Uitwerking Talen en Automaten, 6 juli 2006

Tijdsduur 3 uur. Gesloten boek tentamen.

Opgave 1 (16 %). Beschouw het alfabet $\Sigma = \{a, b, c, d\}$ en de taal L_1 over Σ die bestaat uit de strings $a^i b^k c^{i+k}$ voor getallen i, $k \in \mathbb{N}$.
(a) Laat zien dat de taal L_1 contextvrij is en geef er een contextvrije grammatica voor.
(b) Formuleer het Pomplemma voor *reguliere* talen.
(c) Bewijs dat de taal L_0 niet regulier is.

Uitwerking. (a) Beschouw de grammatica

$$\begin{aligned} S &\to aSc \mid T \\ T &\to bTc \mid \varepsilon \ . \end{aligned}$$

De nonterminal T brengt de taal $L(T) = \{b^k c^k \mid k \in \mathbb{N}\}$ voort. Het startsymbool S brengt voort $L(S) = \{a^i w c^i \mid i \in \mathbb{N}, w \in L(T)\}$. Dit laatste is precies de taal L_1. Omdat L_1 dus voortgebracht wordt door een contextvrije grammatica, is de taal L_1 contextvrij.

(b) Als L een reguliere taal is, is er een getal $n \in \mathbb{N}$ zodanig dat voor elke string $w \in L$ met $|w| \geq n$ er strings x, y, z zijn met $w = xyz$ en $|xy| \leq n$ en $|y| \geq 1$ en voor alle $k \in \mathbb{N}$ de eigenschap $xy^k z \in L$.

(c) Stel dat L_1 regulier is. Dan zou er een $n \in \mathbb{N}$ zijn als in (b). Beschouw bij deze n de string $w = a^n c^n$. Deze voldoet aan $w \in L$ en $|w| \geq n$. Dus er zijn strings x, y, z als beschreven in (b). Hiervoor geldt $xyz = w = a^n c^n$ en $|xy| \leq n$ en $|y| \geq 1$. Dus x en y bestaan uit symbolen a, zeg $y = a^r$ voor zekere $r > 0$. Dan geldt dat $xy^2 z \in L$, en dus $a^{n+r} c^n \in L$, wat niet kan voor $r > 0$. Dus L_1 is niet regulier.

Opgave 2 (12 %). Geef een algoritme om bij een gegeven reguliere expressie E over Σ te beslissen of de taal van E de lege string ε bevat.

Uitwerking. We doen dit met inductie naar de structuur van de regulier expressie E. Er zijn drie basisgevallen: $E = \emptyset$, $E = \varepsilon$, en $E = \mathbf{a}$ voor zeker symbool a. De taal van $E = \emptyset$ is leeg en bevat de lege string dus *niet*. De taal van $E = \varepsilon$ bevat (alleen) de lege string. De taal van $E = \mathbf{a}$ bevat de lege string niet.

Er zijn drie stap-gevallen: $E + F$, EF en E^*. De taal van $E + F$ bevat ε dan en slechts dan als de taal van E **of** de taal van F de lege string bevat. De taal van de concatenatie EF bevat ε dan en slechts dan als **zowel** de taal van E **als** de taal van F de lege string bevat. De taal van E^* bevat de lege string altijd.

Alternatief: converteer de reguliere expressie naar een ε-NFA; converteer deze naar een DFA. De lege string hoort tot de taal dan en slechts dan als de starttoestand van de DFA accepterend is.

Opgave 3 (16 %). Beschouw de taal L_3 die wordt voortgebracht door de grammatica G volgens

$$\begin{aligned} S &\to AA \mid C \mid cSd \\ A &\to da \mid \varepsilon \\ B &\to aB \mid BbCc \mid cDd \\ C &\to B \mid aC \mid S \ . \end{aligned}$$

Leid hieruit 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. De nonterminals B en D zijn niet voortbrengend (*generating*), de nonterminals S, A, C wel (het is overigens onjuist te zeggen dat D niet bestaat, D heeft alleen geen producties). We kunnen ons dus beperken tot de grammatica

$$
\begin{aligned}
S &\rightarrow\ AA \mid C \mid cSd \\
A &\rightarrow\ da \mid \varepsilon \\
C &\rightarrow\ aC \mid S\ .
\end{aligned}
$$

We zien nu dat A, C en S alle drie *nullable* zijn. Overeenkomstig het boek laten we nu de ε-producties weg en voegen producties toe om gedeeltelijke verdwijningen te compenseren. We krijgen aldus:

$$
\begin{aligned}
S &\rightarrow\ A \mid AA \mid C \mid cd \mid cSd \\
A &\rightarrow\ da \\
C &\rightarrow\ a \mid aC \mid S\ .
\end{aligned}
$$

Opgave 4 (14 %). Gegeven zijn twee contextvrije talen L_1 en L_2. Bewijs dat hun vereniging $L_1 \cup L_2$ ook contextvrij is.

Uitwerking. De talen L_1 en L_2 zijn contextvrij en worden dus voortgebraht door contextvrije grammatica's, zeg door $G_1 = (V_1, T_1, P_1, S_1)$ en $G_2 = (V_2, T_2, P_2, S_2)$. We mogen aannemen, dat V_1 en V_2 disjunct zijn (anders hernoemen we de nonterminals uit V_2). We kiezen een nieuw startsymbool S en vormen $V = V_1 \cup V_2 \cup \{S\}$. Als terminal alphabet nemen we $T = T_1 \cup T_2$. Als verzameling producties nemen we $P = P_1 \cup P_2 \cup \{S \rightarrow S_1, S \rightarrow S_2\}$. De nieuwe grammatica wordt nu $G = (V, T, P, S)$.

