

HERTENTAMEN FUNDAMENTELE INFORMATICA 3

Maandag 8 augustus 2011, 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 PDA's en Turing machines door middel van hun toestandsdiagram.

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.

Opgave 1 [9 pt]

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

- Construeer volgens de top-down methode een PDA M , zó dat $L(M) = L(G)$.
- Geef een afleidingsboom in G voor het woord abc .
- Geef een succesvolle berekening in de PDA van onderdeel (a) voor het woord abc . Dat wil zeggen: als q_0 de initiële toestand is en Z_0 het initiële stapelsymbool, een berekening van (q_0, abc, Z_0) naar een eindconfiguratie.

Opgave 2 [25 pt]

Laat $L_1 = \{a^i b^j c^k \mid i, j, k \geq 0 \text{ en } i \neq j\}$, $L_2 = \{a^i b^j c^k \mid i, j, k \geq 0 \text{ en } i \neq k\}$, en $L_3 = \{a^i b^j c^k \mid i, j, k \geq 0 \text{ en } (i \neq j \text{ of } i \neq k)\}$.

- Geef een deterministische PDA M_1 die L_1 accepteert (met eindtoestanden).
- Geef een deterministische PDA M_2 die L_2 accepteert (met eindtoestanden).
- Geef een PDA M_3 die L_3 accepteert (met eindtoestanden).
- Toon aan dat $L_3 \notin \text{DCFL}$. Je mag hierbij gebruik maken van resultaten als
 - als $L_1, L_2 \in \text{CFL}$, dan is $L_1 \cup L_2 \in \text{CFL}$,
 - als $L \in \text{CFL}$ en R is regulier, dan is $L \cap R \in \text{CFL}$,
 - als $L \in \text{DCFL}$, dan is $L' \in \text{DCFL}$,
 - het pomplemma voor context-vrije talen.
- Toon aan dat DCFL niet gesloten is onder vereniging.

Opgave 3 [21 pt]

In deze opgave mag je gebruik maken van de componenten InsertSymbol (σ), DeleteSymbol, Copy en Equal (en niet van andere componenten, tenzij je ze zelf uitwerkt).

InsertSymbol (σ) verandert de tape-inhoud van een Turing machine van $y\underline{z}$ in $y\underline{\sigma}z$ (waarbij z geen Δ bevat) en DeleteSymbol doet precies het omgekeerde. Copy verandert de inhoud van de tape van $\underline{\Delta}x$ in $\underline{\Delta}x\underline{\Delta}x$ (waarbij x geen Δ bevat). Equal accepteert twee strings x en y over een bepaald alfabet (zonder Δ) precies dan als $x = y$.

- (a) Construeer een Turing machine T_1 die als invoer twee unaire getallen x en y heeft, met $x \geq 0$ en $y \geq 1$, en berekent of x deelbaar is door y . Om precies te zijn, T_1 berekent de volgende partiële functie f :

$$f(x, y) = \begin{cases} (x, y, 1) & \text{als } x \text{ deelbaar is door } y \\ (x, y, 0) & \text{als } x \text{ niet deelbaar is door } y \end{cases}$$

De uitvoer bestaat dus uit de twee getallen x en y uit de invoer, én de letter 0 of 1 (afhankelijk van het antwoord). Op de tape van de Turing machine worden deze drie onderdelen uiteraard gescheiden door Δ 's.

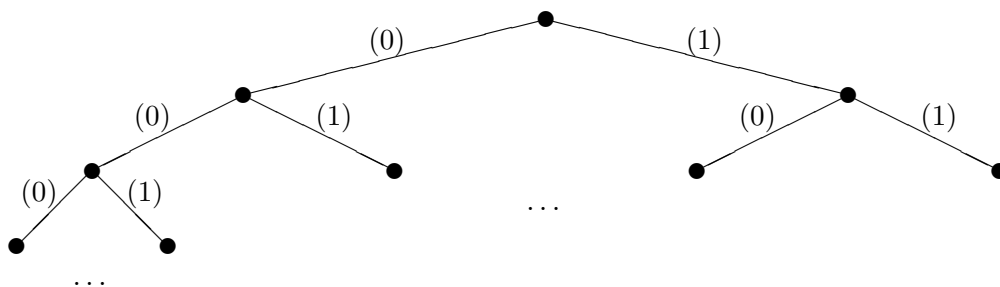
Leg ook duidelijk uit hoe T_1 werkt.

