Opdracht 3

Topics on Parsing and Formal Languages - fall 2010

Rick van der Zwet
<hvdzwet@liacs.nl>

17 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:

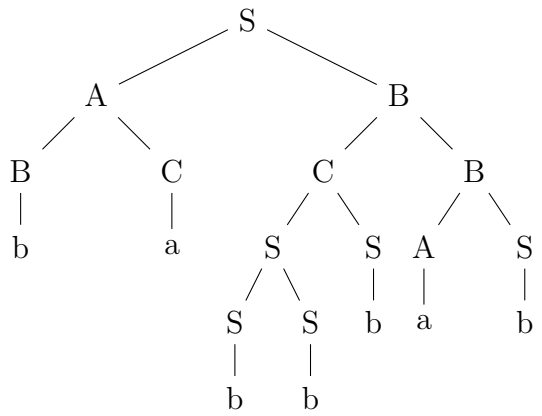
$$S \to AB \mid b$$
$$A \to BC \mid a$$
$$B \to AS \mid CB \mid b$$
$$C \to SS \mid a$$

Gebruikmakend van het CYK algoritme gaan we aantonen dat $x = babbbab \in L(G)$ zit. De ondersteunende tabel is van de grootte 6×6 omdat dit de lengte van het woord x is. In tabel 1 staat[1] 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.

[1] Om de LaTeX tabel automatisch te gegenereren 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: $CYK(L(G), a)$. Algoritme beschreven 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:

$$S \to aSbS \mid bSaS \mid \emptyset$$

Om aan te tonen dat de grammatica correct is, is het eerst belangrijk om te zien dat elke keer dat een a genereerd word 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 herkent.

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 terminereren. Bijvoorbeeld $abab \to abaSbS \to ababSaSbS \to abababab$.

In de betere gevallen bestaat er een $2 \leq j \leq 2N + 2$ zodaning 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 respectivelijk $\{a\}, \{b\}, \{\emptyset\}$ 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 \x	a	b	$
S	$S \to aSbS$	$S \to bSaS$	$S \to \emptyset$
a	pop		
b		pop	
#			accept

Tabel 2: Ontleedtabel for Opdracht 5.5

3 Opgave 5.6

Laat G een *CFG* zijn zonder 'nutteloze symbolen. Als G een LL(1) grammatica is dan en slechts als, voor willekeurig twee ongelijke producties van de vorm $X \to \alpha$ en $X \to \beta$, dan is het volgende geldig, als $x, y \in FOLLOW(X)$ dan $FIRST(\alpha x) \cap FIRST(\beta y) = \emptyset$. De symbolen x en y hoeven niet unique te zijn.

$\Rightarrow$ Als G LL(1) dan moet met behulp van 'e'e symbool de juiste transitie gekozen worden. Als we bijvoorbeeld in toestand X zijn dan moet onze volgende stap (de $FIRST$) unique [2] zijn onafhankelijk wat hier achter wordt gezet.

$\Leftarrow$ Als voor alle willekeurige x en y beiden in $FOLLOW(X)$ beiden geen gemeenschapelijke start symbool hebben ($FIRST(\alpha x) \cap FIRST(\beta y) = \emptyset$), betekend dat de transities $X \to \alpha$ en $X \to \beta$ door elkaar te onderscheiden zijn door het eerste symbool. Omdat dit geldt voor alle transities is de taal dus herkenbaar door enkel het eerste symbool in de transities en dus LL(1). Als ze wel een gemeenschapelijk start symbool hebben zijn er minimaal 2 symbolen nodig om de taal te herkennen en is deze niet LL(1).

[2] De eigenschap wordt afgedwongen door Stelling 5.3.4 en de bovenstaande definitie op [JS2009][pg. 157]

4 Opgave 5.10

Een voorbeeld van een grammatica welke wel $LL(k+1)$ is maar niet $LL(k)$, is gegeven in context van het aantonen dat voor elke $k > 0$ de $L(k+1)$ talen niet de $L(k)$ talen zijn. [STOC69][pg. 174] en is de grammatica van de taal $\{a^n(b^k d|b|cc)^n : n \geq 1\}$:

$$S \to aSA$$
$$S \to aA$$
$$A \to cc$$
$$A \to bB$$
$$A \to \epsilon$$
$$B \to b^{k-1}d$$

5 Opgave 5.14

Laat G een LR(0) grammatica zijn met $A \to \alpha\bullet, \alpha \neq \epsilon$, welke geldig is voor een levensvatbare prefix γ er geen enkel ander *item* kan geldig zijn voor γ. Als er een *item* van de vorm $A \to \alpha B$ is levert dit een shift-reduce error op. Als er een *item* van de vorm $A \to \beta\bullet : \alpha \neq \beta$ zal het betekenen dat er een *item* 'vooraf' aan gegaan $A \to \bullet\alpha$ en $A \to \bullet\beta$, wat niet mogelijk is omdat enkel de eerste geldig kan zijn per definitie.

Referenties

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

[STOC69] Rosenkrantz, D. J. and Stearns, R. E., Properties of deterministic top down grammars, Proceedings of the first annual ACM symposium on Theory of computing, STOC '69, Marina del Rey, California, United States, 165–180