Discrete Event Systems

Exercise Session 2

Roland Schmid

nsg.ee.ethz.ch

ETH Zürich (D-ITET)

30 September 2021

1 Nondeterministic Finite Automata

a) Consider the alphabet $\{a, b\}$. Construct an NFA that accepts all strings containing the substring $a\,b\,b\,a$ at least twice. (This means that words containing $a\,b\,b\,a\,b\,b\,a$ as a substring should also be accepted!)

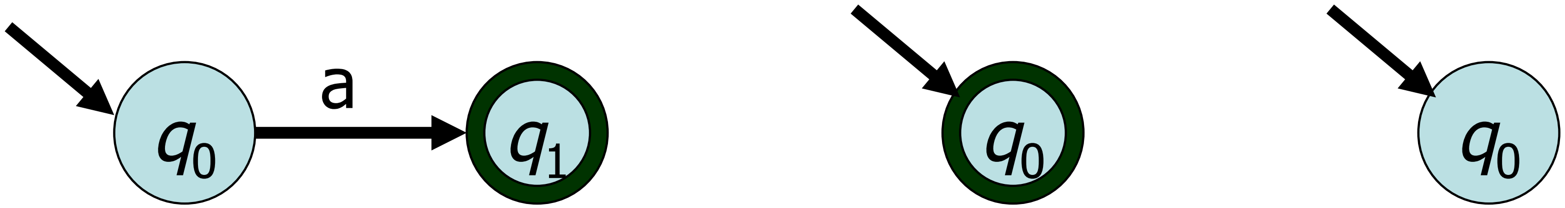
b) Construct an NFA which accepts the following regular expression: $(00 \cup (0(0 \cup 1)^*))^*$.

c) Construct an NFA accepting $1^*0^*1^+$ with as few states as possible. (cf. Exercise 1.1.a)

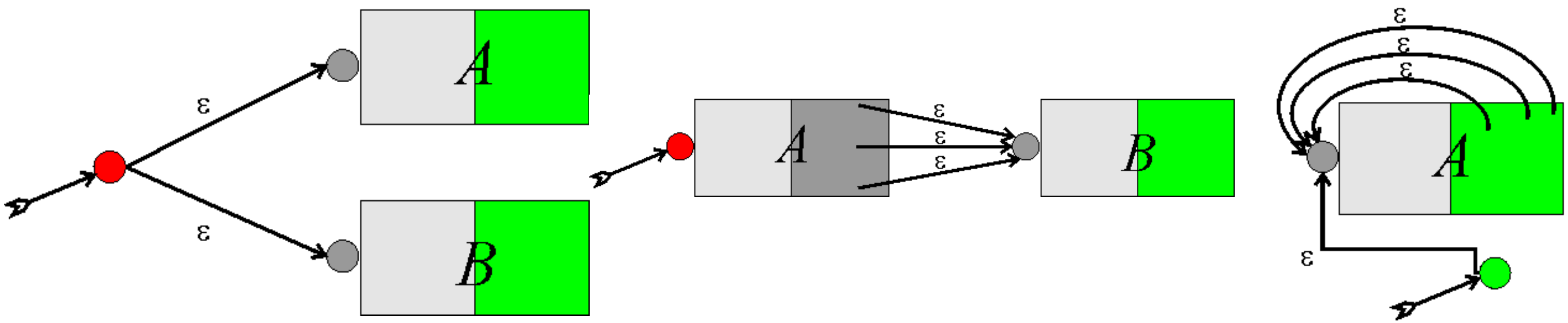
d) Consider a machine $M := (Q, \Sigma, \delta, q_0, Q)$. Is it possible to make a statement about the strings being accepted by M? Does it make a difference whether M is deterministic or not?

