Formal Reasoning 2019
Solutions Test Block 2: Languages and Automata
(06/11/19)

1. The equality (15 points)

$$\overline{L \cup L'} = \overline{L} \cup \overline{L'}$$

does not hold for all languages L and L' over the alphabet $\{a, b\}$. Give languages L and L' over this alphabet for which this equality does not hold, and *explain* why this is the case.

No, it doesn't. Let

$$\begin{aligned} L &= \{w \in \{a,b\}^* \mid w \text{ contains an even number of } a\text{'s}\} \\ L' &= \{w \in \{a,b\}^* \mid w \text{ contains an odd number of } a\text{'s}\} \end{aligned}$$

Then

$$\begin{aligned} L \cup L' &= \{a,b\}^* \\ \overline{L \cup L'} &= \emptyset \\ \overline{L} &= \{w \in \{a,b\}^* \mid w \text{ contains an odd number of } a\text{'s}\} \\ \overline{L'} &= \{w \in \{a,b\}^* \mid w \text{ contains an even number of } a\text{'s}\} \\ \overline{L} \cup \overline{L'} &= \{a,b\}^* \end{aligned}$$

And clearly $a \in \{a,b\}^*$, so $\{a,b\}^* \neq \emptyset$.

Alternatively, show that there is some $w \in L$ such that $w \notin L'$. Then $w \in L \cup L'$ and therefore $w \notin \overline{L \cup L'}$. However, $w \in \overline{L'}$ because $w \notin L$, so $w \in \overline{L} \cup \overline{L'}$. For instance, pick $w = aa$.

Common mistakes:

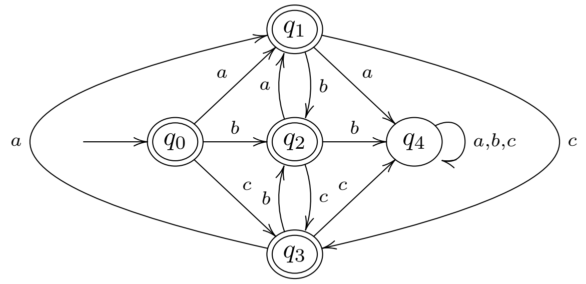
- Students have interpreted $\cup$ as an intersection instead of a union.
- Students assume that language is a set in Σ instead of Σ^*, so the complement is often done wrong.
- The given languages L and L' is equal, though written in a different order.
- No explanation was given.

2. (a) Give a deterministic finite automaton with alphabet $\{a, b, c\}$ for the language: (15 points)

$$L_2 := \{w \in \{a,b,c\}^* \mid \text{adjacent symbols in } w \text{ differ}\}$$

We have $abacb \in L_2$ because in $abacb$ all symbols differ from their predecessor, but $abba \notin L_2$, because there are two bs next to each other in $abba$. We also have $\lambda \in L_2$, because there are no adjacent symbols in λ at all.

Hint: Let the states of the automaton correspond to the last symbol that has been read thus far.



Common mistakes:

- Either the initial state or addition of outgoing arrows to the sink itself were forgotten.
- Students often had a similar automaton, but instead of going to the state corresponding to the last-read symbol, they return to the initial state. For example, if state q_0 would transition to q_1 with a, then q_1 would go to the sink with a, but back to q_0 with b and c.

(b) Give the right linear context-free grammar associated with the automaton from the previous sub-exercise. For this exercise it does not matter whether that automaton was correct for the language. (15 points)

$$
\begin{aligned}
S &\rightarrow aA \mid bB \mid cC \mid \lambda \\
A &\rightarrow aD \mid bB \mid cC \mid \lambda \\
B &\rightarrow aA \mid bD \mid cC \mid \lambda \\
C &\rightarrow aA \mid bB \mid cD \mid \lambda \\
D &\rightarrow aD \mid bD \mid cD
\end{aligned}
$$

Common mistakes:

Students often forgot to add the sink to the grammar or wrote a correct grammar that wasn't equivalent to their automaton.

3. Give a regular expression for the language: (15 points)

$$L_3 := \{w \in \{a, b, c\}^* \mid w \text{ does not contain } ab\}$$

Words in L_3 can be:

- Without any a: $(b \cup c)^*$
- Consisting of consecutive blocks starting with a series of b's and c's, followed by one or more a's, followed by one c, and these blocks are followed by a series of b's and c's, followed again by zero or more a's: $((b \cup c)^* aa^* c)^* (b \cup c)^* a^*$

Combined this gives the expression:

$$(b \cup c)^* \cup ((b \cup c)^* aa^* c)^* (b \cup c)^* a^*$$

Which is equivalent to:

$$(a^* c \cup b)^* a^*$$

A sequence of a's either has to be followed by a c, or it has to be at the end of the word. This leads to the regular expression:

$$(aa^*c \cup b \cup c)^*a^*$$

In the union under the Kleene star there are three cases, depending on whether the first symbol of the remainder of the word is an a, b or c. This solution can be simplified to the short solution:

$$(a^*c \cup b)^*a^*$$

The same solution, but 'backwards', is:

$$b^*(cb^* \cup a)^*$$

A nice solution that combines the b^* start and a^* ending in a symmetric way is:

$$b^*(a^*cb^*)^*a^*$$

A variant of this solution is:

$$(b^*a^*c)^*b^*a^*$$

A way to understand these solutions is that the string has to consist of zero or more c's, with strings that match b^*a^* in between each of them.

Common mistakes:

- Often in an otherwise correct answer the final a^* or initial b^* is forgotten.
- Although it is not a mistake, often regular expressions contain redundant parts, like $(a^*c \cup c \cup \dots)$ or $(a \cup \lambda)^*$ and so on.
- Regularly the $\cup$ symbol is written as a capital letter U.
- Occasionally 'regular expressions' contain braces ('{' and '}'), commas (','), intersection signs ('$\cap$') and complement signs ('$^{-}$'), which of course are not allowed in regular expressions.

4. Consider the context-free grammar G_4 : (15 points)

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

We want to show that $ab \notin \mathcal{L}(G_4)$ and are considering the predicate

$$P_4(w) := \big(w \text{ does not contain any of: } ab,\ aS,\ Ab,\ Sb,\ SS\big)$$

but this does not work. Explain why.

Take the word $v = aAS$. Then clearly $P(v)$ holds. However, $v \to v'$ where $v' = aS$ and then $P(v')$ does not hold. So this predicate is not a proper invariant.

5. Explain why each language that can be recognized by a non-deterministic finite automaton, also can be recognized by a non-deterministic finite automaton that has a *single* final state. (15 points)

This can be arranged by adding a new final state and connect all original final states by a λ-transition to the new final state. More formally, consider the automaton

$$M := \langle \Sigma, Q, q_0, F, \delta \rangle$$

Then we can define a new automaton

$$M' := \langle \Sigma, Q \cup \{q_f\}, q_0, \{q_f\}, \delta' \rangle$$

where δ' is defined as:

$$\begin{array}{rcl} \delta'(q_i, x) & = & \delta(q_i, x) \text{ if } q_i \in Q \text{ and } x \in \Sigma \\ \delta'(q_i, \lambda) & = & \delta(q_i, \lambda) \cup \{q_f\} \text{ if } q_i \in F \\ \delta'(q_i, \lambda) & = & \delta(q_i, \lambda) \text{ if } q_i \notin F \end{array}$$

Hence word w is accepted by automaton M if and only if word w is accepted by automaton M'.