Xiaohui Bei

Algorithms and Theory of Computing

MAS714 2021

Exercises on Regular Language

Exercise 1:
If $L \subseteq \{a, b, c\}^*$ is finite, does there always exist a DFA D that accepts L?

Solution: Yes.

- A language L that contains only one string is always regular. One can easily construct a DFA that check every character of this string in sequence.
- Because regular language is closed under union operations. Hence any language of finite size, which can be seen as the union of a finite number of single-string languages, is also regular.

Exercise 2:
Suppose a language $L \subseteq \Sigma^*$ is accepted by a DFA D. Construct a DFA D^C that accepts the language $L^C = \{w \in \Sigma^* \mid w \notin L\}$.

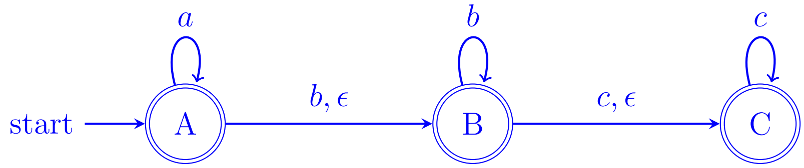
Solution: Let $D = \{Q, \Sigma, \delta, q, F\}$ be the DFA that accepts L. It's easy to check that $D^C = \{Q, \Sigma, \delta, q, Q \backslash F\}$ accepts exactly the language L^C.

Exercise 3:
Construct an NFA N that accepts the language

$$L = \{a^n b^m c^\ell \mid n, m, \ell \geq 0\}.$$

Try to construct N with no more than three states.

Solution:



Exercise 4:
Construct finite automata, deterministic or non-deterministic for the following regular expressions.

(a). $a(a+b)^*b$

(b). $1(1+0)^*0+0$

(c). 0^*10^*

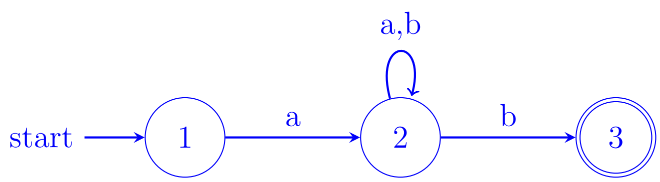
(d). $(0+1)^*1(0+1)^*$

(e). $0^*(1+\epsilon)0^*$

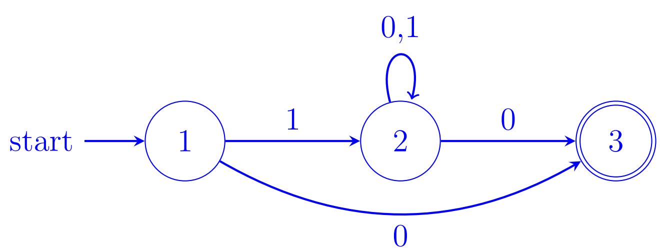
(f). $(0+1)(0+1)(0+1)^*$

Solution:

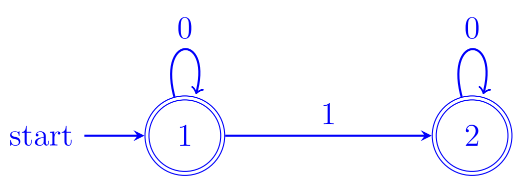
(a).



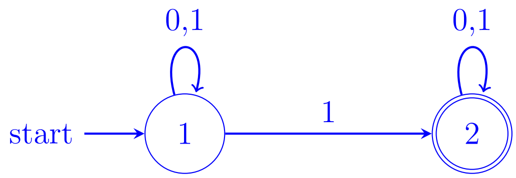
(b).



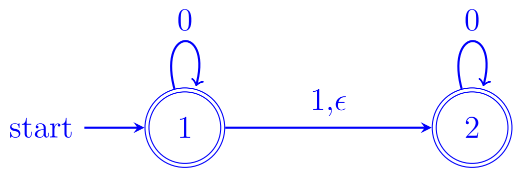
(c).



