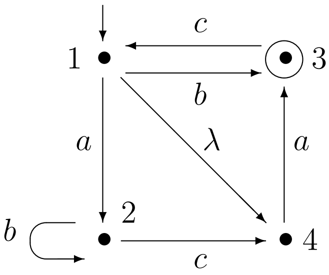
Uitwerking Talen en Automaten, 30 juni 2008

Tijdsduur 3 uur. Gesloten boek tentamen.

Opgave 1 (11 %). Beschouw de NFA-λ met het alphabet $\{a, b, c\}$, de toestanden 1, 2, 3, 4, en de starttoestand 1.



(a) Bepaal de λ-afsluitingen van de toestanden van deze automaat.
(b) Construeer uit deze NFA-λ op systematische wijze een DFA voor dezelfde taal. Geef de volledige overgangstabel van de DFA inclusief starttoestand en accepterende toestand(en). Hoeveel toestanden heeft deze DFA?

Uitwerking. (a: 2 %) Volgens 5.6. De λ-afsluiting van 1 is $\{1, 4\}$. De λ-afsluiingen van 2, 3, 4 zijn respectievelijk $\{2\}$, $\{3\}$ en $\{4\}$.

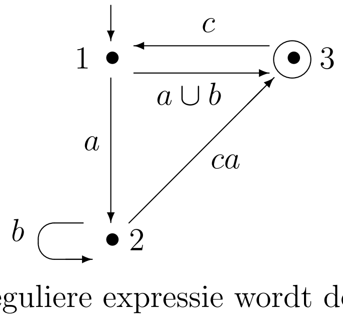
```
(b: 9 %)               a        b        c
      -> {1,4}  |   {2,3}  |  {3}  |    {}  |
        *{2,3}  |     {}   |  {2}  | {1,4}  |
        * {3}   |     {}   |   {}  | {1,4}  |
          {2}   |     {}   |  {2}  |  {4}   |
          {4}   |    {3}   |   {}  |    {}  |
           {}   |     {}   |   {}  |   {}
```

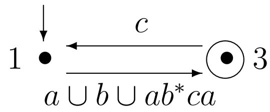
Er zijn dus zes toestanden.

Opgave 2 (10 %). Bepaal op systematische wijze een reguliere expressie voor de taal van de automaat van opgave 1. Een antwoord alleen is niet genoeg.

Uitwerking. Volgens 6.2 en whh408. We kunnen het toevoegen van extra starttoestand en eindtoestand ook uitstellen of nalaten.

We laten eerst knoop 4 weg. Dat geeft de linker graaf. Weglaten van knoop 2 geeft dan de rechter graaf.





De reguliere expressie wordt derhalve $(a \cup b \cup ab^*ca)(c(a \cup b \cup ab^*ca))^*$.

Opgave 3 (18 %).
(a) Formuleer het Pomplemma voor *reguliere* talen.
(b) De taal L_3 over het alfabet $\{a, b\}$ wordt gegeven door de contextvrije grammatica

$$S \rightarrow aS \mid aSa \mid b\ .$$

Beschrijf de taal L_3 als verzameling. Bewijs dat de taal L_3 niet regulier is.

Uitwerking. (a: 5 %) Zie boek.
(b: 13 %) $L_3 = \{a^iba^j \mid i \geq j \geq 0\}$.

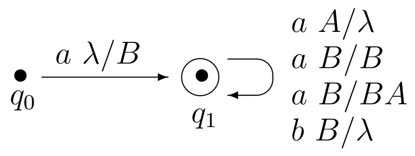
Stel dat L_3 regulier is. Dan is er volgens het pomplemma een constante k (het aantal toestanden van een DFA voor L) zo dat elke string $z \in L_3$ met lengte $|z| \geq k$ een ontbinding $z = uvw$ heeft met $|uv| \leq k$ en $v \neq \lambda$, en $uv^mw \in L_3$ voor elke $m \in \mathbb{N}$. Kies $z = a^kba^k$. Dan geldt inderdaad $z \in L_3$ en $|z| \geq k$. Dus z heeft een ontbinding $a^kb^k = z = uvw$ als beschreven. Wegens $|uv| \leq k$ bestaat uv geheel uit symbolen a. Dus $v = a^r$ voor zekere $r \geq 1$. We nemen $m = 0$. Dit geeft $uv^0w \in L_3$. Maar $uv^0w = a^{k-r}ba^k \notin L_3$. Dit is een tegenspraak. Dus L_3 is niet regulier.

