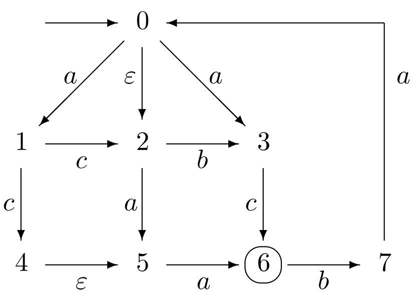
Uitwerking Talen en Automaten, 6 november 2006

Tijdsduur 3 uur. Gesloten boek tentamen.

Opgave 1 (16 %). Beschouw de ε-NFA A_1 over het alfabet $\Sigma = \{a, b, c, d\}$ gegeven in de figuur:

A_1 :



De toestanden zijn genummerd 0 tot en met 7. Toestand 0 is de starttoestand. Toestand 6 is de enige accepterende toestand.
(a) Zet de automaat A_1 volgens het standaardalgoritme om in een deterministische eindige automaat B_1 die dezelfde taal accepteert. Bepaal alle bereikbare toestanden van B_1, en alle accepterende toestanden van B_1.
(b) Bepaal een reguliere expressie voor de taal van de automaten A_1 en B_1.

Uitwerking. (a) Allereeerst, deterministische eindige automaten hebben geen ε-overgangen, dus B_1 mag dat ook niet hebben.

De ε-afsluiting van $\{0\}$ is $\{0, 2\}$. Dat is dus de nieuwe starttoestand. De ε-afsluiting van $\{4\}$ is $\{4, 5\}$. Alle andere knopen zijn hun eigen ε-afsluiting. Vervolgens voeren we het subset-algoritme uit, waarbij de resulterende verzameling steeds ε-afgesloten wordt.

	a	b	c	d
-> {0, 2}	{1, 3, 5}	{3}	{}	{}
{1, 3, 5}	{6}	{}	{2, 4, 5, 6}	{}
{3}	{}	{}	{6}	{}
* {6}	{}	{7}	{}	{}
* {2, 4, 5, 6}	{5, 6}	{3, 7}	{}	{}
{7}	{0, 2}	{}	{}	{}
* {5, 6}	{6}	{7}	{}	{}
{3, 7}	{0, 2}	{}	{6}	{}

Inclusief de lege toestand `{}` zijn dit dus 9 toestanden; de drie accepterende zijn aangeduid met `*`.

(b) De paden van 0 naar 6 die 7 niet aandoen, geven aanleiding tot de reguliere expressie $X = ac(a+\varepsilon)+(ac+\varepsilon)(aa+bc)$ (de tweede summand komt van de paden die toestand 2 aandoen, de eerste van de paden die 2 vermijden). Het pad van 6 naar 0 geeft de reguliere expressie ba. Omdat 6 de enige accepterende toestand is geeft dit dus $X(baX)^*$.

Opgave 2 (18 %). Beschouw het alfabet $\Sigma = \{0, 1, 2\}$ en de taal L_2 over Σ die bestaat uit de strings waarvan het laatste symbool ook strikt vóór het midden voorkomt: $L_2 = \{xaya \mid x, y \in \Sigma^*, a \in \Sigma : |x| < |y|\}$.
(a) Formuleer het Pomplemma voor *reguliere* talen.
(b) Bewijs dat de taal L_2 niet regulier is.
(c) Bewijs dat de taal L_2 contextvrij is en geef er een contextvrije grammatica voor.

Uitwerking. (a) Zie boek (of zie (b)).

(b) Stel dat L_2 regulier is. Dan is er volgens het Pomplemma een getal n zo, dat voor elke string $w \in L_2$ met $|w| \geq n$ er strings x, y, z zijn met $|xy| \leq n$ en $|y| \geq 1$ en met $xy^kz \in L_2$ voor elke $k \in \mathbb{N}$.

Gebruik makend van dit getal n, kies ik nu (bv.) de string $w = 0^n 1 2^{n+1} 1$. Hiervoor geldt $w \in L_2$ en $|w| = 2n + 3 \geq n$. Er zijn dus strings x, y, z zijn met $|xy| \leq n$ en $|y| \geq 1$ en met $xy^kz \in L_2$ voor elke $k \in \mathbb{N}$. Omdat $w = xyz$ met n nullen begint en $|xy| \leq n$, bestaan x en y uitsluitend uit nullen. Zeg dat y bestaat uit i nullen, dus $y = 0^i$ en $i \geq 1$. Er geldt dan $xy^2z = 0^{n+i}12^{n+1}1$, maar ook $xy^2z \in L_2$. Dus $n + i < n + 1$, wat in strijd is met $i \geq 1$. Deze tegenspraak impliceert dat L_2 niet regulier is.