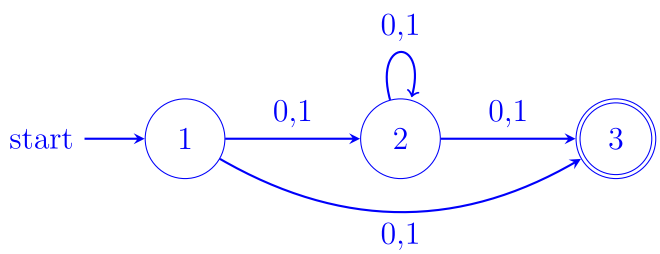
(d).



(e).



(f).



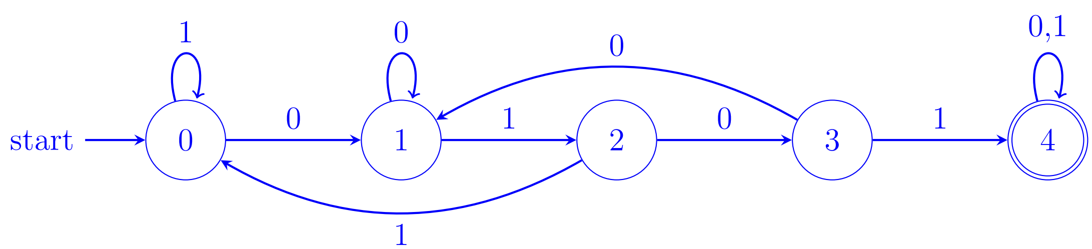
Exercise 5:
Construct a DFA which accepts the following language:

$$L = \{w \in \Sigma^* \mid w \text{ contains the substring } 0101\}$$

That is, $w = x0101y$ for two arbitrary strings x and y.

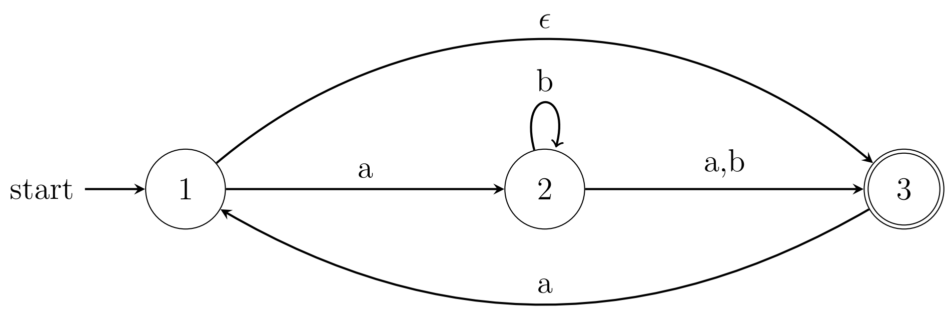
Solution: Note that if the input string w exists in the language L, we eventually will find the substring 0101 in w. We define the states corresponding to how much of the substring 0101 has been observed:

- State 0 is the initial state. In this state no part of the substring 0101 has been observed.
- In state 1, the first zero of 0101 has been observed. If a zero turns up, we continue to search for the next one in the substring 0101.
- In state 2, the substring 01 of 0101 has been observed. If the next character is a one, we return to the initial state since 011 is not a substring of 0101.
- In state 3, the string 010 has been observed. If we add another zero, we return to state 1 - this because the new zero may be the start of a new substring 0101.
- In state 4 we have observed the full substring 0101. Thus, it doesn't matter what other characters there are in the string, since we already have concluded that the string exists in the language. Consequently, state 4 is an accepting state.

