

Zad. 10

a) $1^*(01^*)?(01^*)?(01^*)?$

b) $(00|0)^(1+(00|0)^*)$

c) Regexy nie mają możliwości zliczania wystąpień znaków, więc nie da się utworzyć regexu, który będzie określał język ogółu sekwencji binarnych, w których liczba zer jest podzielna przez dwa, a liczba jedynek przez 3.

Takie wyrażenie regularne musiałoby akceptować wyrażenia typu:

- [illegible]

Biorąc pod uwagę chociażby ten ostatni ciąg, musielibyśmy znaleźć sposób, żeby po pierwszym zerze zagwarantować również drugie, które jest oddalone od niego o 48 jedynek, z tego też powodu hipotetyczne wyrażenie regularne, które byłoby zgodne z wymaganiami musiałoby wyglądać tak:

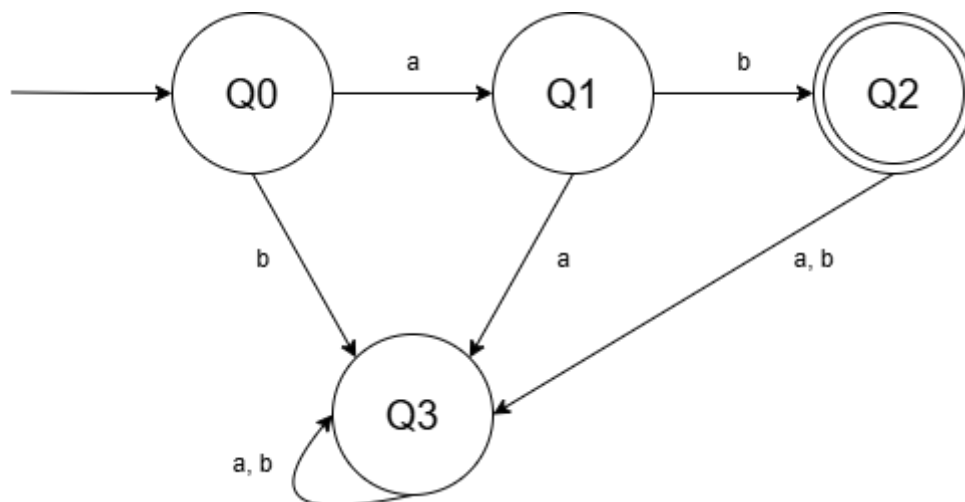
(nieskończenie długa lista akceptowanych podciągów połączonych ze sobą alternatywami)*.

Najkrótsze podciągi w tym regexie miałyby 5 znaków i byłyby permutacjami 11100, a najdłuższe miałyby nieskończenie wiele znaków.

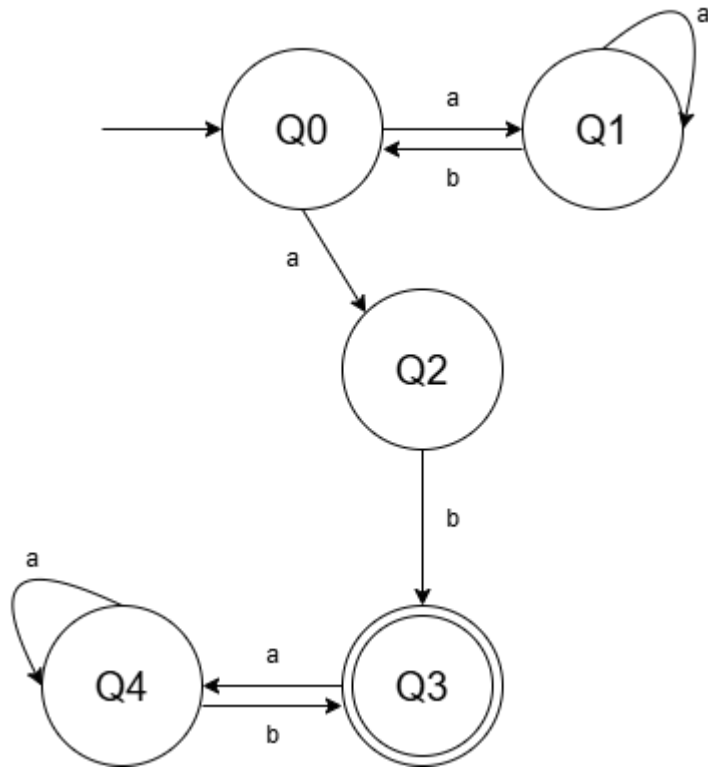
d) $(a|b|c)\{1,4\}@ (a|b|c)\{1,4\}\backslash.(a|b|c)\{2\}$

Zad. 11

a)



b)



Zad. 12

Język L ciągów binarnych, w których sekwencja 11 występuje dokładnie raz da się określić wyrażeniem regularnym: $(10|0)^*11(01|0)^*$