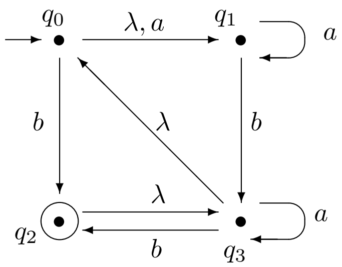
Uitwerking Talen en Automaten, 22 januari 2010

Tijdsduur 3 uur. Gesloten boek tentamen.

Opgave 1 (10 %). Beschouw een taal L over het alphabet Σ. Vul voor de puntjes (. . .) één van de volgende machinetypes in:
A: een lineair begrensde automaat (LBA),
B: een nondeterministische eindige automaat,
C: een Turingmachine die voor elke invoer eindigt,
D: een stapelautomaat,
E: een Turingmachine.
(a) L is contextvrij $\equiv$ L wordt geaccepteerd door . . .
(b) L is recursief opsombaar $\equiv$ L wordt geaccepteerd door . . .
(c) L is regulier $\equiv$ L wordt geaccepteerd door . . .
(d) L is recursief $\equiv$ L wordt geaccepteerd door . . .

Uitwerking
(a) Contextvrij — stapelautomaat.
(b) Recursief opsombaar — TM.
(c) Regulier — NFA.
(d) Recursief — eindigende TM.

Opgave 2 (12 %). Beschouw de NFAλ M gegeven door

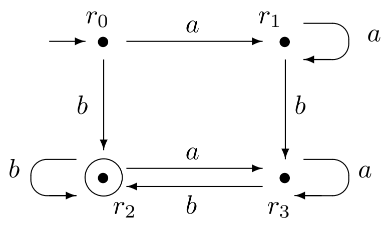


Construeer volgens het standaardalgoritme het toestandsdiagram van de DFA equivalent met M.

Uitwerking. De λ-afsluiting van q_0 is $\{q_0, q_1\}$. Dit is de nieuwe starttoestand. We krijgen aldus de tabel:

δ'	a	b
$\rightarrow \{q_0, q_1\} = r_0$	$\{q_1\}$	$\{q_0, q_1, q_2, q_3\}$
$\{q_1\} = r_1$	$\{q_1\}$	$\{q_0, q_1, q_3\}$
$* \{q_0, q_1, q_2, q_3\} = r_2$	$\{q_0, q_1, q_3\}$	$\{q_0, q_1, q_2, q_3\}$
$\{q_0, q_1, q_3\} = r_3$	$\{q_0, q_1, q_3\}$	$\{q_0, q_1, q_2, q_3\}$

Het toestandsdiagram wordt:



Opgave 3 (16 %). (a) Formuleer het Pomplemma voor *reguliere* talen.
De taal L_3 over het alfabet $\{0, 1\}$ wordt gegeven door de grammatica

$$S \rightarrow 00S0 \mid 1S11 \mid \lambda \ .$$

(b) Bewijs, dat als $w \in L_3$ tenminste één 1 bevat en een staartstuk 0^k heeft, dan heeft w een beginstuk 0^{2k}.
(c) Bewijs dat de taal L_3 niet regulier is.

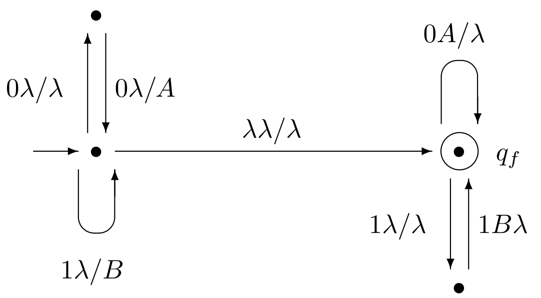
Uitwerking. (a) Zie boek of diktaat.

