Uitwerking tentamen Fundamentele Informatica 3, maandag 16 april 2018

11.16

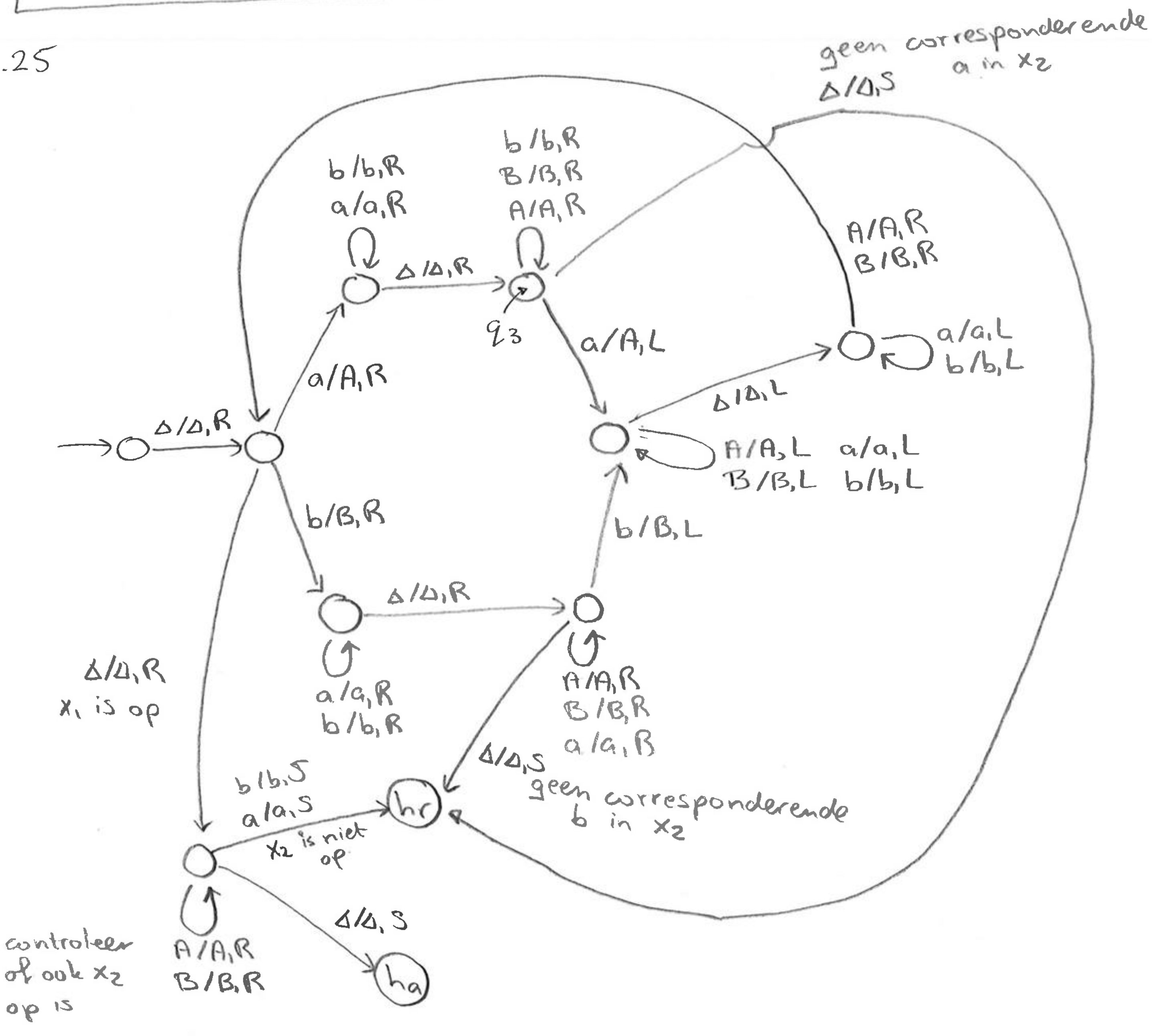
2(a)

T_1 loopt de letters van x_1 een voor een van links naar rechts af. Elke letter wordt gemarkeerd als hoofdletter, waarna T_1 in x_2 op zoek gaat naar dezelfde, ongemarkeerde letter. Als T_1 die vindt, wordt ook die gemarkeerd als hoofdletter. Wanneer T_1 alle letters van x_1 gehad (gemarkeerd) heeft, controleert hij of ook alle letters in x_2 gemarkeerd zijn. Zo ja, dan accepteert T_1; zo nee, dan verwerpt T_1.

