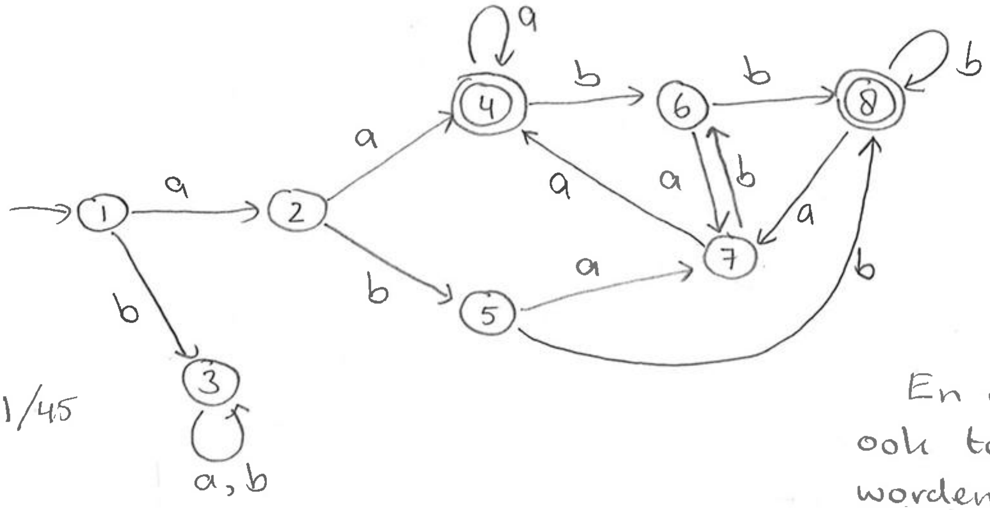
Uitwerking tentamen Automata Theory, maandag 19 december 2022 (1)

19.37

1)



19.41/45

Toestanden 5 en 6 kunnen nog worden samengevoegd.
En vervolgens kunnen ook toestanden 2 en 7 worden samengevoegd.

19.48

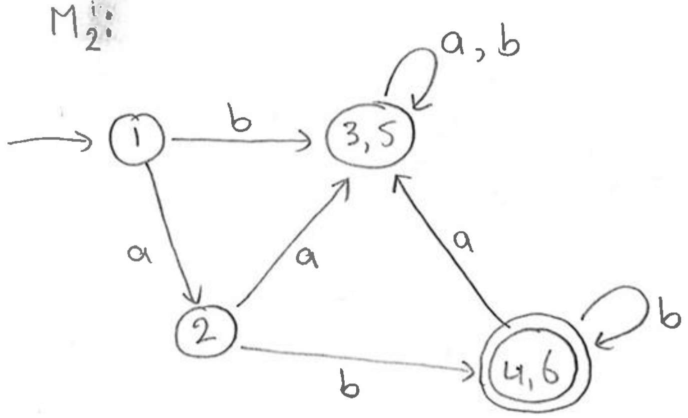
2 (a)

$i =$					
2	2b				
3	3a	2b			
4	1	1	1		
5	3a	2b	·	1	
6	1	1	1	·	1
$j =$	1	2	3	4	5

19.54

(b) We voegen toestanden 3 en 5 en toestanden 4 en 6 samen.

M_2:



19.57

3 (a) We hoeven alleen de accepterende toestanden aan te passen:

$Q_2 = Q_1$

$q_2 = q_1$

$A_2 = Q_1 - A_1$ (alle toestanden die niet accepterend zijn in M_1, zijn accepterend in M_2, en vice versa)

$\delta_2 = \delta_1$

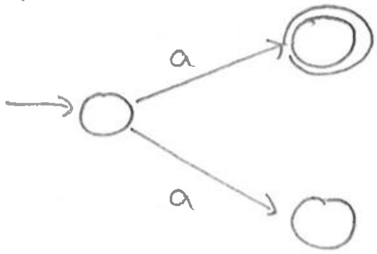
19.59

20.24

(b) Drie redenen:

* De stapelautomaat M_1 kan niet-deterministisch zijn, zodat een string x zowel een pad door M_1 heeft dat in een accepterende toestand leidt, als een pad dat in een niet-accepterende toestand leidt:

  Bijvoorbeeld M_1:

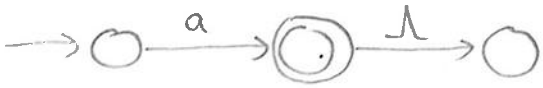


String $x = a$ wordt door M_1 geaccepteerd, en zou ook door M_2 geaccepteerd worden.