Opgave 4 (11 %). De taal L_4 over $\Sigma = \{a, b\}$ wordt voortgebracht door de grammatica

$$\begin{aligned} S &\rightarrow aB \\ A &\rightarrow a \\ B &\rightarrow aB \mid aBA \mid b\ . \end{aligned}$$

Beschrijf een desgewenst uitgebreide stapelautomaat (*extended PDA*) M die taal L_4 accepteert. Geef duidelijk aan, wat de toestandsruimte Q, het stapelalfabet Γ en de overgangsfunctie δ zijn.

Uitwerking. $L_4 = \{a^iba^j \mid i > j \geq 0\}$. We kunnen de constructie in het bewijs van Stelling 7.3.1 gebruiken, maar het kan ook anders. De gegeven grammatica is al in Greibach normaalvorm. Genoemde constructie geeft:



De productie van S geeft de linker pijl. De zelflus van q_1 correspondeert met de productie van A en de drie producties van B.

Opgave 5 (10 %). Beschouw de taal L_5 over het alfabet $\{a, b, c\}$ die voortgebracht wordt door de contextvrije grammatica G met de nonterminals S (startsymbool), A, B, en de productieregels

$$\begin{aligned} S &\rightarrow aA \mid B \\ A &\rightarrow cA \mid Bc \\ B &\rightarrow \lambda \mid Sb\ . \end{aligned}$$

Leid uit grammatica G een nieuwe grammatica G' af met $L(G') = L_5$ waarin het startsymbool geen recursieve variabele is en die geen λ-producties heeft behalve eventueel vanuit het startsymbool. Motiveer je antwoord.

Uitwerking. Volgens 4.1 en 4.2. We voegen allereerst een niet-recursief startsymbool T toe met de productieregel $T \to S$.

We bepalen vervolgens de nullable nonterminals: dat zijn B, S en T. We voegen derhalve de productieregel $T \to \lambda$ toe en laten nullable nonterminals optioneel in de rechterkanten weg, en laten de λ-producties buiten het startsymbool weg. We krijgen dan:

$$\begin{array}{lcl} T & \to & S \mid \lambda \\ S & \to & aA \mid B \\ A & \to & cA \mid Bc \mid c \\ B & \to & Sb \mid b\ . \end{array}$$

Opmerking. De taal L_5 is niet regulier, want hij bevat $a^n(bc)^n$ voor elke $n \in \mathbb{N}$, en voor elke $w \in L_5$ is $n_a(w) \leq n_c(w)$. Dan pomplemma.

Opgave 6 (10 %). Ontwerp een meerbands Turing machine M met invoeralfabet $\Sigma = \{a, b\}$, die de taal $L_6 = \{a^m b^n \mid m^2 = n\}$ accepteert, en die voor elke invoer eindigt. Geef aan welke toestanden $q \in Q$ je gebruikt en wat het betekent als de TM in een gegeven toestand is. Zorg dat duidelijk blijkt dat de $L(M) = L_6$ en dat M op elke invoer eindigt.

Uitwerking. We nemen een driebands TM met eenzijdige banden x, y en z, en bandindices respectievelijk i, j, k. We kopiëren eerst de vooroplopende deelstring van symbolen a naar de banden 2 en 3. Daar staat dan a^m terwijl op band 1 de kop op een b of een B staat. Met een geneste for-lus over de banden 2 en 3, verifiëren we dan dat er op band 1 rechts van a^m nog $n = m^2$ symbolen b staan.

```
q0:     i++ ; j++ ; k++ ;
q1:     while x[i] = a do y[j]:= a ; z[j]:= a ; i++ ; j++ ; k++ end ;
        j-- ;
q2:     while y[j] = a do
          k-- ;
q3:       while z[k] = a and x[i] = b do k-- ; i++ end ;
          assert z[k] = B ; k++ ;
q4:       while z[k] = a do k++ end ;
          j-- ;
        end ;
q2:     if y[j] = B and x[i] = B then goto q5 end ;
q5:     accepteer .
```