T_1 verwerpt ook als hij geen corresponderende letter in x_2 kan vinden voor een zojuist gemarkeerde letter in x_1. Dan heeft x_1 namelijk meer voorkomens van een bepaalde letter dan x_2

Dan heeft x_2 namelijk meer letters dan x_1

11.25



11.33

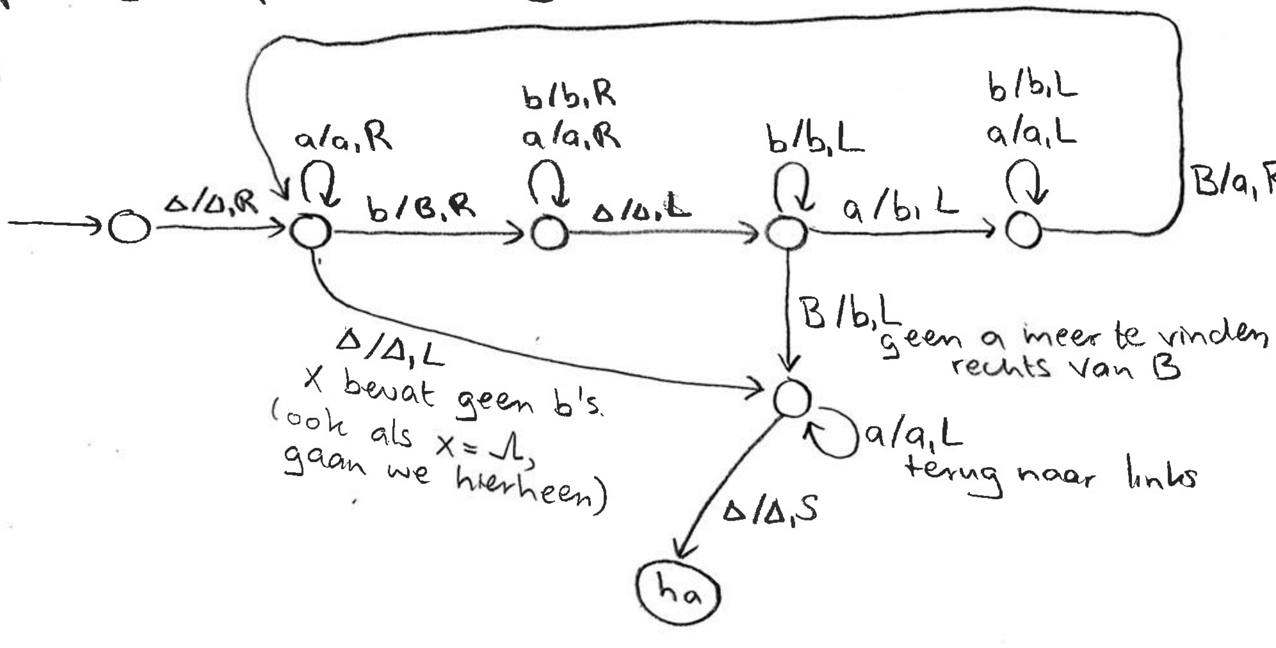
11.38

(b)

T_2 doorloopt zijn invoer x van links naar rechts. Als hij een b tegenkomt, markeert hij die als hoofdletter B, gaat naar het eind van de string, en zoekt dan van achteren naar voren naar de eerste (= achterste) a. Als die gevonden wordt, voordat de B wordt aangetroffen, wordt de a vervangen door b en de B door a. Zo zijn een a en een b verwisseld. Vervolgens gaat T_2 op zoek naar de volgende b vanaf links. Dit gaat zo door, totdat er van achter naar voren geen a meer gevonden wordt, die met de gemarkeerde b verwisseld kan worden, of totdat er bij van links naar rechts lopen (op zoek naar een b) een Δ wordt gevonden. Dat laatste kan alleen als x geen b's bevat.

In beide gevallen gaat T_1 terug naar links, om met de leeskop op de juiste plek te eindigen.

11.49



11.54

Uitwerking tentamen Fundamentele Informatica 3, maandag 16 april 2018

3(a) G bevat de volgende producties:

$S \to LTR$ genereer L (linkerkant) en R (rechterkant) en een T die A^nB^n gaat genereren.

$T \to ATB \mid \Lambda$ genereer dus A^nB^n voor $n \geq 0$

Elke A gaat nu door de B's heen naar rechts lopen, en laat voor elke B een 1 achter. Zo genereert elke A n 1'en (want er zijn n B's):

$AB \to B1A$

$A1 \to 1A$ loop voorbij een eerder gegenereerde 1

$AR \to R$ A heeft werk gedaan en verdwijnt.

Met n A's krijgen we zo $n * n = n^2$ 1'en.
Nu loopt L nog naar rechts, naar de R toe, en ruimt onderweg de B's op.

