TENTAMEN FUNDAMENTELE INFORMATICA 3
Maandag 11 juni 2012, 10.00 - 13.00 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 stapelautomaten en Turing machines 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. `[15 pt]`

 (a) Beschrijf hoe je in het algemeen bij een context-vrije grammatica $G = (V, \Sigma, S, P)$ volgens de top-down methode een stapelautomaat $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$ construeert, zó dat $L(M) = L(G)$.
 De beschrijving mag in woorden of met een plaatje, als ze maar duidelijk is.

 Laat G een context-vrije grammatica zijn, met startsymbool S, terminaal alfabet $\Sigma = \{a, b, c\}$ en producties $S \to aTS \mid c \quad T \to aTb \mid b \mid \Lambda$

 (b) Construeer volgens de bottom-up methode (dus niet top-down) een stapelautomaat $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$, zó dat $L(M) = L(G)$.

 (c) Geef een afleidingsboom in G voor het woord $aabc$.

 (d) Geef een succesvolle berekening in de stapelautomaat van onderdeel (1b) voor het woord $aabc$. Dat wil zeggen: een berekening van $(q_0, aabc, Z_0)$ naar een eindconfiguratie.

---

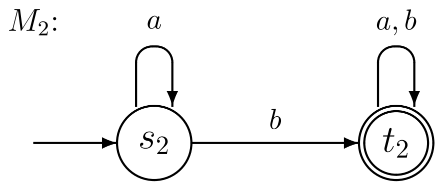
2. `[22 pt]`

 Laat $L_1 = \{x \in \{a, b\}^* \mid n_a(x) = n_b(x) + 1\}$.

 (a) Geef een stapelautomaat M_1 die L_1 accepteert (met eindtoestanden).
 Probeer er voor te zorgen dat M_1 deterministisch is, geen Λ-transities kent, en zo weinig mogelijk toestanden heeft. Lukt dit niet dan kun je nog wel een deel van de punten verdienen.

 (b) In het bewijs van Stelling 6.13 in het boek wordt een constructie beschreven, om een willekeurige stapelautomaat $M_1 = (Q_1, \Sigma, \Gamma, q_1, Z_0, A_1, \delta_1)$, met een willekeurige eindige automaat $M_2 = (Q_2, \Sigma, q_2, A_2, \delta_2)$ te combineren tot een stapelautomaat $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$, die $L(M_1) \cap L(M_2)$ accepteert. Beschrijf deze constructie in je eigen woorden en/of formules en/of plaatjes. Beschrijf met name hoe je toestanden combineert, wat de begin- en eindtoestanden worden, en hoe je transities combineert.

 (c) Pas de constructie uit onderdeel (2b) toe op de stapelautomaat M_1 van onderdeel (2a) en de eindige automaat M_2 hieronder (die overigens de taal $L_2 = \{x \in \{a, b\}^* \mid x \text{ bevat minstens één } b\}$ accepteert):



(d) Bij de constructie van onderdeel (2b) kan het voorkomen dat de resulterende stapelautomaat toestanden en/of transities bevat die nooit bereikt en/of gebruikt zullen worden. Welke toestanden en/of transities in je antwoord op onderdeel (2c) zullen nooit bereikt en/of gebruikt worden. Leg ook uit waarom die toestanden en/of transities nooit bereikt en/of gebruikt zullen worden.

---

3. [13 pt]

Construeer een Turing machine T die als invoer een woord $x \in \{a, b, c\}^*$ krijgt en de letters daarin 'in de goede volgorde zet'. Om precies te zijn: T berekent de functie f, gedefinieerd door

$$f(x) = a^{n_a(x)} b^{n_b(x)} c^{n_c(x)}$$

Dus eerst alle a's, dan alle b's, en dan alle c's.

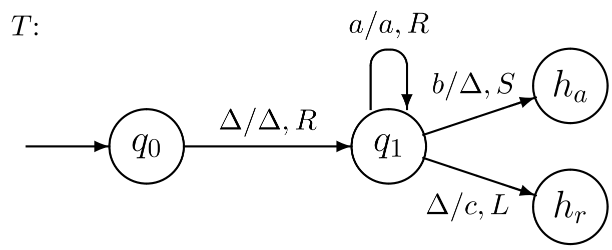
Indien T gebruik maakt van componenten, zul je ook die componenten moeten geven. Leg ook duidelijk uit hoe T werkt.

