TGI WS 2016/17 Sheet 7

Sebastian Löhr, Thure Dührsen 2017-02-14

Exercise 1

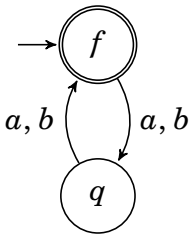
Fix $L_1 := \{w \in \{a,b\}^* : |w|_a \equiv_2 |w|_b\}$ and $L_2 := \{w \in \{a,b\}^* : |w|_a \not\equiv_2 |w|_b\}$.

Claim

(a) A regular expression describing L_1 is given by

$$((a|b)\cdot(a|b))^*.$$

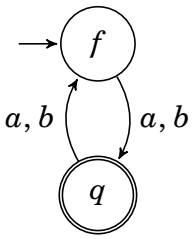
A DFA describing L_1 is shown to the right.



(b) A regular expression describing L_2 is given by

$$(a|b)\cdot((a|b)\cdot(a|b))^*.$$

A DFA describing L_2 is shown to the right.



Proof

(a) L_1 describes the the language that holds all words from $\{a,b\}^*$ where if the count of a is even, also the count of b must be even.
Also if the count of a is odd, it has to have an odd count of b's.
So if we read an a or an b we need to read an a or an b to accept the word.

(b) L_2 describes the the language that holds all words from $\{a,b\}^*$ where if the count of a is even, the count of b must be odd.
Also if the count of a is odd, it has to have an even count of b's.
We have to start with an a or b, after that it's the same as L_1.

□

Exercise 2

Let n be a natural number, let $\Sigma := \{a\}$, and let $\mathscr{A}$ be a DFA that accepts the single word $a^n \in \Sigma^*$.

Claim The automaton $\mathscr{A}$ has at least n states.

Proof

Suppose there is a DFA $\mathscr{A}$ with less than n states that accepts the word a^n, and choose one. In order to accept the word a^n, exactly n transitions must occur, because every symbol read from the input corresponds to exactly one transition and we must read exactly n symbols. As $\mathscr{A}$ has less than n states, at least one of those states must be visited more than once, i. e. we have a loop or even a cycle in the graph representing $\mathscr{A}$, and we must traverse the loop or the cycle in order to accept a^n.

But this means we can traverse the loop or the cycle more than once, which means that there exist $k, l \in \mathbb{N}$ such that

$$\forall\, m \in \mathbb{N} : a^{m \cdot k + l} \in L(\mathscr{A}) \quad ,$$

i. e. $L(\mathscr{A})$ is infinite, contradicting the fact that $\mathscr{A}$ accepts the single word a^n.

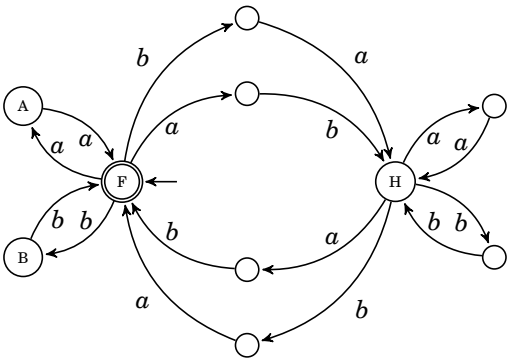
Therefore every DFA that accepts exactly the language $\{a^n\}$ has at least n states.

□

Exercise 3

Let $\mathscr{A}$ be the automaton shown to the right. Let

$$R = (aa|bb)^* \cdot ((ab|ba) \cdot (aa|bb)^* \cdot (ba|ab))^*$$



Claim R is a regular expression for the automaton.

Proof Obviously the automaton accepts the empty word ε. Starting at state F we can use two paths if we read an a and two other paths if we read an b.
Assume we read an a so we can use the path to A, we get back in the final state by reading another a. So we know $(aa)^*$ is part of R.
Similiar to that we get to B reading a b and need another b to accept it. So we get $(bb)^*$ is part of R too. We don't want to fix an order, we write $(aa|bb)^*$ as part of R.
To get to H we have two possible paths, reading ab or ba. based from H we have the same situation as in F seen before, so we can read $(aa|bb)^*$ at this point.
Now we have a path back to F using ab or ba again. So we can setup our regular expression: $R = (aa|bb)^* \cdot ((ab|ba) \cdot (aa|bb)^* \cdot (ba|ab))^*$.

