

Lógica da Computação

Aula 11

Raul Ikeda

2º semestre de 2018

Aula Passada

1. Gramáticas LL(k) e LR(k)
2. Pumping Lemma para CFL
3. Hierarquia de Chomsky

Esta Aula

1. Blocos de Programa
2. Variáveis
3. Tabela de Símbolos Simplificada
4. Print

Objetivo

Hoje vamos dar um grande upgrade para a linguagem, que deixa de ser uma mera calculadora para um interpretador de programas.

Meta:

```
{  
    x = 3;  
    y = 4;  
    z = x + y;  
    printf(z);  
}
```

Variáveis

Nossa linguagem vai aceitar identificadores (variáveis):

Atribuição:

`x = 3;`

Leitura:

`y = x + 3;`

O identificador, ao contrário do `IntVal`, não existe realmente até a execução da AST. Como que o interpretador vai lidar com a natureza volátil do seu valor?

Tabela de Símbolos

É uma tabela que existe durante a execução da AST, que armazena os valores de cada variável e pode ser representada de várias formas:

- Lista Ligada
- Árvore
- Pilhas
- *Hashmap* (OPA!)

Portanto vamos implementar uma classe *SymbolTable* que manipula um dicionário e contém um *getter* e um *setter* para as variáveis.

Quais são as possíveis exceções que podem ser disparadas por essa classe?

Blocos

Agora a linguagem vai passar a aceitar múltiplos comandos. Para isso, precisamos definir 2 diretivas:

1. Bloco de programas: { [...] }
2. Separadores de comandos: ;

A nova EBNF:

```
comandos = "{", comando, ";", { comando, ";" }, "}" ;
comando = atribuição | comandos ;
atribuição = identificador, "=", expressão ;
expressão = termo, { ("+" | "-"), termo } ;
termo = fator, { ("*" | "/"), fator } ;
fator = ("+" | "-"), fator | número | "(", expressão, ")" | identificador ;
identificador = letra, { letra | dígito | "_" } ;
número = dígito, { dígito } ;
letra = ( a | ... | z | A | ... | Z ) ;
dígito = ( 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 ) ;
```

Print

Implementar também:

```
printf(5);  
printf(5+2);  
printf(x);
```

Pensar na estrutura da AST correspondente (quantos filhos, valor, etc).

Como será o método Evaluate() desse nó?

Atividade: Roteiro 5

- Roteiro Impresso ou PDF no Blackboard.
- **Atualizar EBNF e DS!**

Próxima Aula

- Gramáticas sensíveis de contexto
- Máquina de Turing de Fita Finita
- Referências:
 - Ramos et al. Cap. 5.1 e 5.4
 - Sipser Cap. 3.1