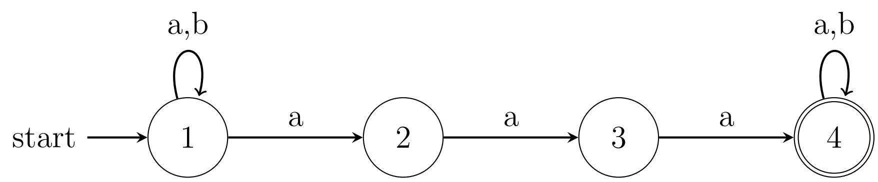


Exercise 6:
Convert each of the NFA's below to an equivalent DFA by subset construction.

(a).

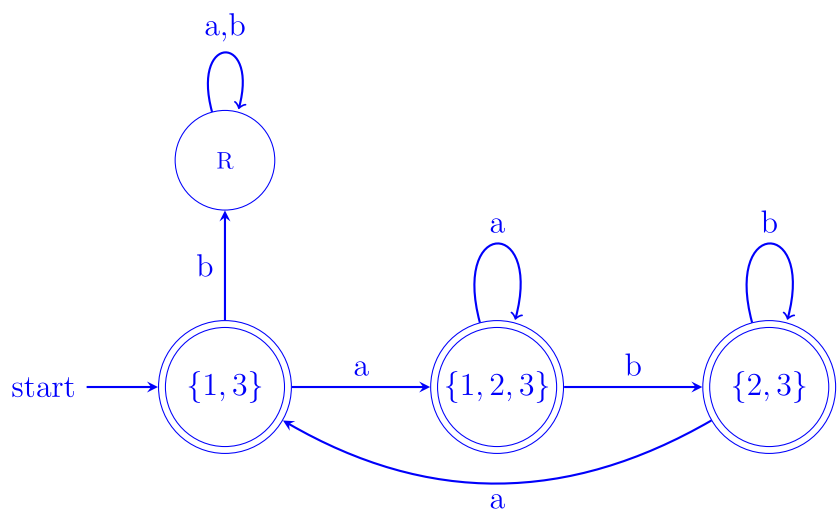


(b).

