

9795 Paradigma de Programação Imperativa e Orientada a Objetos

Edição 2019/1 - Ciência da Computação

Avaliação comparativa de duas linguagens de programação

O objetivo deste trabalho é a avaliação comparativa de duas linguagens de programação.

O trabalho é em equipe de até três pessoas. O compartilhamento de informações entre as equipes é permitido e aconselhado, mas o compartilhamento de código não é permitido. Trabalhos que tenham porções significativas de código iguais, ou copiados da internet, serão anulados.

Descrição

A avaliação comparativa deve ser feita utilizando os seguintes critérios:

- Facilidade de leitura e escrita
- Confiabilidade
- Custo de execução

Algumas características que afetam estes critérios e devem ser considerados na avaliação são: simplicidade, expressividade, tipos de dados e verificação de tipos.

Para dar subsídios para a comparação um avaliador passo a passo de expressões aritméticas deve ser escrito em duas linguagens de programação, Rust e uma das seguintes linguagens: C, C++, Go, Java, Python, JavaScript ou TypeScript. As duas versões devem ter testes automatizados.

As expressões são constituídas de números inteiros, operadores de adição (+), subtração (-), multiplicação (*) e divisão (/) e parênteses ((e)) e seguem a precedência e associativa comumente usada na matemática.

O programa deve ler as expressões da entrada padrão. Cada linha da entrada representa uma expressão. Para cada linha da entrada o programa deve dividi-la em tokens (números, operadores e abre e fecha parênteses) e construir uma árvore que representa a expressão (veja o algoritmo [Shunting Yard](#)). Em seguida, o programa deve exibir passo a passo a avaliação da expressão, executando sempre a expressão mais a esquerda (que pode ser avaliada). Por exemplo:

Unset

```
> (10 / 3 + 23) * (1 - 4)
(3 + 23) * (1 - 4)
26 * (1 - 4)
26 * -3
-78
```

Para ajudar na avaliação comparativa é sugerido escrever uma versão inicial (com os testes e o programa principal) apenas com as operações de adição e multiplicação. Cada versão inicial pode ser escrita por membros diferentes da equipe que depois completam a outra implementação.

A lista a seguir sugere algumas funções que podem ser implementadas nos programas:

- **lexer**: recebe uma string e devolve um arranjo de tokens. Exemplo

Unset

```
"31 * (4 + 10)" -> [31, "*", "(", 4, "+", 10, ")"]
```

-

parser: recebe uma lista de tokens e devolve uma árvore que representa a expressão. Exemplo

Unset

```
[31, "*", "(", 4, "+", 10, ")"] ->
```

```
      *
     / \
    31  +
       / \
      4  10
```

-

eval-step: recebe uma árvore que representa uma expressão e devolve uma árvore com um operador avaliado. Exemplo

Unset



-

`to-string`: recebe uma árvore que representa um expressão e devolve uma string que representa a mesma expressão. Exemplo

Unset



Entrega

A avaliação comparativa deve ser entregue impressa em papel A4. Os diretórios dos programas devem ser zipados e enviados para o e-mail malbarbo arroba gmail.com respeitando o limite de data e horário de entrega. Não envie arquivos compilados, apenas os arquivos de código fonte.

Avaliação

O trabalho será avaliado de acordo com os critérios:

- Fundamentação: a avaliação comparativa deve estar fundamenta nos aspectos estudados em sala.
- Completude: o programa deve estar completo, incluindo testes automatizados.
- Corretude: o programa deve funcionar de acordo com a descrição e deve passar em todos os testes.
- Entendimento: o aluno deve entender o programa que entregou e ser capaz de fazer alterações no código.

Casos de testes

Unset

> 1 + 3

4

> 1 + 2 * 3

7

7

> 4 / 2 + 7

9

9

> 1 + 2 + 3 * 4

15

3 + 3 * 4

15

> (1 + 2 + 3) * 4

24

6 * 4

24

> (10 / 3 + 23) * (1 - 4)

-78

(3 + 23) * (1 - 4)

-78

26 * (1 - 4)

-78

> ((1 + 3) * 8 + 1) / 3

11

(4 * 8 + 1) / 3

11

(32 + 1) / 3

11

33 / 3

11

11

> 58 - -8 * (58 + 31) - -14

784

58 - -8 * 89 - -14

770 - -14

784

> -71 * (-76 * 91 * (10 - 5 - -82) - -79)

42714523

-71 * (-6916 * (10 - 5 - -82) - -79)

-71 * (-6916 * (5 - -82) - -79)

-71 * (-6916 * 87 - -79)

-71 * (-601692 - -79)

