

Compiladores II
Profa. Respons. Andreia Bonfante

Projeto de um Compilador

A linguagem LALG¹

Especificação da Sintaxe da Linguagem LALG

Observações:

- Comentários na LALG: entre { } ou /* */
- Identificadores e números são itens léxicos da forma:
 - ident: sequência de letras e dígitos, começando por letra
 - número inteiro: sequência de dígitos (0 a 9)
 - número real: sequência de um ou mais dígitos seguido de um ponto decimal seguido de um ou mais dígitos.
- palavras reservadas – são os tokens usados para fins específicos, ou seja, que são previamente definidos na linguagem.
- símbolos simples e duplos – são aqueles também definidos na linguagem (<, \$, >, etc. como exemplo de simples, e := como exemplo de duplo).

$G_{Lalg} = \{N, T, P, S\}$ -

$N = \{<programa>, <corpo>, <dc>, <comando>, <comandos>, <dc_v>, <mais_dc>, <dc_p>, <variaveis>, <tipo_var>, <mais_var>, <parametros>, <corpo_p>, <lista_par>, <mais_par>, <dc_loc>, <mais_dcloc>, <lista_arg>, <argumentos>, <pfalsa>, <condicao>, <expressao>, <relacao>, <termo>, <outros_termos>, <op_ad>, <op_un>, <fator>, <mais_fatores>, <op_mul>\}$

$T = \{ident, numero_int, numero_real, (,), *, /, +, -, <>, >=, >, <, if, then, \$, while, do, write, read, ;, else, begin, end, :, , , \}$

¹ <http://www.icmc.usp.br/~gracan/download/sce126/SubItem91SCE126.html>

Regras de Produção da LALG

<programa> ::= program ident <corpo> .
<corpo> ::= <dc> begin <comandos> end
<dc> ::= <dc_v> <mais_dc> | <dc_p> <mais_dc> | λ
<mais_dc> ::= ; <dc> | λ
<dc_v> ::= var <variaveis> : <tipo_var>
<tipo_var> ::= real | integer
<variaveis> ::= ident <mais_var>
<mais_var> ::= , <variaveis> | λ
<dc_p> ::= procedure ident <parametros> <corpo_p>
<parametros> ::= (<lista_par>) | λ
<lista_par> ::= <variaveis> : <tipo_var> <mais_par>
<mais_par> ::= ; <lista_par> | λ
<corpo_p> ::= <dc_loc> begin <comandos> end
<dc_loc> ::= <dc_v> <mais_dcloc> | λ
<mais_dcloc> ::= ; <dc_loc> | λ
<lista_arg> ::= (<argumentos>) | λ
<argumentos> ::= ident <mais_ident>
<mais_ident> ::= ; <argumentos> | λ
<pfalsa> ::= else <comandos> | λ
<comandos> ::= <comando> <mais_comandos>
<mais_comandos> ::= ; <comandos> | λ
<comando> ::= read (<variaveis>) |
 write (<variaveis>) |
 while <condicao> do <comandos> \$ |
 if <condicao> then <comandos> <pfalsa> \$ |
 ident <restoldent>
<restoldent> ::= := <expressao> | <lista_arg>
<condicao> ::= <expressao> <relacao> <expressao>
<relacao> ::= = | <> | >= | <= | > | <
<expressao> ::= <termo> <outros_termos>
<op_un> ::= + | - | λ
<outros_termos> ::= <op_ad> <termo> <outros_termos> | λ
<op_ad> ::= + | -
<termo> ::= <op_un> <fator> <mais_fatores>
<mais_fatores> ::= <op_mul> <fator> <mais_fatores> | λ
<op_mul> ::= * | /
<fator> ::= ident | numero_int | numero_real | (<expressao>)