□

Exercise 4

Many of the ideas (in particular, the claim in part (a)) are the combined work of Etienne, Daniel, Caro and myself.
All of the mistakes (in particular, those in part (b)) are, of course, my own. tdu

Let Σ be an alphabet. Let L and M be languages over Σ. Define $\sqcup\!\sqcup : \Sigma^* \to \mathfrak{P}(\Sigma^*)$ by

$$\left.\begin{array}{rcll} u & \sqcup\!\sqcup & \varepsilon & := \{u\} \\ \varepsilon & \sqcup\!\sqcup & v & := \{v\} \\ ua & \sqcup\!\sqcup & vb & := \Big((u \sqcup\!\sqcup vb) \circ \{a\}\Big) \cup \Big((ua \sqcup\!\sqcup v) \circ \{b\}\Big) \end{array}\right\} \text{ for all } u, v \in \Sigma^* \text{ and for all } a, b \in \Sigma.$$

Furthermore, define

$$L \sqcup\!\sqcup M := \bigcup_{\substack{u \in L \\ v \in M}} u \sqcup\!\sqcup v \quad .$$

Claim

(a) $(ab)^* \sqcup\!\sqcup (ba)^* = \big((ab) \cup (ba)\big)^*$

(b) If L and M are regular languages, then $L \sqcup\!\sqcup M$ is also a regular language.

Proof

(a) Intuitively clear from the "deck of cards" metaphor, but lack of time prevents us from giving a formal proof at this time. But just wait until after Christmas, for Sheet seven-and-a-half! ☺

(b) Assume that L and M are both regular languages.

As L is regular, choose a DFA $\mathscr{A} = (Q_1, \Sigma, \delta_1, q_1, F_1)$ such that $L(\mathscr{A}) = L$.

As M is regular, choose a DFA $\mathscr{B} = (Q_2, \Sigma, \delta_2, q_2, F_2)$ such that $L(\mathscr{B}) = M$.

Construct a NFA $\mathscr{A}^{(N)}$ from $\mathscr{A}$ by setting $\mathscr{A}^{(N)} := (Q_1, \Sigma, \delta_1^{(N)}, q_1, F_1)$, where

$$\forall\, q \in Q_1 \quad \forall\, a \in \Sigma^* : \quad \big((q, a), \{q, q'\}\big) \in \delta_1^{(N)} \iff \big((q, a), q'\big) \in \delta_1$$

and construct a NFA $\mathscr{B}^{(N)}$ from $\mathscr{B}$ by setting $\mathscr{B}^{(N)} := (Q_2, \Sigma, \delta_2^{(N)}, q_2, F_2)$, where

$$\forall\, q \in Q_2 \quad \forall\, a \in \Sigma^* : \quad \big((q, a), \{q, q'\}\big) \in \delta_2^{(N)} \iff \big((q, a), q'\big) \in \delta_2 \quad .$$

For ease of presentation, let $q(\mathscr{A}^{(N)})$ stand for the state that $\mathscr{A}^{(N)}$ is currently in, and let $q(\mathscr{B}^{(N)})$ stand for the state that $\mathscr{B}^{(N)}$ is currently in.

Now, construct another NFA $\mathscr{C} = (Q_3, \Sigma, \delta_3, A, K)$ by setting