---

4. [14 pt]

De invoer van een universele Turing machine bestaat uit een codering van een Turing machine T, gevolgd door een codering van een invoerstring z voor T. Beide coderingen zijn strings over $\{0, 1\}$.

(a) Beschouw onderstaande Turing machine T (die overigens de taal L_2 uit opgave 2 accepteert):



Geef een codering $e(T)$ van T, waarbij je de functie e uit het boek gebruikt. Geef duidelijk aan, welk onderdeel van de Turing machine je waarmee codeert. Vermeld ook expliciet de gebruikte codering voor de individuele toestanden, tapeletters en richtingen.

(b) Aan welke eisen moet een string x over $\{0, 1\}$ voldoen om de codering van een Turing machine te kunnen zijn? Leg ook bij elke eis uit waarom x eraan moet voldoen.

---

5. [20 pt]

   In het bewijs van Stelling 8.14 in het boek wordt beschreven, hoe je bij een willekeurige Turing machine $T = (Q, \Sigma, \Gamma, q_0, \delta)$ 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 komt aan bod bij onderdeel (5a). De tweede soort is voor het simuleren van de Turing machine. De derde soort is voor het reconstrueren van de invoer van de Turing machine als die invoer geaccepteerd wordt.

   (a) Stel dat het invoer alphabet van de Turing machine $\Sigma = \{a, b\}$ is. Dan zijn de eerste soort producties:

   $$S \to S(\Delta\Delta) \mid T \qquad T \to T(aa) \mid T(bb) \mid q_0(\Delta\Delta)$$

   Leg uit wat 'het grote doel' is van deze producties, binnen het geheel van de constructie. Beantwoord hierbij tevens de volgende vragen

   - Waarom worden symbolen steeds dubbel gegenereerd (bijvoorbeeld aa) ?
   - Wat is het doel van de productie $S \to S(\Delta\Delta)$?
   - Wat is het doel van de productie $T \to q_0(\Delta\Delta)$?

   (b) Geef voor de Turing machine T van opgave 4 nu ook de producties van de tweede en derde soort in de bijbehorende unrestricted grammar G.

   (c) Geef een afleiding in de bij onderdeel (5b) resulterende unrestricted grammar, voor het woord aba. Geef bij elke stap in de afleiding duidelijk aan welk deel van de string je herschrijft.

6. [16 pt]

De stelling van Rice luidt als volgt:

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

(a) Voor welk van de volgende drie beslissingsproblemen is de stelling van Rice (direct) toepasbaar?

- AcceptsAllLengths: Gegeven een Turing machine T, accepteert T woorden van alle lengtes ≥ 0 ?
  Dat wil zeggen: is er voor elk geheel getal $N \geq 0$ een woord x van lengte N dat door T geaccepteerd wordt?
- AcceptsLengthN: Gegeven een Turing machine T, accepteert T een woord van lengte N ?
  Dat wil zeggen: is er een woord x van lengte N dat door T geaccepteerd wordt? Hierbij is N een geheel getal ≥ 0.
- AcceptsLength: Gegeven een Turing machine T en een geheel getal $N \geq 0$, accepteert T een woord van lengte N?
  Dat wil zeggen: is er een woord x van lengte N dat door T geaccepteerd wordt?

Motiveer je antwoorden. Dat wil zeggen: laat bij elk van de problemen zien waarom er wel of niet aan de voorwaarden van de stelling van Rice is voldaan. *Hint: bij minstens één van de problemen is de stelling van Rice niet (direct) toepasbaar.*

(b) Kies een beslissingsprobleem hierboven waarop de stelling van Rice niet direct toepasbaar is, en toon aan dat dat probleem toch niet beslisbaar is, met behulp van een reductie met een ander beslissingsprobleem.

Laat hierbij uiteraard zien dat aan alle eisen van een reductie voldaan is, en vergeet niet om de conclusie te trekken.

`einde tentamen`