$LB \to L$ ruim B op

$L1 \to 1L$ L gaat naar rechts

$LR \to \Lambda$ klaar, L en R verdwijnen.

12.04

(b) Een afleiding in G voor het woord 1111:

$$\underline{S} \Rightarrow L\underline{T}R \Rightarrow LA\underline{T}BR \Rightarrow LAA\underline{T}BBR \Rightarrow LA\underline{AB}BR \Rightarrow$$
$$LAB1\underline{AB}R \Rightarrow LAB1B1\underline{AR} \Rightarrow L\underline{AB}1B1R \Rightarrow LB1\underline{A1}B1R \Rightarrow$$
$$LB11\underline{AB}1R \Rightarrow LB11B1\underline{A1}R \Rightarrow LB11B11\underline{AR} \Rightarrow \underline{LB}11B11R \Rightarrow$$
$$L11B11R \Rightarrow^2 11\underline{LB}11R \Rightarrow 11L11R \Rightarrow^2 1111\underline{LR} \Rightarrow 1111$$

12.08

4(b)

Als T de invoer $a_1a_2 \cdots a_n$ accepteert, dan ziet de string in de grammatica G er, direct na het simuleren van de berekening van T als volgt uit:

$$(\Delta\sigma_0)(a_1\sigma_1)(a_2\sigma_2)\cdots(a_n\sigma_n)(\Delta\sigma_{n+1})(\Delta\sigma_{n+2})\cdots(\Delta\sigma_{n+m})$$

waarbij m het aantal paren $(\Delta\Delta)$ achteraan de string is in $(\Delta\Delta)(a_1a_1)(a_2a_2)\cdots(a_na_n)\underbrace{(\Delta\Delta)(\Delta\Delta)\cdots(\Delta\Delta)}_{m \text{ keer } (\Delta\Delta)}$

Verder is elke $\sigma_i \in \Gamma \cup \{\Delta\}$. Samen vormen $\sigma_0\sigma_1\sigma_2 \ldots \sigma_n\sigma_{n+1}\sigma_{n+2} \ldots \sigma_{n+m}$ de tape inhoud bij accepteren.

Ten slotte staat ergens in de string, vóór een openingshaakje (, nog een symbool ha. De positie komt overeen met de positie van de leeskop bij accepteren.

12.16

4(a)

De letters a_i komen steeds dubbel voor in de string, omdat we * aan de ene kant de letters willen gebruiken om de berekening van T voor invoer $a_1a_2 \ldots a_n$ te simuleren. Bij dit simuleren kunnen de letters overschreven worden door andere symbolen

* aan de andere kant de invoer $a_1a_2 \ldots a_n$ willen reconstrueren als T de invoer accepteert. In dat geval is namelijk $a_1a_2 \ldots a_n \in L(T)$, en moet ook $a_1a_2 \ldots a_n \in L(G)$ worden.

De tweede kopie van a_i is bedoeld voor het simuleren. Die kan dus in de loop van het simuleren veranderen.

De eerste kopie van a_i is bedoeld voor het reconstrueren (indien nodig) van $a_1a_2 \ldots a_n$. Die blijft dus in de loop van het simuleren onveranderd.

12.24

(c) Allereerst hebben we producties om het symbool ha over de hele string te verspreiden:

$ha\,(\sigma_1\sigma_2) \to ha\,(\sigma_1\sigma_2)\,ha$ ha naar rechts kopiëren

$(\sigma_1\sigma_2)\,ha \to ha\,(\sigma_1\sigma_2)\,ha$ ha naar links kopiëren

voor $\sigma_1 \in \Sigma \cup \{\Delta\}$ $\sigma_2 \in \Gamma \cup \{\Delta\}$

Vervolgens hebben we producties om van de ontstane string alleen de letters $a_1a_2 \ldots a_n$ uit de eerste componenten van de paren $(\sigma_1\sigma_2)$ over te houden:

$ha\,(\sigma_1\sigma_2) \to \sigma_1$ voor $\sigma_1 \in \Sigma$ $\sigma_2 \in \Gamma \cup \{\Delta\}$

$ha\,(\Delta\,\sigma_2) \to \Lambda$ voor $\sigma_2 \in \Gamma \cup \{\Delta\}$.

12.31.

