

Analísadores Léxicos e Sintáticos

Projeto Final

Prof. Edward Hermann Haeusler

2022.1

Uma das técnicas para implementar linguagens de programação é conhecida como “bootstrapping”. Trata-se de escrever um compilador para as construções mais poderosas de linguagem de programação L e depois usar este núcleo compilável de L para implementar o compilador do restante de L .

Uma variação do *bootstrapping* é estender a linguagem L usando elementos dela mesma, ou seja, usar o núcleo mais expressivo como código objeto. Por exemplo, o comando **case expr of I1:cmds I2:cmds... esac** pode ser obtido com o uso de diversos comandos de **if then else** aninhados. Neste projeto, sugerimos o uso desta variação do método de Bootstrapping para a implementação de um compilador de uma linguagem bem simples que gera código objeto em alguma linguagem imperativa de sua escolha, C, Python, Lua ou Java.

1 Do objetivo

O Projeto Final desta disciplina (INF1022) consiste em usar o conjunto de ferramentas Flex/Bison (Lex/Yacc) para geração de compiladores com o objetivo de gerar código de uma linguagem imperativa (Provol-One) em uma linguagem de sua escolha. O objetivo deste projeto final é fazer com que os alunos da disciplina apliquem os conceitos aprendidos em um projeto mais prático.

A linguagem Provol-One possui somente os comandos de repetição indefinida (Enquanto X Faça C Fim) e atribuição ($v = \langle \text{expr} \rangle$). Provol-One tem somente tipo numérico inteiro não negativo. Pede-se fazer um compilador de Provol-One que gera código objeto em C, Python, Lua, Java, ou qualquer outra linguagem de sua escolha. Deve-se usar um gerador de analisador sintático que implemente o método LaLR(1) ou outro ascendente, por exemplo, Yacc/Lex, Bison/Flex, etc. Uma vez com este compilador de Provol-One pronto, pede-se que sejam feitas algumas extensões de Provol-One, gerando código na própria Provol-One. Sugerimos a inclusão dos comandos de seleção (if-then e if-then-else) comando de repetição definida (Faça $\langle \text{cmds} \rangle$ X vezes), etc.

2 Do desenvolvimento

A sintaxe da linguagem Provol-One é dada pela gramática abaixo:

<i>program</i>	\rightarrow	<i>ENTRADA varlist SAIDA varlist cmds FIM</i>
<i>varlist</i>	\rightarrow	<i>id varlist</i> <i>id</i>
<i>cmds</i>	\rightarrow	<i>cmd cmds</i> <i>cmd</i>
<i>cmd</i>	\rightarrow	<i>ENQUANTO id FAÇA cmds FIM</i>
<i>cmd</i>	\rightarrow	<i>id = id</i> <i>INC(id)</i> <i>ZERA(id)</i>

Todas as variáveis são do tipo número natural (inteiro não-negativo). O comando *Inc(id)* incrementa em um o conteúdo da variável *id*, *Zera(id)* faz o conteúdo da variável *id* ser 0 (zero), *id = id* copia o conteúdo da variável a direita na variável da esquerda, ou seja é um comando de

atribuição. Os booleanos, usados nos comandos *Enquanto* e de desvio condicional (*Se – Entao* e *Se – Entao – Senao*), são relacionados aos valores numéricos na forma: *falso* é 0 (zero) e *verdadeiro* é qualquer valor positivo. O comando de repetição definido *Faca id vezes < cmds > FIM* repete a execução de *cmds* o número de vezes que for o valor de *id*. Este valor é constante e avaliado no início da execução do comando *Faca*. Assim, um trecho de código na forma *Faca X vezes $X = X + 3$* será executado o mesmo número de vezes que for o valor de *X* no início da execução do *Faca*. Por exemplo, se *X* tiver valor 5, este será o número de repetições de *cmds* para o trecho acima. Quando *X* tem valor zero, nenhuma execução de *cmds* é realizada. Se houver necessidade de fazer pequenas modificações na sintaxe de Provol-One, fiquem a vontade.

Um exemplo de programa em Provol-One é:

```
Program ENTRADA X, Y
      SAIDA  Z
      Z=Y
      ENQUANTO X FACA
          INC(Z)
      FIM
```

A entrega do trabalho constará de:

- (i) Arquivos do lex e yacc (flex e bison) usados para gerar o código na linguagem objeto, assim como sua descrição.
- (ii) Arquivos de exemplos de uso e execução comparada.
- (iii) Equipes com até 2 alunos.
- (iv) Uma folha com o nome dos componentes da equipe (PDF).
- (v) Apenas 1 membro da equipe deverá submeter os arquivos (ead).

A entrega deste trabalho deve ser feita até o dia ~~29 de junho~~ 10hs. 04 de julho - 23:59hs. O cronograma das apresentações (publicado no ead) será definido após a entrega dos artefatos. Em geral, a ordem da "apresentação" será a ordem de entrega (FIFO).