(b) Omdat w een afleiding volgens de gegeven grammatica heeft, met 0^k eindigt en toch tenminste één 1 bevat, moet deze afleiding beginnen met k toepassingen van de herschrijfregel $S \rightarrow 00S0$ (gevolgd door andere herschrijvingen waaronder tenminste eenmaal $S \rightarrow 1S11$). De afleiding van w begint dus met $S \Rightarrow^* 0^{2k}S0^k$. Dus $w = 0^{2k}u0^k$ met $u \in L_3$.

(c) Stel, dat L_3 regulier is. Dan is er een DFA M die L_3 accepteert. Stel dat k het aantal toestanden van M is. Volgens het pomplemma geldt dan dat elke $z \in L_3$ met $|z| \geq k$ een opsplitsing $z = uvw$ heeft met $|uv| \leq k$ en $v \neq \lambda$, en $uv^rw \in L_3$ voor elke $r \geq 0$. Ik kies $z = 0^{2k}1^30^k$. Dan geldt $z \in L_3$ en $|z| = 3k + 3 \geq k$. Dus heeft z een opsplitsing $0^{2k}1^30^k = uvw$ als boven. Wegens $|uv| \leq k$ en $uvw = z$ bestaan u en v uitsluitend uit nullen, zeg $v = 0^m$. Dan is $m > 0$ wegens $v \neq \lambda$. Er geldt $uw = uv^0w \in L_3$, maar ook $uw = 0^{2k-m}1^30^k$. Volgens onderdeel (b) moet hieruit volgen dat $2k - m \geq 2k$. Dit is in strijd met $m > 0$. We hebben hier een tegenspraak, dus L_3 is niet regulier.

Opgave 4 (10 %). Beschouw nogmaals de taal L_3 uit de vorige opgave. Construeer een stapelautomaat die de taal L_3 accepteert. Het is voldoende het toestandsdiagram (de gelabelde graaf) te geven en duidelijk te maken waarom deze stapelautomaat de taal L_3 accepteert.

Uitwerking. Er zijn zeer verschillende oplossingen. Eén ervan gebruikt een stapelsymbool A voor de productie $S \rightarrow 00S0$ om een 0 te onthouden, en B voor de productie $S \rightarrow 1B11$ om 11 te onthouden.



De overgang tussen de twee toestanden correspondeert met de productieregel $S \rightarrow \lambda$. Als de invoer en de stapel leeg zijn, wordt de invoer in q_f geaccepteerd.

Een stapelautomaat met slechts één toestand is fout, omdat de taal L van zo'n automaat voldoet aan $L = L^*$; en L_3 voldoet daar niet aan.

Opgave 5 (10 %). (a) Geef de definities van nuttige en nutteloze (useful and useless) symbolen voor een grammatica G.
(b) Gegeven is de grammatica G volgens

$$\begin{aligned}
S &\rightarrow aB \mid aE \mid bF \\
A &\rightarrow aA \mid a \\
B &\rightarrow dBd \mid C \mid bF \\
C &\rightarrow cC \mid \lambda \\
D &\rightarrow dAD \mid BF \\
E &\rightarrow E \mid bS \\
F &\rightarrow BD \mid AF \ .
\end{aligned}$$

Construeer volgens het standaardalgoritme een hiermee equivalente grammatica zonder nutteloze symbolen.

Uitwerking. (a) Zie boek of diktaat.

(b) Voortbrengend zijn in eerste instantie de symbolen A en C (wegens $A \to a$ en $C \to \lambda$), en vervolgens B (wegens $B \to C$) en S (wegens $S \to aB$) en E (wegens $E \to bS$). De nonterminals D en F herschrijven steeds naar elkaar en zijn dus niet voortbrengend. Deze moeten dus weggelaten worden. Dit geeft:

$$\begin{array}{lcl} S & \to & aB \mid aE \\ A & \to & aA \mid a \\ B & \to & dBd \mid C \\ C & \to & cC \mid \lambda \\ E & \to & E \mid bS \ . \end{array}$$

