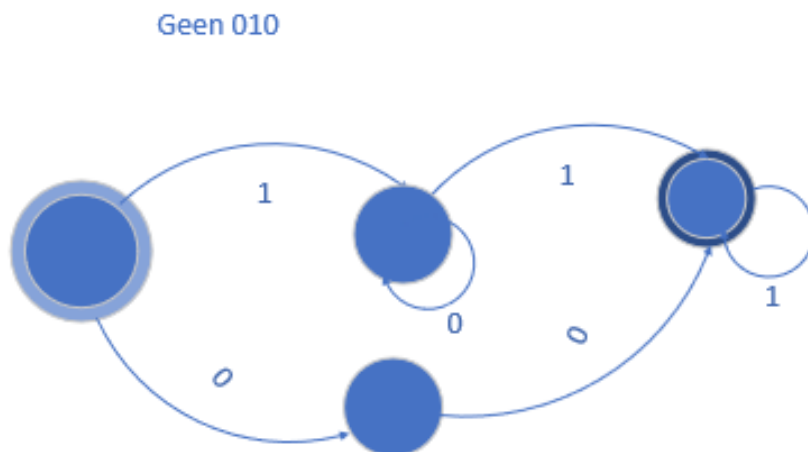
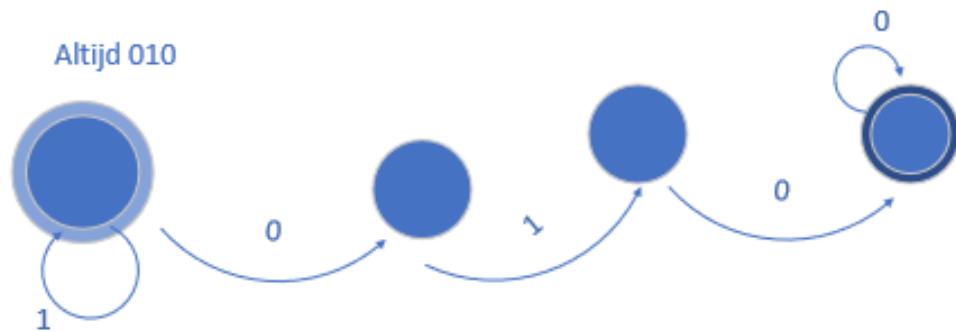
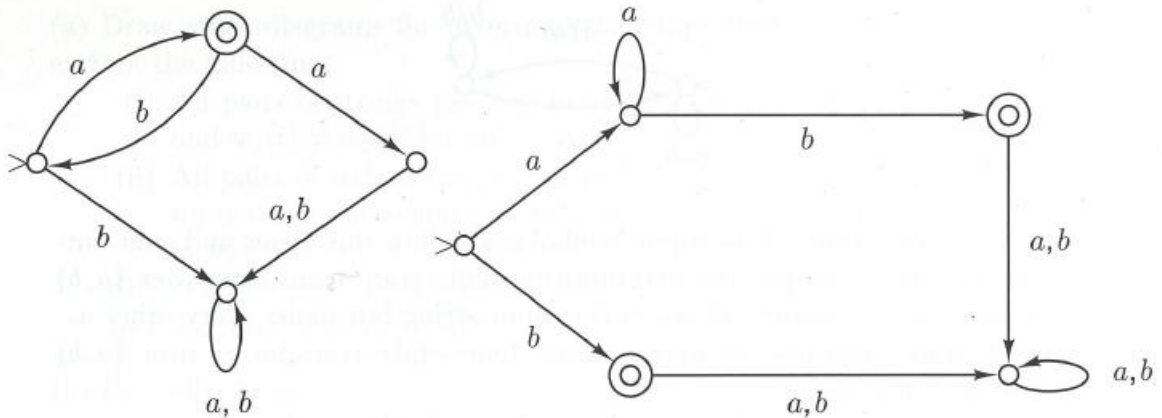


Automata:

1. Ontwerp een deterministische automaton die alleen maar die strings met het alfabet $\{0,1\}$ accepteert die de substring "010" bevatten. Ontwerp vervolgens ook de tegenovergestelde automaat, die juist geen substring "010" wil aantreffen.

