Marcar como feito

Descrição Enviar Editar Visualizar envios

Data de entrega: terça-feira, 30 set. 2025, 23:55
Arquivos requeridos: tradutor.l (Baixar)
Tipo de trabalho: Trabalho individual

Gerador de Forma Intermediária

Nesse trabalho, utilize o código apresentado em Ações Semânticas para Gerar NPR e gere um compilador (junte as peças...) que receberá como entrada expressões com atribuição, e irá gerar o código da forma intermediária que é interpretado na página uma forma Intermediária simples. As principais diferenças são que aceitaremos identificadores com mais de uma letra, bem como números inteiros e reais, strings, o comando "print" e algumas funções com zero ou mais parâmetros, os operadores "**-**" e "**+**" unários e o fatorial "**!**", que é unário mas é pós-fixado. Por último, temos também o operador "**^**", que é associativo à direita. As funções estarão definidas no interpretador, no map funcao<string,Funcao>, onde Funcao é um ponteiro para uma funcao que recebe void e retorna void (ela desempilha o número de parâmetros corretos e empilha o seu resultado).

Exemplo 1:

```
a = "hello";
b = a + " world" + "!";
print b;
```

Código a ser gerado:

```
a "hello" = ^
b a @ " world" + "!" + = ^
b @ print #
```

Exemplo 2:

```
x = 3.14;
a = max( 2, 3 * 7 + x );
print a;
```

Saída:

```
x 3.14 = ^
a 2 3 7 * x @ + max # = ^
a @ print #
```

Exemplo 3:

```
x = - 3.14;
a = 2 ^ 3 ^ x;
b = 8!;
print - a;
```

Saída:

```
x 0 3.14 - = ^
a 2 3 x @ power # power # = ^
b 8 fat # = ^
0 a @ - print #
```

O símbolo "**@**" será usado para buscar o valor de uma variável na tabela de variáveis. Já o símbolo "**#**" será usado para chamar uma função na tabela de funções. A função **power** implementa uma chamada à função **pow** do C, que eleva um número à potência de outro. Você pode assumir

que o número de parâmetros está correto, e que todas as funções estão definidas - se não estiverem, o erro será de execução. E será assumido também que todas as entradas (programas) estão corretos.

Aqui os passos sugeridos para a execução da tarefa:

1. Modificar a gramática de expressões original (Ações Semânticas para Gerar NPR) para incluir funções, print e mais de uma expressão separadas por ';' ;
2. Acrescentar as ações semânticas necessárias para gerar a forma intermediária;
3. Remover o início comum e a recursividade à esquerda;
4. Criar um arquivo lex para lidar com os novos tokens: num, id e string;
5. Escrever o analisador recursivo descendente no arquivo do lex, após o '%%';
6. Para submeter, copie o arquivo todo do lex ***incluindo a função main***.

A máquina de pilha a ser utilizada encontra-se aqui.

Arquivos requeridos

tradutor.l

```
%%

%%

// Deve ser um arquivo lex com o seu analisador sintático na parte final
```

VPL