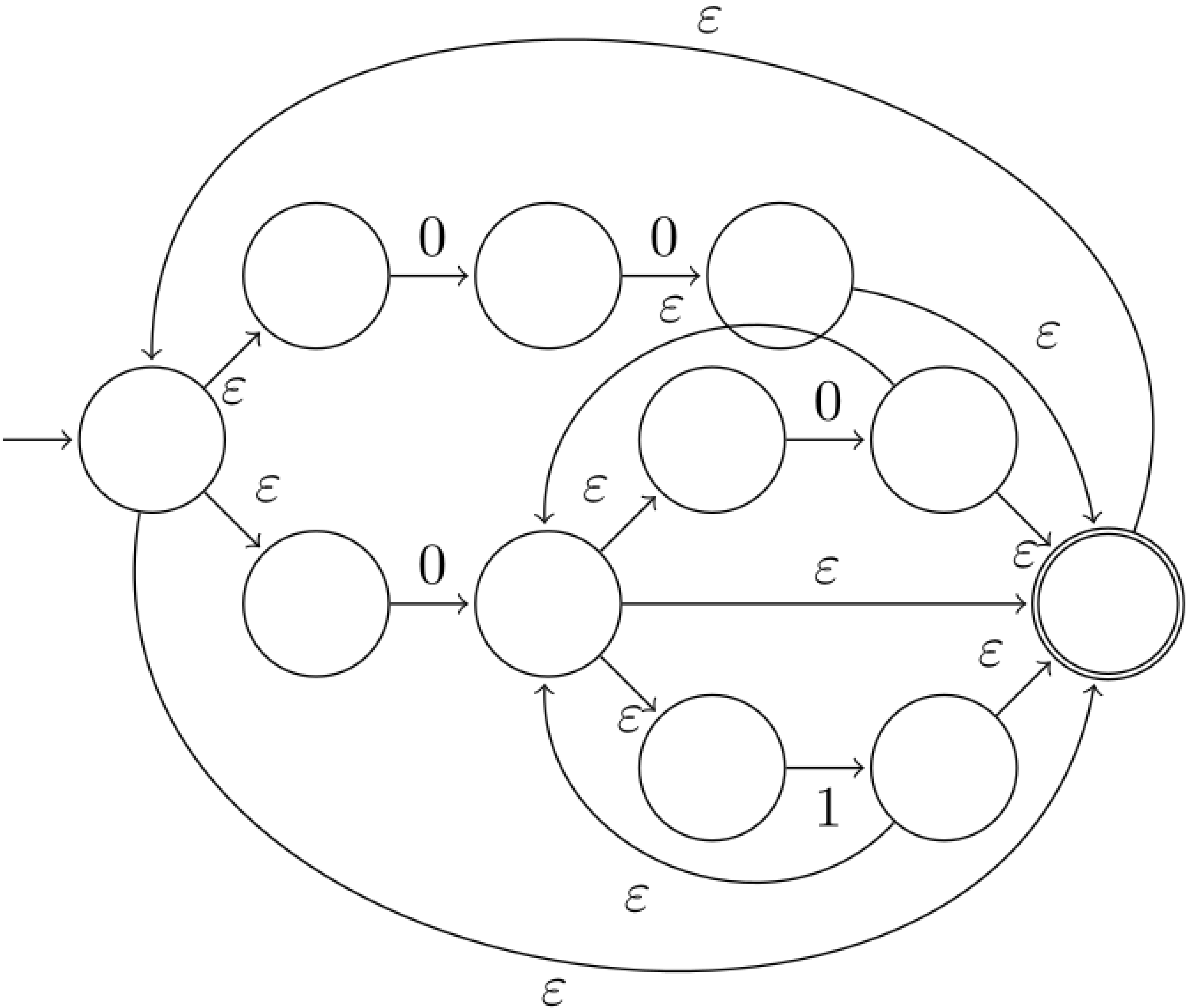
REX → NFA

- *Proof*: The proof works by induction, using the recursive definition of regular expressions. First we need to show how to accept the base case regular expressions $a \in \Sigma$, ε and $\emptyset$. These are respectively accepted by the NFA's:



