Formal Reasoning 2021
Solutions Test Block 2: Languages and Automata
(10/11/21)

Languages

1. Give a regular expression for the language:

$$L_1 := \{w \in \{a, b\}^* \mid w \text{ contains neither } aa \text{ nor } bb\}$$

This means that the words in L_1 have to alternate between a and b.

Since you have to type the expression in the Cirrus editor, please adhere to these rules:

- Type the alphabet symbols and parentheses simply as '`a`', '`b`', '`(`', and '`)`'.
- Type the regular expression $\emptyset$ as '`0`' (a zero).
- Type the regular expression λ as '`^`'.
- Type the operator $\cup$ as '`U`'.
- Type the operator * as '`*`', so without putting it in superscript.

$$(a \cup \lambda)(ba)^* \cup (b \cup \lambda)(ab)^*$$
$$(ab)^*(a \cup \lambda) \cup (ba)^*(b \cup \lambda)$$
$$(a \cup \lambda)(ba)^*(b \cup \lambda)$$
$$(b \cup \lambda)(ab)^*(a \cup \lambda)$$

2. Consider the context-free grammar

$$G_2: \quad S \to a \mid SbS$$

Is the language $\mathcal{L}(G_2)$ regular?

Note that in order to get partial points for a partially correct answer, you actually have to give two answers: first a simple 'Yes' or 'No' and then a more detailed answer.

   (a) Yes.
   (b) No.

Choose the correct explanation:

   (a) Yes, because G_2 is right linear.
   (b) Yes, although G_2 is not right linear, the language $\mathcal{L}(G_2)$ also has a grammar that *is* right linear.
   (c) No, because G_2 is not right linear.
   (d) No, although G_2 is right linear, the language $\mathcal{L}(G_2)$ also has a grammar that is *not* right linear.

(b) is correct

Answer (b) is correct.

The first and the fourth answer make no sense because the grammar is not right linear due to the rule $S \to SbS$.

The third answer makes no sense either. Although G_2 is not right linear, that is not a sufficient condition for generating a non-regular language.

In fact, there exists a right linear grammar that generates the same language:

$$\begin{aligned} G_2' : \quad & S \to aA \\ & A \to bS \mid \lambda \end{aligned}$$

So the remaining second answer is indeed correct.

3. Consider the same context-free grammar

$$G_2 : \quad S \to a \mid SbS$$

Someone proposes the following two properties as potential invariants for this grammar:

$$\begin{aligned} P_{aa}(w) &:= w \text{ does not contain } aa \\ P_{bb}(w) &:= w \text{ does not contain } bb \end{aligned}$$

Is P_{aa} an invariant of G_2?

(a) Yes.

(b) is correct (b) No.

Answer (b) is correct.

Note that $P_{aa}(w)$ is not an invariant. We have that $P_{aa}(aS)$ holds, but $aS \to aa$ and then clearly $P_{aa}(aa)$ does not hold.

Is P_{bb} an invariant of G_2?

(a) is correct (a) Yes.

(b) No.

Answer (a) is correct.

Note that $P_{bb}(w)$ is indeed an invariant:

- $P_{bb}(S)$ clearly holds as S does not contain bb.
- Let v and v' be such that $P_{bb}(v)$ holds and that $v \to v'$. Then v must be of the form wSw' for some words w and w' that both do not contain any bb. From $v \to v'$ it follows that $v' = waw$ or $v' = wSbSw'$. Because w and w' do not contain bb, the only problem could arise at the end of w and the start of w'. However, in the first case, the a prevents that a w that ends on a b is suddenly followed by a b at the start of w'. And in the second case, we have a similar situation. Although we do get a b in between w and w', this b is 'protected' on both sides by an S, so also in this case it cannot happen that a b at the end of w gets connected to this b in SbS, and the b in SbS cannot get connected to a b at the start of w'.

4. Is it the case that if $L = L'^*$ for some language L', that then $L^* = L$?

(a) is correct

(a) Yes, because $(L'^*)^* = L'^*$.

(b) Yes, because you take L to be L'.

(c) No, because $L \neq L'$.

(d) No, because $L^* = L$ is never true.

If $L = L'^*$ and $(L'^*)^* = L'^*$ then we get:

$$L^* = (L'^*)^* = L'^* = L$$

So if we can prove that $(L'^*)^* = L'^*$ then it is clear that $L^* = L$.

