Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

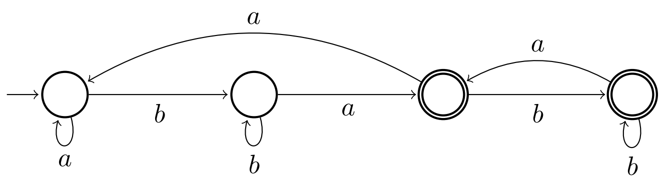
Distributed Computing Group

HS 2009 Prof. Dr. R. Wattenhofer / Jasmin Smula / Raphael Eidenbenz

Discrete Event Systems
Solution to Exercise 2

1 Filter for an Input Stream [exam problem]

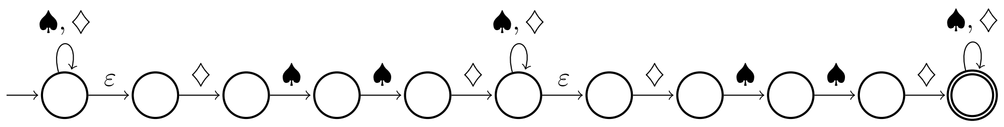
The following figure gives an example with four states.



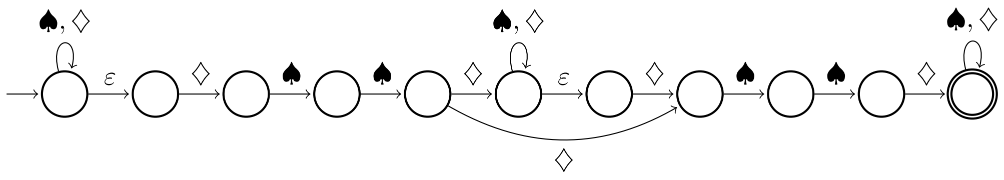
The main idea here is to use two different accepting states after having read a: one for zero b's and one for more than zero b's.

2 Nondeterministic Finite Automata

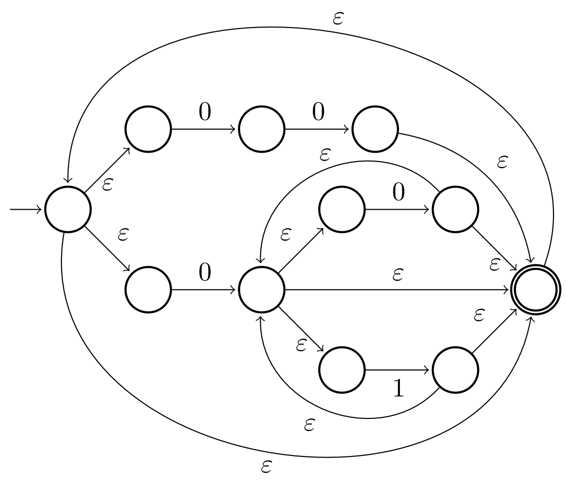
a) The following automaton is an example. (Note that this is not the minimal solution!)



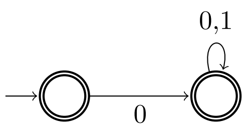
If you also want to allow strings containing ♢♠♠♢♠♠♢ as a substring, the automaton may look as follows:



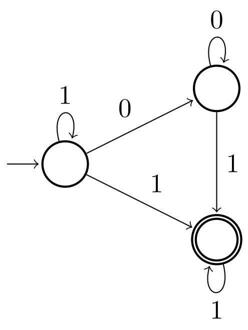
b) The following automaton is an example.



Of course this automaton is not minimal. If you take a closer look at the regular expression $(00 \cup (0(0 \cup 1)^*))^*$, you will see that it can be simplified to $0(0 \cup 1)^*$. A minimal automaton for this regular expression is given by



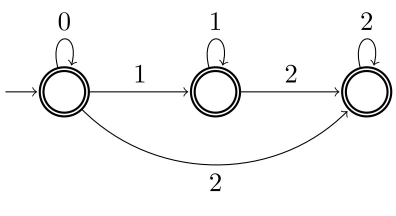
c) Three states are enough if we can use NFAs. With FAs, we needed at least four states.



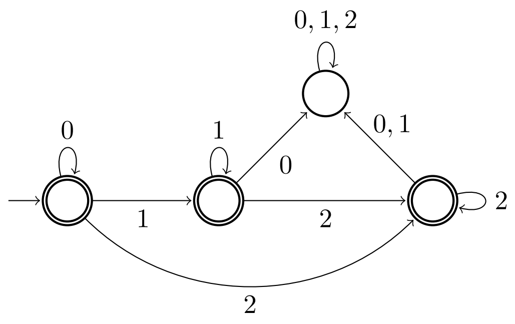
d) A deterministic machine whose states are all accepting accepts *every* string of the corresponding alphabet. However, this does not hold for a nondeterministic automaton, namely if it is under-determined.

3 De-randomization

a) The automaton accepts the strings $0^*1^*2^*$. Without ε-transitions we have

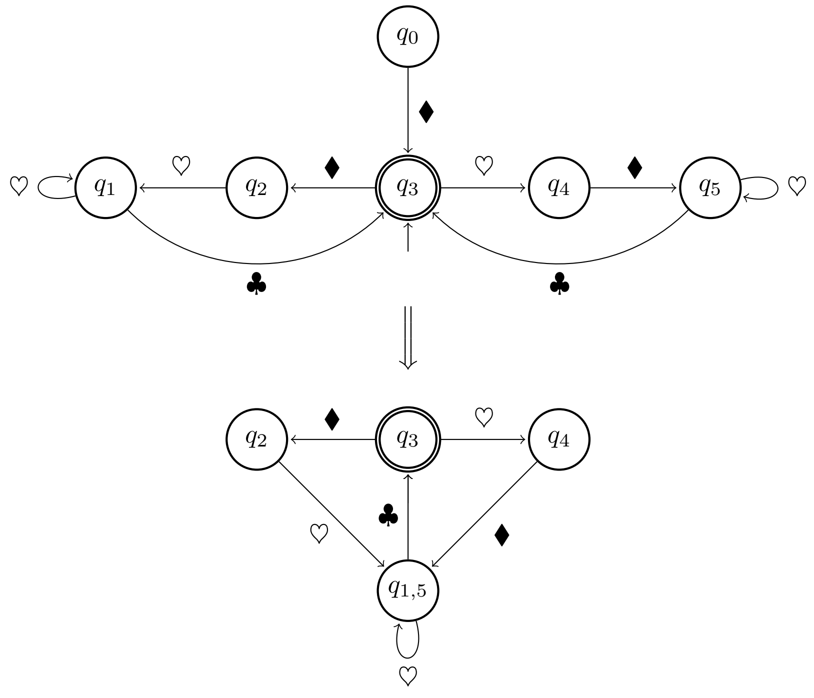


b) The nondeterministic automaton can be found by applying the power set construction presented in the lecture followed by the state minimization algorithm. However, it is obvious that the automaton shown below does the job.



4 States Minimization

State q_0 can be omitted as it is not reachable. Moreover, states q_1 and q_5 can be merged, as there is no input sequence which will show a difference between these two states. The REX of the language is $((\blacklozenge\heartsuit \cup \heartsuit\blacklozenge)\heartsuit^*\clubsuit)^*$.



5 "Regular" Operations in UNIX

In UNIX, the special symbol "$" stands for the end of a line. We have:

```
egrep '(password|passwort)(a|e|i|o|u|A|E|I|O|U)*$'
```