-71 * -601613

42714523

> 10 * 20 + 3 * 7 + 2 * 3 + 10 / 3 * 4

$$200 + 3 * 7 + 2 * 3 + 10 / 3 * 4$$

$$200 + 21 + 2 * 3 + 10 / 3 * 4$$

$$221 + 2 * 3 + 10 / 3 * 4$$

$$221 + 6 + 10 / 3 * 4$$

$$221 + 6 + 3 * 4$$

$$221 + 6 + 12$$

$$221 + 18$$

$$239$$

$$> (-13 - -73) * (44 - -78 - 77 + 42 - -32)$$

$$60 * (44 - -78 - 77 + 42 - -32)$$

$$60 * (122 - 77 + 42 - -32)$$

$$60 * (45 + 42 - -32)$$

$$60 * (87 - -32)$$

$$60 * 119$$

$$7140$$

$$> -29 * 49 + 47 - 29 + 74 - -85 - -27 + 4 - 28$$

$$-1421 + 47 - 29 + 74 - -85 - -27 + 4 - 28$$

$$-1374 - 29 + 74 - -85 - -27 + 4 - 28$$

$$-1403 + 74 - -85 - -27 + 4 - 28$$

$$-1329 - -85 - -27 + 4 - 28$$

$$-1244 - -27 + 4 - 28$$

$$-1217 + 4 - 28$$

$$-1213 - 28$$

$$-1241$$

$$> -74 - -14 + 42 - -4 + -78 + -50 * -35 * -81 + -41$$

$$-60 + 42 - -4 + -78 + -50 * -35 * -81 + -41$$

$$-18 - -4 + -78 + -50 * -35 * -81 + -41$$

$$-14 + -78 + -50 * -35 * -81 + -41$$

$$-92 + -50 * -35 * -81 + -41$$

$$-92 + 1750 * -81 + -41$$

$$-92 + -141750 + -41$$

$$-141842 + -41$$

$$-141883$$

$$> 80 * -18 * (85 * (-46 + -71) - 12 + 26 - 59) + 84$$

$$-1440 * (85 * (-46 + -71) - 12 + 26 - 59) + 84$$

$$-1440 * (85 * -117 - 12 + 26 - 59) + 84$$

$$-1440 * (-9945 - 12 + 26 - 59) + 84$$

$$-1440 * (-9957 + 26 - 59) + 84$$

$$-1440 * (-9931 - 59) + 84$$

$$-1440 * -9990 + 84$$

$$14385600 + 84$$

$$14385684$$

$$> 25 + 38 + 88 + (-6 - -73) * (-83 + (53 + 97) * 14)$$

$$63 + 88 + (-6 - -73) * (-83 + (53 + 97) * 14)$$

$$151 + (-6 - -73) * (-83 + (53 + 97) * 14)$$

$$151 + 67 * (-83 + (53 + 97) * 14)$$

$$151 + 67 * (-83 + 150 * 14)$$

$$151 + 67 * (-83 + 2100)$$

$$151 + 67 * 2017$$

$$151 + 135139$$

$$135290$$

$$> (84 - 90) * (-8 - 75 + -83 * (56 - -77) + 4 + -94)$$

$$-6 * (-8 - 75 + -83 * (56 - -77) + 4 + -94)$$

$$-6 * (-83 + -83 * (56 - -77) + 4 + -94)$$

$$-6 * (-83 + -83 * 133 + 4 + -94)$$

$$-6 * (-83 + -11039 + 4 + -94)$$

$$-6 * (-11122 + 4 + -94)$$

$$-6 * (-11118 + -94)$$

$$-6 * -11212$$

$$67272$$

$$> (54 - -8 - -35 + -68 - -90) * -39 + -43 + -91 * -30$$

$$(62 - -35 + -68 - -90) * -39 + -43 + -91 * -30$$

$$(97 + -68 - -90) * -39 + -43 + -91 * -30$$

$$(29 - -90) * -39 + -43 + -91 * -30$$

$$119 * -39 + -43 + -91 * -30$$

$$-4641 + -43 + -91 * -30$$

$$-4684 + -91 * -30$$

$$-4684 + 2730$$

$$-1954$$

$$> -13 - -74 + (66 + -57) * -93 * -9 * 77 + 79 - 66 + -53$$

$$61 + (66 + -57) * -93 * -9 * 77 + 79 - 66 + -53$$

$$61 + 9 * -93 * -9 * 77 + 79 - 66 + -53$$

$$61 + -837 * -9 * 77 + 79 - 66 + -53$$

$$61 + 7533 * 77 + 79 - 66 + -53$$

$$61 + 580041 + 79 - 66 + -53$$

$$580102 + 79 - 66 + -53$$

$$580181 - 66 + -53$$

$$580115 + -53$$

580062

> $(-72 - 50 * -74 + -45) * 92 * 21 * 5 * (-13 - 66 - 18)$

$(-72 - -3700 + -45) * 92 * 21 * 5 * (-13 - 66 - 18)$

$(3628 + -45) * 92 * 21 * 5 * (-13 - 66 - 18)$

$3583 * 92 * 21 * 5 * (-13 - 66 - 18)$

