

Respostas às questões da Prova de Compilação – 18/11/2017

Questão 1 – A

Questão 2 – D

Questão 3 – B

Questão 4 – D

Questão 5 – C

Questão 6 – E

Questão 7 – C

Questão 8 – B

Questão 9

A gramática G é recursiva à esquerda o que a impede de ser analisada pelo método “top-down”. Vamos fazer transformações sistemáticas para remover a recursividade à esquerda das produções dos símbolos não terminais S e R. A gramática G dada é:

$$\begin{aligned} S &\rightarrow S b B \mid S c B \mid b B \mid c B \mid B \\ B &\rightarrow B d A \mid B h A \mid A \\ A &\rightarrow f c d \mid f i S \mid a \end{aligned}$$

Realizando transformações para remover a recursividade à esquerda:

$$\begin{aligned} S &\rightarrow b B S' \mid c B S' \mid B S' \\ S' &\rightarrow b B S' \mid c B S' \mid \lambda \\ B &\rightarrow A B' \\ B' &\rightarrow d A B' \mid h A B' \mid \lambda \end{aligned}$$

Além disso, as produções de A impedem que a decisão sobre a escolha da produção seja feita com apenas um símbolo de lookahead. Precisamos fatorar, da seguinte forma:

$$\begin{aligned} A &\rightarrow a \mid f A' \\ A' &\rightarrow c d \mid i S \end{aligned}$$

Portanto a gramática G' é expressa através das seguintes produções:

$$\begin{aligned} S &\rightarrow b B S' \mid c B S' \mid B S' \\ S' &\rightarrow b B S' \mid c B S' \mid \lambda \\ B &\rightarrow A B' \\ B' &\rightarrow d A B' \mid h A B' \mid \lambda \\ A &\rightarrow a \mid f A' \\ A' &\rightarrow c d \mid i S \end{aligned}$$

Questão 10

Do ponto de vista sintático, os símbolos terminais da gramática são tokens e não cadeias de caracteres. Deste modo, os símbolos terminais em $\{a, b, c\}$ representam tokens e assim são considerados na parte que segue.

Gramática dada:

$S \rightarrow b M a$
 $M \rightarrow a S b \mid b$

Função auxiliar:

```
void match(token t) {  
    se (lookahead == t)  
        lookahead = lexan(...);  
    else erro();  
}
```

Uma função/procedimento por símbolo não terminal:

```
void S(){  
    se (lookahead == b) {  
        match(b); M(); match(a);  
    } else erro();  
}  
  
void M(){  
    se (lookahead == a){  
        match(a); S(); match(b);  
    } else  
        se (lookahead == b)  
            match(b);  
    else  
        erro();  
}
```

Questão 11

Gramática G:

$S \rightarrow b S c S \mid A C$
 $A \rightarrow B A \mid \lambda$
 $B \rightarrow d a$
 $C \rightarrow f a \mid \lambda$

$FIRST(S) = FIRST(AC) \cup FIRST(bScS)$

Como $\lambda \in FIRST(A)$, $FIRST(AC) = FIRST(A) \cup FIRST(C)$

Como $\lambda \in FIRST(C)$, λ deve ser incluído no $FIRST(AC)$

Logo, $FIRST(S) = \{b, d, f, \lambda\}$

$FIRST(A) = FIRST(BA) \cup FIRST(\lambda)$

Como $\lambda \notin FIRST(B)$, $FIRST(BA) = FIRST(B)$

Logo, $FIRST(A) = \{d, \lambda\}$

$FIRST(B) = FIRST(da) = \{d\}$

$FIRST(C) = FIRST(fa) \cup FIRST(\lambda) = \{f, \lambda\}$

$FIRST(AC) = \{d, f, \lambda\}$

$FIRST(BA) = \{d\}$

$\text{FOLLOW}(S) = \{ \$, c \}$
 $\text{FOLLOW}(A) = \{ f, \$, c \}$
 $\text{FOLLOW}(B) = \{ d, f, \$, c \}$
 $\text{FOLLOW}(C) = \{ \$, c \}$

	b	c	d	f	a	\$
S	$S \rightarrow b S c S$	$S \rightarrow AC$	$S \rightarrow AC$	$S \rightarrow AC$		$S \rightarrow AC$
A		$A \rightarrow \lambda$	$A \rightarrow BA$	$A \rightarrow \lambda$		$A \rightarrow \lambda$
B			$B \rightarrow da$			
C		$C \rightarrow \lambda$		$C \rightarrow fa$		$C \rightarrow \lambda$

LL(1) Parsing table

Pilha	Cadeia	Ação
\$S	dadafa\$	imprime $S \rightarrow AC$
\$CA	dadafa\$	imprime $a \rightarrow BA$
\$CAB	dadafa\$	imprime $B \rightarrow da$
\$CAad	dadafa\$	match d
\$CAa	adafa\$	match a
\$CA	dafa\$	imprime $A \rightarrow BA$
\$CAB	dafa\$	imprime $B \rightarrow da$
\$CAad	dafa\$	match d
\$CAa	afa\$	match a
\$CA	fa\$	imprime $A \rightarrow \lambda$
\$C	fa\$	imprime $C \rightarrow fa$
\$af	fa\$	match f
\$a	a\$	match a
\$	\$	Reconheceu