TENTAMEN FUNDAMENTELE INFORMATICA 3
Maandag 20 augustus 2007, 10.00 - 13.00 uur

Dit tentamen bestaat uit 5 opgaven, waarbij steeds tussen [en] staat hoeveel punten er ongeveer mee te verdienen zijn. In totaal zijn er 100 punten te verdienen.

Opgave 1 [`23 pt`]
Laat $L_1 = \{a^m b^m b^n a^{2n} \mid m, n \geq 0\}$.

(a) Geef een deterministische PDA die L_1 accepteert met eindtoestanden.

Als je geen deterministische PDA kunt vinden, geef dan een niet-deterministische PDA.

(b) Zelfde vraag als bij onderdeel (a), maar probeer nu zo weinig mogelijk toestanden (eindtoestanden en niet-eindtoestanden samen) te gebruiken.

(c) Stel dat je een PDA M hebt die een taal L accepteert met eindtoestanden. Laat M' uit M ontstaan door eindtoestanden en niet-eindtoestanden te verwisselen. Je zou kunnen denken dat M' het complement L' van L accepteert. Geef drie redenen waarom dit niet in het algemeen werkt.

(d) Construeer een PDA die het complement L_1' van onze taal L_1 accepteert met eindtoestanden.

Opgave 2 [`22 pt`]
De context-vrije grammatica G heeft startsymbool S, terminaal alfabet $\{a, b, c\}$ en producties $S \to aSb \mid aTVc \quad T \to ba \mid bTa \quad V \to Va \mid a \mid b$.

(a) Beschrijf de taal die gegenereerd wordt door G.

(b) Elimineer de linksrecursie in G, zonder de gegenereerde taal te veranderen.

(c) Pas in G factorisatie toe op die producties bij eenzelfde niet-terminaal, waarvoor de rechterzijden met hetzelfde symbool beginnen.

Het resultaat van de vorige twee onderdelen noemen we H.

(d) Geef voor elk van de producties van H de bijbehorende lookahead, d.w.z. geef aan bij welke terminalen (als volgend invoersymbool) de productie hoort.

(e) Geef een DPDA die als top-down parser voor $L(H)$ kan fungeren.

Opgave 3 [`23 pt`]
In de eerste twee onderdelen van deze opgave mag je gebruik maken van de componenten InsertSymbol (σ) en DeleteSymbol (en niet van andere componenten, tenzij je ze zelf uitwerkt).

InsertSymbox (σ) verandert de tape-inhoud van een Turing machine van $y\underline{z}$ in $y\underline{\sigma}z$, en DeleteSymbol doet precies het omgekeerde. Hierbij is $y \in (\Gamma \cup \{\Delta\})^*$, is $\sigma \in \Gamma \cup \{\Delta\}$, en is $z \in \Gamma^*$. Zoals gebruikelijk is Γ hier het tape-alfabet.

(a) Geef een Turing machine T_{ab} voor de partiële functie $f_{ab} : \{a,b,c\}^* \to \{a,b,c\}^*$ zodat $f_{ab}(x) = x$ als $x = a^i b^j$ met $0 \leq i < j$, en $f_{ab}(x)$ is ongedefinieerd voor alle andere x.

Geef T_{ab} door middel van zijn toestandsdiagram en leg duidelijk uit hoe hij werkt.

(b) Construeer een Turing machine T die de taal

$$L = \{x \in \{a,b,c\}^* \mid x = a^i b^j c^k \text{ met } 0 \leq i < j < k\}$$

accepteert.

Geef T door middel van zijn toestandsdiagram en leg duidelijk uit hoe hij werkt.

(c) Geef een *unrestricted grammar* G die de taal L uit onderdeel (b) genereert.

Leg duidelijk uit wat de bedoeling is van de verschillende niet-terminalen en producties in G.

Opgave 4 [`12 pt`]
Laat T_1 een Turing machine zijn die een taal L_1 accepteert, en laat T_2 een Turing machine zijn die een taal L_2 accepteert. Beschrijf een algemene procedure om een Turing machine T te construeren die de vereniging $L_1 \cup L_2$ accepteert. T mag meerdere tapes hebben.

Wees nauwkeurig in je beschrijving. Toon ook aan dat T inderdaad $L_1 \cup L_2$ accepteert.

Opgave 5 [`20 pt`]

(a) Laat P_1 en P_2 twee beslissingsproblemen zijn. Leg uit hoe je aantoont dat $P_1 \leq P_2$.

Bekijk nu de volgende twee beslissingsproblemen.

Accepts-*aaaa*:
Gegeven een Turing machine T. Is de string $aaaa \in L(T)$?

Accepts
Gegeven een Turing machine T en een string x over het invoeralfabet van T. Is $x \in L(T)$?

(b) Toon aan dat Accepts-*aaaa* $\leq$ Accepts.

(c) Toon aan dat Accepts $\leq$ Accepts-*aaaa*.

(d) Gegeven is dat Accepts onbeslisbaar is. Toon aan dat dan ook Accepts-*aaaa* onbeslisbaar is.

(e) Bij Accepts-*aaaa* en Accepts geldt dus dat Accepts-*aaaa* $\leq$ Accepts en dat Accepts $\leq$ Accepts-*aaaa*.

Geldt zoiets altijd? Dat wil zeggen: als $P_1 \leq P_2$ voor beslissingsproblemen P_1 en P_2, geldt dan automatisch ook $P_2 \leq P_1$?

Zo ja, geef een bewijs. Zo nee, geef een tegenvoorbeeld, waarbij je uiteraard duidelijk maakt dat het inderdaad een tegenvoorbeeld is.

`einde tentamen`