E 4.1

Exercise

Prove, by using the pumping lemma, that the language

$$L = \{ a^j b^j \mid j \geq 1 \}$$

is not regular.

solution

Remember that the pumping lemma tells us that if the words of a language cannot be "pumped", then the language is not regular.

Let m be a sufficiently large integer for the pumping lemma, and let w a word in L, with $|w| \geq m$; this can always hold if we choose j large enough.

$$w = xyz$$

There are three cases:

(a) $y = a^k$ for some $k \in \mathbb{N}$;
in this case $xy^2z \notin L$ because xy^2z has more a's than b's.

(b) $y = b^k$ for some $k \in \mathbb{N}$;
analogous to the previous case

(c) $y = a^k b^k$ for some k;
in this case $xy^2z \notin$ because simmetry is lost

(d) $y = a^h b^k$ for some $h, k \in \mathbb{N}$; $h \neq k$
also in this case symmetry is lost in xy^2z.

We have seen that there is no way to pump words of L, therefore L is not regular.

Exercise

E 4.2

Prove that the language

$$L = \{ww^R \mid w \in \{a,b\}^*\}$$

is not regular.

Note that L is the language of palindromes on $\{a,b\}$.

Remember that w^R denotes the string w after being inverted

solution

It is immediate to notice that we can pump strings of L only in the middle of the string; this for symmetry reasons.

Let $z = ww^R \in L$; we denote $w = uv$; therefore $w^R = v^R u^R$, and $ww^R = uvv^Ru^R$. We can pump vv^R at will, still obtaining palindrome strings (which obviously are in L):

$$uu^R \in L$$
$$uv\,v^Ru^R \in L$$
$$uvv^Rvv^Ru^R \in L$$
$$uvv^Rvv^Rvv^Rvv^R \in L$$

and so on.

Notice that this is the only way we can pump: in general, to prove that a language is not regular by showing that strings cannot be pumped, one has to consider all possible ways of pumping.

In this case, however, the condition $|xy| \leq n$ for the first part of the generic string $w = xyz$ does not hold: in fact, the string that is pumped is arbitrarily far from the beginning of the string. We can finally conclude that L is not regular.

E 4.3

Exercise

Prove that the language $L = \{ a^k \mid k \text{ is prime} \}$ is not regular.

solution

We apply the pumping lemma and show that there is no way to pump. Without loss of generality, we choose $|w| = m \geq n$, where $w \in L$, n is the constant of the pumping lemma, and $m \in \mathbb{N}$.

Now let $w = xyz$, with $|xy| \leq n$. From the pumping lemma $xy^iz \in L$ for all $i \in \mathbb{N}$; we choose $i = m+1$ and we have $w' = xy^{m+1}z$;

$$|w'| = |w| + |y^m| = |w| + m|y| = m(1+|y|)$$

which is not prime. Therefore, there is no way to pump and L is not regular.

Exercise (4.3.4 from textbook) E 4.4

Give an algorithm to tell whether two regular languages have at least one string in common.

Solution

It suffices to check whether the intersection L of the two languages, that we denote with L_1 and L_2, is empty.

We have
$$L = L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

An automaton accepting L can be easily constructed from automata accepting L_1 and L_2; emptiness of the language accepted by an automaton can be checked by checking reachability of a final state.

Notice that when constructing, for example, the automaton accepting $\overline{L_1}$, the automaton accepting L_1 of which we invert final and non-final states has to be <u>deterministic</u>. Otherwise, we do not obtain an automaton accepting $\overline{L_1}$.

Another solution consists in the direct construction of the automaton accepting $L_1 \cap L_2$.

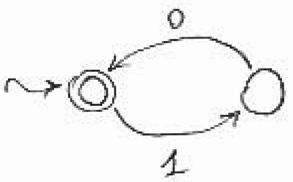
E 1.5

Exercise

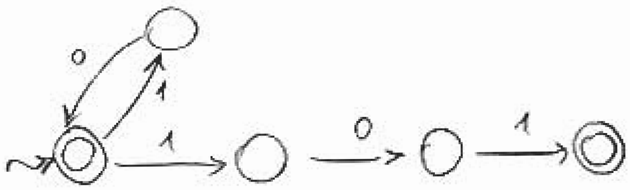
Construct a DFA accepting the language denoted by the regular expression

$$1(00+1)^*((10)^*+101)^*$$

solution

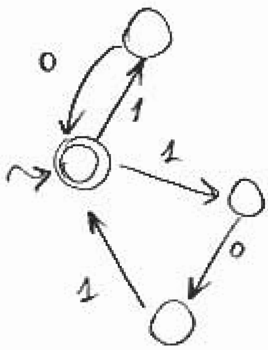


accepts 10^*



accepts $(10)^*+101$

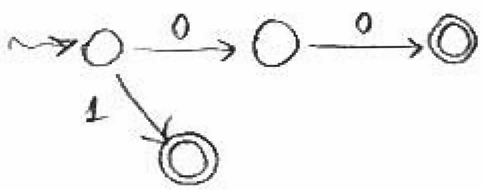
We want to iterate the expression $(10)^*+101$; we can make initial state and final states collapse.