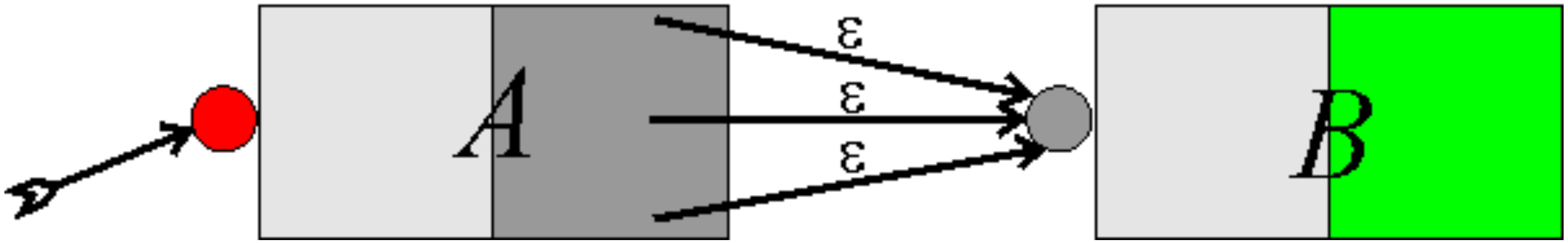
- Finally, we need to show how to inductively accept regular expressions formed by using the regular operations. These are just the constructions that we saw before, encapsulated by:




ε
0
0
ε
ε
ε
0
ε
ε
0
ε
ε
ε
ε
1
ε
ε

NFA: Concatenation

- The concatenation *A•B* is formed by putting the automata in serial. The start state comes from *A* while the accept states come from *B*. *A*'s accept states are turned off and connected via ε-edges to *B* 's start state:

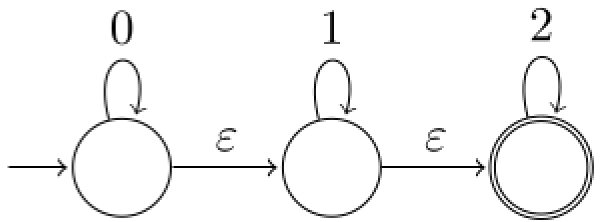


2 Exam question [2018]

Assume that the alphabet Σ is $\{0, 1\}$ and consider the language $L = \{w \mid$ there exist two zeros in w that are separated by a string whose length is $4i$ 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

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

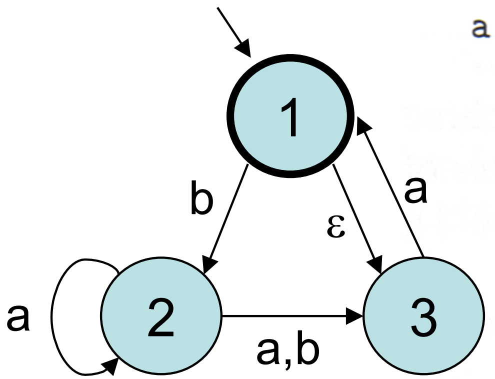


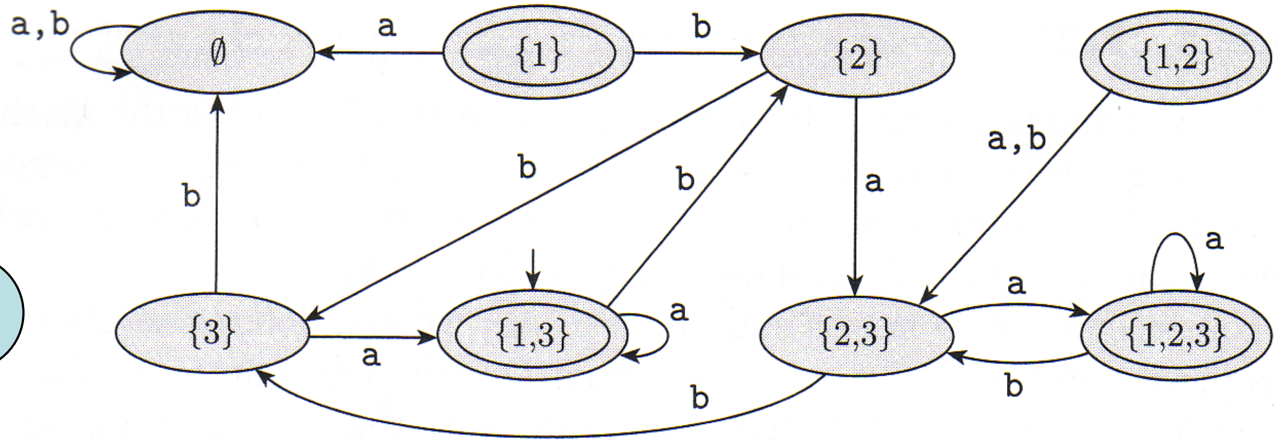
b) Finally, transform the machine into a deterministic automaton.