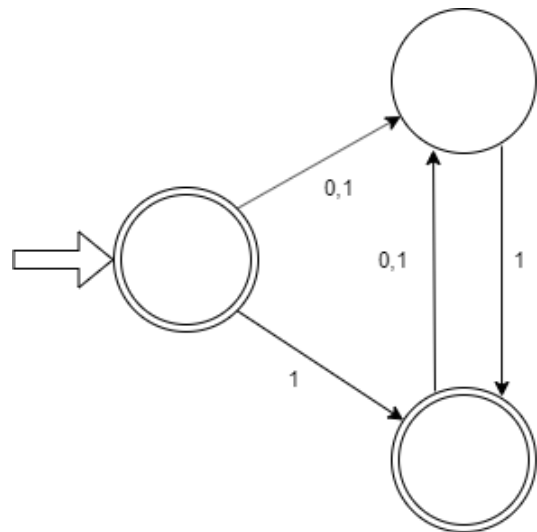


2. Hieronder staan een aantal automata. Beschrijf in je eigen woorden welke talen deze beschrijven.

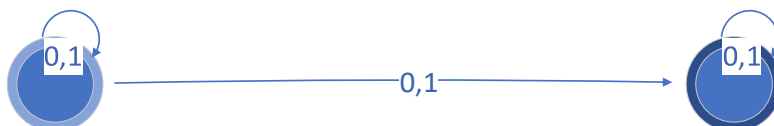


1.
Er kunnen alleen string van $(AB^*)A$ gemaakt worden, of A .
2.
Er kunnen alleen strings van B , of AB , of $(A^*)B$ gemaakt worden. Alleen eindigen op B .

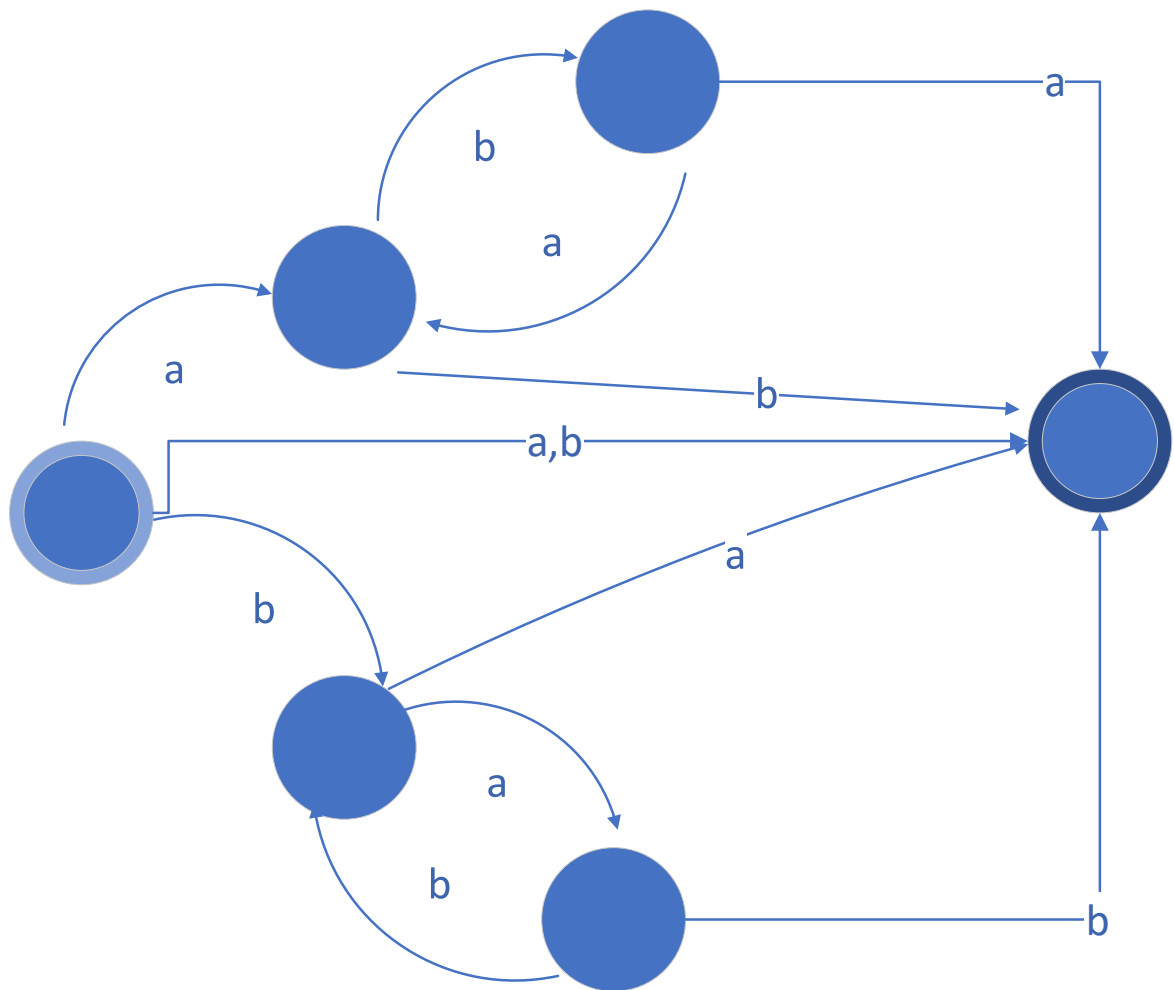
3. Hieronder staat een niet-deterministische automaton (niet alle overgangen zijn beschikbaar). Beschrijf in je eigen woorden welke taal deze machine accepteert en ontwerp vervolgens een eigen deterministische automaton (dus met voor elke staat overgangen bij 0 en bij 1).



