EXTRA TENTAMEN FUNDAMENTELE INFORMATICA 3
Donderdag 22 juni 2017, 14.00 - 17.00 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.
Geef de gevraagde Turingmachines door middel van hun transitiediagram.
Wanneer er bij een vraag om uitleg of toelichting gevraagd wordt, is het belangrijk om die ook te geven.
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.

---

1. [24pt] Bij deze opgave moet je twee Turingmachines construeren. Als je daarbij gebruik wilt maken van componenten, zul je die ook moeten uitwerken.

   (a) Construeer een Turingmachine T_1 die de functie $f_1 : \mathbb{N} \to \mathbb{N}$ berekent, gedefinieerd door
   $$f_1(x) = \lceil x/3 \rceil$$
   ('x gedeeld door 3, naar boven afgerond'). Ga hierbij uit van de unaire representatie van de natuurlijke getallen.

   (b) Laat
   $$L = \{yzy \mid y, z \in \{a, b\}^* \text{ en } |y| = |z|\}$$
   Ofwel: L bevat strings x in $\{a, b\}^*$ die in drie even lange substrings zijn te splitsen, waarbij de eerste en de laatste substring aan elkaar gelijk zijn.
   Construeer een Turingmachine T_2 die strings $x \in \{a, b\}^*$ als invoer heeft en x accepteert, dan en slechts dan als $x \in L$.
   Leg ook duidelijk uit hoe T_2 werkt.

---

2. [16pt of 18pt]

   (a) Wanneer noemen we een *unrestricted* grammatica $G = (V, \Sigma, S, P)$ context-gevoelig?

   (b) Kies een van de volgende twee onderdelen (b1) of (b2), die feitelijk alleen verschillen in de taal L_i.

   (b1) (12 pt)

      i. Geef een context-gevoelige grammatica G_1 voor de taal

      $$L_1 = \{a^{2n}db^ndc^{3n} \mid n \geq 1\}$$

      Leg uit wat de functie is van de diverse variabelen en producties in G_1.
      Als het niet lukt om een context-gevoelige grammatica te bedenken voor de gekozen taal, kun je het opgegeven aantal punten -1 verdienen met een unrestricted grammatica voor de taal.

      ii. Geef een afleiding in G_1 voor het woord *aadbdccc*.

   (b2) (14 pt)

      i. Geef een context-gevoelige grammatica G_2 voor de taal

      $$L_2 = \{a^{2n}da^nda^{3n} \mid n \geq 1\}$$

      Leg uit wat de functie is van de diverse variabelen en producties in G_2.
      Als het niet lukt om een context-gevoelige grammatica te bedenken voor de gekozen taal, kun je het opgegeven aantal punten -1 verdienen met een unrestricted grammatica voor de taal.

      ii. Geef een afleiding in G_2 voor het woord *aadadaaa*.

---

3. [18pt]

   (a) Leg duidelijk uit in welke opzichten een lineair begrensde automaat (LBA) afwijkt van een niet-deterministische Turingmachine. Besteed hierbij ook aandacht aan de initiële configuratie voor een invoer x.

   In het boek wordt een constructie beschreven om bij een gegeven context-gevoelige grammatica G een LBA M te construeren, zó dat $L(M) = L(G)$. M accepteert dus precies die strings die gegenereerd worden door G. De werking van M bestaat uit drie fases, en de tweede fase bestaat uit het (voorwaarts) simuleren van een afleiding in G.

   (b) Waar vindt dat simuleren van een afleiding precies plaats? Het antwoord 'op de tape' is niet voldoende.

   (c) Beschrijf (in woorden) hoe het simuleren van een afleiding in G gebeurt door M. Besteed hierbij aandacht aan de volgende vragen: Hoe begint de simulatie? Wat gebeurt er iedere stap van de simulatie? Wanneer stopt de simulatie? Waar is er sprake van niet-determinisme?

---

4. [17pt] Beschouw de volgende twee beslissingsproblemen:

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

   AcceptsEverything:
   Gegeven een Turingmachine T met invoeralfabet Σ, is $L(T) = \Sigma^*$?

