite automaton M_2' with $L(M_2') = L_2$.



3. If for a language L is given that $\lambda \in L$ and $LL = L$, does it always hold that $L^* = L$?

   If so, explain why. If not, give an example of a language L_3 for which this does not hold, and explain why it is a counterexample.

   Yes, it always holds.

   Since $L \subseteq L^*$ for any language L, we only have to prove that $L^* \subseteq L$.

   - Let $w \in L^*$.
   - Then there exists $k \in \mathbf{N}$ such that $w = w_1 w_2 \cdots w_{k-1} w_k$ where $w_i \in L$ for all i.
     - If $k = 0$ then $w = \lambda$ and it was given that $\lambda \in L$, so in this case $w \in L$.
     - If $k = 1$ then $w = w_1$, where $w_1 \in L$, so also in this case $w \in L$.
     - If $k \geq 2$ we know that $w_{k-1} w_k \in L$, because $LL = L$.
     - But this means that we can write $w = w_1 w_2 \cdots w'_{k-1}$ where $w_i \in L$ and $w'_{k-1} \in L$.
     - So we have shown that if we can split w in k parts that are all in L, then we can also split w in $k - 1$ parts that are all in L.
     - Now if $k - 1 = 1$ we can repeat the argument we mentioned above for this case. (Note that if $k \geq 2$ it cannot happen that $k - 1 = 0$.)
     - And if $k - 1 \geq 2$ we can repeat this trick and get that $w'_{k-2} = w_{k-2} w'_{k-1} = w_{k-2} w_{k-1} w_k \in L$, so we can split w also in $k - 2$ parts that are all in L.
     - Because k is finite, we know that after applying this trick $k - 1$ times we have that $w = w'_1 \in L$, so also in this case $w \in L$.
   - So in all cases we get that $w \in L$.

   Hence $L^* \in L$, and together with $L \subseteq L^*$ we get that $L^* = L$.