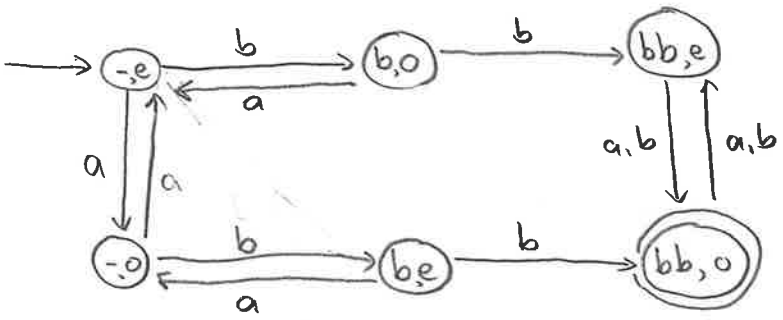
15.45

1]



Namen van toestanden geven aan:
- hoe ver we zijn gekomen bij het vinden van een substring bb:
  λ, b, of bb
- of de lengte van de string tot nu toe even of oneven is: e of o

15.53

2] Stel dat L wél door een eindige automaat M geaccepteerd kan worden, en dat het aantal toestanden van M n is.

Laat dan $x = a^n b a^{n+2}$. Er geldt $|x| \geq n$ en $x \in L$, want met $i=n$, $j=1$ en $k=n+2$ geldt $x = a^i b^j a^k$ met $i+j = n+1 < n+2 = k$.

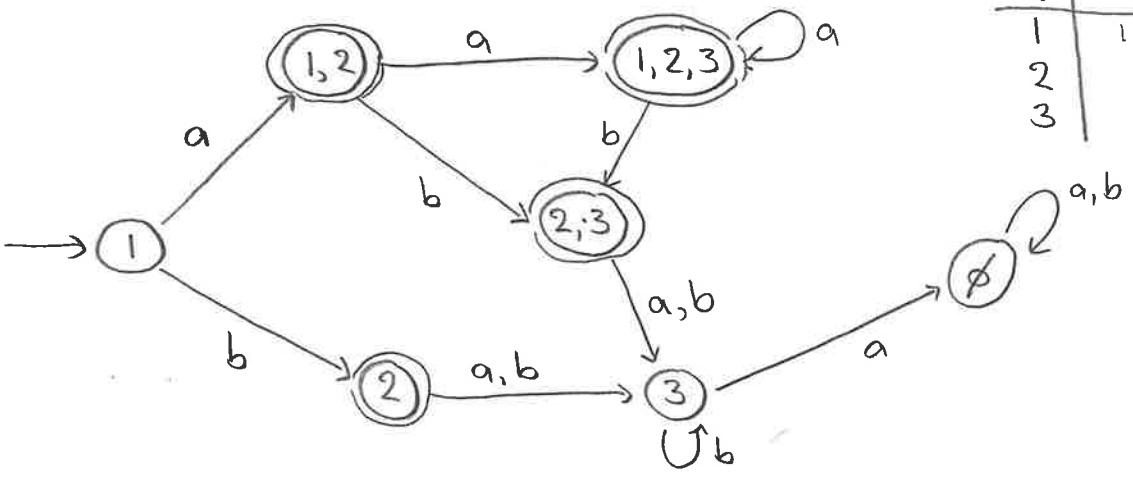
Volgens het pomplemma voor reguliere talen is x te schrijven als $x = uvw$ met
1) $|uv| \leq n$
2) $|v| \geq 1$
3) $\forall m \geq 0$, is $uv^m w \in L$

Echter, omdat $|uv| \leq n$, bestaan u en v uit alleen a's, hoogstens n
Omdat $|v| \geq 1$, geldt $v = a^k$ met $1 \leq k \leq n$
Maar dan is (neem $m=2$) $uv^2w = a^{n+k} b a^{n+2} = a^{i'} b^{j'} a^{k'}$
met $i' = n+k \geq n+1$, $j' = 1$, $k' = n+2$, zodat $i'+j' \geq n+2 = k'$,
terwijl $i'+j' < k'$ moet zijn om in L te zitten.
Dit is in tegenspraak met het pomplemma.
De veronderstelling dat L wel door een eindige automaat geaccepteerd kan worden, kan dus niet kloppen: L is niet regulier.

