

V.4 - Especificação Sintática da LSI-111

Considerações preliminares

Objetivo didático

Paradigma imperativo (estruturado)

- apenas construções básicas, comuns a maioria das linguagens, independentemente de paradigma.

Especificação preliminar

- Identificação das construções

Declarações

Constantes e variáveis

Procedimentos e Funções (com/sem parâmetros)

Comandos

Atribuição, seleção e repetição, E/S

Expressões

operandos e operadores convencionais

Recursos avançados

nenhum (apenas recursos básicos da Ling.)

- Detalhamento (informal) das construções

Declarações:

Variáveis

- declaração explícita
- tipos
 - simples
 - inteiro, real, booleano e caracter
 - cadeia (string) de caracteres
 - Intervalo (inteiros e caracteres)

- vetor (array) uni-dimensional, com elementos de tipo simples

Constantes

- declaração explícita
- Tipos simples

Procedimentos/Funções

- Proc (sem tipo)
- Função (com tipo pré-definido)
- Ambos com e sem parâmetros
 - parâmetros por valor
 - e por referência
 - apenas de tipos simples
- permitir proc/funções aninhados
- permitir declarações locais de tudo

Comandos

Atribuição

Chamada de procedimento

E/S: (sem arquivos e sem formatação)

- Entrada – lista de identificadores
- Saída – lista de expressões

Seleção: se-então, (if-then)

se-então-senão (if-then-else)

Repetição: enquanto-faça (while-do)

repita-até (repeat-until)

Comando composto: início “comandos” fim

Comando vazio: ϵ

Expressões:

Operadores

aritméticos:

Binários: +, -, *, / e DIV

Unário: -

lógicos: e, ou, não

relacionais: =, <, >, <=, <>, <>,

Operandos

variáveis: simples ou indexadas

constantes: numéricas (inteiras e reais),
booleanas (verdadeiro/falso),
literais (cadeia ou char)

designadores de função:

