Geração de Código para LALG **(continuação)**

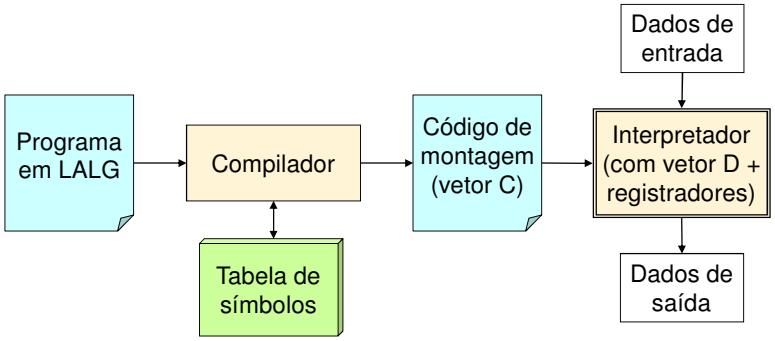
Ambiente de execução para LALG
Máquina hipotética
Repertório de instruções

Prof. Thiago A. S. Pardo

1

Geração de código para LALG

- Interpretador



2

Repertório de instruções

- Grupos de instruções que funcionam para LALG, mas que funcionariam também para o básico de outras linguagens, como Pascal e C
  - Avaliação de expressões
  - Atribuição
  - Comandos condicionais e iterativos
  - Entrada e saída
  - Alocação de memória
  - Inicialização e finalização de programa
  - Procedimentos
- Junto com cada instrução
  - **interpretação durante a execução em cor verde**
  - **ações durante compilação em cor vermelha**

3

Instruções para procedimentos

- Características da LALG
  - Passagem de parâmetros por valor
  - Somente procedimentos globais

4

Instruções para procedimentos

- Ao chamar procedimento
  - PUSHER e
  - {PARAM n}
  - {.....}
  - CHPR p
- No início do procedimento
  - {DSVI k}
  - {COPVL} → Deve levar em conta a posição usada pelo endereço de retorno na pilha D, isto é, soma-se 1 ao endereço obtido para o primeiro parâmetro ou variável local
  - {.....}
- No fim do procedimento
  - DESM n
  - RTPR

5

Exemplo: com parâmetros

```
Program ex3;
var a,b: integer;
procedure proc(x,y:integer);
        var l: integer;
        begin
                l:= x + y;
                x:= l;
        end;
begin
        read(a,b);
        proc(a,b);
end.
```

```
1.INPP
2. ALME 1
3. ALME 1
4. DSVI 16
5. COPVL
6. COPVL
7. ALME1
8.CRVL 3
9. CRVL 4
10. SOMA
11. ARMZ 5
12. CRVL 5
13. ARMZ 3
14. DESM 3
15. RTPR
16. LEIT
17. ARMZ 0
18. LEIT
19. ARMZ 1
20. PUSHER 24
21. PARAM 0
22. PARAM 1
23. CHPR 5
24. PARA
```

Exercício

- Gere código para o programa ao lado e interprete o código gerado

```
program p;
var x: integer;
procedure nomep(a:real);
var y: integer;
begin
y:=1;
end;
procedure teste(b,c:real);
var d: integer;
begin
d:=1;
if (b>c) then
begin
b:=b-c;
c:=2;
end;
end;
begin
read(x);
nomep(x);
teste(x,5);
write(x);
end.
```

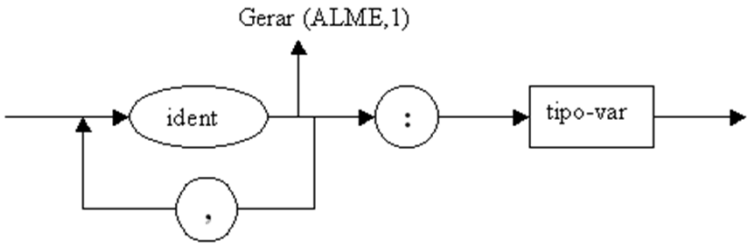
Exercício

- E se procedimento recursivo?
  - Gere o código e interprete
    - Funciona? Justifique.
    - E procedimentos que chamam outros procedimentos?

```
program p;
var x: integer;
procedure nomep(a:real);
var y: integer;
begin
y:=a+2;
if y<10 then
nomep(y);
end;
begin
read(x);
nomep(x);
end.
```