16.06

3]

q	δ(q,a)	δ(q,b)
1	1, 2	2
2	3	3
3	–	3



16.33

4 (a)

(i) Nee, $L_1 \not\subseteq L$, want $x = bb \in L_1$, terwijl $bb \notin L$.

16.36

(ii) Nee, $L \not\subseteq L_1$, want $x = ababa \in L$, terwijl $ababa \notin L_1$

16.38

(b)

(i) $\exists a,\ L_2 \subseteq L$

16.39

(ii) Nee, $L \not\subseteq L_2$, want $x = ab \in L$, terwijl $ab \notin L_2$

16.42

5 (a)

We noemen een context-vrije grammatica $G = (V, \Sigma, S, P)$ regulier, als elke productie in P een van de volgende vormen heeft:

$A \to \sigma B$ met $A, B \in V$, $\sigma \in \Sigma$

$A \to \Lambda$ met $A \in V$

16.44

(b)

G heeft startvariabele A en de volgende producties:

$A \to aB \mid bD \mid \Lambda$

$B \to aB \mid bC$

$C \to aB \mid bC \mid \Lambda$

$D \to aD \mid bD$

16.48

6 (a) G heeft startvariabele S en de volgende producties:

$S \to A_1 \mid A_2$ A_1 voor strings zonder b ($j=0$, en we kunnen alle a's onder a^k scharen)
A_2 voor strings met b

$A_1 \to aA_1 \mid a$ één of meer a's

$A_2 \to aA_2a \mid B$ i en k ophogen, of overstappen op b's

$B \to bBa \mid bAa$ j en k ophogen, of afronden met laatste b, en dan met A_1 nog één of meer extra a's.

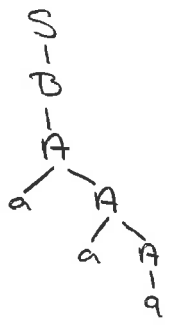
17.03

Een voudige dubbelzinnige grammatica:

$S \to aSa \mid B$

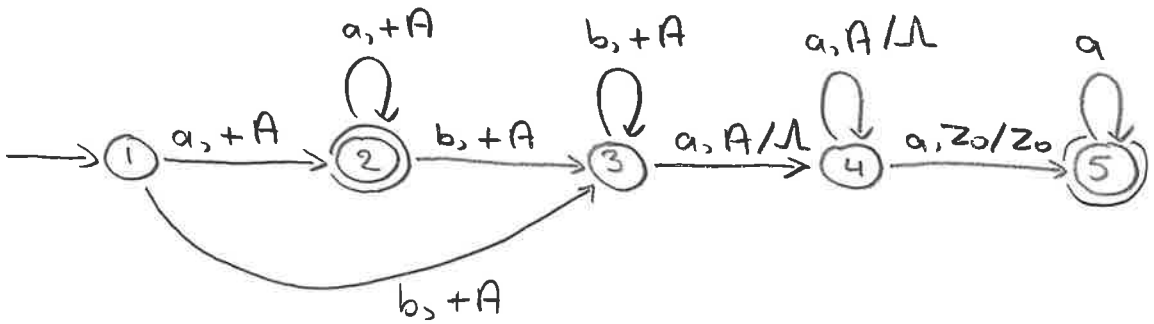
$B \to bBa \mid A$

$A \to aA \mid a$



17.06

6 (b)