Bereikbaar zijn nu de nonterminals S (per definitie), en dus B en E wegens de producties van S, en dus C wegens $B \to C$. Daar houdt het op. De nonterminal A is dus niet meer bereikbaar, de andere vier S, B, C, E wel. We krijgen dus als eindresultaat de grammatica

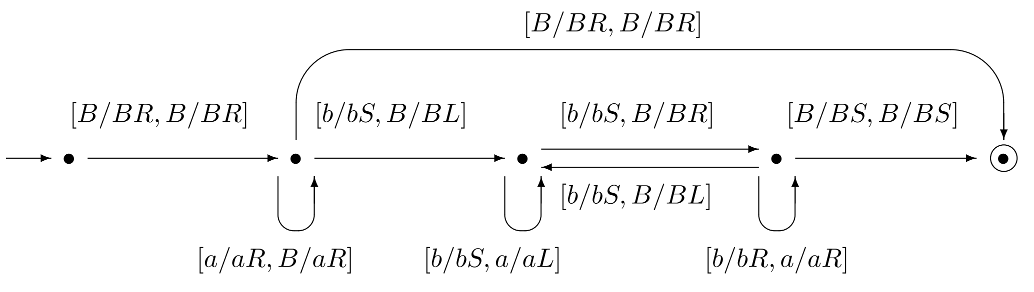
$$\begin{array}{lcl} S & \to & aB \mid aE \\ B & \to & dBd \mid C \\ C & \to & cC \mid \lambda \\ E & \to & E \mid bS \ . \end{array}$$

Opgave 6 (10 %). De taal L_6 over een alfabet $\{a, b\}$ bestaat uit de strings $a^k b^n$ voor natuurlijke getallen k en n waarbij n een veelvoud is van k; dus

$$L_6 = \{a^k b^n \mid \exists\, m : n = m \cdot k\} \ .$$

Construeer een Turingmachine die deze taal accepteert en op elke invoer eindigt. Je mag meer banden gebruiken en/of nondeterminisme. Beschrijf de werking van je Turingmachine, geef het toestandsdiagram (de gelabelde graaf) en maak duidelijk waarom deze Turingmachine aan de gestelde eisen voldoet.

Uitwerking. We gebruiken een tweebands Turingmachine, kopiëren de rij a's naar de tweede band, en passeren vervolgens telkens evenveel b's op band 1 als we a's op band 2 hebben; waarbij na elke doorgang de kop op band 2 eerst helemaal naar links gaat.



Opgave 7 (10 %). (a) Geef de definities van *recursieve* en *recursief opsombare* talen.

Het boek/diktaat beschrijft een manier om een Turingmachine M te coderen tot een string $R(M) \in \{0, 1\}^*$ en een universele Turingmachine die gebruikt kan worden om een aldus gecodeerde M te simuleren. We beschouwen nu de taal L_7 die bestaat uit de strings $R(M)1^n$ zodanig dat executie van Turingmachine M bij lege invoer (λ) termineert binnen n stappen.

(b) Is de taal L_7 recursief? Geef een argumentatie.
(c) Is de taal L_7 recursief opsombaar? Geef een argumentatie.

Uitwerking. (a) Zie boek of diktaat.

(b/c) We maken een Turingmachine M_7 om L_7 te accepteren. Deze gebruikt de universele Turingmachine UTM en een extra band. Machine M_7 controleert eerst dat de invoer van de vorm $R(M)1^n$ is. Zo niet, dan wordt de invoer verworpen. Anders wordt de string 1^n op de extra band gezet. M_7 gebruikt vervolgens UTM om de Turingmachine M te simuleren, waarbij in elke gesimuleerde stap van M één symbool 1 van de extra band gewist wordt. Als de enen op de extra band op zijn, stopt M_7 en wordt de invoer verworpen. Als UTM voordien gestopt is, wordt de invoer geaccepteerd. Machine M_7 accepteert aldus L_7. Hij termineert altijd. De taal L_7 is dus recursief, en dus ook recursief opsombaar.

