universidade de aveiro
theoria poiesis praxis

DETI

Linguagens Formais e Autómatos - 2020

Linguagem Para Análise Dimensional

2º Semestre

2019/2020

Afonso Cardoso - 88964

António Domingues - 89007

Pedro Gonçalves - 88859

Pedro Silva - 89228

Tiago Barros – 88963

Conteúdo

Introdução

A análise dimensional é imprescindível no ramo da Física nos dias que correm. É especialmente utilizada para definir relações entre diferentes grandezas, dando origem às dimensões que são utilizadas nas operações algébricas.

Neste projeto desenvolvemos uma linguagem, começando por criar programas exemplo e definindo os padrões de geração de código para as instruções da mesma, posteriormente passámos à resolução da sintaxe e da semântica da gramática e por fim do interpretador (linguagem declarativa) e do compilador (linguagem compilada), simplificando a criação e a utilização de variáveis de diferentes dimensões.

Linguagem Declarativa

A partir de alguns exemplos demonstrativos criados no início do trabalho, desenvolvemos a linguagem declarativa que torna possível a declaração de diferentes dimensões, unidades ou constantes (e.g. aceleração da gravidade) conforme as preferências do utilizador.

```
program : decl* EOF
        ;

decl : 'dim' WORD '[' WORD ']' 'as' ('real' | 'integer') ';' #DimFromUnit
     | 'dim' WORD 'as' oper ';'                              #DimFromDim
     | 'unit' WORD 'as' oper ';'                             #UnitCreate
     | 'const' WORD 'as' NUM 'as' ('real' | 'integer') ';'   #ConstFromNumb
     | 'const' WORD 'as' NUM oper ';'                        #ConstFromOper
     ;


oper: a1=oper op=('*'|'/') a2=oper                          #OperAritm     //Oper unit
     | '(' oper ')'                                         #OperSub
     | (NUM | WORD)                                         #OperBasic
     ;
```

Linguagem Compilada

Nesta linguagem o principal objetivo é facilitar operações entre dimensões previamente estabelecidas. Ou seja, o interpretador permite uma operação entre metros e centímetros, visto que estes são da mesma dimensão, apenas de escalas diferentes, enquanto que uma entre metros e segundos, seria interrompida.

Para o armazenamento dos 3 tipos de variáveis (dimensão, constante e unidade) criadas pelo utilizador, recorremos à criação de um 'HashMap'.

Esta linguagem está munida de operações que possibilitam ciclos iterativos, condições, atualizar o valor de uma variável existente, imprimir um resultado ou uma frase, entre outras.

```
instructions :
             define ';'
            | print ';'
            | update ';'
            | decision ';'
            | incremento ';'
            | forLoop
        ;

define :
          tipo=WORD id=WORD                                          #defineID
        | tipo=WORD id=WORD '=' valor=operacao (unidades=units)?     #defineOperacao
        ;
```

Instruções

Neste projeto o utilizador pode criar variáveis a partir de dimensões previamente estabelecidas, seguindo o exemplo:

```
distance alturaInicial;
distance dist1 = 10 m;
velocity velInicial = 0 m/s;
velocity velInst;

velInst = 10 m/s;
```

Para realizar operações, deverá seguir a seguinte estrutura:

```
alturaInicial = velInst * 10;
```

Para realizar ciclos 'for' deverá seguir a seguinte estrutura:

```
for(int i=tempoInicial; i<tempoFinal ; i++){
   velInst = 10 * i;
   print("A velocidade instantanea no segundo " + i + " e de " + velInst);
}
```

Para proceder à impressão de algum dado:

```
print("A altura a que a pedra foi lancada foi de " + alturaInicial);
print("A velocidade instantanea no segundo " + i + " e de " + velInst);
```

Processo de Utilização

Para iniciar a utilização do programa, após a linguagem declarativa estar criada, é necessário executar no diretório "*Declare/*" os seguintes comandos: "antlr4-build" seguido de "antlr4-run [filename]". Após estarem efetuados, é criado um ficheiro binário que armazena as características da linguagem declarada.

Em seguida, o utilizador deve mudar para o diretório "Compiler/" e editar o ficheiro que deseja utilizar como exemplo, tendo em conta a linguagem previamente definida. Por fim, é necessário executar os comandos: "antlr4-build" seguido de "antlr4-run [filename]".

Após esta sequência de comandos, será importado o ficheiro binário pelo compilador.

No final, a geração de código é armazenada num ficheiro java com o nome "Output.java". Posteriormente, executa-se "java Output" para ver o resultado obtido do código criado pelo utilizador.

```
cardoso@cardoso-VirtualBox:~/Desktop/LFAprojeto/lfa1920-g15/Compiler$ antlr4-build
Processing ./Compiler
Note: ././VisitCompile.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
cardoso@cardoso-VirtualBox:~/Desktop/LFAprojeto/lfa1920-g15/Compiler$ antlr4-run quedalivre_comp
public class Output{
    public static void main(String[] args){
        Value alturaInicial = new Value();
        Value alturaFinal = new DoubleVar("0.0");
        Value tempoInicial = new DoubleVar("0.0");
        Value tempoFinal = new DoubleVar("5.0");
        Value velInicial = new DoubleVar("0.0");
        Value velInst = new Value();
        Value gravity = new DoubleVar("9.8");
        for(double i = tempoInicial.getValueDouble();i < tempoFinal.getValueDouble();i++){
            velInst.setValueDouble(new Double(gravity.getValueDouble() * i));
            System.out.println("A velocidade instantanea no segundo "+i+" e de "+velInst);

        };
        alturaInicial.setValueDouble(new Double(gravity.getValueDouble() * tempoFinal.getValueDouble() * tempoFinal.getValueDouble() / 2));
        System.out.println("A altura a que a pedra foi lancada foi de "+alturaInicial);
    }
}
cardoso@cardoso-VirtualBox:~/Desktop/LFAprojeto/lfa1920-g15/Compiler$ java Output.java
Note: Output.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
A velocidade instantanea no segundo 0.0 e de 0.0
A velocidade instantanea no segundo 1.0 e de 9.8
A velocidade instantanea no segundo 2.0 e de 19.6
A velocidade instantanea no segundo 3.0 e de 29.400000000000002
A velocidade instantanea no segundo 4.0 e de 39.2
A altura a que a pedra foi lancada foi de 122.5
```

Conclusão

Através deste projeto facilitamos a manipulação de dados criados pelo utilizador, tendo em conta as restrições estabelecidas, e o manuseamento dessas através de diversas operações.

Foi realçada a importância dos fundamentos que originam as linguagens.

Contribuição dos Autores

O aluno António procedeu à realização das gramáticas assim como o início do desenvolvimento dos 'visitors'.

Os alunos Pedro Silva e Afonso desenvolveram os programas exemplos e as suas funcionalidades, bem como o relatório.

O aluno Pedro Gonçalves desenvolveu o interpretador (linguagem declarada).

O aluno Tiago desenvolveu o compilador (linguagem compilada).

Não obstante da divisão das tarefas, os elementos do grupo sempre que necessário cooperaram entre si.

Deste modo, atribuímos a cada elemento a cotação de 20%.