   (a) Toon aan dat *Accepts-*$\Lambda \leq$ *AcceptsEverything*. Laat hierbij uiteraard ook zien dat aan alle eisen van een reductie is voldaan.
   Als je geen geschikte reductie tussen Accepts-Λ *en* AcceptsEverything *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) Wat is er mis met de volgende, voorgestelde, omgekeerde reductie *AcceptsEverything* $\leq$ *Accepts-*Λ:

   > Laat T_1 een willekeurige instantie van *AcceptsEverything* zijn. Dan construeren we een instantie T_2 van *Accepts-*Λ als volgt:
   >
   > Uitgaande van een lege tape gaat T_2 op zijn tape T_1 simuleren voor achtereenvolgens alle mogelijke invoerstrings x uit Σ^*, in canonieke volgorde: $x = \Lambda, a, b, aa, ab, \ldots$ (voor het geval dat $\Sigma = \{a, b\}$). Als T_1 bij dit simuleren een invoerstring accepteert, veegt T_2 de tape schoon en gaat hij T_1 voor de volgende invoerstring simuleren. Als T_1 een invoerstring niet accepteert, gaat T_2 naar h_r. Als T_1 op deze manier alle invoerstrings accepteert, gaat T_2 naar h_a.

   Motiveer je antwoord.

Z.O.Z.

5. [23pt] Bij geen enkel onderdeel van deze opgave is het nodig om projecties en dergelijke te gebruiken bij de beschrijving van de gevraagde functies. Verder mag je er in deze opgave van uitgaan

   - dat operaties als optellen, aftrekken en vermenigvuldigen primitief recursief zijn,
   - dat eenvoudige vergelijkingen tussen twee getallen primitief recursief zijn,
   - dat operaties als $\wedge$, $\vee$ en $\neg$ de primitieve recursiviteit behouden,
   - dat volledige gevalsonderscheiding de primitieve recursiviteit behoudt.

   (a) Laat P een $(n+1)$-place predikaat zijn. De begrensde existentiële kwantificatie van P is het $(n+1)$-place predikaat E_P gedefinieerd door

   $$E_P(X,k) = (\text{er bestaat een } y \text{ met } 0 \leq y \leq k, \text{ zó dat } P(X,y) \text{ is true})$$

   Toon aan dat als P primitief recursief is, ook E_P primitief recursief is. Doe dit door van de karakteristieke functie χ_{E_P} aan te tonen dat die primitief recursief is.
   Indien je gebruik maakt van de operatie primitieve recursie, geef dan de gebruikte functies g en h ook voor algemene parameters $X, x_2, x_3, \ldots$

   Laat $T = (Q, \Sigma, \Gamma, q_0, \delta)$ een Turingmachine zijn.

   (b) Leg precies uit wat het *tape number* behorend bij een configuratie xqy van T is. Hierin is $x = x_0x_1 \ldots x_{i-1} \in (\Gamma \cup \{\Delta\})^*$, q een toestand en $y = x_ix_{i+1} \ldots x_j \in (\Gamma \cup \{\Delta\})^*$.
   N.B.: het gaat dus alleen om het tape number, en niet om het *configuration number*.

   (c) Aan welke eisen moet een natuurlijk getal t voldoen om een geldig tape number voor T te zijn?

   (d) Toon aan dat het predikaat

   $$t \text{ is een geldig tape number voor } T$$

   primitief recursief is.
   Bij dit onderdeel mag je er tevens vanuitgaan

   - dat begrensde kwantificatie de primitieve recursiviteit behoudt,
   - dat de functies $PrNo(i)$ en $Exponent(i,m)$ primitief recursief zijn. Wellicht ten overvloede: De functie $PrNo(i)$ levert het i-de priemgetal op, ofwel: $PrNo(0) = 2, PrNo(1) = 3, PrNo(2) = 5, \ldots$.
     De functie $Exponent(i,m)$ levert de exponent van $PrNo(i)$ in de priemfactorisatie van m op, bijvoorbeeld $Exponent(1,18) = 2$, want de exponent van priemgetal 3 in $m = 18$ is 2. Als $m = 0$, definiëren we $Exponent(i,m) = 0$.

`einde tentamen`