HERTENTAMEN COMPUTABILITY
Vrijdag 14 juni 2024, 13.00 - 16.00 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. Wanneer er bij een vraag om uitleg, motivatie of toelichting gevraagd wordt, is het belangrijk om die ook te geven.

---

1. [31 pt]

   (a) Laat $f_1 : \{0,1\}^* \to \{0,1\}^*$ gedefinieerd zijn door $f_1(x) = x01y$, waarbij y de langste suffix van x is die alleen uit 1'en bestaat. Bijvoorbeeld, $f_1(10111011) = 101110110111$, $f_1(\Lambda) = 01$ en $f_1(01) = 01011$.

   Teken een gewone (deterministische, 1-tape) Turingmachine T_1 die de functie f_1 berekent.

   Als je bij dit onderdeel gebruik wilt maken van componenten, zul je die ook moeten uitwerken (tekenen dus).

   Leg ook duidelijk uit hoe T_1 werkt.

   (b) Laat nu $f_2 : \mathbb{N} \to \mathbb{N}$ gedefinieerd zijn door $f_2(n) = n(n+1)/2 = 1 + 2 + 3 + \cdots + n$.

   Teken een gewone (deterministische, 1-tape) Turingmachine T_2 die de functie f_2 berekent. T_2 moet gebruik maken van de unaire representatie van de natuurlijke getallen.

   Bij dit onderdeel mag je gebruik maken van de component T_1 uit onderdeel (a) en van de componenten *NB*, *PB*, *Insert*(σ) en *Delete* zoals die in het boek beschreven zijn. Als je T_1 daarbij iets moet aanpassen, beschrijf de benodigde aanpassing dan precies, op het niveau van toestanden en transities. Andere componenten mag je alleen gebruiken als je ze zelf uitwerkt (tekent dus). Wellicht ten overvloede:

   - *NB* verplaatst de leeskop naar de eerste Δ rechts van de huidige positie,
   - *PB* verplaatst de leeskop (zo mogelijk) naar de eerste Δ links van de huidige positie,
   - *Insert*(σ) verandert de tape-inhoud van $y\underline{z}$ in $y\underline{\sigma}z$ (waarbij z geen Δ bevat),
   - *Delete* verandert de tape-inhoud van $y\underline{\sigma}z$ in $y\underline{z}$ (waarbij z geen Δ bevat).

   Leg ook duidelijk uit hoe T_2 werkt.

2. [18 pt] Laat de taal XX als volgt gedefinieerd zijn:

$$XX = \{ss \mid s \in \{a,b\}^*\}$$

   Een 1-tape Turingmachine voor taal XX vereist voor een invoer x al gauw een aantal stappen dat kwadratisch is in de lengte van x. Teken een **2-tapes** Turingmachine T die in een **lineair** aantal stappen bepaalt of een invoer $x \in \{a,b\}^*$ wel of niet in XX zit. In het eerste geval moet T (uiteraard) accepteren, in het tweede geval (impliciet of expliciet) verwerpen of crashen.

   Als je voor T gebruik wilt maken van componenten, moeten dat wel 2-tapes componenten zijn, en zul je ze ook moeten uitwerken (tekenen dus).

   Leg ook duidelijk uit hoe T werkt.

3. [20 pt] Bij een eerder tentamen werd gevraagd om een unrestricted grammar voor de taal
$$L = \{a^i b^j a^k \mid 0 \leq i < j < k\}$$
Wellicht ten overvloede: de eerste vijf elementen in de canonieke volgorde van L zijn: $baa, baaa, baaaa, bbaaa, abbaaa$.

Hieronder moet je vier keer een vraag met 'ja' of 'nee' beantwoorden. Als het antwoord 'ja' is, hoef je dat niet toe te lichten.

