

**Werkcollege Compilerconstructie**  
**Dinsdag 24 september 2013**

1. Beschouw de context-vrije grammatica  $G$  met startvariabele  $stmt$  en de volgende producties:

$$\begin{aligned} stmt &\rightarrow matchedstmt \\ &\quad | \quad openstmt \\ matchedstmt &\rightarrow \mathbf{if\ expr\ then\ } matchedstmt \mathbf{\ else\ } matchedstmt \\ &\quad | \quad \mathbf{other} \\ openstmt &\rightarrow \mathbf{if\ expr\ then\ } stmt \\ &\quad | \quad \mathbf{if\ expr\ then\ } matchedstmt \mathbf{\ else\ } openstmt \end{aligned}$$

- (a) Is  $G$  een LL(1)-grammatica? Motiveer je antwoord.
- (b) Construeer vanuit  $G$  een nieuwe context-vrije grammatica  $G'$  met  $L(G') = L(G)$ , door
- (indien van toepassing) links-recursie te elimineren, en
  - (indien van toepassing) links-factorisatie toe te passen.
- Leg uit hoe je  $G'$  uit  $G$  verkrijgt.
- (c) Bepaal voor elke variabele in  $G'$  de FIRST-verzameling. Is  $G'$  een LL(1)-grammatica? Motiveer je antwoord.
2. Beschouw de context-vrije grammatica  $G$  met startvariabele  $S$  en de volgende producties:

$$\begin{aligned} S &\rightarrow XCd \\ X &\rightarrow Cb \mid d \\ C &\rightarrow a \mid \epsilon \end{aligned}$$

Bepaal voor elke variabele in  $G$  de FIRST-verzameling en de FOLLOW-verzameling.

Is  $G$  een LL(1) grammatica? Motiveer je antwoord.

3. Beschouw de context-vrije grammatica  $G$  met startvariabele  $S$  en de volgende producties:

$$\begin{aligned} S &\rightarrow \mathbf{if\ } B \mathbf{\ } S \mid \mathbf{id = id} \\ B &\rightarrow B \mathbf{boolop\ } E \mid E \\ E &\rightarrow \mathbf{id\ relop\ id} \mid \mathbf{id} \end{aligned}$$

$G$  is dus een eenvoudige grammatica voor een instructie in een programmeertaal.  $S, B, E$  zijn de variabelen en **if**, **id**, **=**, **boolop**, **relop** zijn de terminalen in  $G$ .

- (a) Construeer vanuit  $G$  een nieuwe context-vrije grammatica  $G'$  met  $L(G') = L(G)$ , door

- (indien van toepassing) links-recursie te elimineren, en
- (indien van toepassing) links-factorisatie toe te passen.

Leg uit hoe je  $G'$  uit  $G$  verkrijgt.

- Bepaal voor elke variabele in de nieuwe grammatica  $G'$  zowel de FIRST- als de FOLLOW-verzameling.
- Construeer de LR(0)-automaat bij grammatica  $G'$ .
- Construeer de SLR *parsing table* bij grammatica  $G'$ .
- Is  $G'$  een SLR grammatica? Motiveer je antwoord.

Beschouw nu de context-vrije grammatica  $H$  met startvariabele  $S$  en de volgende producties:

- $S \rightarrow S - A$
- $S \rightarrow A$
- $A \rightarrow AB$
- $A \rightarrow B$
- $B \rightarrow B+$
- $B \rightarrow p$
- $B \rightarrow q$

- Geef (ad hoc) een afleidingsboom in  $H$  voor de string  $q + p$ .
- Gegeven is dat de SLR parsing table bij grammatica  $H$  er als volgt uit ziet:

State	Action					Goto		
	-	+	p	q	\$	S	A	B
0			s6	s7		1	9	8
1	s2				acc			
2			s6	s7			3	8
3	r1		s6	s7	r1			4
4	r3	s5	r3	r3	r3			
5	r5	r5	r5	r5	r5			
6	r6	r6	r6	r6	r6			
7	r7	r7	r7	r7	r7			
8	r4	s5	r4	r4	r4			
9	r2		s6	s7	r2			4

Parse de string  $q + p$  met deze tabel. Laat bij iedere stap duidelijk zien wat je doet, bijvoorbeeld met behulp van een tabel van de volgende vorm:

States on stack	Corresponding Symbols on stack	Input	Action
...	...	...	...