TENTAMEN

Basismodellen in de Informatica

vakcode: 211180
datum: 26 juni 2008
tijd: 9:00–12:30 uur

Algemeen

- Bij dit tentamen mag gebruik worden gemaakt van het boek van Sudkamp, van de handleiding van Basismodellen in de Informatica, en van de handouts van de colleges.
- Dit tentamen bestaat uit 6 opgaven, waarvoor in het totaal 90 punten behaald kunnen worden. Het minimale aantal punten per opgave bedraagt 0. Het cijfer is het aantal punten gedeeld door 10, afgerond tot een geheel getal, plus 1.

Opgave 1 (20 punten)

Gegeven is de volgende NFA-λ:

$$\begin{aligned} M &= \big(\,\{q_0, q_1, q_2, q_3, q_4\}\ ,\ \{a, b\}\ ,\ \delta\ ,\ q_0\ ,\ \{q_1\}\,\big),\ \text{waarbij} \\ \delta &= \{(q_0, a, q_1), (q_1, \lambda, q_2), (q_2, a, q_4), (q_2, b, q_3), (q_3, \lambda, q_1), (q_1, a, q_3), (q_4, b, q_3)\} \end{aligned}$$

a. Teken het toestandsdiagram van M.

b. Beschrijf $\mathcal{L}(M)$, de taal die door M geaccepteerd wordt, door middel van een reguliere expressie. Gebruik hierbij het standaard algoritme gebaseerd op “expression graphs”.

Opgave 2 (20 punten)

Gegeven de reguliere expressie $E = a^*(a \cup b)^*b$.

a. Beschrijf een stappenplan om een reguliere expressie om te zetten in een equivalente DFA met een minimaal aantal toestanden.

b. Voer deze stappen uit om E om te zetten in een minimale DFA. Geef alle tussenresultaten, en maak gebruik van de standaardalgoritmes.

Opgave 3 (15 punten)

Gegeven de volgende contextvrije grammatica G, met $V = \{S, P, Q\}$, $\Sigma = \{a, b\}$ en startsymbool S:

$$\begin{aligned} S &\rightarrow aP \\ P &\rightarrow aP \mid bQ \mid b \\ Q &\rightarrow bP \mid a \end{aligned}$$

a. G is regulier. Waarom?

b. Geef de bijbehorende eindige automaat

c. Geef in deze eindige automaat de berekening van het woorde $aabbb$

d. G is in Greibach normaalvorm. Waarom?

e. Geef de bijbehorende stapelautomaat

f. Geef in deze stapelautomaat de berekening van het woord $aabbb$

Opgave 4 (20 punten)

Gegeven de volgende contextvrije grammatica, met $V = \{E, F\}$, $\Sigma = \{1, +, (,)\}$ en startsymbool E:

$$\begin{array}{l} E \to F \mid F{+}F \\ F \to 1 \mid (E) \end{array}$$

Deze grammatica genereert de taal van (zeer eenvoudige) *expressies*; bijvoorbeeld horen $(1 + 1) + 1$ en $1 + (1 + (1 + (1)))$ tot de taal. Als we voor de symbolen letters gebruiken, zoals gebruikelijk in het boek, bijvoorbeeld door de substitutie

$$1 \leftrightarrow e \qquad + \leftrightarrow p \qquad (\leftrightarrow a \qquad) \leftrightarrow b$$

dan verandert het alfabet in $\{a, b, e, p\}$, de grammatica in

$$\begin{array}{l} E \to F \mid FpF \\ F \to e \mid aEb \end{array}$$

en de bovengenoemde voorbeeldwoorden worden $aepebpe$ en $epaepaepaebbb$. De opgaven hieronder gaan over deze, laatste, grammatica, die we G noemen.

a. Geef in G een linkerafleiding, een rechterafleiding en een afleidingsboom voor $aepebpe$.

b. Zet G om naar Chomsky normaalvorm, volgens de in het boek beschreven methode. Licht de stappen die u neemt toe!

c. Zet voor de verkregen CNF-grammatica een CYK-matrix op die het woord $aepebpe$ ontleedt.

Opgave 5 (10 punten)

U wordt de nieuwe directeur van Softwarehouse BadIQ en moet saneren. Geef aan waarom de volgende twee projecten maar beter per direct gestopt kunnen worden:

a. Terminator: Om in een klap af te zijn van software die hangt, worden Java programma's automatisch omgezet naar Turingmachines. Die Turingmachines worden vervolgens automatisch geanalyseerd op terminatie.

b. Determinator: Om context-vrije grammatica's (CFG) te simuleren, wordt een willekeurige CFG eerst omgezet in een stapelautomaat (PDA), die vervolgens gedeterminiseerd wordt. Uit een deterministische PDA is immers eenvoudig code te genereren.

Opgave 6 (15 punten)

Beschouw de volgende twee niet-reguliere talen:

$$\begin{array}{lcl} \mathcal{L}_1 & := & \{a^i b^j \mid i \le j\} \\ \mathcal{L}_2 & := & \{w \in \{a,b\}^* \mid \#_a(w) \le \#_b(w)\} \end{array}$$

waarbij $\#_a(w)$ het aantal voorkomens van a in w betekent.

a. Bewijs met behulp van de pompstelling dat $\mathcal{L}_1$ niet regulier is.

b. Bewijs met behulp van de geslotenheidseigenschappen van reguliere talen dat dan ook $\mathcal{L}_2$ niet regulier is.