

Projeto de Compiladores I

Semestre: 2017-2

Profa. Responsável: Andreia Bonfante

Versão 14 de Novembro de 2017

Considerando a Gramática G descrita abaixo, faça as atividades previstas nos próximos slides.

$G = ( \{Z, I, D, L, X, K, O, S, E, R, T\}, \{var, :, id, ,, integer, real, ;, :=, if, then, +\}, P, Z)$   
P:

$Z \rightarrow I S$

$I \rightarrow var D$

$D \rightarrow L : K O$

$L \rightarrow id X$

$X \rightarrow , L$

$X \rightarrow \epsilon$

$K \rightarrow integer$

$K \rightarrow real$

$O \rightarrow ; D$

$O \rightarrow \epsilon$

$S \rightarrow id := E$

$S \rightarrow if E then S$

$E \rightarrow T R$

$R \rightarrow + T R$

$R \rightarrow \epsilon$

$T \rightarrow id$

Parte 1. Faça os grafos sintáticos de cada uma das regras de G.

Parte 2. Implemente um analisador léxico para G.

Entrada: arquivo texto, contendo uma sequência de código escrito na linguagem G.

Saída: lista de tokens de G ou erro léxico.

Parte 3. Implemente um analisador sintático descendente recursivo para analisar cadeias pertencentes à linguagem gerada por G (utilize funções para cada não terminal).

Utilize a lista de tokens gerada pelo analisador léxico construído na parte 2.

Entrada: arquivo texto, contendo uma sequência de código escrito na linguagem G.

Saída: cadeia sintaticamente correta ou erro sintático (se erro, imprimir qual erro).

Parte 4: Acrescente, aos grafos sintáticos (parte 1), ações semânticas inserção dos identificadores na tabela de símbolos e de verificação de tipos ( tomar como base slides do material de análise semântica, principalmente o 16).

Parte 5: Implemente a tabela de símbolos com rotinas de busca e inserção. Acrescente ao analisador sintático as ações semânticas que você definiu de busca e inserção na tabela de símbolos e de verificação de tipos.

Parte 6: Acrescente aos grafos sintáticos e à sua implementação a geração de código intermediário, com regras tais como previstas abaixo:

$S \rightarrow id := \{E.esq = id.lexval\} E \{gera(':=', E.dir, ' ', E.esq)\}$

$E \rightarrow T \{R.esq = T.dir\} R \{E.dir = R.dir\}$

$R \rightarrow + T \{R_1.esq = T.dir\} R_1 \{R.dir := geratemp; gera('+', R.esq, R_1.dir, R.dir)\}$

$R \rightarrow \epsilon \{R.dir = R.esq\}$

$T \rightarrow id \{T.dir = id.lexval\}$

$S \rightarrow \text{if } E \text{ then } \{S_1.quad := prox; prox := prox+1\} S_1 \{remenda(S_1.quad, JF, E.dir, prox, ' ')\}$

Ps: O índice (<sub>1</sub>) serve apenas para diferenciar qual não terminal está sendo referido nas regras das ações semânticas. Na sintaxe, são os mesmos elementos. Exemplo: Na regra:  $R \rightarrow + T R_1$  a, os dois Rs são o mesmo.



## Parte 6:

Exemplo de entrada

```
var
  a, b, c: integer;
  d: real
if a+b then a:=c
```

Saída produzida

```
1: [+ a b t1]
2: [JF t1 4 -]
3: [:= a c -]
4: [ ... ]
```