In $q2$ geldt de invariant van de buitenlus dat $x = Ba^m b^{(m-j)\cdot m}(a \cup b)^*$ terwijl i op het eerstvolgende symbool van de invoer staat. Hieruit volgt dat acceptatie bij $y[j] = x[i] = B$ correct is. Als de invoer minder dan m^2 b's na a^m heeft, stopt de berekening in $q3$. Als de invoer na m^2 b's nog meer bevat, stopt het in $q2$.

Opgave 7 (6 %). We beschouwen een taal L over een alfabet Σ.
(a) Wanneer is taal L *recursief opsombaar*? Geef de definitie.
(b) Wanneer is taal L *recursief*? Geef de definitie.

Uitwerking. L is recursief opsombaar als er een Turing machine M is met $L = L(M)$. L is recursief als er een Turing machine M is met $L = L(M)$ zodat M bovendien bij elke invoer eindigt.

Opgave 8 (24 %). We beschouwen de klasse van Turing machines met één tweezijdig oneindige band, zonder accepterende toestanden, met het invoeralfabet $\{0, 1\}$ en het bandalfabet $\{0, 1, B\}$.
(a: 6 %) Laat zien hoe een willekeurige TM uit deze klasse gecodeerd kan worden als een string in $\{0, 1\}^*$.

Laat L_{tm} de verzameling van deze coderingen zijn. Voor elke codering $w \in L_{tm}$ definiëren we $T(w)$ als de door w gecodeerde Turing machine, en $L(T(w))$ als de taal die door Turing machine $T(w)$ geaccepteerd wordt.
(b: 2 %) Laat zien dat L_{tm} een reguliere taal is.
(c: 5 %) Is $L(T(w))$ recursief voor elke $w \in L_{tm}$? Is $L(T(w))$ recursief opsombaar voor elke $w \in L_{tm}$? Geef argumentatie.
(d: 4 %) Specificeer "de" universele Turing machine U in termen van L_{tm} en $T(w)$.
(e: 7 %) Toon aan dat $L_8 = \{w \in L_{tm} \mid w \in L(T(w))\}$ recursief opsombaar is.

Uitwerking. (a) Lees 11.5 uit het boek en parafraseer dat.
(b) De taal is regulier want hij wordt beschreven door een reguliere expressie:

$$\begin{aligned}
&\textit{TuringMachine} ::= 0\ 0\ 0\ (\textit{transitie}\ 0)^*\ 0 \\
&\textit{transitie} ::= \textit{state sym state sym dir} \\
&\textit{state} ::= 1^+0 \\
&\textit{sym} ::= 10 \mid 110 \mid 1110 \\
&\textit{dir} ::= 10 \mid 110\ .
\end{aligned}$$

(c) Nee en ja. Volgens de stelling van Turing zijn er recursief opsombare talen over $\{0, 1\}$ die niet recursief zijn. Als L een dergelijke taal is, is er een $w \in L_{tm}$ met $L = L(T(w))$. Elke taal $L(T(w))$ is recursief opsombaar omdat hij geaccepteerd wordt door de TM $T(w)$.
(d) Turing machine U accepteert een string v precies als v de concatenatie is van een $w \in L_{tm}$ en een u die door $T(w)$ geaccepteerd wordt. Dit betekent dat $L(U) = \{wu \mid w \in T_{tm} \ \wedge \ u \in L(T(w))\}$.
(e) Het is voldoende een TM M te ontwerpen waarvoor $L(M) = L_8$. Voor $w \in L_{tm}$ geldt $w \in L_8$ precies dan als $w \in L(T(w))$ oftewel als $ww \in L(U)$. We laten M dus eerst verifiëren dat $w \in L_8$, verdubbelen w tot ww en voeren dat in bij de universele TM U. Er geldt dan $w \in L_8$ precies dan als M op invoer w eindigt. Dus $L(M) = L_8$. Dus L_8 is recursief opsombaar.

NB. Er wordt niet gevraagd te bewijzen dat L_8 niet recursief is.