week 1 les 1

ST opdr.

Opdracht 1 theoretische en oefen opdrachten DES

Theorie vragen

1. * regular languages
Alle finite languages zijn regular.
empty language (Ø), singleton,
union, concatenation, Kleene star

geaccepteerd door:
FSM, NFA, DFA, read-only Turing machine, alternating finite machine

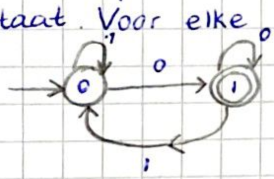
Vb. van geen regular language: $\{a^n b^n \mid n \geq 0\}$.
Een finite automaton heeft een eindig geheugen kan niet het exacte aantal a's onthouden.

* recursively enumerable languages
Een taal waarvoor een Turing machine bestaat die elke string in die taal accepteert, en elke string die niet in de taal voorkomt mag afwijzen. Alle regular en recursieve talen zijn recursively enumerable.

geaccepteerd door:
Turing machine

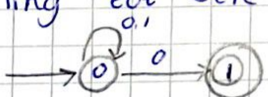
Wikipedia – Automata Theory / Regular language / recursively enumerable languages

2. DFA: op basis van de input gaat de machine naar één bepaalde staat. Voor elke staat en elk symbool is er een transitie gedifineerd.



← accepteert strings eindigend op 0.

NFA: op basis van de input kunnen er meerdere staten geselecteerd worden. Ook kan bij een NFA een null input mee krijgen, in tegenstelling tot een DFA.

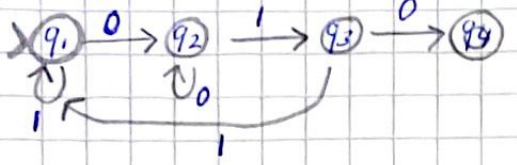


← er zijn meerdere paden voor "00". Omdat één pad leidt tot de final state, is "00" geaccepteerd.

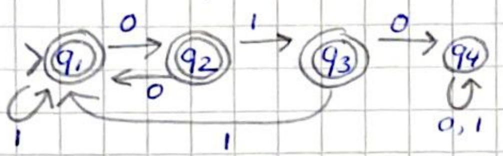
Oefeningen

Automata

1. - accepteer alleen strings met subset 010

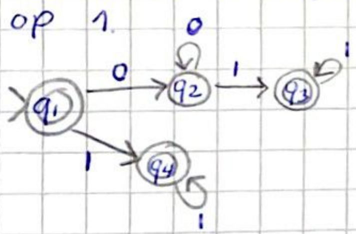


   - accepteer alleen strings zonder 010

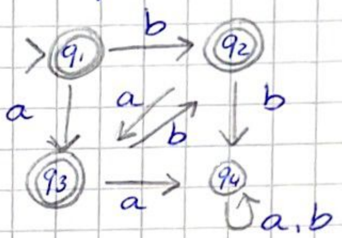


2. - automata accepteert alleen a, ab en een vermenigvuldiging van ab
   a, $(ab)^*$
   - automata aceepteert alleen b en alles wat begint met a en eindigt op b.
   b, $(a)^* + b$

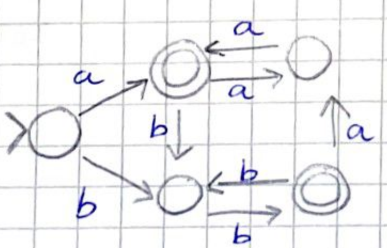
3. Machine accepteert lege strings, alleen 1 en alles eindigend op 1.



4. Accepteer alleen strings die geen aa of bb bevatten.



5. Accepteer alleen strings met oneven aantal a's en even aantal b's.



Turing machines

1.

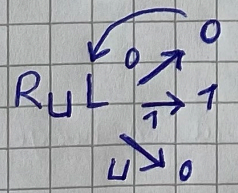
a	q	b	b	⊔	b	b	⊔	⊔	⊔	a	b	a	q_0
a	q	b	b	⊔	b	b	⊔	⊔	⊔	a	b	a	q_0
a	q	b	b	⊔	b	b	⊔	⊔	⊔	a	b	a	q_0
a	q	b	b	⊔	b	b	⊔	⊔	⊔	a	b	a	q_0
a	q	b	b	⊔	b	b	⊔	⊔	⊔	a	b	a	q_0
a	q	b	b	⊔	b	b	⊔	⊔	⊔	a	b	a	q_0
a	q	b	b	⊔	b	b	⊔	⊔	⊔	a	b	a	q_0
a	q	b	b	⊔	b	b	⊔	⊔	⊔	a	b	a	q_0
a	q	b	b	⊔	b	b	⊔	⊔	⊔	a	b	a	q_0
a	q	b	b	⊔	b	b	⊔	⊔	⊔	a	b	a	q_1
a	q	b	b	⊔	b	b	⊔	⊔	⊔	a	b	a	q_1
a	q	b	b	⊔	b	b	⊔	⊔	⊔	a	b	a	q_1
a	q	b	b	⊔	b	b	⊔	⊔	⊔	a	b	a	q_1
a	q	b	b	⊔	b	b	⊔	⊔	⊔	a	b	a	q_2
~~a~~	~~q~~	~~b~~	~~b~~	~~⊔~~	~~b~~	~~b~~	~~⊔~~	~~⊔~~	~~⊔~~	~~a~~	~~b~~	~~a~~	h

De machine blijft de kop op en neer bewegen tussen a en b totdat hij een b tegen komt in q_2.

3. $R_{\underline{\sqcup}} \xrightarrow{a} L_a$

Bij alles wat geen ⊔ is, schrijf a.
a is in deze zin een placeholder
? kwam niet veel verder dan dit

4. De machine verhoogt een binair getal met 1.



Weet niet precies wat ik hier gedaan heb maar het is de bedoeling dat dit een Turing machine is voor het vermenigvuldigen van binaire getallen

$0 \cdot 0 = 0$
$0 \cdot 1 = 0$
$1 \cdot 0 = 0$
$1 \cdot 1 = 1$

2. Ontwerp turing machine die naar rechts scant net zo lang tot je 2 aa's achter elkaar tegenkomt:

q_0	a	$(q_1, \rightarrow)$	$K = \{q_0, q_1, h\}$
q_0	$\sqcup$	$(q_0, \rightarrow)$	$\Sigma = \{a, \sqcup\}$
q_1	a	(h)	$s = q_0$
q_1	$\sqcup$	$(q_0, \rightarrow)$	$H = h$