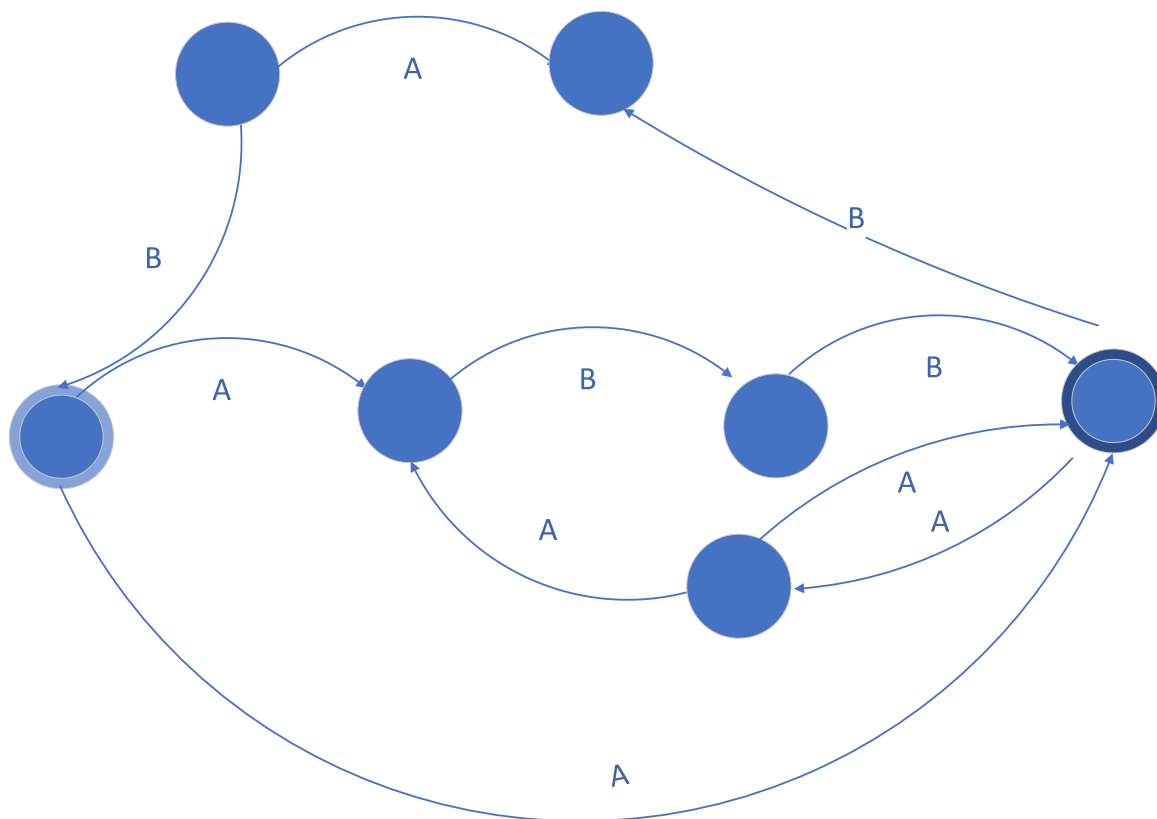
Wat opvalt aan dit niet-deterministische automaton is dat de staat altijd eindigt op 1. Er gaan twee pijlen naar de endnode en die geven beide een 1. Je kunt ook nooit van staat 0 naar 0 gaan. Er komt dan altijd een 1 tussen. De output kan nooit uit alleen nullen bestaan.



4. Teken een automaton, voor de taal $\{a, b\}$ die zowel de substring "aa" als "bb" niet bevat.



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



1. Hieronder is een Turing machine formeel beschreven. Gegeven de startstring “a q b b \sqcup b b \sqcup \sqcup \sqcup a b a q0”, beschrijf wat de machine doet.