$329636 * 21 * 5 * (-13 - 66 - 18)$

$6922356 * 5 * (-13 - 66 - 18)$

$34611780 * (-13 - 66 - 18)$

$34611780 * (-79 - 18)$

$34611780 * -97$

-3357342660

> $-7 - -37 * (90 + 70) - 30 - -44 + -32 - 56 - -48 - -78$

$-7 - -37 * 160 - 30 - -44 + -32 - 56 - -48 - -78$

$-7 - -5920 - 30 - -44 + -32 - 56 - -48 - -78$

$5913 - 30 - -44 + -32 - 56 - -48 - -78$

$5883 - -44 + -32 - 56 - -48 - -78$

$5927 + -32 - 56 - -48 - -78$

$5895 - 56 - -48 - -78$

$5839 - -48 - -78$

$5887 - -78$

5965

> $65 * -83 - -3 + -20 + 24 - 85 * (-24 + -32) * (61 - 20)$

$-5395 - -3 + -20 + 24 - 85 * (-24 + -32) * (61 - 20)$

$-5392 + -20 + 24 - 85 * (-24 + -32) * (61 - 20)$

$-5412 + 24 - 85 * (-24 + -32) * (61 - 20)$

$-5388 - 85 * (-24 + -32) * (61 - 20)$

$-5388 - 85 * -56 * (61 - 20)$

$-5388 - -4760 * (61 - 20)$

$-5388 - -4760 * 41$

$-5388 - -195160$

189772

> $55 * 48 * -44 - -32 + 1 * -80 * -94 - 74 * -53 + -30 + -61$

$2640 * -44 - -32 + 1 * -80 * -94 - 74 * -53 + -30 + -61$

$-116160 - -32 + 1 * -80 * -94 - 74 * -53 + -30 + -61$

$-116128 + 1 * -80 * -94 - 74 * -53 + -30 + -61$

$-116128 + -80 * -94 - 74 * -53 + -30 + -61$

$-116128 + 7520 - 74 * -53 + -30 + -61$

$-108608 - 74 * -53 + -30 + -61$

$-108608 - -3922 + -30 + -61$

$-104686 + -30 + -61$
 $-104716 + -61$
 -104777
 $> -82 * (25 + 62 + 3) - -72 + -65 * -32 * (77 + 12) - -95 + 51$
 $-82 * (87 + 3) - -72 + -65 * -32 * (77 + 12) - -95 + 51$
 $-82 * 90 - -72 + -65 * -32 * (77 + 12) - -95 + 51$
 $-7380 - -72 + -65 * -32 * (77 + 12) - -95 + 51$
 $-7308 + -65 * -32 * (77 + 12) - -95 + 51$
 $-7308 + 2080 * (77 + 12) - -95 + 51$
 $-7308 + 2080 * 89 - -95 + 51$
 $-7308 + 185120 - -95 + 51$
 $177812 - -95 + 51$
 $177907 + 51$
 177958
 $> (2 - 65 - (-24 + -97) * -5 * -61) * (-41 + 85 * 9 * -92 * (75 - 18))$
 $(-63 - (-24 + -97) * -5 * -61) * (-41 + 85 * 9 * -92 * (75 - 18))$
 $(-63 - -121 * -5 * -61) * (-41 + 85 * 9 * -92 * (75 - 18))$
 $(-63 - 605 * -61) * (-41 + 85 * 9 * -92 * (75 - 18))$
 $(-63 - -36905) * (-41 + 85 * 9 * -92 * (75 - 18))$
 $36842 * (-41 + 85 * 9 * -92 * (75 - 18))$
 $36842 * (-41 + 765 * -92 * (75 - 18))$
 $36842 * (-41 + -70380 * (75 - 18))$
 $36842 * (-41 + -70380 * 57)$
 $36842 * (-41 + -4011660)$
 $36842 * -4011701$
 -147799088242
 $> -20 + -51 + 20 + -68 * -11 + -35 * -14 - 95 - 32 + -52 * -23$
 $- -90 * -42$
 $-71 + 20 + -68 * -11 + -35 * -14 - 95 - 32 + -52 * -23 - -90 * -42$
 $-51 + -68 * -11 + -35 * -14 - 95 - 32 + -52 * -23 - -90 * -42$
 $-51 + 748 + -35 * -14 - 95 - 32 + -52 * -23 - -90 * -42$
 $697 + -35 * -14 - 95 - 32 + -52 * -23 - -90 * -42$
 $697 + 490 - 95 - 32 + -52 * -23 - -90 * -42$
 $1187 - 95 - 32 + -52 * -23 - -90 * -42$
 $1092 - 32 + -52 * -23 - -90 * -42$
 $1060 + -52 * -23 - -90 * -42$

$$1060 + 1196 - -90 * -42$$

$$2256 - -90 * -42$$

$$2256 - 3780$$

$$-1524$$