Trabalho Final INF1022 2024.2

Profs. Vitor Pinheiro e Edward Hermann

30 de outubro de 2024

1 Enunciado

O trabalho pode ser feito em dupla ou de forma individual. Neste trabalho deve ser desenvolvido um analisador sintático para a linguagem Matemágica. O analisador sintático deve ser capaz de compilar programas escritos utilizando a linguagem Matemágica para uma outra linguagem a sua escolha (como ilustrado na Figura 1). Ou seja, o analisador sintático recebe como entrada um programa na linguagem Matemágica e produz como saída um outro programa escrito em uma outra linguagem. A linguagem do programa de saída pode ser escolhida por você, ela pode ser qualquer linguagem a sua escolha. Por exemplo, mas não restrita a essas: C, C++, Java, Lua, Pyhton ou Assembly.

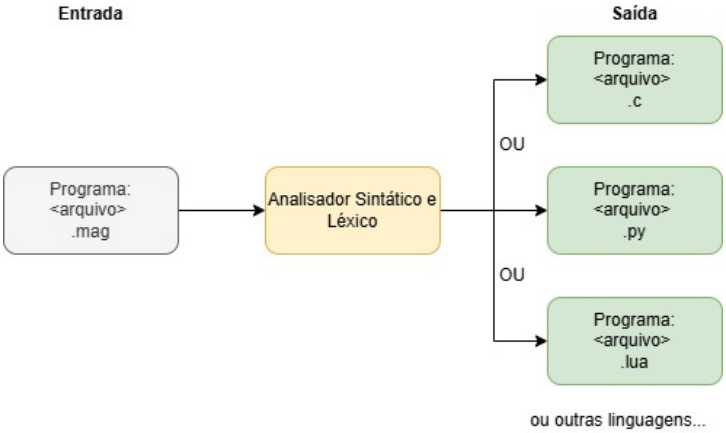

Figura 1: Analisador sintático para a linguagem Matemágica.

Para a implementação do analisador sintático, deve-se usar um gerador de analisador sintático que implemente o método LaLR(1) ou outro ascendente, por exemplo, Yacc/Lex,Bison/Flex (usa LaLR(1)), JavaCC (que usa o descendente LL(1)), etc.

1.1 Desenvolvimento

A sintaxe da linguagem *Matemágica* é dada pela gramática abaixo:

$$
\begin{array}{lcl}
programa & \longrightarrow & cmds \\
cmds & \longrightarrow & cmd\ \ cmds \mid\ cmd \\
cmd & \longrightarrow & atribuicao \mid impressao \mid operacao \mid repeticao \\
atribuicao & \longrightarrow & FACA\ \ var\ \ SER\ \ num. \\
impressao & \longrightarrow & MOSTRE\ \ var. \mid MOSTRE\ \ operacao. \\
operacao & \longrightarrow & SOME\ \ var\ \ COM\ \ var. \mid SOME\ \ var\ \ COM\ \ num. \mid \\
 & & SOME\ \ num\ \ COM\ \ num. \\
repeticao & \longrightarrow & REPITA\ \ num\ \ VEZES:\ \ cmds\ \ FIM
\end{array}
$$

- Todas as variáveis são do tipo inteiro e não negativo.
- *var* só pode ser formado por letras e precisa ter no mínimo uma letra.
- *num* representa um número.
- O terminal $SOME$ indica que deve ser feita uma soma.
- Perceba que o terminal . sinaliza o fim dos comandos *atribuicao*, *impressao* e *operacao*.
- O terminal FIM sinaliza o fim de uma repetição.

Além disso, a linguagem Matemágica será capaz de executar um loop que é definido pela regra:

$$repeticao \longrightarrow REPITA\ \ num\ \ VEZES:\ \ cmds\ \ FIM$$

A regra acima, que define o loop, diz para repetir *num* vezes a sequencia de comandos que esta em *cmds*.

A gramática acima não esta completa, neste trabalho você vai precisar definir algumas regras a mais para incluir algumas funcionalidades. Mais importante que isso, fique a vontade para alterar a gramática caso ache necessário para poder gerar a linguagem que esta sendo pedida. Por exemplo, você pode retirar recursão a esquerda, fatorar ou até alterar as regras acima. Desde que a linguagem final gerada seja a mesma.

Além disso, a gramática descrita acima deve ser complementada para que a linguagem Matemágica seja capaz de executar comandos do tipo:

- SE-ENTAO e SE-ENTAO-SENAO: Comandos de seleção do tipo SE-ENTÃO. Por exemplo: SE condicao ENTAO comando. Ou SE condicao ENTAO comando SENAO comando. A condição deve ser representada por um número. O número 0 significa FALSO e qualquer número diferente de zero significa VERDADEIRO.
- Multiplique A por B: Funcao para multiplicar A por B. Deve ser implementada no formatos: $MULTIPLIQUE\ \ A\ \ POR\ \ B$.
  Onde A e B podem ser variáveis ou números.

Alguns exemplos de programas em Matemágica são:

```
FACA x SER 10.
MOSTRE x.
```

```
FACA a SER 3.
FACA b SER 7.
SOME a COM b.
MOSTRE a.
```

```
FACA total SER 0.
REPITA 5 VEZES:
SOME total COM 2.
FIM
MOSTRE total.
```

```
FACA total SER 0.
REPITA 5 VEZES:
SOME total COM 2.
FIM
MOSTRE total.
```

```
FACA n SER 4.
SE n ENTAO
MOSTRE n.
FIM
```

```
FACA num SER 0.
SE num ENTAO
MOSTRE num.
SENAO
MOSTRE 10.
FIM
```

```
FACA x SER 3.
FACA y SER 5.
MULTIPLIQUE x POR y.
MOSTRE x.
```

```
FACA resultado SER 1.
REPITA 3 VEZES:
MULTIPLIQUE resultado POR 2.
FIM
MOSTRE resultado.
```

1.2 Observação

Você tem a permissão de expandir a gramática para permitir comandos adicionais ou demais operações que julgar interessantes e/ou necessárias.

Deixe explícitas as alterações realizadas na gramática descrita neste documento (documente no relatório do trabalho). Tanto as alterações obrigatórias (solicitadas neste trabalho) quanto alterações que você julgar interessantes.

2 Entrega

Você deve entregar um arquivo contendo seu relatório, no qual estarão descritos seu trabalho - o que foi implementado, como foi implementado, o que funciona, o que não funciona, quais os testes utilizados etc. - e quaisquer outras informações que você considere relevantes, como a maneira de executá-lo.

No seu relatório final deve estar contido a gramática final utilizada no trabalho. A gramática deve estar descrita de maneira similar ao que a gramatica inicial foi descrita neste documento. Deixe claro quais regras foram adicionadas na gramática descrita neste relatório para que seja possível permitir todas as funcionalidades solicitadas. Caso tenha implementado alguma funcionalidade adicional, ou seja, uma que não foi solicitada, por favor deixe explícito.

Além disso, entregue um .zip com os casos de teste utilizados.

Por fim, entregue o código do seu analisador bem como quaisquer outros arquivos/módulos auxiliares que tenha criado para a execução do trabalho.

Indique nome e matrícula de **ambos os componentes da dupla**, se for o caso.

Caso tenha arguições, a ordem das arguições será definida com base na ordem das entregas no EAD.

Dica

- Comece pequeno: garanta que seu analisador funcione para um fragmento da linguagem, depois vá incrementando.