Geração de código para LALG

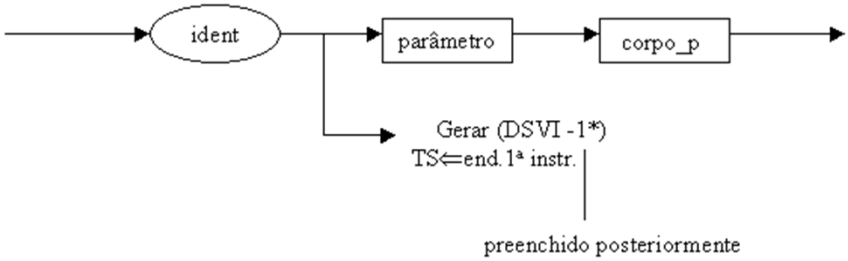
- Alguns exemplos de onde se gerar código
  - Declaração de variáveis



9

Geração de código para LALG

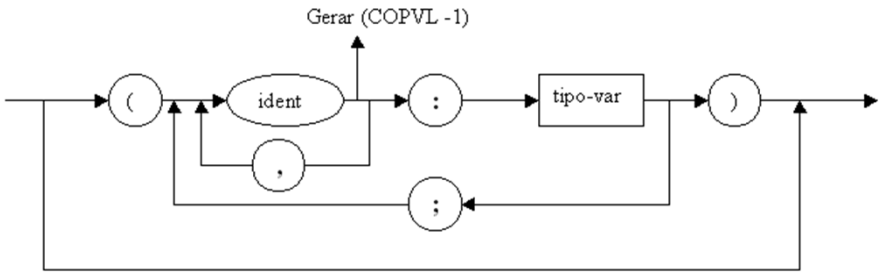
- Alguns exemplos de onde se gerar código
  - Declaração de procedimentos



10

Geração de código para LALG

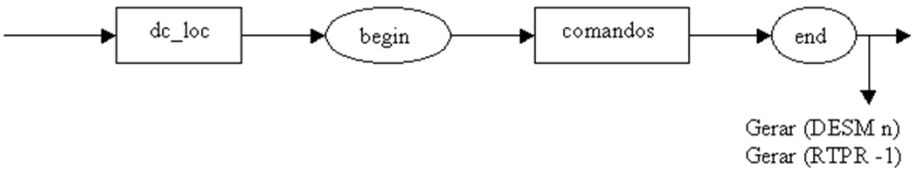
- Alguns exemplos de onde se gerar código
  - Parâmetros de procedimentos



11

Geração de código para LALG

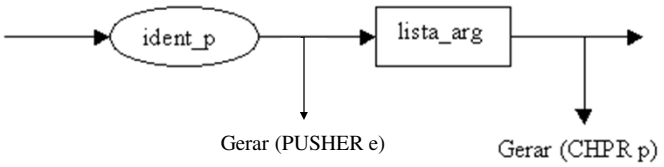
- Alguns exemplos de onde se gerar código
  - Corpo do procedimento



12

Geração de código para LALG

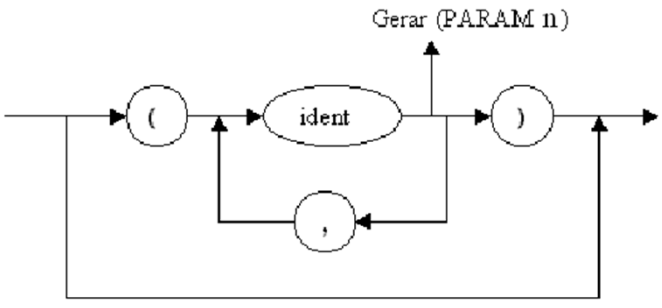
- Alguns exemplos de onde se gerar código
  - Chamada de procedimento



