

Substitutiva da Segunda Prova de Construção de Compiladores.

Primeiro Semestre de 2001. Dept. de Computação – UFSCar.

José de Oliveira Guimarães.

Turma A (Terça).

Lembre-se: justifique tudo a menos de menção em contrário.

1. (2.0) Seja a gramática

$P ::= \text{VarDec Assigns} \mid \text{Assigns} \mid N$
 $\text{VarDec} ::= \text{"var"} \text{ ID IDList ";"}$
 $\text{IDList} ::= \mid \text{"," ID IDList}$
 $\text{Assigns} ::= \text{Assigns Assign} \mid \text{Assign}$
 $\text{Assign} ::= \text{ID "=" N}$

Onde N e ID são terminais. Calcule a função first para cada não terminal. Lembre-se: $\text{first}(X) = \{ X \}$ para X terminal, se $X ::= \epsilon$ for uma produção, adicione ϵ a $\text{first}(X)$ e se X for não terminal e $X ::= Y_1 Y_2 \dots Y_k$ for uma produção, então coloque f em $\text{first}(X)$ se, para algum i , $f \in \text{first}(Y_i)$ e $\epsilon \in \text{first}(Y_j)$, $j = 1, 2, \dots, i-1$. Se $\epsilon \in \text{first}(Y_j)$, $j = 1, 2, \dots, k$, então coloque ϵ em $\text{first}(X)$.

Não é necessário justificar, naturalmente.

2. (1.0) Elimine a recursão à esquerda da seguinte gramática:

$E ::= E \text{"or"} T \mid T$
 $T ::= T \text{"and"} F \mid F$
 $F ::= \text{"true"} \mid \text{"false"} \mid \text{"(" E ")"}$

Não é necessário justificar explicitamente, mas deixe claro os passos que você seguiu até a gramática resultante.

3. (1.0) Fatore à esquerda a seguinte gramática:

$E ::= T \text{"or"} E \mid T$
 $T ::= F \text{"and"} T \mid F$
 $F ::= \text{"true"} \mid \text{"false"} \mid \text{"(" E ")"}$

Não é necessário justificar explicitamente, mas deixe claro os passos que você seguiu até a gramática resultante.

4. (2.0) Faça o autômato finito que reconheça a seguinte expressão regular. Faça ambos o desenho

e o código em Java.

$[\text{^}0\text{A}] [0-9] * [\text{a-z}] +$

Não é necessário justificar.

5. (2.0) Pode ser provado que se uma gramática G é $\text{LL}(1)$ então se $A ::= \alpha \mid \beta$ forem duas produções distintas de G , não há um terminal b tal que $b \in \text{first}(\alpha)$ e $b \in \text{first}(\beta)$.

Explique porque esta regra é necessária para que G não seja ambígua. Para auxiliá-lo na resposta, encontre uma sentença que possua duas derivações à esquerda diferentes se esta regra for violada.

6. (2.0) Dada a gramática

1. $E ::= T E'$ 2. $E' ::= + T E'$ 3. $E' ::= \epsilon$
 4. $T ::= F T'$ 5. $T' ::= * F T'$ 6. $T' ::= \epsilon$
 7. $F ::= (E)$ 8. $F ::= \text{id}$

onde $+$, $*$, $($, $)$ e id são terminais, a tabela associada, que chamaremos de \mathbf{M} , é

	id	()	+	*	eof
E	1	1	-	-	-	-
E'	-	-	3	2	-	3
T	4	4	-	-	-	-
T'	-	-	6	6	5	6
F	8	7	-	-	-	-

O símbolo $-$ deve ser lido 0 (zero).

Faça a análise da sentença $\text{id} * \text{id id}$. Utilize uma tabela como a abaixo. A ação pode ser “emp n” para utilizar a regra n para substituição, “de-semp” para reconhecer o token corrente (avança-se para o próximo token) e “aceita” para terminar a análise. Naturalmente, haverá um erro de análise.

Pilha	Entrada	Ação
E	id + id id	

Não é necessário justificar.