TENTAMEN COMPUTABILITY
Donderdag 30 maart 2023, 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. Wanneer er bij een vraag om uitleg, motivatie of toelichting gevraagd wordt, is het belangrijk om die ook te geven.

---

1. [30 pt] Bij onderdelen (b) en (c) van deze opgave mag je gebruik maken van de componenten *NB*, *PB*, *Insert*(σ) en *Delete* zoals die in het boek beschreven zijn. 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).

   (a) Geef de eerste zes strings in de canonieke volgorde van $\{a, b\}^*$.
   Als je het antwoord op dit onderdeel niet weet, dan kun je het 'kopen' bij de docent. Wellicht kun je dan wel onderdelen (b) en (c) maken.

   (b) Teken een gewone (deterministische, 1-tape) Turing machine T_1 die de functie $f_1 : \{a, b\}^* \to \{a, b\}^*$ berekent, gedefinieerd door

   $$f_1(x) = \text{ de opvolger van } x \text{ in de canonieke volgorde van } \{a, b\}^*$$

   (c) Teken een gewone (deterministische, 1-tape) Turing machine T_2 die de partiële functie $f_2 : \mathbb{N} \to \{a, b\}^*$ berekent, gedefinieerd door

   $$f_2(n) = \begin{cases} \text{niet gedefinieerd} & \text{als } n = 0 \\ \text{het } n\text{-de element in de canonieke volgorde van } \{a, b\}^* & \text{als } n \geq 1 \end{cases}$$

   Voor de goede orde: we kennen geen nul-de element in de canonieke volgorde van $\{a, b\}^*$. Bij onderdeel (a) heb je dus $f_2(1), f_2(2), \ldots, f_2(6)$ gegeven.
   Voor het natuurlijke getal n gebruiken we de unaire representatie (we zouden f_2 dus ook kunnen zien als een partiële functie van $\{1\}^*$ naar $\{a, b\}^*$).
   Je mag voor T_2 ook gebruik maken van T_1 als component. Als je T_1 daarbij iets moet aanpassen, beschrijf de benodigde aanpassing dan precies.
   Leg ook duidelijk uit hoe T_2 werkt.

---

2. [19 pt] Laat $L \subseteq \{a, b\}^*$ een taal zijn. We definiëren $P(L)$ als de taal van alle prefixen van elementen van L. Formeel:

$$P(L) = \{x \in \{a, b\}^* \mid \ xy \in L \text{ voor zekere } y \in \{a, b\}^*\}$$

   (a) Geef $P(L_1)$ voor de taal $L_1 = \{ab, babb\}$.

   Stel dat T_1 een Turingmachine is, zó dat $L(T_1) = L$ (L is dus weer willekeurig). We willen nu ook $P(L)$ accepteren. Daarvoor kunnen we gebruik maken van T_1.

   (b) Leg duidelijk uit waarom het volgende algoritme (met invoer x) niet per se werkt om $P(L)$ te accepteren:
   $y = \Lambda$;
   do
   voer T_1 uit met invoer xy;
   if (T_1 hierbij niet accepteert)
   y is volgende string in $\{a, b\}^*$ (in canonieke volgorde);
   while (T_1 heeft niet geaccepteerd);
   accepteer;

   (c) Teken een niet-deterministische Turingmachine T_2, zó dat $L(T_2) = P(L)$. T_2 moet gebruik maken van T_1 als component, en daadwerkelijk de kracht van niet-determinisme gebruiken bij het accepteren van $P(L)$. Andere componenten mag je alleen gebruiken als je ze zelf uitwerkt (tekent dus).

---

3. [21 pt]

   (a) Laat de taal XX als volgt gedefinieerd zijn:

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

   Geef een *unrestricted grammar* G_1, zó dat $L(G_1) = XX$.
   Leg uit wat de functie is van de diverse variabelen en producties in G_1.

   (b) Stel dat G_2 een unrestricted grammar is, met startvariabele T, zó dat $L(G_2) = L$ voor zekere taal $L \subseteq \{a, b\}^*$.
   Laat nu G_3 een unrestricted grammar zijn die alle variabelen en producties van G_2 bevat, en daarbovenop twee extra variabelen S en E (met S als startvariabele van G_3) en de volgende extra producties:

$$S \to ET \qquad Ea \to E \qquad Eb \to E \qquad E \to \Lambda$$

   Beschrijf de elementen van $L(G_3)$, in termen van L. Motiveer je antwoord.

---

4. [8 pt]

   (a) We noemen een taal $L \subseteq \Sigma^*$ *recursief opsombaar* als er een Turingmachine bestaat die L accepteert. Wanneer noemen we L (volgens onze definitie) *recursief*?

   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.

   (b) Toon aan dat als L recursief is, dat L dan ook recursief opsombaar is. Doe dit door de werking van een Turingmachine te beschrijven die L accepteert.

---

5. [22 pt] Beschouw de volgende twee beslissingsproblemen:

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

   ExistsXReachesState:
   Gegeven een Turingmachine T_2 en een nonhalting state q van T_2, bestaat er een invoer x_2 voor T_2, zó dat T_2 in de berekening voor x_2 ooit toestand q bereikt ?

   (a) De stelling van Rice luidt als volgt:

   > Als R een niet-triviale taaleigenschap (language property) van Turingmachines is, dan is het beslissingsprobleem
   > P_R: Gegeven een Turingmachine T, heeft T eigenschap R ?
   > niet beslisbaar.

   Toon aan dat beslissingsprobleem *Accepts-*Λ aan alle voorwaarden voor de stelling van Rice voldoet. Gebruik (en teken) voor het aantonen van niet-trivialiteit concrete Turingmachines.

   (b) Voor *Accepts-*Λ volgt de niet-beslisbaarheid dus rechtstreeks uit de stelling van Rice.

   Toon aan dat ook *ExistsXReachesState* niet beslisbaar is, met behulp van een reductie met *Accepts-*Λ. Laat uiteraard ook zien dat aan alle eisen van een reductie is voldaan, en vergeet niet om de conclusie te trekken.

`einde tentamen`