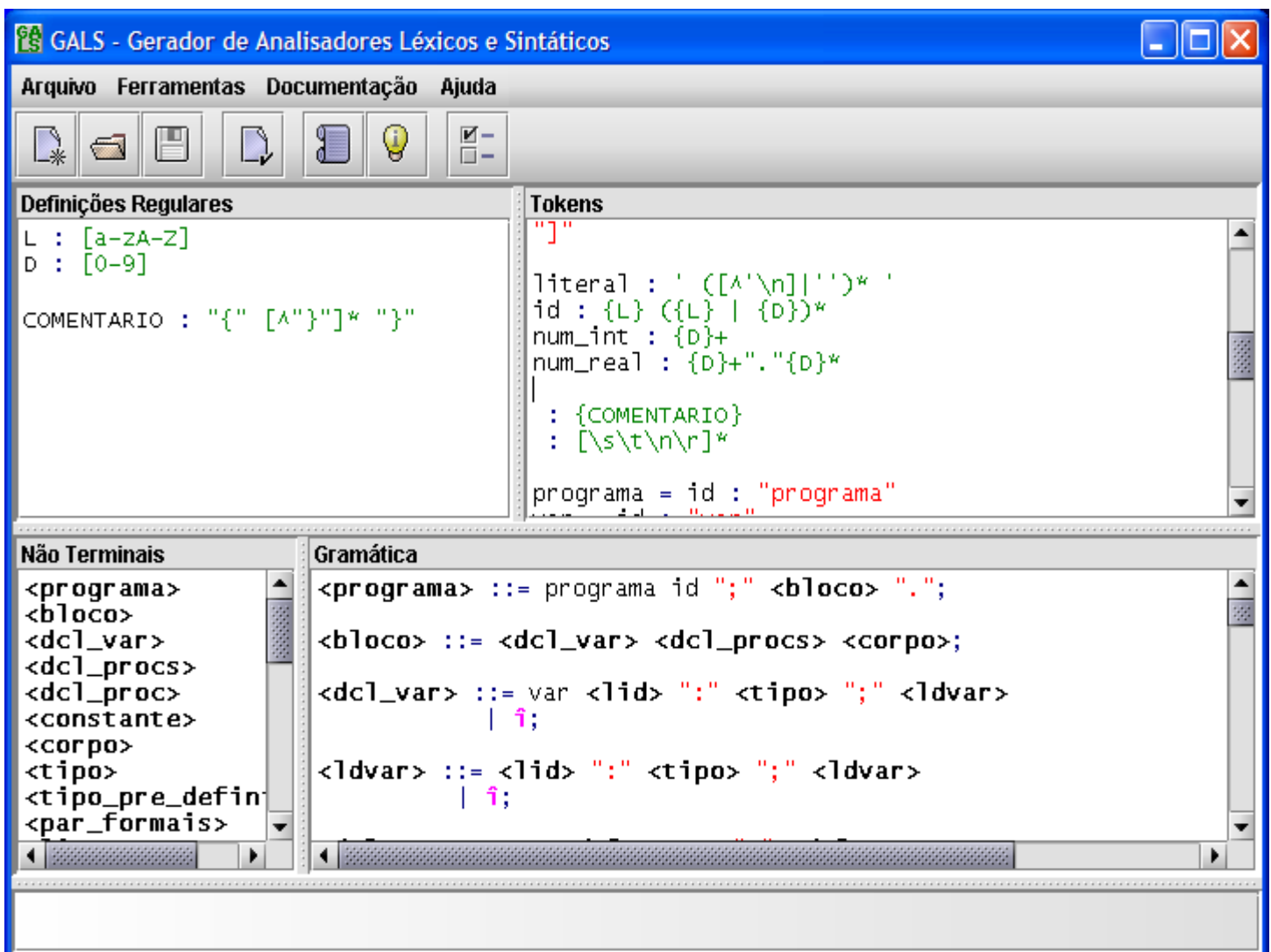
ativação de função

Especificação formal

- Mecanismo: GLC
- Notação: BNF pura (a ser usada no GALS)

V.5 – Implementação do PARSER da LSI-111

- Técnica de análise sintática: PREDITIVA (LL(1))
- Estrutura: 1 PASSO
 - A.S. como núcleo do compilador
- Estratégia de implementação:
 - Uso do gerador GALS
- Adaptar o Léxico (de acordo com a GLC padrão)
- Integrar Léxico e Sintático (via GALS)
- Linguagem de implementação: mesma do léxico
- Equipes: 2 componentes (mesma do léxico)
- Data de entrega: 11/07/2011
- Forma de entrega: Encaminhar por e-mail arquivo zipado contendo: fonte, executável e testes



GRAMÁTICA PADRÃO LSI-111

<programa> ::= <declaracoes> <comandos> "." ;

<declaracoes> ::= <dcl_const> <dcl_var> <dcl_procs> ;

<dcl_const> ::= const <tipo_pre_def> id "=" <constante> ";" <dcl_const>
| ^ ;

<dcl_var> ::= var <lid> ":" <tipo> ";" <dcl_var> | ^ ;

<dcl_procs> ::= <dcl_proc> ";" <dcl_procs> | ^ ;

<dcl_proc> ::= proc id <parametros> ";" <declaracoes> <comandos>
| funcao id <parametros> ":" <tipo_pre_def> ";" <declaracoes> <comandos>
;

<parametros> ::= "(" <listapar> ")" | ^ ;

<listapar> ::= ref <lid> ":" <tipo_pre_def> <rep_listapar>
| val <lid> ":" <tipo_pre_def> <rep_listapar> ;

<rep_listapar> ::= ";" <listapar> | ^ ;

<lid> ::= id <rep_id> ;

<rep_id> ::= "," id <rep_id> | ^ ;

<constante> ::= id | <constante_explicita> ;

<tipo> ::= <tipo_pre_def>
| intervalo <constante> ".." <constante>
| cadeia "[" <constante> "]"
| vetor "[" <constante> "]" de <tipo_pre_def> ;

<tipo_pre_def> ::= inteiro | real | booleano | caracter ;

<comandos> ::= inicio <comando> <rep_comando> fim ;

<rep_comando> ::= ";" <comando> <rep_comando> | ^ ;

<comando> ::= id <rcomid>
| inicio <comando> <rep_comando> fim
| se <expressao> entao <comando> <senaoparte>
| enquanto <expressao> faca <comando>

```

| repita <comando> ate <expressao>
| leia "(" <lid> ")"
| escreva "(" <expressao> <rep_expressao> ")" |  $\hat{1}$ ;

```

<senaoparte> ::= senao <comando> | $\hat{1}$;

```

<rcomid> ::= ":" <expressao>
| "[" <expressao> "]" ":" <expressao>
| "(" <expressao> <rep_expressao> ")" |  $\hat{1}$  ;

```

<rep_expressao> ::= "," <expressao> <rep_expressao> | $\hat{1}$;

<expressao> ::= <expsimp> <resto_expressao> ;

<resto_expressao> ::= <oprel> <expsimp> | $\hat{1}$;

<oprel> ::= "=" | "<" | ">" | ">=" | "<=" | "<>" ;

<expsimp> ::= <termo> <rep_expsimp> ;

<rep_expsimp> ::= <op_add> <termo> <rep_expsimp> | $\hat{1}$;

<op_add> ::= "+" | "-" | ou ;

<termo> ::= <fator> <rep_termo> ;

<rep_termo> ::= <op_mult> <fator> <rep_termo> | $\hat{1}$;

<op_mult> ::= "*" | "/" | e ;

```

<fator> ::= nao <fator> | "-" <fator> | "(" <expressao> ")"
| <variavel> | <constante_explicita> ;

```

<variavel> ::= id <rvar> ;

```

<rvar> ::= "(" <expressao> <rep_expressao> ")"
| "[" <expressao> "]" |  $\hat{1}$  ;

```

<constante_explicita> ::= num_int | num_real | falso | verdadeiro | literal ;