

Discrete Event Systems

Exercise Sheet 2

1 Nondeterministic Finite Automata

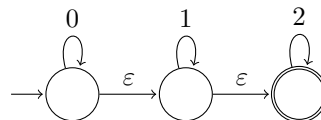
- Consider the alphabet $\{a, b\}$. Construct an NFA that accepts all strings containing the substring $abba$ at least twice. (This means that words containing $abbaabba$ as a substring should also be accepted!)
- Construct an NFA which accepts the following regular expression: $(00 \cup (0(0 \cup 1)^*))^*$.
- Construct an NFA accepting $1^*0^*1^+$ with as few states as possible. (cf. Exercise 1.1.a)
- Consider a machine $M := (Q, \Sigma, \delta, q_0, Q)$. Is it possible to make a statement about the strings being accepted by M ? Does it make a difference whether M is deterministic or not?

2 Exam question [2018]

Assume that the alphabet Σ is $\{0, 1\}$ and consider the language $L = \{w \mid \text{there exists two zeros in } w \text{ that are separated by a string whose length is } 4i \text{ for some } i \geq 0\}$. For example the strings 1001 and 10110101 belong to L , whereas the strings 101 and 010101 do not. Design an NFA that recognizes L with 6 states or less.

3 De-randomization

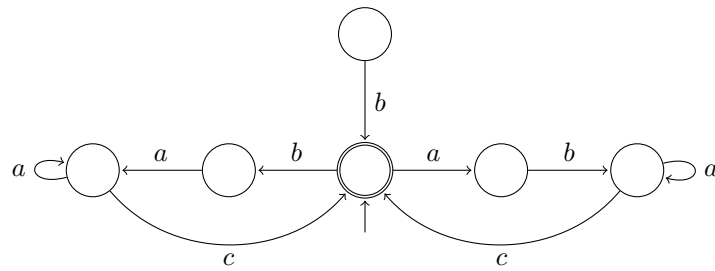
- Give a regular expression for the following NFA and construct an equivalent NFA *without* ε -transitions.



- Finally, transform the machine into a deterministic automaton.

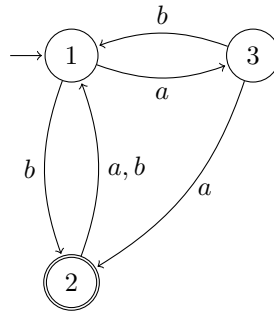
4 States Minimization

Simplify the following automaton. Explain why your changes are allowed. Finally, give the corresponding regular expression.



5 From DFA to Regular Expression

Build a language-equivalent GNFA for the DFA below. Then, as seen during the lecture, eliminate the intermediate states of the GNFA and derive the regular expression that represents the language accepted by the DFA.



6 “Regular” Operations in UNIX

In this exercise you are asked to provide a UNIX command to output all lines in a file ending with “password” or “passwort”, followed by an unknown number (potentially zero) of vowels.