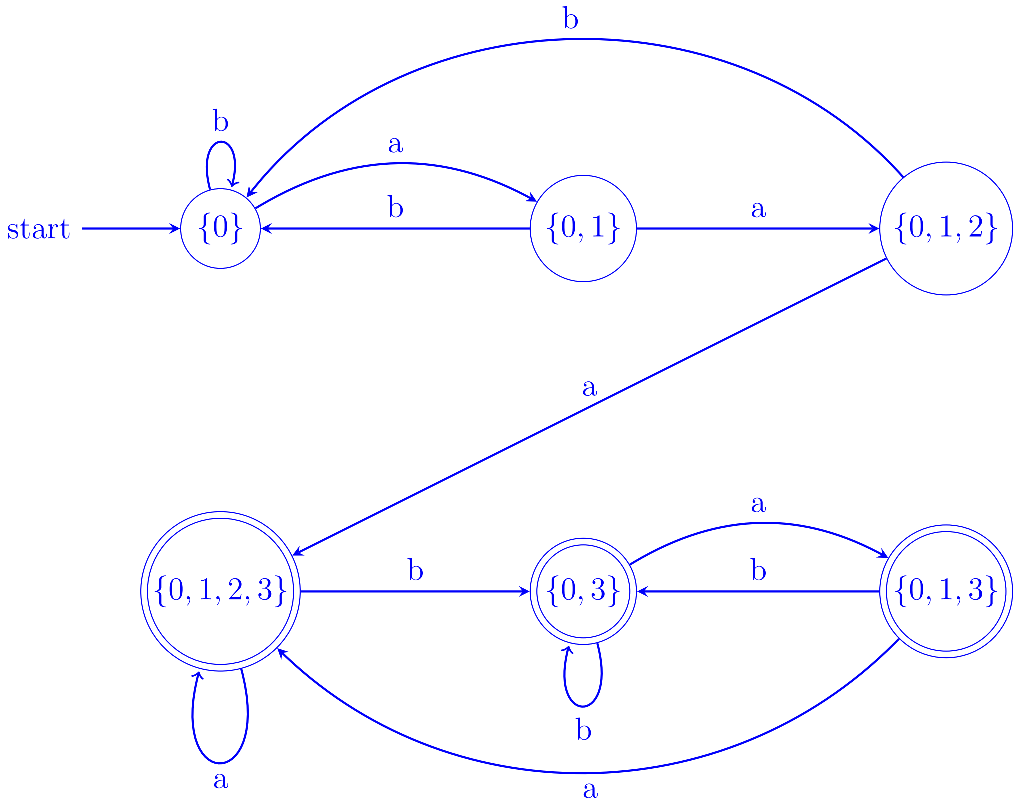


Solution:

(a).



(b).



Exercise 7:
Construct regular expressions representing languages, over the alphabet $\{a, b, c\}$, in which for every string w it holds that:

(a). The number of a's in w is even.

(b). $|w| = 3i.(i \geq 0)$

Solution:

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

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

Exercise 8:
Find a regular expression for the language consisting of alternating zeroes and ones.

Solution:

- $(01)^*$ is the language of zero or more 01.
- $(1+\epsilon)(01)^*$ is the language of alternating zeros and ones which ends in 1.
- $(1+\epsilon)(01)^*(0+\epsilon)$ is the language of alternating zeroes and ones.

Alternatively, $(0+\epsilon)(10)^*(1+\epsilon)$ is also correct.

Exercise 9:
Describe which languages the following regular expressions represents, using common english.

(a). $(0+1)^*01$

(b). 1^*01^*

(c). $(11)^*$

(d). $(0+1)^*01(0+1)^*$

(e). 1^*0^*

(f). $(10+0)^*(1+10)^*$

(g). $0^*(1+000^*)^*0^*$

Solution:

(a). All binary strings that end with the substring 01.

(b). Binary strings that contain exactly a zero.

(c). Strings consisting only of ones and which lengths are even.

(d). Strings containing the substring 01.

(e). Strings on the form $111\ldots000\ldots$, that is, strings that begins with zero or more ones followed by zero or more zeroes.

(f).
- $(10+0)^*$ is all strings which doesn't contain the substring 11.
- $(1+10)^*$ is all strings which doesn't contain the substring 00.

so the concatenation of these is all strings where each occurrence of 00 precedes all occurrences of 11.

(g). All strings which doesn't contain the substring 101.

Exercise 10:
Are the following languages regular? Prove your claims.

(a). $L = \{a^i b^j a^{ij} \mid i, j \geq 0\}$

(b). $L = \{b^2 a^n b^m a^3 \mid m, n \geq 0\}$

(c). $L = \{a^{k^3} \mid k \geq 0\}$

Solution:

(a). The language is not regular. To show this, let's suppose L to be a regular language with pumping length $p > 0$. Furthermore, let's consider the string $w = a^p b^p a^{p^2}$. It is apparent that $|w| \geq p$ and $w \in L$. According to the pumping lemma, $w = xyz$ where

- $|xy| \leq p$.
- $y \neq \epsilon$.
- $xy^k z \in L$ for all $k > 0$.

Consequently, $xy^0z = xz$ must belong to L. Since $|xy| \leq p$ and $|y| > 0$, then it is easy to see that $xy^0z = a^{p-|y|}b^p a^{p^2}$ is not a member of L. Thus, L is not regular.

(b). The Language is regular. Indeed, it can be expressed by the regular expression $bba*b*aaa$.

(c). The language is not regular. Again, let's suppose that L is regular with pumping length $p > 0$. The string $w = a^{p^3}$ contradicts the pumping lemma. Indeed, if $w = xyz$ so that the statement of the pumping lemma holds, then it is easy to see that $xy^k z = a^{p^3+(k-1)|y|}$. However, if such y existed, then $p^3 + (k-1)|y| = n(k)^3$ for every $k \geq 0$, where $n(k) \in \mathbb{N}$ depends upon k, which is trivially false.