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)

a q b b ⊔ b b ⊔ ⊔ ⊔ a b a	q0
a b q b ⊔ b b ⊔ ⊔ ⊔ a b a	q0
a b b q ⊔ b b ⊔ ⊔ ⊔ a b a	q0
a b b ⊔ q b b ⊔ ⊔ ⊔ a b a	q0
a b b ⊔ b q b ⊔ ⊔ ⊔ a b a	q0
a b b ⊔ b b q ⊔ ⊔ ⊔ a b a	q0
a b b ⊔ b b ⊔ ⊔ ⊔ q a b a	q0
a b b ⊔ b q b ⊔ ⊔ ⊔ a b a	q1
a b b ⊔ b q b ⊔ ⊔ ⊔ a b a	q1
a b b ⊔ b b q ⊔ ⊔ ⊔ a b a	q2
a b b ⊔ b b q ⊔ ⊔ ⊔ a b a	h

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.

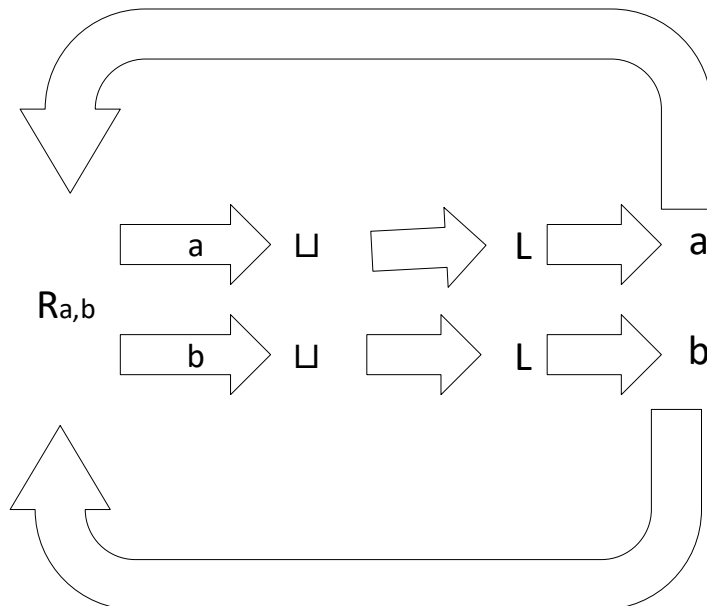
De machine $M = (K, \Sigma, \delta, s, F)$, waarbij:

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

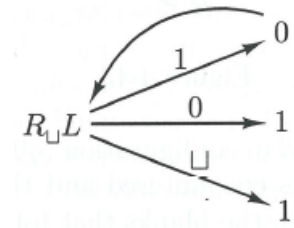
En δ is gegeven door de volgende tabel.

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

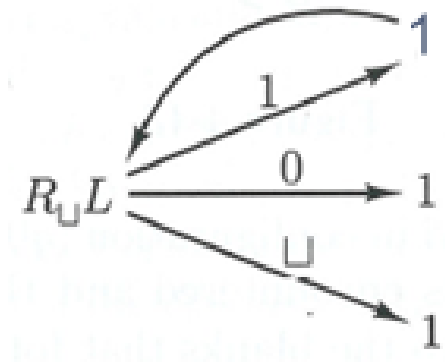
3. Ontwerp een Turing machine (in recursieve notatie) die een invoerstring (zonder blanks) een positie naar links verschuift. De startpositie is links van de string. De machine moet dus " $\sqcup q \sqcup w \sqcup$ " vervangen door " $q \sqcup w \sqcup \sqcup$ ", waarbij "w" een willekeurige string is, bijvoorbeeld "a a b b"



4. Hieronder is een Turing machine recursief beschreven. Beschrijf wat deze machine doet. Bedenk een aanpassing op deze machine, geef wat de aangepaste machine doet en teken die machine.



Ga naar rechts totdat de cursor een \sqcup ziet, zet dan een stap naar links. Als de cursor dan een 1 ziet, zet er een 0 neer.
 Als de cursor dan een 0 ziet, zet er een 1 neer.
 Als de cursor dan een \sqcup ziet, zet er een 1 neer.



Ga naar rechts totdat de cursor een \sqcup ziet, zet dan een stap naar links. Als de cursor dan een 1 ziet, zet er een 1 neer.
 Als de cursor dan een 0 ziet, zet er een 1 neer.
 Als de cursor dan een \sqcup ziet, zet er een 1 neer.
 Hiermee komt er een 1 2 plekken naast elke \sqcup .