

Construção de Compiladores

Período Especial

Aula 14: Rótulos e Comandos de Desvio

Bruno Müller Junior

Departamento de Informática
UFPR

2020

- 1 Introdução
- 2 Escopo e visibilidade em Pascal
 - Semântica
 - Funcionamento na prática
 - Comandos MEPA
 - Comandos MEPA
 - Esquema de Tradução
 - Exemplo
- 3 Código MEPA

Introdução

- Há muita discussão sobre o uso de rótulos e comandos de desvio:
 - O principal argumento dos contrários é o que se chama “programação macarrônica”.
 - Os favoráveis apresentam situações pontuais onde o uso de goto pode deixar o código mais claro.
- Sugerimos a leitura de dois artigos clássicos, de dois dos maiores nomes da história da computação (tem um referência na wikipedia que resume, mas os artigos originais são espetaculares).
 - “Go To Statement Considered Harmful”, *Edsger W. Dijkstra*, Communications of the ACM, 1968.
 - “Structured Programming with go to Statements”, *Donald Knuth*, Computing Surveys 6 (4): 261–301, 1974.

Escopo e visibilidade em Pascal

- Um rótulo é um número.
- Um rótulo deve ser declarado (rótulos tem tempo de vida).
- Um rótulo só pode ser instanciado uma vez.
- Um desvio deve ser para um rótulo declarado e instanciado.

```
program escopoProc (input, output);  
|   label 100, 200;  
|   procedure p;  
|   |   label 100, 300;  
|   |   ... goto 300; ...  
|   |   300: ...  
|   |   100: ...  
|   |   goto 100; ... goto 200;  
|   |   ...  
|   ...  
|   200: ... goto 200 ...  
|   ...  
|   100:  
|   ...
```

Escopo e visibilidade em Pascal

- Normalmente os compiladores Pascal colocam uma regra adicional: o rótulo-alvo do desvio tem de estar declarado e instanciado no mesmo procedimento do desvio (goto). Isto permite simplificar o modelo de execução utilizando somente um comando semelhante ao DSVS.
- O nosso Pascal simplificado permite desvio para qualquer rótulo que esteja visível.
- A implementação do mecanismo em tempo de execução é mais complicada, porém muito mais interessante do ponto de vista de projeto de execução de uma linguagem.

Funcionamento na prática

- Para entender como funciona o modelo na MEPA, considere que o programa abaixo foi executado na seguinte seqüência:
 $r \rightarrow s \rightarrow p \rightarrow q \rightarrow q \rightarrow q \rightarrow \text{goto } 100$

```
program modeloGoto (input, output);
| var a: integer;
| procedure p;
| | label 100;
| | var b: integer;
| | procedure q;
| | | var c: integer;
| | | begin ... q(1) ... goto 100; ... end;
| | begin ... q(2) ... 100: ... end;
| procedure r;
| | var d: integer;
| | procedure s;
| | | var e: integer;
| | | begin ... p(3) ... end;
| | begin ... s(4) ... end;
| begin ... r(5) ... end.
```

Comandos MEPA

- Quando foi executado o goto 100, a situação da pilha (M) e do vetor de regs. de base (D) voltou à mesma situação que estavam quando o R.A. “ativo” era p.
- Isto pode ser feito com duas operações:
 - Desempilha todos os R.A. com $k > k_p$, restaurando todos os $D[k]$.
 - Restaura topo da pilha (s) à mesma altura que estava no procedimento p (onde está o 100:);
- Cada operação é um comando MEPA novo.

Comandos MEPA

Instrução	Ação	Significado
DSVR p,j,k	$\text{temp1} := k$ enquanto $\text{temp1} \neq j$ $\text{temp2} := M[D[\text{temp1}] - 2];$ $D[\text{temp1}] := M[D[\text{temp1}] - 1];$ $\text{temp1} := \text{temp2}$	Desvia para rótulo p: rótulo alvo j: n.l. destino k: n.l. atual
ENRT k,n	$s := D[j] + n - 1$	Entra em Rotulo k: n.l. atual n: num. v.l.

Esquema de Tradução

label <rotulo>



Inserir na TS

goto <rotulo>



DSVR p,j,k

<rotulo>:



ENRT k, n

Exemplo

Exemplo

```

1 program exemplo12 (input, output);
2   var x : integer;
3   procedure p (var t : integer);
4     label 100, 200;
5     var s : integer;
6     function f(z : integer): integer;
7       begin (* f *)
8         if z<0 then goto 200
9           else if z=0 then f:=2
10            else f:=f(z-2)①*z+1
11       end; (* f *)
12   begin (* p *)
13     100: s:=f(t)②; t:=s;
14     if t<x then goto 100;
15     200: x:=x-1
16   end; (* p *)
17   procedure r;
18     procedure q;
19       var y : integer;
20       begin read(y); p(y)③; write(y) end; (* q *)
21     begin q④ end; (* r *)
22   begin read(x); r⑤ end.

```

Código MEPA

1	INPP	24	R10: NADA	47	CRVL 0,0	70	CHPR R01,2
2	AMEM 1	25	AMEM 1	48	CMME	71	CRVL 2,0
3	DSVS R00	26	CRVL 2,-4	49	DSVF R12	72	IMPR
4	R01: ENPR 1	27	CRCT 2	50	DSVR R02,1,1	73	DMEM 1
5	AMEM 1	28	SUBT	51	DSVS R11	74	RTPR 2,0
6	DSVS R04	29	CHPR R05,2	52	R12: NADA	75	R14: NADA
7	R05: ENPR 2	30	CRVL 2,-4	53	R11: NADA	76	CHPR R15,1
8	DSVS R06	31	MULT	54	R03: ENRT 1 1	77	RTPR 1,0
9	R06: NADA	32	CRCT 1	55	CRVL 0,0	78	R00: NADA
10	CRVL 2,-4	33	SOMA	56	CRCT 1	79	LEIT
11	CRCT 0	34	ARMZ 2,-5	57	SUBT	80	ARMZ 0,0
12	CMME	35	R09: NADA	58	ARMZ 0,0	81	CHPR R13,0
13	DSVF R08	36	R07: NADA	59	DMEM 1	82	DMEM 1
14	DSVR R03,1,2	37	RTPR 2,1	60	RTPR 1,1	83	PARA
15	DSVS R07	38	R04: NADA	61	R13: ENPR 1		
16	R08: NADA	39	R02: ENRT 1 1	62	DSVS R14		
17	CRVL 2,-4	40	AMEM 1	63	R15: ENPR 2		
18	CRCT 0	41	CRVI 1,-4	64	AMEM 1		
19	CMIG	42	CHPR R05,1	65	DSVS R16		
20	DSVF R10	43	ARMZ 1,0	66	R16: NADA		
21	CRCT 2	44	CRVL 1,0	67	LEIT		
22	ARMZ 2,-5	45	ARMI 1,-4	68	ARMZ 2,0		
23	DSVS R09	46	CRVI 1,-4	69	CREN 2,0		

- Página para anotações

Licença

- Slides desenvolvidos somente com software livre:
 - \LaTeX usando beamer;
 - Inkscape.
- Licença:
 - Creative Commons Atribuição-Use Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License. <http://creativecommons.org/licenses/by-nc-nd/2.5/br/>