

Theorievraag 1:

1. Finite state machine; alfabet van letters
2. Turingmachine; alfabet van letters inclusief de twee symbolen $|>$ en $|_<$

Theorievraag 2:

1. Een deterministisch proces is een opeenvolging van uitkomsten die een causaal verband hebben. (=wet van oorzaak gevolg). Voor de taal/automaton betekent dit dat elke staat invloed heeft op de volgende gebeurtenis.

Niet-deterministische finite automaton (NFA)

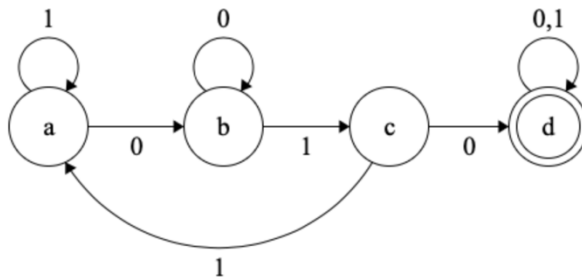
Niet alle overgangen zijn beschikbaar

Deterministische finite automaton (DFA)

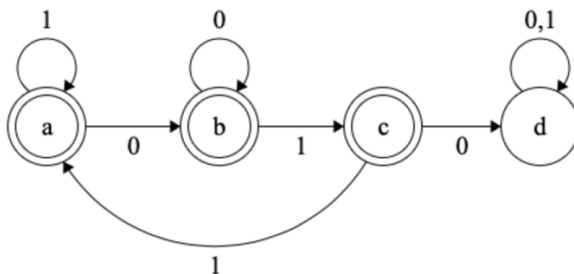
Met voor elke staat een overgang bij 0 en 1.

Oefeningen

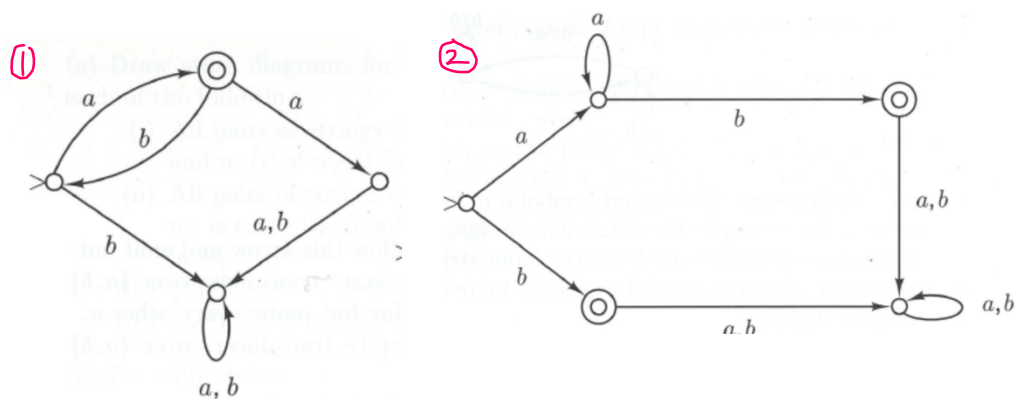
1. DFA die alleen string $\{0,1\}$ accepteert met substring 010.



2. DFA met string $\{0,1\}$ die substring 010 niet accepteert.



3.



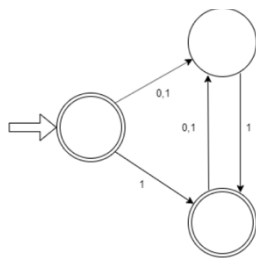
1. Er is hier een final state waarbij die altijd moet eindigen op een a. De uitkomsten hier kunnen zijn oneindig vaak ababa.... en alleen a.

2. Hier zijn er twee final states, onderin kun je alleen eindigen op b voor een juiste uitkomst en rechts bovenin moet het eindigen op b.
aab, ab, b,

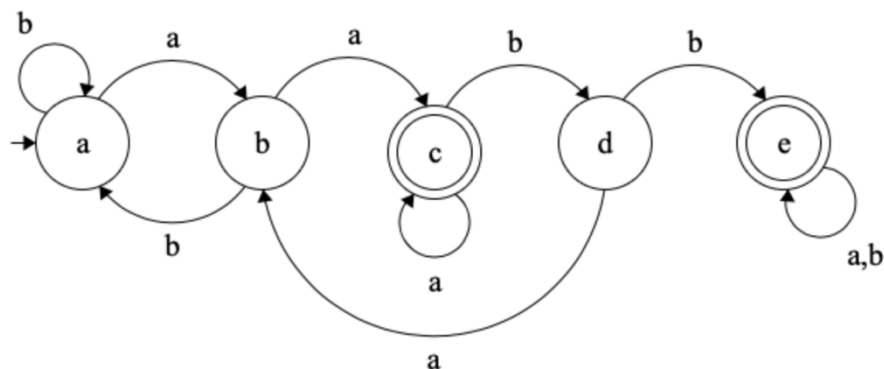
In beide gevallen heb je te maken met determinisme omdat alle staten naar a en b kunnen.