Uitwerking tentamen Fundamentele Informatica 3, maandag 16 april 2018

12.35

5 (a)

Allereerst kunnen we $\mathbb{N} \times \mathbb{N}$ als volgt beschrijven:

$$\begin{array}{ccccc}
(0,0) & (0,1) & (0,2) & (0,3) & \cdots \\
(1,0) & (1,1) & (1,2) & (1,3) & \cdots \\
(2,0) & (2,1) & (2,2) & (2,3) & \cdots \\
(3,0) & (3,1) & (3,2) & (3,3) & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{array}$$

Vervolgens gaan we een Cantor wandeling door deze beschrijving maken, waarbij we alle elementen precies één keer tegenkomen:

$$\begin{array}{ccccc}
(0,0) & (0,1) & (0,2) & (0,3) & \cdots \\
(1,0) & (1,1) & (1,2) & (1,3) & \cdots \\
(2,0) & (2,1) & (2,2) & (2,3) & \cdots \\
(3,0) & (3,1) & (3,2) & (3,3) & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{array}$$

De lijst wordt dan:

$(0,0)\ (0,1)\ (1,0)\ (0,2)\ (1,1)\ (2,0)\ (0,3)\ \cdots$

12.41

(b) (i)

Overaftelbaar, want elke functie f van $\mathbb{N}$ naar $\{0,1\}$ correspondeert met een deelverzameling S_f van $\mathbb{N}$, namelijk

$$S_f = \{ i \in \mathbb{N} \mid f(i) = 1 \}.$$

Er zijn overaftelbaar veel deelverzamelingen van $\mathbb{N}$, en dus ook overaftelbaar veel functies van $\mathbb{N}$ naar $\{0,1\}$.

12.45

(ii) Aftelbaar, want elke functie f van $\{0,1\}$ naar $\mathbb{N}$ correspondeert met een element $(f(0), f(1))$ van $\mathbb{N} \times \mathbb{N}$.

Volgens (a) is $\mathbb{N} \times \mathbb{N}$ aftelbaar, en dus is ook de verzameling functies van $\{0,1\}$ naar $\mathbb{N}$ aftelbaar.

12.48

6 (a)

Een instantie van AcceptsILengthI (AILI) bestaat uit een Turingmachine T en een geheel getal $i \geq 0$.

12.50
(b)

Ja-instantie: $(T_1, 0)$ waarbij T_1 is: $\rightarrow \bigcirc \xrightarrow{\Delta/\Delta,S} (h_r)$

T_1 accepteert geen enkele string, en dus in het bijzonder 0 strings van lengte 0.

Nee-instantie: $(T_2, 0)$ waarbij T_2 is $\rightarrow \bigcirc \xrightarrow{\Delta/\Delta,S} (h_a)$

T_2 accepteert alle invoerstrings, en dus in het bijzonder ook de lege string Λ van lengte 0. T_2 accepteert dus niet 0 strings van lengte 0 (maar 1 string van lengte 0).

12.54
(c)

We moeten aantonen:

$$\begin{array}{lccc} & \text{Accepts-ab} & \leq & \text{AILI} \\ \text{instanties} & T_1 & & (T_2, i) \end{array}$$

We moeten een willekeurige instantie T_1 van Accepts-ab ombouwen naar een instantie (T_2, i) van AILI, zó dat T_1 is ja-instantie van Accepts-ab, dan en slechts dan als (T_2, i) is ja-instantie van AILI.

Omdat $|ab| = 2$, ligt het voor de hand om $i = 2$ te nemen.
Er moet dus gelden: T_1 accepteert ab, d.e.s.d. als T_2 accepteert precies 2 strings van lengte 2.

We geven T_2 hetzelfde invoeralfabet Σ als T_1.

T_2 controleert
- of zijn invoer de string aa is,
- zo ja, accepteer
- zo nee, controleer of zijn invoerstring ab is,
  - zo ja, simuleer T_1
  - zo nee, verwerp.

Het is duidelijk dat (T_2, i) op algoritmische wijze uit T_1 is te construeren.
Verder geldt:

T_1 is ja-instantie van Accepts-ab $\Leftrightarrow$ T_1 accepteert ab $\Leftrightarrow$ T_2 accepteert ab $\Leftrightarrow$ $L(T_2) = \{aa, ab\}$, want alle andere invoerstrings worden verworpen $\Leftrightarrow$ T_2 accepteert precies 2 strings van lengte 2 $\Leftrightarrow$ (T_2, i) is ja-instantie van AILI.

Inderdaad hebben we dus een reductie van Accepts-ab naar AILI: Accepts-ab $\leq$ AILI.
Omdat gegeven is dat Accepts-ab niet beslisbaar is, is ook AILI niet beslisbaar.

± 13.10

14.15

1)

Bij een Turingmachine is δ een functie van $Q \times (\Gamma \cup \{\Delta\})$ naar $(Q \cup \{h_a, h_r\}) \times (\Gamma \cup \{\Delta\}) \times \{R, L, S\}$

14.17.