

Relatório sobre Tratamento de Erros

Luca Campelli

Ricardo Almeida

Rodrigo Pedro Marques

Maio de 2017

Objetivos

A partir da definição sintática da linguagem “Nossa Linguagem” e do analisador sintático correspondente, deve-se mostrar a representação da árvore sintática e os exemplos com erros.

Desenvolvimento

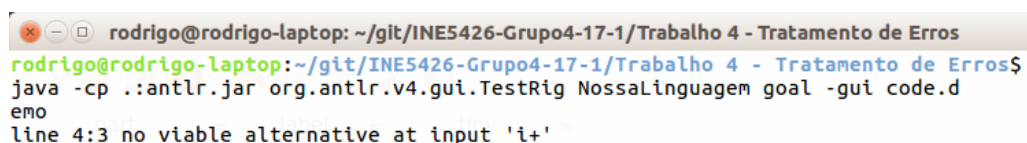
O desenvolvimento se deu por meio de estudos e pesquisas de como o *ANTLR* funciona e quais os comandos seriam necessários para gerar a árvore sintática com os exemplos de erros. Primeiramente, foi necessário compilar a linguagem e gerar o “.jar”, em seguida geramos o analisador sintático da “Nossa Linguagem”.

Para efeitos de apresentação, os códigos “code.demo”, “code2.demo” e “code3.demo” (onde este último abrange boa parte da definição da linguagem) foram alterados para que gerassem erros de sintaxe. É possível encontrar esses arquivos como apêndices e na pasta “codigos” acompanhada deste relatório.

Para gerar as árvores sintáticas de cada código, foram necessários os seguintes comandos:

```
java -cp ./antlr.jar org.antlr.v4.gui.TestRig NossaLinguagem goal -gui code.demo
java -cp ./antlr.jar org.antlr.v4.gui.TestRig NossaLinguagem goal -gui code2.demo
java -cp ./antlr.jar org.antlr.v4.gui.TestRig NossaLinguagem goal -gui code3.demo
```

Estes comandos fazem com o que o *ANTLR* gere o analisador sintático da linguagem, junto da árvore sintática do código dado como entrada. Ao executar este comando dando um arquivo como entrada com erros de sintaxe, ele irá apontar os erros. Nas figuras 1, 2 e 3 respectivamente.



```
rodrigo@rodrigo-laptop: ~/git/INE5426-Grupo4-17-1/Trabalho 4 - Tratamento de Erros
rodrigo@rodrigo-laptop:~/git/INE5426-Grupo4-17-1/Trabalho 4 - Tratamento de Erros$
java -cp ./antlr.jar org.antlr.v4.gui.TestRig NossaLinguagem goal -gui code.d
emo
line 4:3 no viable alternative at input 'i+'

```

Figura 1 – Code

```

rodrigo@rodrigo-laptop:~/git/INE5426-Grupo4-17-1/Trabalho 4 - Tratamento de Erros$
java -cp ./antlr.jar org.antlr.v4.gui.TestRig NossaLinguagem goal -gui code2.demo
line 10:1 mismatched input 'quant' expecting {'extends', '{'}

```

Figura 2 – Code2

Como é possível observar na figura 1, um erro é apontado na variável “i++” pois ela não foi declarada anteriormente e também não reconhece o comando.

Na figura 2 é possível observar outro erro devido ao esquecimento da abertura de chaves depois da declaração da classe.

```

rodrigo@rodrigo-laptop:~/git/INE5426-Grupo4-17-1/Trabalho 4 - Tratamento de Erro
s$ java -cp ./antlr.jar org.antlr.v4.gui.TestRig NossaLinguagem goal -gui code3.
demo
line 3:9 token recognition error at: '$'
line 4:1 missing ';' at '}'
line 10:10 extraneous input 's' expecting ';'
line 26:10 no viable alternative at input 'if(dois==3){System.'
line 26:10 no viable alternative at input 'System.'

```

Figura 3 – Code3

E por último, na figura 3 é possível observar vários erros que são apontados devido ao desconhecimento de símbolos, falta de ponto e vírgula e comandos incompletos.

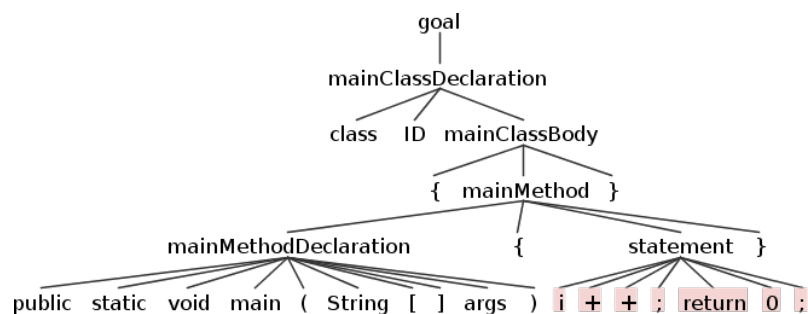


Figura 4 – Árvore sintática do code.demo

Na figura 4 encontra-se a árvore sintática gerada pelo *ANTLR* do código “code.demo”. Como é possível observar, ele apresentada em fundo vermelho onde está o erro de sintaxe apontado. As árvores sintáticas dos códigos “code2.demo” e “code3.demo” geradas, encontram-se na pasta *arvores* acompanhada deste relatório.

```
class ID
{
    public static void main (String[] args) {
        i++;
        return 0;
    }
}
```

```
class ID
{
    public static void main (String[] args) {
        return WORK();
    }
}

class ID2
//{
    quant = maybe;

    public boolean WORK(){
        int i;
        int j;
        char c;
        int[] a;

        double notDouble;

        if(quant){
            i = 40;
            j = 25;
        } else {
            c = 'a';
        }

        recur (quant == true || quant == false) ? () : j = 10;

        return j;
    }
}
```

```
class minhaClassePrincipal {
    public static void main (String[] meusArgumentos) {
        return $ algumNumero()
    }
}

class ID {
    boolean falso = true;
    public int algumNumero(){
        return 2s;
    }
}

/* Isso eh um header */
class teste3 extends ID {
    int um = 1; //meu numero
    char letraA = 'a';
    boolean verdadeiro = false;
    quantum myQuantum = maybe;

    public int blackHole(int numero, boolean sera){
        int dois = 2;
        int auxiliar = 999;
        while(dois < auxiliar){
            if(dois == 3){
                System.out.(letraA);
            }
            dois = dois + 1;
        }
        return dois;
    }
}
```