20.27

* De stapelautomaat M_1 kan Λ-transities bevatten die leiden van een accepterende toestand naar een niet-accepterende toestand of vice versa.

  Bijvoorbeeld M_1:

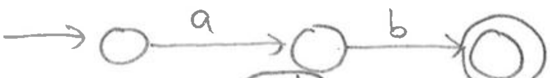


String $x = a$ wordt door M_1 geaccepteerd, en zou ook door M_2 geaccepteerd worden.

20.30

* De stapelautomaat M_1 kan transities 'missen', waardoor hij altijd crasht voor een bepaalde invoer, wat de accepterende toestanden ook zijn.

  Bijvoorbeeld M_1:



String $x = b$ wordt door M_1 niet geaccepteerd, en zou ook door M_2 niet geaccepteerd worden.

20.33

(c) Bijvoorbeeld $L_1 = XX' = \{ss \mid s \in \{a,b\}^*\}'$

20.34

Of $L_1 = AnBnCn' = \{a^i b^i c^i \mid i \geq 0\}'$

Of $L_1 = \{x \in \{a,b,c\}^* \mid n_a(x) = n_b(x) = n_c(x)\}'$

20.35

4)
(a)
Een reguliere expressie voor L_1: $aa^*bb^*cc^*$

20.37

(b)
Een reguliere expressie voor L_1':
$(a+b+c)^*(ba+ca+cb)(a+b+c)^* + (a+b)^* + (a+c)^* + (b+c)^*$

Als $x \in \{a,b,c\}^*$ niet in L_1 zit, zijn er twee mogelijkheden

* x bevat wel a's, b's en c's, maar niet in de juiste volgorde. In dat geval moet x een substring ba of ca of cb bevatten, met de letters in de verkeerde volgorde.
* x bevat niet alle drie de soorten letters, dus x bevat alleen a's en b's, alleen a's en c's of alleen b's en c's. Dit wordt beschreven door het tweede deel van de reguliere expressie (waarbij het overigens ook mogelijk is dat x maar één of nul soorten letters bevat)

Dit wordt beschreven door het eerste deel van de reguliere expressie (waarbij het overigens ook mogelijk is dat x niet alle drie de soorten letters bevat)

20.46/21.00

21.01

5 (ai) Nee, $L \not\subseteq L(G_1)$, want b.v. $x = abbbbbc$ zit wel in L, maar niet in $L(G_1)$. De b's die 'bij de a's horen', komen namelijk in G_1 helemaal achteraan te staan.

(aii) Nee, $L(G_1) \not\subseteq L$, want b.v. $x = abbbbcb$ zit wel in $L(G_1)$, maar niet in L: $S \Rightarrow aSb \Rightarrow abbbCb \Rightarrow abbbbCcb \Rightarrow abbbbcb$

(bi) Nee, $L \not\subseteq L(G_2)$, want b.v. $x = bbb$ zit wel in L, maar niet in $L(G_2)$. In G_2 wordt er minstens een a of c gegenereerd, via A of C.

(bii) Ja, $L(G_2) \subseteq L$

(ci) Ja, $L \subseteq L(G_3)$

(cii) Ja, $L(G_3) \subseteq L$

21.13

6 (a)

$N_0 = \emptyset$

$N_1 = N_0 \cup \{B\} = \{B\}$

$N_2 = N_1 \cup \emptyset = N_1$ $\Longrightarrow$ alleen B is nullable

21.15

(b) B is nullable, dus we voegen producties toe aan G_1, waarin we voorkomens van B weglaten. Dat geeft de volgende grammatica G_2:

$S \rightarrow Sa \mid bb \mid AB \mid A$

$A \rightarrow aAb \mid BBa \mid Ba \mid a$

$B \rightarrow SB \mid a \mid \cancel{\Lambda} \mid S$

↳ we schrappen de Λ-producties want is niet meer nodig

21.18

(c) S-derivable variabelen : $\{A\}$

A-derivable variabelen : $\emptyset$

B-derivable variabelen : $\{S, A\}$

21.20

(d) We voegen de producties van de X-derivable variabelen toe aan X zelf, en schrappen de unitproducties. Dat geeft de volgende grammatica G_3:

$S \rightarrow Sa \mid bb \mid AB \mid \cancel{A} \mid aAb \mid BBa \mid Ba \mid a$

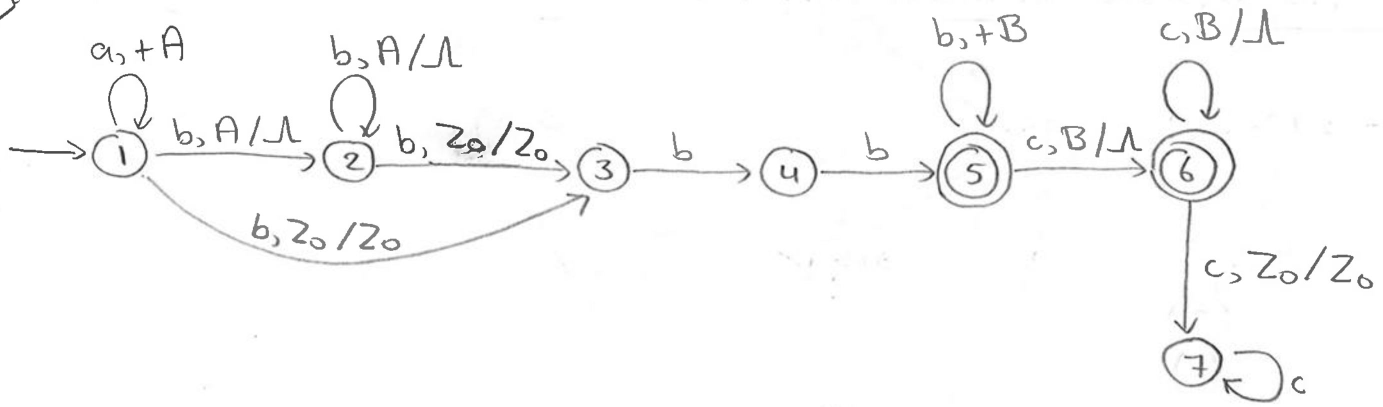
$A \rightarrow aAb \mid BBa \mid Ba \mid a$

$B \rightarrow SB \mid a \mid \cancel{S} \mid Sa \mid bb \mid AB \mid \cancel{A} \mid aAb \mid BBa \mid Ba \mid \cancel{a}$

dubbel.

21.23.

7)



M_1 leest in toestand 1 a^i, en zet voor elke a die hij leest een A op de stapel.
In toestand 2 worden net zoveel b's gelezen als er a's gelezen zijn. Voor elke gelezen b wordt een A van de stapel gehaald.
Als de stapel leeg is, lezen we naar toestanden 3, 4 en 5 nog drie b's, het minimum aantal extra b's dat we nodig hebben voor een string in L.
Dan kunnen we accepteren in toestand 5, maar we kunnen ook nog extra b's lezen. Deze extra b's tellen we met B's op de stapel.
Vervolgens mogen we nog net zoveel c's lezen als er B's op de stapel staan. Dat doen we in toestand 6. Voor elke c die we lezen halen we een B van de stapel. We mogen de string nog steeds accepteren.
Als we nog een c lezen, terwijl alle B's van de stapel zijn, hebben we teveel c's. We gaan dan naar de niet-accepterende toestand 7, waar we nog meer c's kunnen lezen, maar ook die niet meer zullen accepteren.
Als de letters in de verkeerde volgorde staan, crasht de automaat.

21.38

8) u_1 is niet geschikt, want zit niet in L (we hebben een b te weinig)
u_2 is wel geschikt.
u_3 is niet geschikt, want b.v. de opsplitsing

$$v = a^n \quad w = b \quad x = \Lambda \quad y = \Lambda \quad z = b^{3n-1}c^n$$

voldoet keurig aan het pomplemma. Je kunt $wy = b$ zonder probleem weg- of bijpompen.
u_4 is niet geschikt, want b.v. de opsplitsing

$$v = \Lambda \quad w = abbc \quad x = \Lambda, \quad y = \Lambda$$
$$z = (abbc)^{n-1}abc$$

voldoet keurig aan het pomplemma. Je kunt $wy = abbc$ zonder probleem weg- of bijpompen.

21.50