

Opdracht 2

Topics on Parsing and Formal Languages - fall 2010

Rick van der Zwet
<hvdzwet@liacs.nl>

20 december 2010

Samenvatting

Dit schrijven zal uitwerkingen van opgaven behandelen uit het boek [JS2009] gebruikt bij het college. In deze opdracht zullen zeven opgaven (9, 18, 24, 26, 32, 37, 41) van hoofdstuk 4 behandeld worden.

1 Opgave 4.9

Laat $L \subseteq \Sigma^*$ de taal zijn en definieer $\sigma(L) = \{x \in \Sigma^* : xy \in L \text{ voor alle } y \in \Sigma^*\}$. Als L context-vrij is dan is $\sigma(L)$ ook context-vrij. Als naar de taal L gekeken wordt moet de taal uit twee delen ($A \cdot B$) opgebouwd zijn wil $xy \in L$ accepteren. A zal x beschrijven. B moet een reguliere taal zijn van de vorm Σ^* .

Een *CFL* waarbij het *RL* suffix afgehaald wordt zal wederom een *CFL* zijn (zie Opgave 4.18) en daarom is $\sigma(L)$ ook weer context-vrij.

2 Opgave 4.18

Als L een context-vrije taal is en R een reguliere taal dan is de rechter quotient $L/R = \{x \in \Sigma^* : \exists y \in R \text{ zodat } xy \in L\}$ ook een context-vrije taal. xy

zijn dus de woorden in L waarbij geldt dat het ‘staartje’ y wat eraf gehaald wordt in de reguliere taal R zit.

Om dit te bewijzen gaan we naar de PDA van L kijken. Hierbij kan R in de omgekeerde volgorde door de taal gehaald worden om zo de eindtoestand naar ‘voren’ te halen. Dit levert een kortere PDA op.

3 Opgave 4.24

Als $s^{[-1]}(L) := \{x : s(x) \cap L \neq \emptyset\}$ en s verbindt letters met reguliere talen en L is context-vrij. Dan is $s^{[-1]}(L)$ niet altijd CFL. $s^{[-1]}(L)$ kan een verzameling kan zijn van de talen waarbij de lengte van de letters even moet zijn, welke geen context-vrije verzameling is.

Hetzelfde probleem doet voor als s letters aan context-vrije talen verbindt.

4 Opgave 4.26

Gegeven een PDA M kan je altijd een M' maken welke de eigenschap heeft dat geen enkele stap het huidige symbool op de stapel vervangt (geen transitie van de vorm $(p, \gamma Y) \in \delta(q, a, X)$ waarbij $X \neq Y$). Dus dat alle stappen, of een symbool van de stapel afhalen, of een string van symbolen op de stapel zetten. Dit is te doen door de huidige substitutie’ transitie te vervangen door een tweetal (gekoppelde) transitie. Waarbij de eerste het symbool eraf haalt en de tweede de gewenste symbolen er weer op zet.

De bovengenoemde methode zorgt voor maar één mogelijk pad. Het zal de $DPDA$ structuur (indien present) dus niet aantasten.

5 Opgave 4.32

Als ψ een Parikh map is, dan zijn er lange Engelse woorden over een bepaald alfabet voor specifieke eigenschappen. Voorbeelden zijn te vinden in de on-

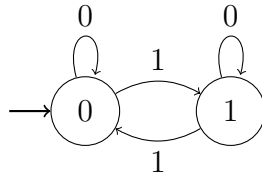
derstaande tabel.

eigenschap	woorden	alfabet
$\psi(w)$ alle items gelijk aan 1	dutchwomen	$\{d, u, t, c, h, w, o, m, e, n\}$
$\psi(w)$ alle items gelijk aan 2	tomtom	$\{t, o, m\}$
$\psi(w)$ alle items gelijk aan 3	sestettes	$\{s, e, t\}$
$\psi(w)$ alle items ≥ 2	unprosperousness	$\{e, n, o, p, r, s, u\}$
$\psi(w)$ (1, 2, 2, 3, 3, 3)	chinchinchee	$\{r, i, n, c, h, e\}$

6 Opgave 4.37

a) $\text{perm}(x)$ is de set van alle permutaties van de letters van x . Voor talen is $\text{perm}(L) := \bigcup_{x \in L} \text{perm}(x)$. $\text{perm}(L)$ is niet context-vrij als L de reguliere taal is $\{x \in a^*b^*c^* : |x|_a, |x|_b, |x|_c \text{ is even}\}$, omdat dit onder andere $\{a^n b^n c^n : n = 2 * i, i \geq 0\}$ bevat, welke niet context-vrij [JS2009][pg. 108] is.

b) Als L echter een reguliere taal over een alfabet van twee symbolen is, dan zal $\text{perm}(L)$ context-vrij zijn. Welke de reguliere taal, maximaal één complexe relatie kan aangaan en wel namelijk tussen de twee symbolen in. Alle andere gevallen (enkel relatie één symbool) is er maar een enkelvoudige afhankelijk gecodeerd. Pak bijvoorbeeld de ingewikkelde $\{x \in a^*b^* : |x|_a, |x|_b \text{ even zijn}\}$. De permutatie van bijvoorbeeld $aabb$ zullen zijn $aabb, abba, abab, baba, baab, bbaa$, waarbij in het context-vrije domein een analoge taal ontstaan, waarbij het aantal a ‘even’ moet zijn om te accepteren en het aantal b gelijk aan het aantal ‘a’.



Figuur 1: Automaton generating the Thue-Morse sequence as the sequence generated by the following DFA, where one inputs n expressed in base 2

7 Opgave 4.41

Om te bewijzen dat, de taal van alle woorden over de $\{0, 1\}$ welke geen prefixen zijn van een Thue-Morse woord context-vrij is. Thue-Morse woorden zijn overlap-vrij en kunnen door een automaat gegenereert worden, zie hiervoor Figuur 1¹. Een paar eigenschappen het Thue-Morse woord zijn het feit dat er geen overlap in voorkomt en ook geen ‘blokken’ in de vorm van XXX waarbij $X \in \Sigma^*$.

Woorden met een overlap bevatten de serie $cxxc$ waarbij c een letter is en x een woord, de overlap is dan cx . De taal van de woorden ziet er dus als volgt uit: $\{0, 1\}^*1x1x\{0, 1\}^*$ of $\{0, 1\}^*0x0x\{0, 1\}^*$ waarbij $x := \{0, 1\}^*\{0, 1\}^*$. Dit is niet een context-vrije taal. Welke het *pumping-lemma* hier faalt. Er moet dus gekeken worden naar een meer generiek geval. Dit kunnen we door verschillende talen te generen en gebruik te maken van het gegeven feit dat de klasse van de context-vrije-talen gesloten zijn onder een eindige verzameling. Deze talen zijn te vinden in [BS2009][pg. 89 Theorem 2.24].

Referenties

[JS2009] Jeffrey Shallit, *A second course in formal languages and automata theory*, Cambridge University Press, 2009.

[BS2009] J. Berstel, *Combinatorics on words: Christoffel words and repetitions in words*, American Mathematical Society, 2009.

¹<http://www.cs.uwaterloo.ca/~shallit/Talks/canad4.pdf>