(c) De taal L_2 is contextvrij omdat we er een contextvrije grammatica voor kunnen maken. Dit gebeurt alsvolgt. We laten de nonterminals A, B, C staan voor oneven strings met in het midden respectievelijk een 0, 1, of 2. We gebruiken D voor willekeurige niet-lege strings en E voor strings ter lengte 1.

$$
\begin{array}{lcl}
S & \rightarrow & AD0 \mid BD1 \mid CD2 \\
A & \rightarrow & 0 \mid EAE \\
B & \rightarrow & 1 \mid EBE \\
C & \rightarrow & 2 \mid ECE \\
D & \rightarrow & E \mid ED \\
E & \rightarrow & 0 \mid 1 \mid 2 \ .
\end{array}
$$

Opgave 3 (16 %). Beschouw de taal L_3 die wordt voortgebracht door de grammatica $G = (V, T, P, S)$ met de alfabetten $V = \{A, B, C, D, S\}$ en $T = \{a, b, c, d\}$ en de productieregels

$$
\begin{array}{lcl}
S & \rightarrow & bA \mid BB \\
A & \rightarrow & aA \mid AbD \\
B & \rightarrow & \varepsilon \mid AS \mid cCc \\
C & \rightarrow & B \mid aC \mid Sd \\
D & \rightarrow & \varepsilon \mid dD \ .
\end{array}
$$

Leid uit G een grammatica G' af, die geen overbodige (*useless*) symbolen bevat en ook geen ε-producties bevat en waarvoor geldt dat $L(G') = L_3 - \{\varepsilon\}$.

Uitwerking. Het enig niet-voortbrengende symbool is A. Als we dat weglaten krijgen we de grammatica

$$
\begin{array}{lcl}
S & \rightarrow & BB \\
B & \rightarrow & \varepsilon \mid cCc \\
C & \rightarrow & B \mid aC \mid Sd \\
D & \rightarrow & \varepsilon \mid dD \ .
\end{array}
$$

De bereikbare symbolen zijn nu in $V' = \{S, B, C\}$ en $T' = \{a, c, d\}$. We kunnen nu dus D en zijn producties weglaten en krijgen:

$$
\begin{array}{lcl}
S & \rightarrow & BB \\
B & \rightarrow & \varepsilon \mid cCc \\
C & \rightarrow & B \mid aC \mid Sd \ .
\end{array}
$$

De nonterminals B, C en S zijn alle drie nullable. Eliminatie van de ε-producties levert dus op:

$$
\begin{array}{lcl}
S & \rightarrow & B \mid BB \\
B & \rightarrow & cc \mid cCc \\
C & \rightarrow & B \mid a \mid aC \mid d \mid Sd \ .
\end{array}
$$

Opgave 4 (16 %). Gegeven is de taal L_4 over het alfabet $\Sigma = \{0, 1, 2\}$ die bestaat uit de strings $w \in \Sigma^*$ die meer nullen dan enen bevatten. Schrijven we dus $cnt(a, w)$ voor het aantal symbolen a in string w, dan is

$$L_4 = \{w \in \Sigma^* \mid cnt(0, w) > cnt(1, w)\} \ .$$

(a) Ontwerp een stapelautomaat P_4 die de taal L_4 accepteert bij lege stapel, dwz. met $L_4 = N(P_4)$. Licht toe waarom je oplossing correct is.
(b) Bewijs (bv. hiermee) dat L_4 een contextvrije taal is.

Uitwerking. (a) We zetten de invoersymbolen op de stapel, maar laten nullen en enen elkaar opheffen. We zorgen aldus dat de stapel alleen nullen of alleen enen bevat, uitgezonderd het onderste symbool Z_0. We houden invariant bij invoer w, dat de concatenatie van de stapel en de rest van de invoer w' voldoet aan $cnt(0, w') - cnt(1, w') = cnt(0, w) - cnt(1, w)$. Naast de starttoestand q_0 gebruiken we een tweede toestand waarin de invoer ongewijzigd blijft en de stapel leeg gemaakt wordt. Dus $\Gamma = \{0, 1, Z_0\}$ en $Q = \{q_0, q_1\}$. In toestand q_0 wordt een 2 op de invoer gewist zonder effect op de stapel of de toestand.

$$\begin{array}{l}
\delta(q_0, 0, Z_0) = \{(q_0, 0Z_0)\} \ , \\
\delta(q_0, 1, Z_0) = \{(q_0, 1Z_0)\} \ , \\
\delta(q_0, 0, 0) = \{(q_0, 00)\} \ , \\
\delta(q_0, 0, 1) = \{(q_0, \varepsilon)\} \ , \\
\delta(q_0, 1, 0) = \{(q_0, \varepsilon)\} \ , \\
\delta(q_0, 1, 1) = \{(q_0, 11)\} \ , \\
\delta(q_0, 2, X) = \{(q_0, X)\} \ \text{ voor elke } X \in \Gamma, \\
\delta(q_0, \varepsilon, 0) = \{(q_1, \varepsilon)\} \ , \\
\delta(q_1, \varepsilon, X) = \{(q_1, \varepsilon)\} \ \text{ voor elke } X \in \Gamma.
\end{array}$$

Als de invoer op is, hoeven we vanwege de invariant alleen maar na te gaan of de stapel tenminste één 0 bevat. We gaan dan naar q_1 en maken de stapel leeg.