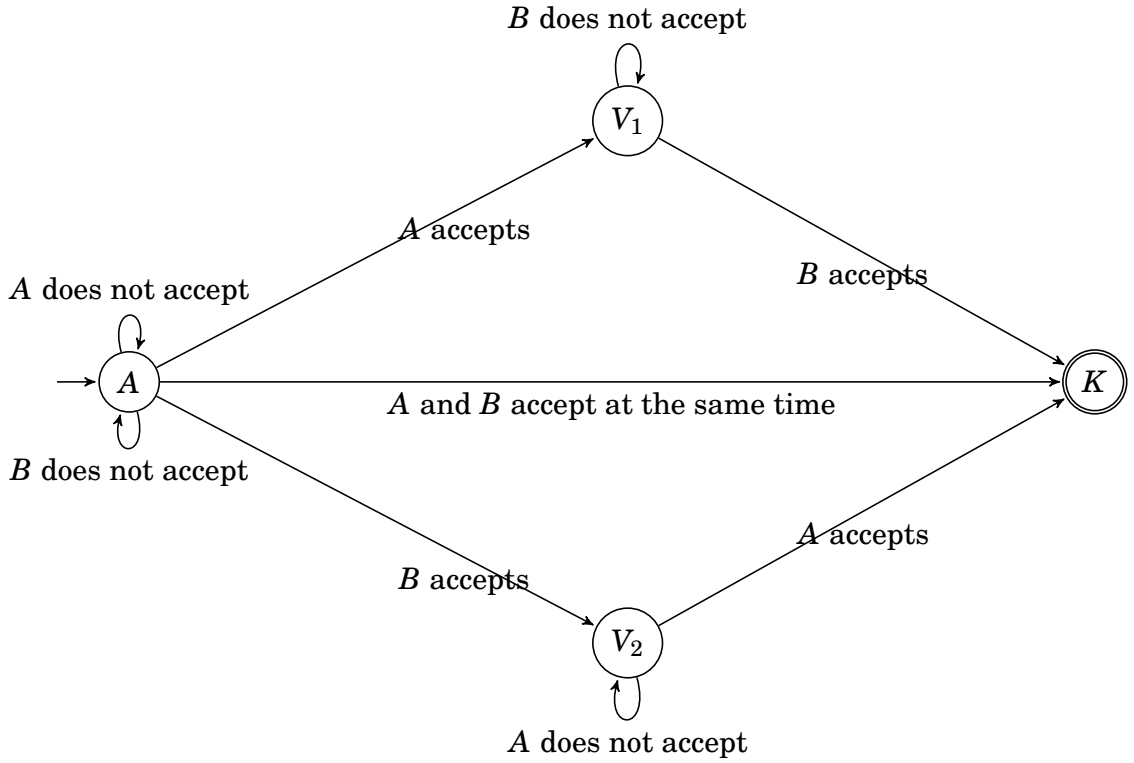
$$Q_3 := \{A, V_1, V_2, K\}$$

and by letting δ_3 be defined by

$$\forall\, q \in Q_3 \quad \forall\, a \in \Sigma^* : \delta_3(q,a) = \begin{cases} A & \text{if } \delta_1^{(N)}(q(\mathscr{A}^{(N)}),a) \notin F_1 \text{ and } \delta_2^{(N)}(q(\mathscr{B}^{(N)}),a) \notin F_2 \\ V_1 & \text{if } \delta_1^{(N)}(q(\mathscr{A}^{(N)}),a) \in F_1 \text{ and } \delta_2^{(N)}(q(\mathscr{B}^{(N)}),a) \notin F_2 \\ V_2 & \text{if } \delta_1^{(N)}(q(\mathscr{A}^{(N)}),a) \notin F_1 \text{ and } \delta_2^{(N)}(q(\mathscr{B}^{(N)}),a) \in F_2 \\ K & \text{if } \delta_1^{(N)}(q(\mathscr{A}^{(N)}),a) \in F_1 \text{ and } \delta_2^{(N)}(q(\mathscr{B}^{(N)}),a) \in F_2 \end{cases}$$

The construction of $\mathscr{A}^{(N)}$ and $\mathscr{B}^{(N)}$ enables each of the automata $\mathscr{A}^{(N)}$ and $\mathscr{B}^{(N)}$ to have the choice to advance to the next input token or remain at their current state, as might be required to accept the ⧢isation w of two words $u \in L$ and $v \in M$.[1] If none of the automata $\mathscr{A}^{(N)}$ and $\mathscr{B}^{(N)}$ accepts w, then $\mathscr{C}$ remains in A and therefore does not accept the word.

What we are trying to do here is roughly the following (a Ti*k*Zpicture hopefully says more than a thousand words):



This solution is probably not entirely correct, but at least we tried... and we're not in this for the points, anyway.

□

[1] In German I would say „Verschränkung“ or something similar, but in English? “entanglement”? “interlacement”? Something else entirely?