

Opdracht 3

Topics on Parsing and Formal Languages - fall 2010

Rick van der Zwet
<hvdzwet@liacs.nl>

16 december 2010

Samenvatting

Dit schrijven zal uitwerkingen van opgaven behandelen uit het boek [JS2009] gebruikt bij het college. In deze opdracht zullen vijf opgaven (1, 5, 6, 8, 14) van hoofdstuk 5 behandeld worden.

1 Opgave 5.1

De grammatica G bestaat uit de volgende producties:

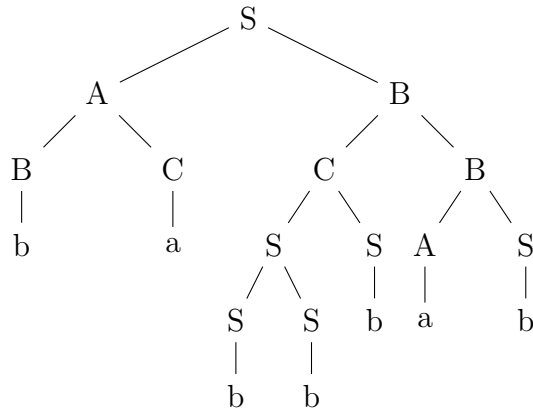
$$\begin{aligned} S &\rightarrow AB \mid b \\ A &\rightarrow BC \mid a \\ B &\rightarrow AS \mid CB \mid b \\ C &\rightarrow SS \mid a \end{aligned}$$

Gebruikmakend van het CYK algoritme gaan we aantonen dat $x = babbab \in L(G)$ zit. De ondersteunende tabel is van de grootte 6×6 omdat dit de lengte van het woord x is. In tabel 1 staat¹ cel i, j voor welke transities er gevolgt moet worden om het subwoord $x[i..j]$ te vormen. Omdat de start transitie S in 1,6 staat zit het woord x in $L(G)$. De ontleedboom is te zien in figuur 1.

¹Om de \LaTeX tabel automatisch te genereren vanuit een woord en een CFG grammatica heb ik <http://rickvanderzwet.nl/svn/personal/liacs/TPFL2010/assignment3/cyk.py> geschreven, vanwege de fouten ik met handwerk maakte.

i \ j	1	2	3	4	5	6	7
1	S B	A: (B,C,1)	C: (S,S,1) S: (A,B,2) B: (A,S,2)	A: (B,C,1) B: (C,B,3) C: (S,S,3)	C: (S,S,1) S: (A,B,2),(A,B,4) A: (B,C,3) B: (A,S,4),(C,B,4)	A: (B,C,5)	A: (B,C,1),(B,C,3),(B,C,4) C: (S,S,1),(S,S,5) S: (A,B,2),(A,B,4) ... (A,B,5),(A,B,6) B: (A,S,2),(C,B,3) ... (A,S,4),(C,B,4),(A,S,5) ... (C,B,5),(A,S,6)
2		A C	S: (A,B,2) B: (A,S,2),(C,B,2)	C: (S,S,3)	S: (A,B,2) B: (C,B,2),(C,B,4) A: (B,C,3)	A: (B,C,5)	S: (A,B,2),(A,B,5),(A,B,6) B: (A,S,2),(C,B,2), ... (C,B,4),(A,S,5),(A,S,6) A: (B,C,3) C: (S,S,5)
3			S B	C: (S,S,3)	A: (B,C,3) B: (C,B,4)	A: (B,C,5)	A: (B,C,3) B: (C,B,4),(A,S,5),(A,S,6) S: (A,B,5),(A,B,6)
4				S B	C: (S,S,4)	\emptyset	A: (B,C,4) C: (S,S,4) B: (C,B,5)
5					S B	A: (B,C,5)	C: (S,S,5) S: (A,B,6) B: (A,S,6)
6						A C	S: (A,B,6) B: (A,S,6),(C,B,6)
7							S B

Tabel 1: $CYK(L(G), a)$. Algorithme beschreiben in [JS2009][pg. 142]



Figuur 1: Ontleedboom voor het woord *babbbab*

2 Opgave 5.5

Om een LL(1) grammatica te generen voor alle woorden in $\{w \in \{a, b\}^* : |w|_a = |w|_b\}$ is:

$$S \rightarrow aSbS \mid bSaS \mid \emptyset$$

Om aan te tonen dat de grammatica correct is, is het eerst belangrijk om te zien dat elke keer dat een *a* genereerd wordt er ook automatisch een *b* genereerd wordt. Deze dus altijd gelijk zijn. Om te laten zien dat deze grammatica *alle* woorden in de taal bevat is bewijzen we met inductie naar lengte van het woord. Als $|w| = 0$ dan is $w = \emptyset$, deze wordt door de taal herkend.

Neem alle woorden tot lengte $2N$ en een gelijk aantal *a* en *b* afleidbaar zijn van *S*. Neem nu de string w' met een gelijk aantal *a* en *b*, een lengte van $2(N + 1)$ en *a* als begin symbool. In het slechte geval is $2N + 2$ weer nieuw woord doordat je altijd een extra *T* kan ontwikkelen en die daarna laat termineren. Bijvoorbeeld $abab \rightarrow abaSbS \rightarrow ababSaSbS \rightarrow ababab$.

In de betere gevallen bestaat er een $2 \leq j \leq 2N + 2$ zodanig dat *j* aangeeft dat $w[1..j]$ een gelijk aantal *a* en *b* heeft, zodanig dat de vorm van $w' = aw_1bw_2$. Met inductie kunnen we bewijzen dat w_1 en w_2 gemaakt kunnen worden van *S*, wat volgt dat w' ook van *S* gemaakt kan worden.

LL(1) eigenschap wordt bereikt, door naar de *FIRST* te kijken, welke respectievelijk $\{a\}$, $\{b\}$, $\{\emptyset\}$ zijn. De $FOLLOW(S) = \{a, b, \emptyset\}$. Deze twee gegevens samen maken dat de LL(1) bereikt wordt, welke ook te zijn is in tabel 2.

$y \backslash x$	a	b	\$
S	$S \rightarrow aSbS$	$S \rightarrow bSaS$	$S \rightarrow \emptyset$
a	pop		
b		pop	
#			accept

Tabel 2: Ontleedtabel for Opdracht 5.5

3 Opgave 5.6

4 Opgave 5.8

5 Opgave 5.14

Referenties

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