**TENTAMEN FUNDAMENTELE INFORMATICA 1 (I&E)**
Maandag 17 december 2012, 14.00 - 17.00 uur

---

Dit tentamen bestaat uit 7 opgaven. waarbij steeds tussen [ en ] staat hoeveel punten er ongeveer mee te verdienen zijn. In totaal zijn er 100 punten te verdienen.

Als je het antwoord op een onderdeel niet weet, en je hebt dat antwoord nodig bij een later onderdeel, dan kun je het antwoord 'kopen' bij de docent.

Geef gevraagde eindige automaten, stapelautomaten en Turing machines door middel van hun transitiediagram (het plaatje dus).

Als er bij een opgave gevraagd wordt om uitleg bij je antwoord, is het belangrijk dat je die ook geeft.

---

1. [10 pt]

   (a) Laat $L_1 = \{\Lambda, ab\}$ en $L_2 = \{aa, b\}$. Wat zijn de eerste zeven elementen in de canonieke volgorde van de taal $L_1 L_2^*$ ? Geef deze zeven elementen ook in de canonieke volgorde.

   (b) Laat nu $L_1$ en $L_2$ twee willekeurige talen zijn. Geldt altijd (dus voor elke $L_1$ en $L_2$) dat

   $$L_1 L_2^* \subseteq (L_1 \cup L_2)^*$$

   Zo ja, toon dit formeel aan.

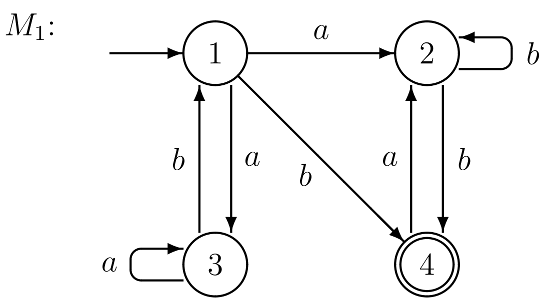
   Zo nee, geef een voorbeeld van talen $L_1$ en $L_2$ waarvoor het niet geldt, en een string $x$ die daar het bewijs van is.

---

2. [19 pt]

   (a) Geef een eindige automaat (FA) of (naar keuze) een niet-deterministische eindige automaat (NFA) voor de taal corresponderend met de reguliere expressie $b^*(a + ab)^*b^*b$.

   (b) Beschouw de volgende NFA $M_1$:




   Gebruik de subset-constructie om een (deterministische) FA $M_2$ te construeren, zó dat $L(M_2) = L(M_1)$. In je antwoord moet de relatie tussen $M_1$ en $M_2$ duidelijk zichtbaar zijn.

   (c) Geef een reguliere expressie corresponderend met de taal $L(M_1)$, voor de NFA $M_1$ hierboven.

---

3. [10 pt] Laat

$$L = \{x \in \{a,b\}^* \mid x \text{ bevat geen substring } abba\}.$$

   (a) Geef een eindige automaat $M$, zó dat $L(M) = L$.

   (b) Geef een context-vrije grammatica $G$, zó dat $L(G) = L$.

---

4. [16 pt]

   (a) Wanneer zeggen we dat een context-vrije grammatica $G = (V, \Sigma, S, P)$ in *Chomsky normaalvorm* is?

   (b) Beschouw de context-vrije grammatica $G$ met alfabet $\{a,b\}$, startsymbool $S$ en producties

$$S \to aSb \mid T \qquad T \to bTa \mid TT \mid \Lambda$$

   Construeer een context-vrije grammatica $G'$ in Chomsky normaalvorm, zó dat $L(G') = L(G) - \{\Lambda\}$. Leg duidelijk uit hoe je aan je antwoord komt, en geef ook tussenresultaten.

---

5. [15 pt] Bij een willekeurige context-vrije grammatica $G = (V, \Sigma, S, P)$ kunnen we de niet-deterministische top-down stapelautomaat $NT(G)$ construeren. Deze stapelautomaat accepteert precies de strings die $G$ genereert, ofwel: $L(NT(G)) = L(G)$.

   (a) Leg duidelijk uit hoe in het algemeen $NT(G)$ uit $G$ geconstrueerd wordt. Gebruik desgewenst een plaatje ter illustratie.

   (b) Met $NT(G)$ kunnen we afleidingen in $G$ simuleren op de stapel, terwijl we letter voor letter de invoerstring $x$ lezen.

      i. Betreft het hier linkspreferente, rechtspreferente of willekeurige afleidingen? Geef een intuïtieve motivatie voor je antwoord.

      ii. Stel dat we tijdens een berekening in $NT(G)$ voor een invoerstring $x$ een configuratie bereikt hebben, met als resterende invoer $y$ en stapelinhoud $\alpha$. Hoe ziet dan de string in de afleiding in $G$ eruit, die met deze configuratie overeenkomt?

---

6. [12 pt] Laat

$$L = \{x \in \{a,b\}^* \mid n_a(x) < n_b(x) + 2\}.$$

(a) Geef de eerste vier elementen in de canonieke volgorde van de taal $L$. Geef deze vier elementen ook in de canonieke volgorde.

(b) Geef een stapelautomaat $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$, zó dat $L(M) = L$.

Probeer er voor te zorgen dat $M$ deterministisch is, geen $\Lambda$-transities kent, en zo weinig mogelijk toestanden heeft. Lukt dit niet, dan kun je nog wel een deel van de punten verdienen.

(c) Geef een berekening in je antwoord $M$ van het vorige onderdeel voor de string $x = aaabbb$. Dat wil zeggen: een berekening vanaf $(q_0, aaabbb, Z_0)$ naar een configuratie waarin heel de string is gelezen.

---

7. [18 pt] Construeer een Turing machine $T$ die als invoer een string $x \in \{a,b\}^*$ heeft, en de volgende partiële functie $f$ berekent:

$$f(x) = \begin{cases} a^n & \text{als } x = a^n b^{2n} a^n \text{ voor zekere } n \geq 0 \\ \text{niet gedefinieerd} & \text{voor elke andere } x \end{cases}$$

Ofwel: $T$ controleert of $x$ van de vorm $a^n b^{2n} a^n$ is, en zo ja, dan laat hij $a^n$ over op de tape (als een unaire codering van het getal $n$).

Leg ook duidelijk uit hoe $T$ werkt. Leg in het bijzonder ook uit wat $T$ doet als zijn invoer $x$ niet van de gewenste vorm is. Wanneer ontdekt $T$ dat en wat doet hij dan?