Insper

Lógica da Computação

Aula 11

Raul Ikeda

1º semestre de 2020

Esta Aula

- Blocos de Programa
- Variáveis
- Tabela de Símbolos Simplificada
- Print

Objetivo

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

Meta:

```
{
  $x1 = 3;
  $y2 = 4;
  $z_final = $x1 + $y2;
  echo $z_final;
}
```

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 (aka Dicionários)

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:

```
BLOCK = "{", { COMMAND }, "}" ;
COMMAND = ( λ | ASSIGNMENT | PRINT), ";" | BLOCK ;
ASSIGNMENT = IDENTIFIER, "=", EXPRESSION, ";" ;
PRINT = "echo", EXPRESSION, ";" ;
EXPRESSION = TERM, { ("+" | "-"), TERM } ;
TERM = FACTOR, { ("*" | "/"), FACTOR } ;
FACTOR = (("+" | "-"), FACTOR) | NUMBER | "(", EXPRESSION, ")" | IDENTIFIER ;
IDENTIFIER = "$", LETTER, { LETTER | DIGIT | "_" } ;
NUMBER = DIGIT, { DIGIT } ;
LETTER = ( a | ... | z | A | ... | Z ) ;
DIGIT = ( 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 ) ;
```

Print

Implementar também:

```
echo 5;
echo 5+2;
echo $x*(4+1);
```

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

Como será o método Evaluate() desse nó?

Detalhe importante: O PHP é *case insensitive* para instruções gerais. Logo, tem que funcionar: echo, ECHO, Echo, EcHo, etc.

Outro detalhe: O PHP **NÃO** é case insensitive para variáveis. Logo, $a é diferente de $A.

Atividade: Roteiro 5

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

Próxima Aula

- Linguagens Sensíveis ao Contexto
- Máquina de Turing de Fita Finita

Referências:

- Ramos et al. Cap. 5.1 e 5.4
- Sipser Cap. 3.1