accepts $((10)^*+101)^*$

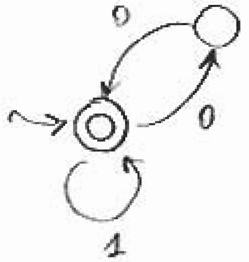
We go on with other sub-expressions:

accepts 1



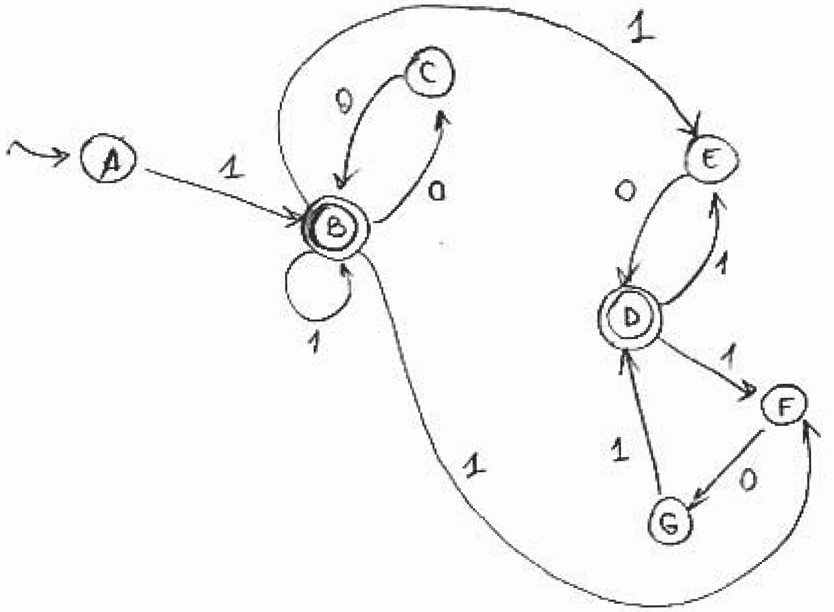
accepts $00+1$

E 4.6



accepts $(00+1)^*$

A. NFA accepting the desired language is the following:



Now we transform this NFA into a DFA.

$\delta(A, 1) = B$

$\delta(B, 0) = C$

$\delta(B, 1) = BEF$

$\delta(C, 0) = B$

$\delta(BEF, 0) = CDG$

$\delta(BEF, 1) = B$

$\delta(CDG, 0) = B$

$\delta(CDG, 1) = DEF$

$\delta(DEF, 0) = DG$

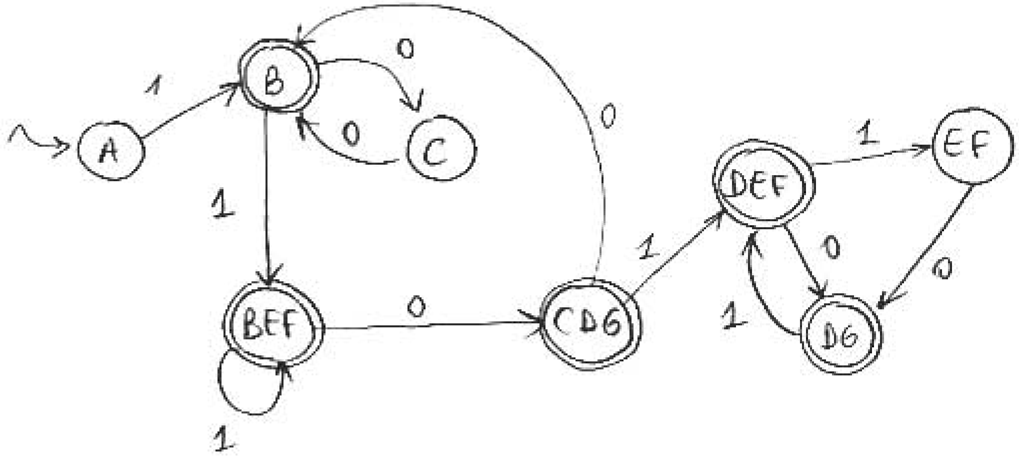
$\delta(DEF, 1) = EF$

$\delta(DG, 1) = DEF$

$\delta(EF, 0) = DG$

The final DFA is :

(E 4.7)



Exercise

Consider a m-state DFA accepting a language L. Prove that if $L \neq \emptyset$ there exists $x \in L$ such that $|x| < m$.

solution

Let w be the shortest string accepted by A. By contradiction, let $|w| \geq m$; then for the pumping lemma $w = xyz$ and $xz \in L$, with $|xz| < |w|$. Contradiction

Exercise

Let A be a DFA with m states, accepting the language L. Prove that

$$L \text{ is infinite } \underline{\text{iff}} \; \exists w \in L \;\middle|\; m \leq w < 2m$$

solution

<u>if</u> By the pumping lemma, L contains infinite strings.

<u>only-if</u>

If L is infinite, there exist $w \in L$ with $|w| \geq m$. If $|w| < 2m$ we are done; otherwise we apply the pumping lemma: $w = xyz$, $xz \in L$. Note that $|xz| < w$ and $|xy| \leq m$. We can repeat this step iteratively until we obtain a string of L of the desired length. Note that the fact that $|xy| \leq m$ implies $|y| \leq m$, so that we are guaranteed that we cannot obtain a string of length $< m$ from a string of length $\geq 2m$.

E 4.9

Corollary (of results of the two previous exercises)

To check whether the language L accepted by a DFA is empty, finite or infinite, we feed the automaton with all strings of length $\leq 2n$, where n is the number of states. We have the following cases:

(a) if no string is accepted, L is empty;

(b) if all accepted strings have length $< n$, the language is finite;

(c) if there is an accepted string of length $\geq n$, the language L is infinite.