(a) Laat G_1 de unrestricted grammar zijn met startvariabele S en de volgende producties:

$$\begin{array}{llllll} & S \to SABC \mid SBC \mid SC \mid LBCC & & & & \\ BA \to AB & CA \to AC & CB \to BC & & & \\ LA \to a & aA \to aa & aB \to ab & bB \to bb & bC \to ba & aC \to aa \end{array}$$

Hierbij worden eerst A's gegenereerd voor a^i, B's voor b^j en C's voor a^k, de hoofdletters worden vervolgens gesorteerd, en ten slotte worden ze van links naar rechts omgezet in de bedoelde terminalen.

i. Is $L(G_1) \subseteq L$? Zo nee, geef een element x van $L(G_1)$ dat niet in L zit. Geef in dat geval ook een afleiding van x in G_1. Als je in die afleiding meerdere 'gelijksoortige' producties achter elkaar moet toepassen, mag je die stappen samenvatten met $\Rightarrow^*$.

ii. Is $L \subseteq L(G_1)$? Zo nee, geef een element x van L dat niet in $L(G_1)$ zit. Beredeneer in dat geval ook waarom x niet afgeleid kan worden in G_1.

(b) Laat G_2 de unrestricted grammar zijn met startvariabele S en de volgende producties:

$$\begin{array}{llll} S \to aSBBCC \mid SBC \mid SC \mid LBCC & & & \\ CB \to BC & & & \\ LB \to b & bB \to bb & bC \to ba & aC \to aa \end{array}$$

Globaal hebben de producties van G_2 dezelfde functie als de producties van G_1. Belangrijk verschil is dat a^i direct links van S wordt gegenereerd.

i. Is $L(G_2) \subseteq L$? Zo nee, geef een element x van $L(G_2)$ dat niet in L zit. Geef in dat geval ook een afleiding van x in G_2. Als je in die afleiding meerdere 'gelijksoortige' producties achter elkaar moet toepassen, mag je die stappen samenvatten met $\Rightarrow^*$.

ii. Is $L \subseteq L(G_2)$? Zo nee, geef een element x van L dat niet in $L(G_2)$ zit. Beredeneer in dat geval ook waarom x niet afgeleid kan worden in G_2.

---

4. [16 pt]

   (a) Laat $T = (Q, \Sigma, \Gamma, q_0, \delta)$ een willekeurige Turingmachine zijn, en laat $a_1 \ldots a_n \in \Sigma^*$ een mogelijke invoer voor T zijn.

   Geef de initiële configuratie van T voor invoer $a_1 \ldots a_n$.

   Als je het antwoord op dit onderdeel niet weet, dan kun je het 'kopen' bij de docent. Wellicht kun je dan wel onderdeel (b) maken.

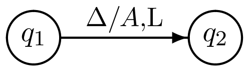
   In het bewijs van Stelling 8.14 in het boek wordt beschreven, hoe je bij een willekeurige Turingmachine $T = (Q, \Sigma, \Gamma, q_0, \delta)$ een unrestricted grammar $G = (V, \Sigma, S, P)$ kunt construeren, zó dat $L(G) = L(T)$.

   De grammatica G bevat drie soorten producties. De eerste soort is bedoeld om initiële configuraties van de Turingmachine na te bouwen, met twee kopieën van de tape-inhoud. De tweede soort producties is voor het simuleren van berekeningen van de Turingmachine.

   (b) Hoe ziet de string in de grammatica G er precies uit, die het resultaat is van de eerste soort producties voor de invoer $a_1 \ldots a_n$, d.w.z. de string corresponderend met de initiële configuratie van T voor deze invoer?

   (c) Wat is het doel van elk van de twee kopieën van de tape-inhoud in de string uit het vorige onderdeel?

   (d) Geef alle producties van de tweede soort in G die volgens de constructie van Stelling 8.14 corresponderen met de volgende transitie van T:



   Je mag de producties algemeen beschrijven in termen van elementen σ_i van Σ en/of Γ en/of $\{\Delta\}$. Maak dan wel duidelijk wat de σ_i's kunnen zijn. Als alternatief mag je ook aannemen dat $\Sigma = \{a, b\}$ en dat $\Gamma = \{a, b, A\}$.

---

5. [15 pt] Beschouw de volgende twee beslissingsproblemen:

   Accepts-ab:
   Gegeven een Turingmachine T_1, is $ab \in L(T_1)$?

   Accepts-$\{ba\}$:
   Gegeven een Turingmachine T_2, is $L(T_2) = \{ba\}$?

   Er is gegeven dat *Accepts-ab* niet beslisbaar is. Toon aan dat ook *Accepts-*$\{ba\}$ niet beslisbaar is, met behulp van een reductie met *Accepts-ab*. Laat uiteraard ook zien dat aan alle eisen van een reductie is voldaan, en vergeet niet om de conclusie te trekken.

`einde tentamen`