- (b) Construeer een Turing machine T_2 die precies alle priemgetallen (2, 3, 5, 7, ...) accepteert. De invoer van T_2 is een unair, niet-negatief getal x .

Leg ook duidelijk uit hoe T_2 werkt.

Opgave 4 [20 pt]

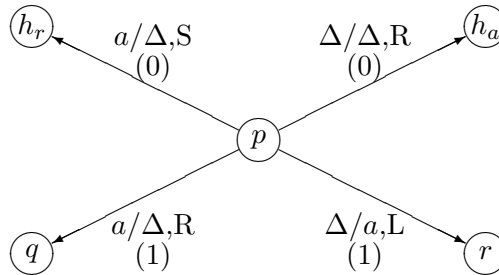
Volgens Stelling 9.2 uit het boek kan iedere niet-deterministische Turing machine T_1 (voor het accepteren van een taal) gesimuleerd worden door een 3-tapes deterministische Turing machine T_2 . In het bewijs wordt er voor het gemak vanuit gegaan dat T_1 op ieder moment in zijn berekening (in iedere toestand met elk symbool op de tape) moet kiezen uit twee mogelijke stappen, aangegeven met 0 of 1. Hij heeft dus bij een gegeven invoer een 'berekeningsboom' (computation tree) van de volgende vorm:



- (a) De deterministische TM T_2 wandelt als het ware door deze berekeningsboom op zoek naar een configuratie met toestand h_a . Moet deze zoektocht in de vorm van een 'Breadth First Search' zijn, of in de vorm van een 'Depth First Search', of maakt het niet uit? Motiveer je antwoord.
- (b) Bij het simuleren van T_1 heeft T_2 op zijn tape 1 altijd de invoer x van T_1 staan. Op zijn tape 2 heeft T_2 een bitstring staan, die aangeeft welke keuzes in T_1 hij achtereenvolgens simuleert. Op tape 3 van T_2 komt dan de tape-inhoud van T_1 te staan die het gevolg is van deze rij keuzes, voorafgegaan door een speciaal symbool $\#$. Waar is dit speciale symbool $\#$ voor (wanneer maak je er gebruik van), en waarom hoeft er *achteraan* tape 3 van T_2 geen $\#$ te staan?

- (c) Voor het simuleren van een rij keuzes in T_1 gebruikt T_2 een component *Execute* die altijd eindigt in h_a . Deze component laat een * onder de leeskop van tape 3 achter, dan en slechts dan als de rij keuzes T_1 naar diens h_a zouden leiden.

Stel voor het gemak dat a en Δ de enige symbolen zijn die op de tape van T_1 voor kunnen komen, en dat vanuit een bepaalde toestand p in T_1 de volgende transities mogelijk zijn:



Geef het hiermee corresponderende stukje van de component *Execute* van T_2 .

Opgave 5 [15 pt]

- (a) Geef een unrestricted grammar G_1 voor de taal

$$L_1 = \{a^i b^j \mid (i, j \geq 0) \text{ en } (i \text{ is een veelvoud van } j)\}.$$

Leg ook uit wat de bedoeling is van de verschillende niet-terminalen en producties in G_1 .

- (b) Leg uit waarom er geen context-gevoelige grammatica voor L_1 bestaat.
 (c) Geef een context-gevoelige grammatica G_2 voor de taal

$$L_2 = \{aa^i b^j \mid (i, j \geq 0) \text{ en } (i \text{ is een veelvoud van } j)\}$$

(dus elk woord in L_2 bevat $i + 1$ a 's). Leg ook uit wat de bedoeling is van de verschillende niet-terminalen en producties in G_2 .

Opgave 6 [10 pt]

- (a) Formuleer de stelling van Rice.
 (b) Gebruik de stelling van Rice om aan te tonen dat het volgende beslissingsprobleem onbeslisbaar is:

Accepts-Even:

Gegeven een Turing machine T . Accepteert T een eindig, even aantal woorden?

Vergeet niet om aan te tonen dat aan de voorwaarden voor de stelling van Rice is voldaan.

- (c) Geef een voorbeeld van een beslissingsprobleem dat over Turing machines gaat, en waar de stelling van Rice niet (direct) op kan worden toegepast. Leg ook uit waarom de stelling van Rice niet (direct) op dat probleem kan worden toegepast.