Opgave 8 (10 %). Gegeven zijn twee recursieve talen L_a en L_b over een alfabet Σ. Bewijs dat de doorsnede $L_a \cap L_b$ recursief is. Gebruik hiertoe Turingmachines en beschrijf hoe en waarom deze werken, en hoe je daarmee het gestelde bewijst.

Uitwerking. Omdat L_a en L_b recursief zijn, zijn er altijd eindigende Turingmachines M_a en M_b die respectievelijk L_a en L_b accepteren. We maken nu een Turingmachine M_8 die eerst zijn invoer naar band 2 kopieert, dan M_a op band 1 loslaat en M_b op band 2 loslaat. Beide berekeningen eindigen, dus dit mag na elkaar of parallel. Machine M_8 accepteert de invoer alleen als zowel M_a als M_b hun kopie van de invoer accepteren. Dat is precies als de invoer een element is van $L_a \cap L_b$. Dus M_8 eindigt altijd en accepteert de taal $L_a \cap L_b$. Dus deze taal is recursief.

Opgave 9 (12 %). Gegeven een recursief opsombare taal L_9. Beschouw de taal L'_9 van de beginstukken van L_9, volgens $L'_9 = \{w \in \Sigma^* \mid \exists\, u \in \Sigma^* : wu \in L_9\}$. Bewijs dat L'_9 recursief opsombaar is.

Uitwerking. Er is een Turingmachine M_9 die de taal L_9 accepteert. We maken hieruit een nondeterministische Turingmachine M'_9 voor L'_9. Deze breidt eerst zijn invoer w op nondeterministische wijze uit tot een string wu en laat daarop M_9 los. Als dit accepterend eindigt, wordt de invoer w geaccepteerd. Als er geen accepterende berekening is, wordt de invoer verworpen.

Bonusopgave (10). Geef een recursieve taal L_{10} waarvoor de taal van de beginstukken $L'_{10} = \{w \in \Sigma^* \mid \exists\, u \in \Sigma^* : wu \in L_{10}\}$ niet recursief is, met argumentatie.

Uitwerking. De taal L_7 is recursief, zie boven. Neem dus $L_{10} = L_7$. De taal L'_{10} bestaat uit de beginstukken van strings $R(M)1^n$ waarbij M een TM is die op lege invoer binnen n stappen eindigt. We bewijzen dat L'_{10} niet recursief is door reductie naar de taal $L_t = \{R(M) \mid \lambda \in L(M)\}$, die volgens boek/diktaat niet recursief is.

De taal van de gecodeerde Turingmachines $L_c = \{R(M) \mid M\}$ is regulier en dus zeker recursief.

Er geldt $L_t \subseteq L'_{10} \cap L_c$, want als $w \in L_t$, dan is $w = R(M)$ voor zekere M met $\lambda \in L(M)$; deze M eindigt op invoer λ in een aantal stappen, zeg binnen n stappen; dus $R(M)1^n \in L_{10}$; dus $w = R(M) \in L'_{10}$; en ook $w \in L_c$.

Omgekeerd geldt ook $L'_{10} \cap L_c \subseteq L_t$, want als $w \in L'_{10} \cap L_c$, dan is w een beginstuk van een string van de vorm $R(M)1^n$ voor een TM M die op lege invoer binnen n stappen eindigt. Tevens is $w = R(M')$ voor zekere TM M'. Dus w eindigt op de deelstring 0^3, die nergens verder in w voorkomt. Dus $w = R(M)$ waarbij M is een TM die op lege invoer eindigt; dus $w \in L_t$.

Dit bewijst $L_t = L'_{10} \cap L_c$. Als L'_{10} recursief was, dan was L_t dus recursief volgens opgave 8. Dus L'_{10} is niet recursief.