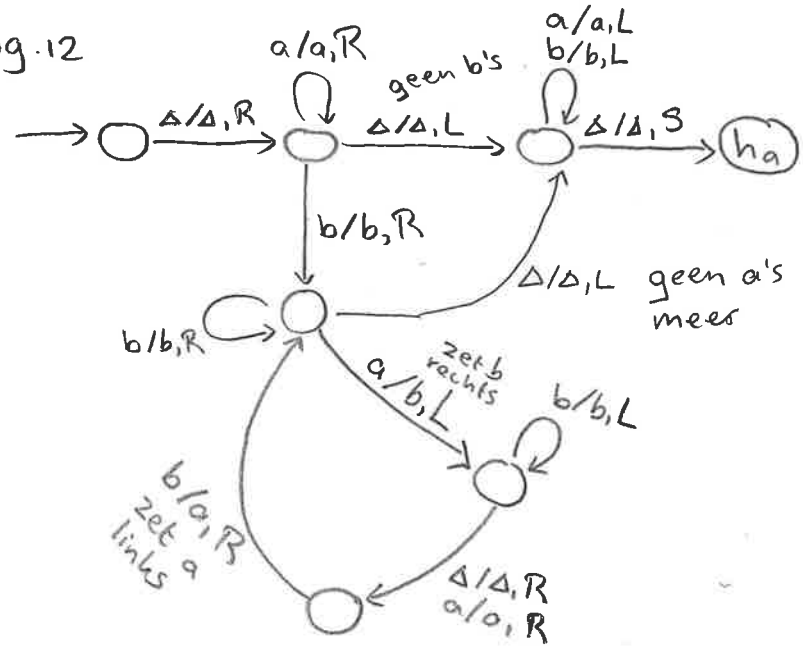


M leest in eerste instantries a's. Omdat a's van a^k kunnen zijn, als $i = j = 0$, gaan we hiervoor naar accepterende toestand 2.
17.10/22 We tellen ondertussen wel deze a's met A's op de stapel, voor het geval er b's komen, en de zojuist gelezen a's tot a^i behoorden, en we ze uiteindelijk moeten wegstrepen tegen a's van a^k.
Als er b's komen, tellen we ook die met (extra) A's op de stapel, in toestand 3.
Als er dan a's van a^k komen, halen we voor elke a een A van de stapel. Als de stapel 'leeg' is (we zien Z_0), geldt $i + j = k$. Met een extra a krijgen we $i + j < k$ en gaan we naar accepterende toestand 5, waar we nog extra a's mogen lezen (daarmee k verder ophogend).
17.28/18.20

7)

T loopt door invoer x (naar rechts), op zoek naar eerste b.
Vanaf daar loopt T verder naar rechts, naar eerstvolgende a.
Deze a borrelt als het ware naar voren: de b's die ervoor staan, schuiven 1 positie naar rechts.
Vervolgens gaat T op zoek naar de volgende a, die ook 'naar voren borrelt', enzovoort, tot er geen a meer te vinden is rechts. Dan gaat de leeshop terug naar links en accepteert T.
Als de string helemaal geen b bevat, gaat de leeshop meteen terug naar links en accepteert T.
18.28

og.12



og.18 / og.19 / og.20

8)

instanties: Accepts $\leq$ AcceptsEverything

(T_1, x) $\quad$ T_2

Laat T_1, x een willekeurige instantie van Accepts zijn.
We definiëren de instantie T_2 van AcceptsEverything als volgt:

T_2 krijgt een invoer x_2, vervangt die invoer door x, en simuleert vervolgens T_1 op die invoer x.

Er geldt:

(T_1, x) is ja-instantie van Accepts $\Leftrightarrow$ T_1 accepteert x $\Rightarrow$
T_2 accepteert zijn invoer x_2, wat die ook is, omdat hij T_1 simuleert op x $\Rightarrow$ $L(T_2) = \Sigma^*$ $\Leftrightarrow$ T_2 is ja-instantie van AcceptsEverything.

(T_1, x) is nee-instantie van Accepts $\Leftrightarrow$ T_1 accepteert x niet. $\Rightarrow$
T_2 accepteert zijn invoer x_2 niet, wat die ook is, omdat hij T_1 simuleert op x $\Rightarrow$ $L(T_2) = \emptyset$ $\Rightarrow$ $L(T_2) \neq \Sigma^*$ $\Leftrightarrow$
T_2 is nee-instantie van AcceptsEverything

Uiteraard is T_2 op algoritmische wijze uit (T_1, x) te construeren.

og.31