TENTAMEN COMPUTABILITY
Vrijdag 26 maart 2021, 09.00 - 12.00 uur

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

Als je voor de Turingmachines die je moet tekenen, gebruik wilt maken van componenten, zul je die componenten ook moeten uitwerken (tekenen dus). Wanneer er bij een vraag om uitleg, motivatie of toelichting gevraagd wordt, is het belangrijk om die ook te geven.

---

1. [26 pt] Laat $L = \{a^i b^j a^k \;\mid\; 0 \leq i, j, k \text{ en } j > \max\{i, k\}\}$

   (a) Geef de eerste zes elementen van L in de canonieke volgorde.

   (b) Teken een gewone (deterministische, 1-tape) Turingmachine T met invoeralfabet $\{a, b\}$, zó dat $L(T) = L$.

   Hint: Controleer eerst of de letters in de invoer in de juiste volgorde staan.

   Leg ook duidelijk uit hoe T werkt.

---

2. [11 pt] Laat T_1 een Turingmachine zijn die een taal L accepteert.

   Beschrijf de werking van een **niet-deterministische** Turingmachine T_2 om L^* te accepteren.

   Hint: Vraag je af wanneer een invoer x in L^* zit.

   Omdat we van L niets anders weten dan dat $L = L(T_1)$, zal T_2 gebruik moeten maken van component T_1, of een variant van T_1. Beschrijf eventueel benodigde aanpassingen van T_1 duidelijk, op het niveau van de functionaliteit (je hoeft geen beschrijving op het niveau van transities te geven).

   N.B.: Je hoeft niet formeel te bewijzen dat T_2 daadwerkelijk L^* accepteert.

---

3. [20 pt] Laat G de *unrestricted grammar* zijn met startsymbool S, terminaalalfabet $\{1\}$, en de volgende producties:

$$
\begin{array}{rcl}
S & \to & TR \\
T & \to & TAB \mid L \\
AB & \to & B1A \qquad A1 \to 1A \qquad AR \to R \\
LB & \to & L \qquad L1 \to 1L \qquad LR \to \Lambda
\end{array}
$$

   (a) Wat zijn de kortste drie strings die door G gegenereerd worden? Geef voor de kortste twee strings ook een afleiding in G.

   (b) Wat is de taal die door G gegenereerd wordt?

   Motiveer je antwoord door het effect te beschrijven dat de verschillende producties op de gegenereerde string hebben.

---

4. [22 pt] In het bewijs van Stelling 8.13 in het boek wordt beschreven hoe je bij een willekeurige *unrestricted grammar* G een niet-deterministische Turingmachine (NTM) T kunt construeren, zó dat $L(T) = L(G)$. De NTM moet dus precies die strings accepteren, die de grammatica genereert. De werking van T voor een invoer x bestaat uit drie fases:

1. MovePastInput
2. Simuleer op de tape een willekeurige afleiding in G.
3. Vergelijk x met de in stap 2 afgeleide string y. Accepteer, dan en slechts dan als x gelijk is aan y.

(a) Beargumenteer dat de drie fases er inderdaad voor zorgen dat $L(T) = L(G)$.

(b) Laat $G = (V, \Sigma, S, P)$ een unrestricted grammar zijn met $V = \{S, B, C\}$, $\Sigma = \{a, b, c\}$ en de volgende producties:

$$S \to aBCS \mid \Lambda$$
$$Ba \to aB \qquad Ca \to aC \qquad CB \to BC \qquad B \to b$$
$$C \to c$$

Geef een NTM T_2 die op zijn tape een willekeurige afleiding in deze grammatica G simuleert (vanuit S) en het resultaat van de afleiding op de tape achterlaat. T_2 kan dus dienen als component voor de tweede fase van T.

Om precies te zijn: laat de string y het resultaat van de afleiding zijn, en laat q_0 de begintoestand van T_2 zijn. Dan moet T_2 beginnen in configuratie $q_0\Delta$ en na het succesvol simuleren van de afleiding eindigen in configuratie $h_a\Delta y$.

Leg ook duidelijk uit hoe T_2 werkt.

---

5. [21 pt] Laat S een (vaste) deelverzameling van $\{a, b\}^*$ zijn. Dan definiëren we het beslissingsprobleem *AcceptsAtLeast-S* als volgt:

> *AcceptsAtLeast-S*: Gegeven een Turingmachine T met invoeralfabet $\{a, b\}$, accepteert T elke string in S ?

Merk op dat als T elke string in S accepteert, en ook nog andere strings, dat T dan gewoon een ja-instantie van *AcceptsAtLeast-S* is.

Laat nu x, y en z drie vaste strings over alfabet $\{a, b\}$ zijn.

(a) Toon aan dat *AcceptsAtLeast*-$\{x\} \leq$ *AcceptsAtLeast*-$\{y, z\}$. Laat uiteraard ook zien dat aan alle eisen van een reductie is voldaan.

Als je geen geschikte reductie tussen de twee beslissingsproblemen weet, kun je een deel van de punten verdienen door uit te leggen hoe je voor algemene beslissingsproblemen P_1 en P_2 aantoont dat $P_1 \leq P_2$.

(b) Toon aan dat **niet** geldt dat *AcceptsAtLeast*-$\{x, y\} \leq$ *AcceptsAtLeast*-$\emptyset$. Bij het laatste beslissingsprobleem is de deelverzameling S dus gelijk aan $\emptyset$.

`einde tentamen`