TENTAMEN FUNDAMENTELE INFORMATICA 3
Vrijdag 6 juni 2014, 14.00 - 17.00 uur

Dit tentamen bestaat uit 6 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 Turing machine door middel van zijn 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. [16 pt] Laat T_1 een Turing machine zijn die twee unaire, niet-negatieve gehele getallen x en y omwerkt tot y en $x + y$ (in die volgorde). T_1 voert dus een berekening uit van configuratie $q_1\Delta x\Delta y$ naar configuratie $h_a\Delta y\Delta z$, waarbij q_1 de begintoestand van T_1 is en z de unaire representatie van $x + y$ is. Bijvoorbeeld: van $q_1\Delta 11\Delta 111$ naar $h_a\Delta 111\Delta 11111$.

   Construeer nu een Turing machine T_2 die als invoer een unair, niet-negatief getal n heeft, en dan het n^{e} Fibonacci-getal F_n berekent. Om precies te zijn: als q_2 de begintoestand van T_2 is, dan moet T_2 een berekening uitvoeren van $q_2\Delta n$ naar $h_a\Delta F_n$, waarbij je n en F_n moet zien als unaire representaties.

   Je mag hierbij gebruik maken van T_1, omdat je die kunt gebruiken om een volgend Fibonacci-getal te berekenen. Verder mag je gebruik maken van de componenten *NB*, *PB*, *Insert*(σ) en *Delete* zoals die in het boek beschreven zijn (en niet van andere componenten, tenzij je ze zelf uitwerkt). 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),
   - de Fibonacci-getallen F_n zijn als volgt gedefinieerd:

$$F_n = \begin{cases} 0 & \text{als } n = 0 \\ 1 & \text{als } n = 1 \\ F_{n-2} + F_{n-1} & \text{als } n \geq 2 \end{cases}$$

   Leg duidelijk uit hoe T_2 werkt.

---

2. [19 pt] In Stelling 7.31 van het boek wordt bewezen dat voor iedere niet-deterministische Turing machine (NTM) T_1 een gewone (deterministische) Turing machine (TM) T_2 bestaat met $L(T_2) = L(T_1)$.

   In het bewijs wordt voor het gemak aangenomen dat er in de NTM T_1 voor iedere combinatie van niet-halting toestand en tapesymbool precies twee mogelijke transities zijn. De paden die T_1 voor een bepaalde invoer x kan doorlopen corresponderen dan mooi met bitstrings. De TM T_2 gaat nu met behulp van vier tapes alle mogelijke paden van T_1 simuleren. T_2 gaat naar h_a zodra hij een pad van T_1 tegenkomt dat naar h_a leidt.

   (a) Moet T_2 dit via een breadth-first-search of via een depth-first-search doen? Motiveer je antwoord.

   Tijdens het simuleren van alle mogelijke paden/bitstrings van T_1 bevat

   - de eerste tape van T_2 steeds de invoer x
   - de tweede tape van T_2 de bitstring die gesimuleerd wordt
   - de derde tape de inhoud van de tape van T_1 tijdens het gevolgde pad, voorafgegaan door een speciaal \$ symbool; voordat we een pad/bitstring gaan simuleren, kopiëren we daarom de inhoud van de eerste tape naar deze tape,
   - de vierde tape de bitstrings waarbij T_1 tijdens het simuleren naar h_r ging, crashte of van de tape afliep.

   (b) Wat is de functie van het speciale symbool \$ op de derde tape van T_2? En *wanneer* komen we dat symbool tegen?

   (c) Wat is de functie van de bitstrings op de vierde tape van T_2? En *wanneer* bekijken we ze?

---

3. [17 pt] Laat F_n het n^{e} Fibonacci-getal zijn (zie opgave 1). Geef een onbeperkte grammatica G, zó dat

$$L(G) = \{1^{F_n} \mid n \geq 0\}$$

   Ofwel: G genereert de Fibonacci-getallen in unaire representatie. Leg uit wat de functie is van de diverse variabelen en producties in G.

   Hint: bij een mogelijke oplossing kunnen we beginnen met de volgende productie: $S \to LM_1 1 M_2 R$.
   Nu staat tussen L en M_1 het getal $F_0 = 0$, tussen M_1 en M_2 staat $F_1 = 1$, en de ruimte tussen M_2 en R kunnen we steeds gebruiken om het volgende Fibonacci-getal te berekenen.