3. NFA;

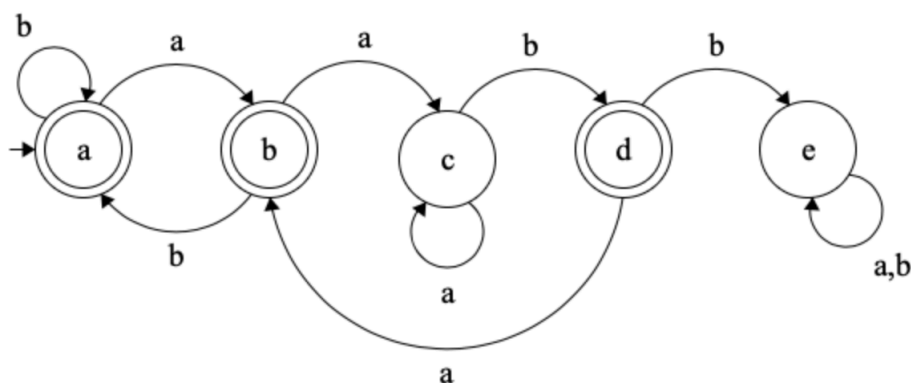
Dit automaton heeft twee final states maar bij de begin state kan die niet stoppen zonder een string aan te nemen, rechts onderin is dus de correcte final state. De string eindigt altijd met een 1 maar kan zowel met een 1 of 0 beginnen. Dit is een niet deterministische automaton omdat niet alle staten naar een 0 en 1 kunnen.



4. Teken een automaton voor de taal {a,b} die zowel substring aa als bb niet bevat.
- Eerst **wel** {aa} en {bb} toestaan als substring

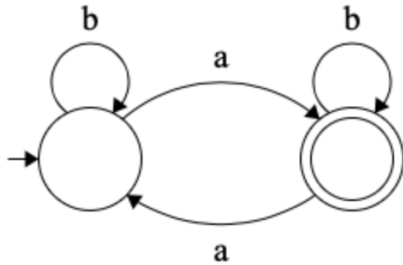


- Convert naar **niet** {aa} en {bb} toestaan als substring

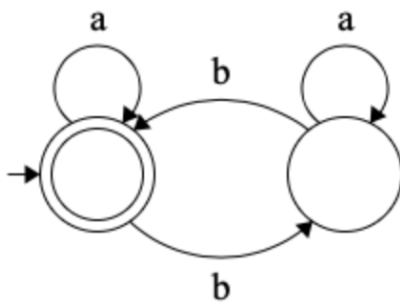


5. Teken een automaton voor de taal $\{a,b\}$ die een oneven aantal 'a' en even aantal 'b' bevat.

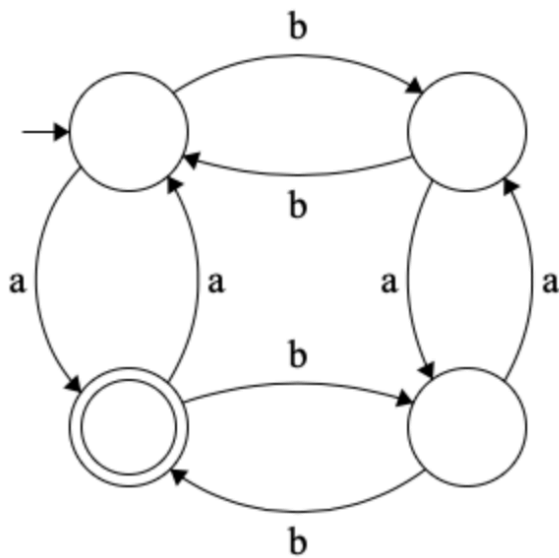
- Oneven aantal 'a'



- Even aantal 'b'



- Merge deze automata met elkaar



a, bab, aabbbbaabbb etc

Turing machine

1. Gegeven een startstring $a \ q \ b \ b \ _ _ \ b \ b \ _ _ _ \ a \ b \ a \ q_0$. Beschrijf wat deze machine doet:
 - TM begint bij q_0 en leest a , transitie links naar q_1 .
 - Bij q_1 heeft die de waarde b en moet rechts naar q_2 .
 - Bij q_2 waarde a , rechts naar q_2 ????

De machine $M = (K, \Sigma, \delta, s, F)$, waarbij:

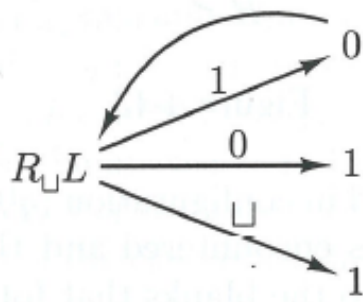
- $K = \{q_0, q_1, q_2, h\}$
- $\Sigma = \{a, b, \sqcup\}$
- $s = q_0$
- $H = \{h\}$

En δ is gegeven door de volgende tabel.

q	σ	$\delta(q, \sigma)$
q_0	a	(q_1, \leftarrow)
q_0	b	(q_0, \rightarrow)
q_0	\sqcup	(q_0, \rightarrow)
q_1	a	(q_1, \leftarrow)
q_1	b	(q_2, \rightarrow)
q_1	\sqcup	(q_1, \leftarrow)
q_2	a	(q_2, \rightarrow)
q_2	b	(q_2, \rightarrow)
q_2	\sqcup	(h, \sqcup)

2. Ontwerp een Turing machine in formele notatie die naar rechts scant net zo lang tot hij twee a 's achter elkaar tegenkomt en dan stopt.
3. Ontwerp een Turing machine in recursieve notatie die een invoerstring zonder blanks een positie naar links verschuift. De startpositie is links van de string.

4. Hieronder is een Turing machine recursief beschreven. Beschrijf wat deze machine doet. Bedenk een aanpassing op deze machine.



(R_ en L)

De machine vindt het rechter einde van de input, gaat naar links tot dat het 1's ziet en verandert het naar 0's. Wanneer het een 0 ziet, verandert het naar een 1 en stopt.

- Eigen aanpassing:

State table for 3 state, 2 symbol busy beaver

Tape symbol	Current state A			Current state B			Current state C		
	Write symbol	Move tape	Next state	Write symbol	Move tape	Next state	Write symbol	Move tape	Next state
0	1	R	B	1	L	A	1	L	B
1	1	L	C	1	R	B	1	R	HALT