(b) De taal L_4 is contextvrij omdat er een stapelautomaat P is met $L_4 = N(P)$, zie stelling boek.

Opgave 5 (16 %). Ontwerp een Turing machine M met invoeralfabet $\Sigma = \{0, 1\}$, die zijn invoerstring w opvat als een binaire codering van een natuurlijk getal n. Als de invoer een rijen nullen is, zodat $n = 0$, moet M de invoer verwerpen. Anders dient executie van M te eindigen in een accepterende toestand, terwijl de band een binaire codering van $n - 1$ bevat en de leeskop op het minst signifikante bit van deze codering van $n - 1$ staat. Geef het volledige zeventupel van M, en geef de overgangsfunctie δ van M in tabelvorm.

Uitwerking. We sturen de kop eerst naar de meest rechtse een. Als die er is, wordt hij omgezet in 0, worden alle nullen rechts daarvan in enen omgezet en wordt de string geaccepteerd. Anders wordt de string verworpen.

```
q0:    while x[j] <> B do j++ end ; j-- ;
q1:    while x[j] = 0 do j-- end ;
       if x[j] = B then verwerp
       else {hier geldt x[j] = 1} x[j] := 0 ; j++ end ;
q2:    while x[j] = 0 do x[j] := 1 ; j++ end ; j-- ;
q3:    accepteer.
```

We gebruiken dus $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ met $Q = \{q_0, q_1, q_2, q_3\}$ en $\Gamma = \Sigma \cup \{B\}$ en $F = \{q_3\}$, met de overgangsfunctie gegeven door de tabel:

```
delta       0            1            B
-> q0  (q0, 0, R)  (q0, 1, R)  (q1, B, L)
   q1  (q1, 0, L)  (q2, 0, R)       -
   q2  (q2, 1, R)       ?      (q3, B, L)
 * q3       -           -           -
```

Op de plek van het vraagteken mag alles staan, omdat die onbereikbaar is.

Opgave 6 (18 %). Alle (deterministische eenbands) Turing machines met invoeralfabet $\Sigma = \{0, 1\}$ kunnen gecodeerd worden in de taal *Ctm* die bepaald wordt door de reguliere expressie $((\mathbf{0}^+\mathbf{1})^5\mathbf{1})^+\mathbf{1}$. We schrijven $D(m)$ voor de Turing machine die door $m \in \mathit{Ctm}$ gecodeerd wordt. Het boek gebruikt dit voor de constructie van de universele taal

$$L_u = \{mv \mid m \in \mathit{Ctm}, v \in \Sigma^* : v \in L(D(m))\} \ .$$

(a) Is L_u recursief opsombaar? Wat betekent dit?
(b) Is L_u beslisbaar (d.i. recursief)? Wat betekent dit?
Er worden in (a) en (b) geen bewijzen gevraagd.

We beschouwen nu hiernaast de taal

$$L_{term} = \{mv \mid m \in \mathit{Ctm}, v \in \Sigma^* : D(m) \text{ eindigt bij invoer } v\} \ .$$

(c) Is L_{term} recursief opsombaar?
(d) Is L_{term} beslisbaar (d.i. recursief)?
Bewijs je beweringen in (c) en (d) met behulp van je antwoorden van (a) en (b).

Uitwerking. (a) L_u is recursief opsombaar (zie het boek), dwz. er bestaat een Turing machine, de zg. universele TM M_u, met de eigenschap $L_u = L(M_u)$.

(b) L_u is niet beslisbaar (zie het boek), dwz. bij elke keuze van M_u zijn er invoerstrings w zodanig dat M_u op invoer w niet eindigt.

(c) L_{term} is recursief opsombaar. Om dit aan te tonen hoeven we vanwege onderdeel (a) alleen een reductie van L_{term} naar L_u te geven. Bij instantie mv met $m \in \mathit{Ctm}$ voor L_{term} wijzigen we m in m' zo dat $D(m')$ precies dan accepteert als $D(m)$ eindigt. We voegen aan mv nu de instantie $m'v$ toe. Dit is een reductie want $mv \in L_{term} \equiv m'v \in L_u$.

(d) L_{term} is niet beslisbaar. Om dit aan te tonen hoeven we vanwege onderdeel (b) alleen een reductie van L_u naar L_{term} te geven. Bij instantie mv met $m \in \mathit{Ctm}$ voor L_u wijzigen we m in m' zo dat $D(m')$ precies dan eindigt als $D(m)$ accepteert. We voegen aan mv nu de instantie $m'v$ toe. Dit is een reductie want $mv \in L_u \equiv m'v \in L_{term}$.