Determinizing NFA's: Example

- Idea: We might keep track of all parallel active states as the input is being called out. If at the end of the input, one of the active states happened to be an accept state, the input was accepted.

- Example, consider the following NFA, and its deterministic FA.





NFA's have 3 types of non-determinism

Nondeterminism type	Machine Analog	δ -function	Easy to fix?	Formally
Under-determined	Crash	No output	yes, fail-state	$\|\delta(q,a)\|= 0$
Over-determined	Random choice	Multi-valued	no	$\|\delta(q,a)\|> 1$
ε	Pause reading	*Redefine alphabet*	no	$\|\delta(q,\varepsilon)\|> 0$

One-Slide-Recipe to Derandomize

- Instead of the states in the NFA, we consider the power-states in the FA. (If the NFA has n states, the FA has 2^n states.)

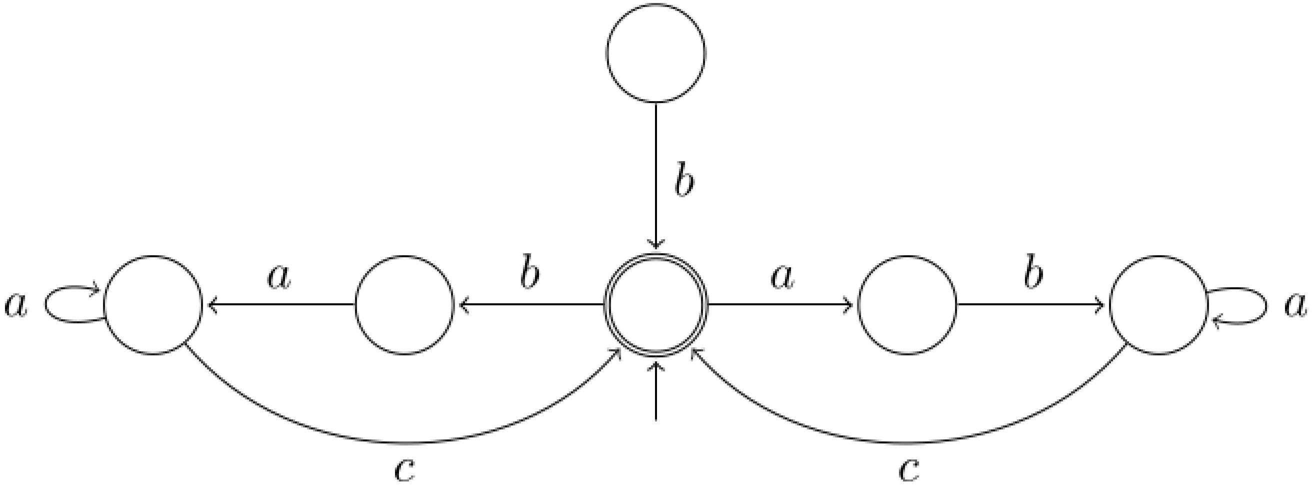
- First we figure out which power-states will reach which power-states in the FA. (Using the rules of the NFA.)

- Then we must add all epsilon-edges: We redirect pointers that are initially pointing to power-state {a,b,c} to power-state {a,b,c,d,e,f}, if and only if there is an epsilon-edge-only-path pointing from any of the states a,b,c to states d,e,f (a.k.a. transitive closure). We do the very same for the starting state: starting state of FA = {starting state of NFA, all NFA states that can recursively be reached from there}

- Accepting states of the FA are all states that include a accepting NFA state.

4 States Minimization

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



Minimization

- Definition: An automaton is irreducible if
  - it contains no useless states, and
  - no two distinct states are equivalent.

- By just following these two rules, you can arrive at an “irreducible” FA. Generally, such a local minimum does not have to be a global minimum.

- It can be shown however, that these minimization rules actually produce the global minimum automaton.

- The idea is that two prefixes u,v are indistinguishable iff for all suffixes x, $ux \in L$ iff $vx \in L$. If u and v are distinguishable, they cannot end up in the same state. Therefore the number of states must be at least as many as the number of pairwise distinguishable prefixes.

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.