As for all languages, it trivially holds that $L'^* \subseteq (L'^*)^*$. So we only have to show that $(L'^*)^* \subseteq L'^*$. Now let $w \in (L'^*)^*$. Then there exists $k \in \mathbb{N}$ such that

$$w = w_1 w_2 \cdots w_k$$

where $w_i \in L'^*$ for each $i = 1, \ldots, k$. But this means that for each w_i there exists $l_i \in \mathbb{N}$ such that

$$w_i = w_{i_1} w_{i_2} \cdots w_{i_{l_i}}$$

where each $w_{i_j} \in L'$ for $j = 1, \ldots, l_i$. So combining this for each $i = 1, \ldots, k$ we get

$$w = w_{1_1} \cdots w_{1_{l_1}} \quad w_{2_1} \cdots w_{2_{l_2}} \quad \cdots \quad w_{k_1} \cdots w_{k_{l_k}}$$

Note that the white space is only added to visualize the original k blocks.

However, since $w_{i_j} \in L'$ for each $i = 1, \ldots, k$ and $j = 1, \ldots, l_i$ it follows that $w \in L'^*$, which we had to prove.

The second answer makes no sense because if you take L' to be the language of words over $\Sigma = \{a, b\}$ that contain an odd number of a's, then $L'^* = \Sigma^*$ which is not equal to L' since it also contains words with an even number of a's, like λ.

The third and the fourth answer make no sense because the answer is 'yes'.

Automata

5. Consider again the context-free grammar

$$G_2 : \quad S \to a \mid SbS$$

Give a deterministic finite automaton M_2 such that $\mathcal{L}(M_2) = \mathcal{L}(G_2)$. You are supposed to draw the automaton on the paper appendix that you should have received. Make sure to include your name and student number at the indicated place, before handing it in at the end of the test.

In addition, you have to provide some metrics within Cirrus about the automaton that you created by filling in the blanks below.

My automaton has ... state(s), of which ... are final state(s).

Take for instance:

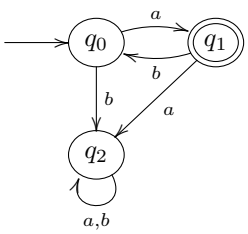
$$M_2 := \langle \{a, b\}, \{q_0, q_1, q_2\}, q_0, \{q_1\}, \delta_2 \rangle$$

with

$$\begin{array}{lll} \delta_2(q_0, a) = q_1 & \delta_2(q_1, a) = q_2 & \delta_2(q_2, a) = q_2 \\ \delta_2(q_0, b) = q_2 & \delta_2(q_1, b) = q_0 & \delta_2(q_2, b) = q_2 \end{array}$$

This automaton has three states, including one final state.

Represented as a LaTeX diagram this gives:



6. Consider the deterministic finite automaton

$$M_6 := \langle \{a, b\}, \{q_0, q_1, q_2, q_3\}, q_0, \{q_0, q_1, q_2\}, \delta_6 \rangle$$

with

$$\begin{array}{llll} \delta_6(q_0, a) = q_1 & \delta_6(q_1, a) = q_3 & \delta_6(q_2, a) = q_1 & \delta_6(q_3, a) = q_3 \\ \delta_6(q_0, b) = q_2 & \delta_6(q_1, b) = q_2 & \delta_6(q_2, b) = q_3 & \delta_6(q_3, b) = q_3 \end{array}$$

Which of the following words is *not* in $\mathcal{L}(M_6)$?

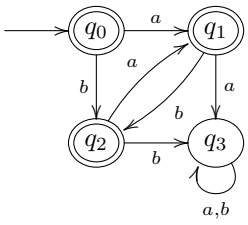
(a) λ

(b) a

(c) is correct (c) aa

(d) ab

If we represent M_6 as a diagram, we get:



The productions for these four words are:

$$\begin{array}{l} q_0 \\ q_0 \xrightarrow{a} q_1 \\ q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_3 \\ q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \end{array}$$

And because states q_0, q_1, and q_2 are final states, and q_3 is not (in fact, it is a sink), the words λ, a, and ab are accepted, but aa is not.

7. Consider again the language

$$L_1 := \{w \in \{a, b\}^* \mid w \text{ contains neither } aa \text{ nor } bb\}$$

What is the minimal number of states in a *non*-deterministic finite automaton that accepts L_1?