---

4. [14 pt] Een alfabet Σ is per definitie een niet-lege, eindige verzameling. We nemen aan dat de symbolen die we willen gebruiken om een taal te genereren of te accepteren (in het bijzonder de symbolen uit een alfabet Σ) allemaal afkomstig zijn uit een (aftelbare) verzameling $\mathcal{S} = \{a_1, a_2, a_3, \ldots\}$.

   (a) Toon met behulp van een Cantorwandeling aan dat er maar aftelbaar veel deelverzamelingen van $\mathcal{S}$ zijn met twee elementen.

   (b) Geef voor elk van de volgende vijf verzamelingen L_i aan of L_i aftelbaar is, overaftelbaar is, of (voor L_1 tot en met L_4) of dat afhankelijk is van het alfabet Σ.

   - L_1 = de verzameling van alle strings over een alfabet Σ
   - L_2 = de verzameling van alle talen over een alfabet Σ
   - L_3 = de verzameling van alle recursief-opsombare talen over een alfabet Σ
   - L_4 = de verzameling van alle niet-recursief-opsombare talen over een alfabet Σ
   - L_5 = de verzameling van alle mogelijke alfabetten Σ

   Je hoeft je antwoorden bij dit onderdeel niet toe te lichten.

---

5. [20 pt]

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

   Beschouw nu de volgende twee beslissingsproblemen:

   Accepts-Λ: gegeven een Turing machine T_1, is $\Lambda \in L(T_1)$?

   WritesSymbol: gegeven een Turing machine T_2 en een symbool σ uit het tape alphabet van T_2, schrijft T_2 ooit σ op de tape bij invoer Λ ?

   (b) Toon aan dat *Accepts*-$\Lambda \leq$ *WritesSymbol* Geef een duidelijke beschrijving van de reductie. Laat uiteraard ook zien dat aan alle eisen van een reductie voldaan is.

   (c) Toon aan dat *WritesSymbol* $\leq$ *Accepts*-Λ Je hoeft nu alleen maar een (duidelijke) beschrijving van de reductie te geven. Je hoeft nu dus niet te laten zien dat aan alle eisen van een reductie is voldaan.

---

6. [14 pt] In deze opgave mag je gebruik maken van de functies *PrNo* en *Exponent* uit het boek. Wellicht ten overvloede:

   - voor $i \geq 0$ is $PrNo(i)$ het i^{e} priemgetal (beginnend bij $PrNo(0) = 2$)
   - voor $i \geq 0$ en $x \geq 1$ is $Exponent(i, x)$ de exponent van het i^{e} priemgetal in de priemfactorisatie van x.

   (a) Wat is het Gödel-nummer van een rij niet-negatieve getallen $(x_0, x_1, \ldots, x_k)$ met $k \geq 0$, ofwel $gn(x_0, x_1, \ldots, x_k)$?

   Stelling 10.19 in het boek luidt als volgt:

   > Stel dat $g : \mathbb{N}^n \to \mathbb{N}$ en $h : \mathbb{N}^{n+2} \to \mathbb{N}$ primitieve recursieve functies zijn, en dat f uit g en h verkregen wordt met 'course-of-values' recursie; dat wil zeggen:
   >
   > $$\begin{aligned} f(X, 0) &= g(X) \\ f(X, k+1) &= h(X, k, gn(f(X, 0), \ldots, f(X, k))) \qquad \text{als } k \geq 0 \end{aligned}$$
   >
   > Dan is f primitief recursief.

   Hierin is $gn(f(X, 0), \ldots, f(X, k))$ het Gödel-nummer van $(f(X, 0), \ldots, f(X, k))$. De waarde van $f(X, k+1)$ mag dus (mede) afhangen van alle voorgaande functiewaardes.

   Een typisch voorbeeld van een functie f die je met 'course-of-values' recursie kunt definiëren, is de functie F van de Fibonacci-getallen:

   $$\begin{aligned} F(0) &= 0 \\ F(1) &= 1 \\ F(k+1) &= F(k-1) + F(k) \qquad \text{als } k \geq 1 \end{aligned}$$

   In dit geval is de n uit Stelling 10.19 dus gelijk aan 0.

   (b) Laat zien dat voor $k \geq 1$, het $(k+1)^{\text{e}}$ Fibonacci-getal $F(k+1)$ inderdaad te schrijven is als functie van k en $gn(F(0), F(1), \ldots, F(k))$.

   (c) Bechrijf nu voor algemene parameters x_1, x_2 de functie $h(x_1, x_2)$ uit Stelling 10.19, die van toepassing is op de functie F.

`einde tentamen`