

**Werkcollege Compilerconstructie**  
**Vrijdag 21 september 2018**

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

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

- (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.
- (d) Bepaal voor elke rechterkant  $\alpha$  van een productie  $A \rightarrow \alpha$  in  $G'$  de verzameling  $FIRST(\alpha)$ .
- (e) Is  $G'$  een LL(1)-grammatica? Motiveer je antwoord.
2. Beschouw de context-vrije grammatica  $G$  met startvariabele  $S$  en de volgende producties:

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

- (a) Bepaal voor elke variabele in  $G$  de FIRST-verzameling en de FOLLOW-verzameling.
- (b) Bepaal voor elke rechterkant  $\alpha$  van een productie  $A \rightarrow \alpha$  in  $G$  de verzameling  $FIRST(\alpha)$ .
- (c) Is  $G$  een LL(1) grammatica? Motiveer je antwoord.
3. (a) Elimineer op een systematische wijze de (directe en indirecte) links-recursie uit de grammatica  $G_1$  met variabelen  $\{S, A, B, C, D\}$ , terminalen  $\{a, b\}$ , startvariabele  $S$  en de volgende producties:

$$\begin{aligned} S &\rightarrow AB \\ A &\rightarrow \epsilon \mid Ca \\ B &\rightarrow bD \mid Bb \\ C &\rightarrow Sb \mid a \\ D &\rightarrow Sa \mid b \end{aligned}$$

Leg uit hoe je te werkt gaat en geef tussenresultaten.

N.B.: hoewel  $G_1$  een  $\epsilon$ -productie bevat, werkt het algoritme uit het boek wel gewoon.

- (b) Laat  $G_2$  het resultaat van het vorige onderdeel zijn. Construeer vanuit  $G_2$  een nieuwe context-vrije grammatica  $G_3$  door (indien van toepassing) links-factorisatie toe te passen.
  - (c) Bepaal voor elke variabele in de resulterende grammatica  $G_3$  de FIRST-verzameling.
  - (d) Is  $G_3$  een LL(1) grammatica? Motiveer je antwoord.
4. Bij het verwijderen van indirecte links-recursie zetten we de variabelen in een bepaalde volgorde en willen we alleen nog maar ‘vooruit wijzen’. Om dat te bereiken, lopen we de variabelen van voor naar achteren af, en passen we producties zo nodig aan.

Kijk eens wat er zou gebeuren bij de grammatica  $G_1$  van de vorige opgave, als je alleen vooruit wil wijzen, maar je zou de variabelen van achteren naar voren aflopen om de producties (zo nodig) aan te passen.

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

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

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

- (b) Bepaal voor elke variabele in de nieuwe grammatica  $G'$  zowel de FIRST- als de FOLLOW-verzameling.
- (c) Construeer de top-down *parsing table* bij de nieuwe grammatica  $G'$ .
- (d) Is de nieuwe grammatica  $G'$  een LL(1) grammatica? Motiveer je antwoord.

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

$$\begin{aligned}
 S &\rightarrow BA \\
 A &\rightarrow -BA \mid \epsilon \\
 B &\rightarrow DC \\
 C &\rightarrow DC \mid \epsilon \\
 D &\rightarrow pE \mid qE \\
 E &\rightarrow +E \mid \epsilon
 \end{aligned}$$

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

Non-terminal	Input Symbol				
	-	+	$p$	$q$	\$
$S$			$S \rightarrow BA$	$S \rightarrow BA$	
$A$	$A \rightarrow -BA$				$A \rightarrow \epsilon$
$B$			$B \rightarrow DC$	$B \rightarrow DC$	
$C$	$C \rightarrow \epsilon$		$C \rightarrow DC$	$C \rightarrow DC$	$C \rightarrow \epsilon$
$D$			$D \rightarrow pE$	$D \rightarrow qE$	
$E$	$E \rightarrow \epsilon$	$E \rightarrow +E$	$E \rightarrow \epsilon$	$E \rightarrow \epsilon$	$E \rightarrow \epsilon$

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:

Matched	Stack	Input	Action
	$S\$$	$q + p\$$	output $S \rightarrow BA$
...	...	...	...