13

Geração de código para LALG

- Alguns exemplos de onde se gerar código
  - Lista de argumentos na chamada de procedimento



14

Exercício em duplas para entregar

- Escreva o procedimento sintático completo para a declaração de variáveis na LALG
  - Tratamento de erros sintáticos pelo modo pânico
  - Análise semântica e tratamento de erros semânticos
  - Geração de código

<dc_v> ::= var <variaveis> : <tipo_var> ;
<tipo_var> ::= real | integer
<variaveis> ::= ident <mais_var>
<mais_var> ::= , <variaveis> | λ

Exercício – possível solução

```
procedimento dc_v(S) {
        se (simb=var)
                então obter_símbolo()
                senão {
                        imprimir("Erro: var esperado");
                        ERRO(S+{id});
                }
        se (simb=id)
                então {
                        se busca_TS(cadeia,token=id,cat=var,escopo=0)=TRUE
                                então imprimir("Erro: identificador declarado novamente")
                                senão inserir_id_TS(cadeia,token=id,cat=var,escopo=0,end=s++);
                        se erro_léxico=FALSE e erro_sintático=FALSE e erro_semântico=FALSE
                                então gera_codigo(contador_linha++ || "ALME 1");
                        obter_símbolo();
                }
                senão {
                        imprimir("Erro: id esperado");
                        ERRO(S+{:}+{,});
                }
...
```

Rosa=interface com léxico, azul=semântica, verde=geração de código

Exercício – possível solução

```
        enquanto (simb=simb_virgula) faça {
                obter_símbolo();
                se (simb=id)
                    então {
                        se busca_TS(cadeia,token=id,cat=var,escopo=0)=TRUE
                            então imprimir("Erro: identificador declarado novamente")
                            senão inserir_id_TS(cadeia,token=id,cat=var,escopo=0,end=s++);
                        se erro_léxico=FALSE e erro_sintático=FALSE e erro_semântico=FALSE
                            então gera_codigo(contador_linha++ || "ALME 1");
                        obter_símbolo();
                    }
                    senão {
                        imprimir("Erro: id esperado");
                        ERRO(S+{:}+{,});
                    }
        }
        se (simb=simb_dp)
                então obter_símbolo()
                senão {
                        imprimir("Erro: ':' esperado");
                        ERRO(S+{real,integer});
                }
...
```

Rosa=interface com léxico, azul=semântica, verde=geração de código

17

Exercício – possível solução

```
        se (simb=real) ou (simb=integer)
                então {
                        inserir_tipo_ids_declarados_TS(cadeia,cat=var,escopo=0);
                        obter_símbolo();
                }
                senão {
                        imprimir("Erro: 'real' ou 'integer' esperado");
                        ERRO(S+{;});
                }
        se (simb=simb_pv)
                então obter_símbolo()
                senão {
                        imprimir("Erro: ';' esperado");
                        ERRO(S+{var});
                }
}
```

Rosa=interface com léxico, azul=semântica, verde=geração de código

18

Exercício

- Escreva o procedimento sintático completo para os comandos da LALG
  - Tratamento de erros sintáticos pelo modo pânico
  - Análise semântica e tratamento de erros semânticos
  - Geração de código

<cmd> ::= read (<variaveis>) |
write (<variaveis>) |
while <condicao> do <cmd> |
if <condicao> then <cmd> <pfalsa> |
ident := <expressao> |
ident <lista_arg> |
begin <comandos> end
<variaveis> ::= ident <mais_var>
<mais_var> ::= , <variaveis> | λ
...