

# Construção de Compiladores

## Avaliação do Compilador

### Comando Goto

Bruno Müller Junior

Departamento de Informática  
UFPR

21 de Junho de 2023

- Este arquivo descreve a alteração a ser efetuada no compilador.
- Os próximos slides apresentam:
  - Contexto
  - Sintaxe
  - Semântica
  - Entrega do Trabalho

# Contexto

- Este pacote contém 4 arquivos: o arquivo PDF que você está lendo e 3 arquivos contendo programas Pascal chamados goto1.pas, goto2.pas e goto3.pas.
- A alteração será a inclusão do comando goto.
- O comando goto permite o desvio incondicional do fluxo para o rótulo indicado.
- Para incluí-lo em nosso no Pascal é necessário alterar o compilador em três locais:
  - 1 Declarar o rótulo. O rótulo é sempre um número.
  - 2 Inserir o rótulo em algum local do programa.
  - 3 Utilizar o comando de desvio (goto).

# Sintaxe

Tomasz - páginas 172-174

- ```
2. <bloco> ::=  
  [ <parte de declarações de rotulos> ]  
  ...  
  
3. <parte de declarações de rotulos> ::=  
  label número {, número };  
  
17. <comando> ::= [<número>:] <comando sem rótulo>  
  
18. <comando sem rótulo> ::=  
  ...  
  | <desvio>  
  ...  
  
21. <desvio> ::= goto <número>
```

# Alteração - Semântica

- Ao encontrar um número na regra 17, seu compilador deverá gerar o comando MEPA "ENRT  $k$ ,  $n$ " onde  $k$  é o nível léxico corrente e  $n$  é o número de variáveis locais do procedimento corrente.
- Ao encontrar um goto na regra 21, seu compilador deverá gerar o comando MEPA "DSVR  $p, j, k$ " onde  $p$  rótulo mepa para onde desviar,  $j$  é o nível léxico onde está declarado o rótulo e  $k$  é o nível léxico corrente ( $k \leq j$ )
- Os programas goto1.pas e goto2.pas devem funcionar, mas goto3.pas deve acusar erro.

# Horário

- Horário de início: 21/06/2023, 17h30
- Horário de entrega sem penalidade: 21/06/2023, 23h59m
- Entrega com penalidade: -0,5 ponto a cada 12 horas após o horário de entrega sem penalidade.

# Avaliação

40 pontos Parte 1: Análise do funcionamento do programa.

- Todos os programas de `AvaliaTrabs.tar.bz2` devem funcionar exceto os últimos (exemploErro que devem dar erro).

# Avaliação

30 pontos Parte 2: Modificação do programa.

- Descreva a idéia original para abordar a alteração;
  - O que deu certo e o que teve de ser modificado no trajeto?
- Quais foram os trechos do código que foram alterados?
- Quais os maiores problemas para fazer funcionar?



# Avaliação

## 30 pontos Parte 3: Questionário

- Relate as principais dificuldades ao longo do desenvolvimento.
- Procure a descrição do comando `record`. Descreva (máximo 30 linhas) como incluí-lo no compilador.

# Formato

- Extraia novamente `AvaliaTrabs.tar.bz2`;
- Modifique o nome do diretório para `<GRRxxxx-GRRyyyy>`,
  - ① inclua um diretório (`./Original`) contendo os fontes de seu compilador originais;
  - ② inclua um diretório (`./Alterado`) contendo os fontes de seu compilador após alteração;
  - ③ crie arquivo funcionamento.txt: relate ali algum problema que eu deva saber quando eu avaliar os 40 pontos;
  - ④ crie arquivo modifica.txt: relate ali algum problema que eu deva saber quando eu avaliar a alteração e as perguntas e respostas sobre modificações;
  - ⑤ crie arquivo perguntas.txt: responda ali o questionário.
  - ⑥ crie arquivo alunos.txt: RA e nomes dos alunos.

# Entrega

- Empacote o diretório <GRRxxxx-GRRyyyy> no arquivo <GRRxxxx-GRRyyyy>.tar.bz2 AvaliaTrabs.tar.bz2;
- Envie para brunomuller65@gmail.com com assunto CI211-Trab-<GRRxxxx-GRRyyyy>.
- Obviamente <GRRxxxx-GRRyyyy> devem ser modificados para os GRRs dos alunos.