



UNIVERSIDADE DE SÃO PAULO - ICMC

Departamento de Ciências de Computação

SCC 206 – Introdução a Compilação - 1º Sem /2010

PROFa: Sandra Aluisio

WIKI: <http://wiki.icmc.usp.br/index.php/SCC-206>

Exercício sobre Tratamento de Erros Sintáticos em A.S.D. com a implementação de Procedimentos Recursivos

Gramática da linguagem MICRO em notação EBNF

1. `<programa> ::= <bloco> .`
2. `<bloco> ::= <decl> inicio <comandos> fim`
3. `<decl> ::= [tipo <idtipo>] [var <idvar>]`
4. `<idtipo> ::= <id> = <id> ; {<idtipo>}`
5. `<idvar> ::= <id> : <id> ; {<idvar>}`
6. `<comandos> ::= <coms> { ; <coms>}`
7. `<coms> ::= <id> := <expr> |
 read (<listaid>) |
 write (<listaexp>) |
 if <exp> then <coms> [else <coms>] |
 inicio <comandos> fim`
8. `<exp> ::= [+|-] <termo> { (+|-) <termo>}`
9. `<termo> ::= <fator> {(*|/) <fator>}`
10. `<fator> ::= <id> | <numero> | (<exp>)`
11. `<listaid> ::= <id> { , <id>}`
12. `<listaexp> ::= <exp> { , <exp>}`

OBS: `<id>` e `<numero>` são considerados terminais.

Exercício

- 1) Levante o conjunto dos seguidores de cada não terminal da gramática acima
- 2) Modifique o Analisador Sintático fornecido para que ele trate erros sintáticos. Siga a estratégia dada em aula que consome símbolos estranhos ao contexto para retornar a um ponto seguro de análise e apresente também alguns reparos.

Considere implementados os procedimentos e funções:

```
function Analex(var s: atomo) : codigo;  
{codigo é do tipo enumerado e atomo é string}
```

```
procedure erro(n : integer);  
{n é o número da mensagem de erro que é impressa}
```

```
procedure teste(n1, n2: set of codigo; n:integer);  
{como fornecida na aula}
```

```
type codigo = (eof, ponto, sinicio, sfim, stipo, svar, igual, pontovirgula, atrib, sread, sif,  
sthen, selse, mais, doispontos, menos, multi, divi, ident, numero, abrepar, fechapar, virgula)
```