We bewijzen dat $L(G) = L_1 \cup L_2$. Omdat in de afleidingen van S_1 volgens G alleen nonterminals uit V_1 voorkomen, brengt de nonterminal S_1 van G precies de taal L_1 voort. De nonterminal S_2 brengt evenzo precies de taal L_2 voort. Omdat S alleen de herschrijfregels $S \rightarrow S_1$ en $S \rightarrow S_2$ heeft, brengt S precies de taal $L_1 \cup L_2$ voort. Dit bewijst $L_1 \cup L_2 = L(G)$. Derhalve is $L_1 \cup L_2$ contextvrij.

Opgave 5 (10 %).
(a) Geef definities voor de begrippen *recursief opsombaar* en *beslisbaar*.
(b) Is elke recursief opsombare taal ook beslisbaar?
(c) Is elke beslisbare taal ook recursief opsombaar?
(NB. Je mag *beslisbaar* beschouwen als hetzelfde als *recursief*, maar het is iets anders dan *recursief opsombaar*.)
Licht je antwoorden bij (b) en (c) kort toe; er worden geen bewijzen gevraagd.

Uitwerking. (a) Een taal L is recursief opsombaar als er een Turing machine M is met $L = L(M)$. Een taal L is beslisbaar als er een voor elke invoer eindigende Turing machine M is met $L = L(M)$.

(b) Nee, want in het boek wordt een taal L_u beschreven die recursief opsombaar en niet beslisbaar is.

(c) Ja. Uit definitie (a) volgt direct dat elke beslisbare taal tevens recursief opsombaar is.

Opgave 6 (10 %). Gegeven zijn de talen L_1 en L_2, en deterministische eenbands Turing machines M_1 en M_2 waarvoor geldt $L(M_1) = L_1$ en $L(M_2) = L_2$. We zijn geïnteresseerd in de vereniging $L_1 \cup L_2$.
(a) Beschrijf een deterministische (mogelijk meerbands) Turing machine M die voldoet aan $L(M) = L_1 \cup L_2$.
(b) De constructie van onderdeel (a) bewijst een bewering van de vorm: "Als de talen L_1 en L_2 ... zijn, dan is $L_1 \cup L_2$ ook ...." Welke bewering is dit?

Uitwerking. (a) We maken uit M_1 en M_2 een tweebands Turing machine M. Pas eerst M_1 en M_2 zo aan dat ze nooit verwerpend eindigen; als een string niet tot de taal behoort, eindigen ze niet. Machine M doet nu het volgende. M kopieert eerst de invoerstring van band 1 naar band 2. Vervolgens simuleert M de Turing machines M_1 en M_2, in stappen om de beurt, respectievelijk op de banden 1 en 2. Als één van beide in een accepterende toestand is, wordt de invoer geaccepteerd.

Dit gebeurt dus precies dan als de invoer element is van $L_1 \cup L_2$. Dit bewijst $L(M) = L_1 \cup L_2$.

(b) "Als L_1 en L_2 recursief opsombaar zijn, is ook $L_1 \cup L_2$ recursief opsombaar."

Opgave 7 (10 %). Turing machines kunnen gecodeerd worden in de ltaal *Ctm* die bepaald wordt door de reguliere expressie $((\mathbf{0}^+\mathbf{1})^5\mathbf{1})^+\mathbf{1}$. Beschrijf een ε-NFA die precies deze taal accepteert en minder dan 18 toestanden bevat.

Uitwerking. We nemen 13 toestanden, genummerd van 0 tot 12. De starttoestand is 0. De enige accepterende toestand is 12. Voor $0 \leq i < 5$ is $\delta(2i, 0) = \{2i+1\}$ en $\delta(2i+1, 0) = \{2i+1\}$ en $\delta(2i+1, 1) = \{2i+2\}$. Verder is $\delta(10, 1) = \{11\}$ en $\delta(11, 1) = \{12\}$ en $\delta(11, \varepsilon) = \{0\}$. In alle andere gevallen is $\delta(q, a)$ leeg. Maak zelf een schets.

Opgave 8 (12 %). Gegeven zijn twee beslisbare talen L_1 en L_2. Concatenatie geeft de taal $L = L_1 L_2$. Laat zien dat L ook beslisbaar is.

Uitwerking. Omdat L_1 en L_2 beslisbaar zijn, zijn er deterministische eenbands Turing machines M_1 en M_2 met $L_1 = L(M_1)$ en $L_2 = L(M_2)$, die beide voor alle invoer eindigen. We maken nu een driebands Turing machine M, die zijn invoer w achtereenvolgens op $|w| + 1$ manieren splitst volgens $w = xy$ en x kopieert naar band 2 en y kopieert naar band 3, vervolgens met M_1 beslist of $x \in L_1$ geldt, en zo ja met M_2 beslist of $y \in L_2$ geldt. Als beide gelden, wordt w geaccepteerd. Anders gaat M door met de volgende splitsing. Op band 1 nemen we twee sporen: het ene spoor voor de invoerstring w, het andere spoor voor een pointer om de splitsing in x en y aan te geven. Als alle splitsingen te vergeefs onderzocht zijn, wordt de string w verworpen. De taal van M is dus $L(M) = L_1 L_2$. Berekening van M eindigt bij elke invoer. Dus L is beslisbaar.