(a) one

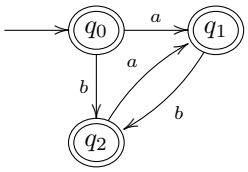
(b) two

(c) is correct (c) three

(d) more than three

Answer (c) is correct.

Note that this automaton accepts the language:



The automaton distinguishes three states:

- q_0: No symbols have been read, both a and b must lead to accepted words.
- q_1: The last symbol read was an a, only a b can lead to an accepted word.
- q_2: The last symbol read was a b, only an a can lead to an accepted word.

A proof sketch that it cannot be done with only two states goes like this:

- Assume that we have an automaton with two states q_0 and q_1, where q_0 is the initial state.
- Note that a is accepted, so there should be a path from q_0 to a final state with exactly one a-transition. (And possibly some λ-transitions.)
- If this a-transition is a loop on q_0 or q_1, then you can always take that loop twice, providing a path to end in a final state after reading aa, which is not allowed.
- So the a-transition must be from q_0 to q_1.[1]
- So $\delta(q_0, a) = \{q_1\}$.
- And either q_1 is a final state or q_0 is a final state and q_0 can be reached with a λ-transition from q_1.
- Likewise, $\delta(q_0, b) = \{q_1\}$.
- However, the word ab must also be accepted.

[1] Actually, this is not correct, because it could be from q_1 to q_0 if there is a λ-transition from q_0 to q_1.

- So from q_1 there must be a path to get to a final state after reading b.
- But since there are no b-loops in q_0 and q_1, there are only two options:
  - There is a λ-transition from q_1 to q_0 and you can do $q_0 \xrightarrow{a} q_1 \xrightarrow{\lambda} q_0 \xrightarrow{b} q_1$ if q_1 is a final state or $q_0 \xrightarrow{a} q_1 \xrightarrow{\lambda} q_0 \xrightarrow{b} q_1 \xrightarrow{\lambda} q_0$ if q_0 is a final state. But then you also get the productions: $q_0 \xrightarrow{a} q_1 \xrightarrow{\lambda} q_0 \xrightarrow{a} q_1$ if q_1 is a final state or $q_0 \xrightarrow{a} q_1 \xrightarrow{\lambda} q_0 \xrightarrow{a} q_1 \xrightarrow{\lambda} q_0$ if q_0 is a final state, implying that aa is accepted, but that is not allowed.
  - There is a b-transition from q_1 to q_0. But then you get the productions: $q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_0 \xrightarrow{b} q_1$ if q_1 is a final state or $q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_0 \xrightarrow{b} q_1 \xrightarrow{\lambda} q_0$ if q_0 is a final state, implying that abb is accepted, which is not allowed.
- So it cannot be done with two states.

And we leave the proof that it cannot be done with a single state to the reader.

So we need at least three states.

8. For a deterministic finite automaton $M = \langle \Sigma, Q, q_0, F, \delta \rangle$, we define $M' = \langle \Sigma, Q, q_0, F \cup \{q_0\}, \delta \rangle$.

   What is the relation between the languages recognized by M and M'?

   (a) $\mathcal{L}(M') = \mathcal{L}(M) \cup \{\lambda\}$

   (b) $\mathcal{L}(M') \subseteq \mathcal{L}(M) \cup \{\lambda\}$, and for some M equality does not hold

(c) is correct (c) $\mathcal{L}(M) \cup \{\lambda\} \subseteq \mathcal{L}(M')$, and for some M equality does not hold

   (d) none of the above

   Answer (c) is correct.

   The difference between M and M' is that the initial state q_0 is now also a final state. Since all original final states are still final states, all accepted words by M are also accepted by M'. So certainly $\mathcal{L}(M) \subseteq \mathcal{L}(M')$. In addition, λ is accepted by M'. (Note that we don't know whether it is also accepted by M, but that doesn't matter.) So we have $\mathcal{L}(M) \cup \{\lambda\} \subseteq \mathcal{L}(M')$. It is also clear that for some languages M the equality does not hold. Take for instance $M = \langle \{a\}, \{q_0\}, q_0, \varnothing, \delta \rangle$, where δ is defined by

$$\delta(q_0, a) = q_0$$

   As M has no final states, it follows that $\mathcal{L}(M) = \varnothing$ and that

$$\mathcal{L}(M) \cup \{\lambda\} = \varnothing \cup \{\lambda\} = \{\lambda\} \subset \{a^n \mid n \in \mathbb{N}\} = \mathcal{L}(M')$$

   In particular, this example also shows why the other options are incorrect.