



Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Departamento de Sistemas e Computação

Graduação em Ciência da Computação

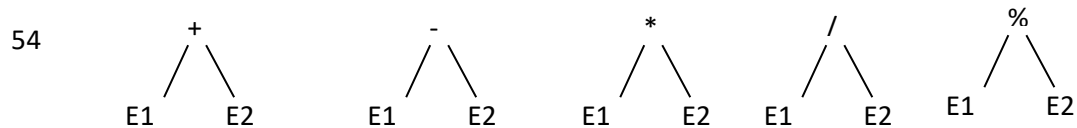
Exercício sobre BT

Objetivo: Praticar a implementação de um uso de árvore binária. Para este exercício você vai precisar de uma implementação de BT.

Uma das aplicações de árvore binária é a possibilidade de representação de expressões aritméticas. Imagine as seguintes expressões aritméticas

- Um número inteiro é uma expressão aritmética
- $E + E$ (soma)
- $E * E$ (multiplicação)
- $E - E$ (subtração)
- E / E (divisão inteira)
- $E \% E$ (resto da divisão)

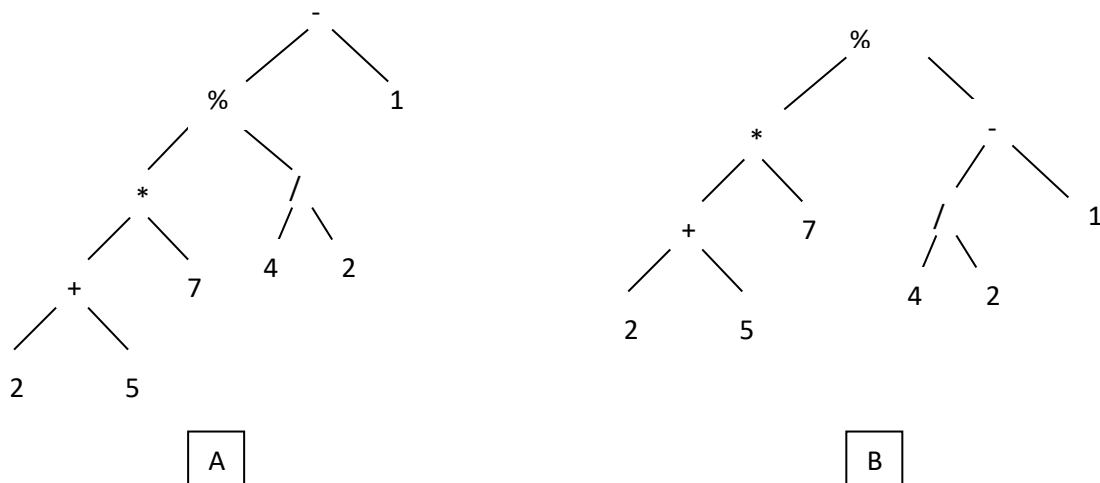
Cada uma dessas expressões é representado pelas seguintes estruturas:



Note que um número é uma árvore contendo apenas um nó (folha). Com essa representação expressões complexas podem ser representadas de forma bem intuitiva. Por exemplo, as expressões:

- A: $(2+5)*7 \% (4/2) - 1$
- B: $(2+5)*7 \% ((4/2) - 1)$

Seriam representadas por:

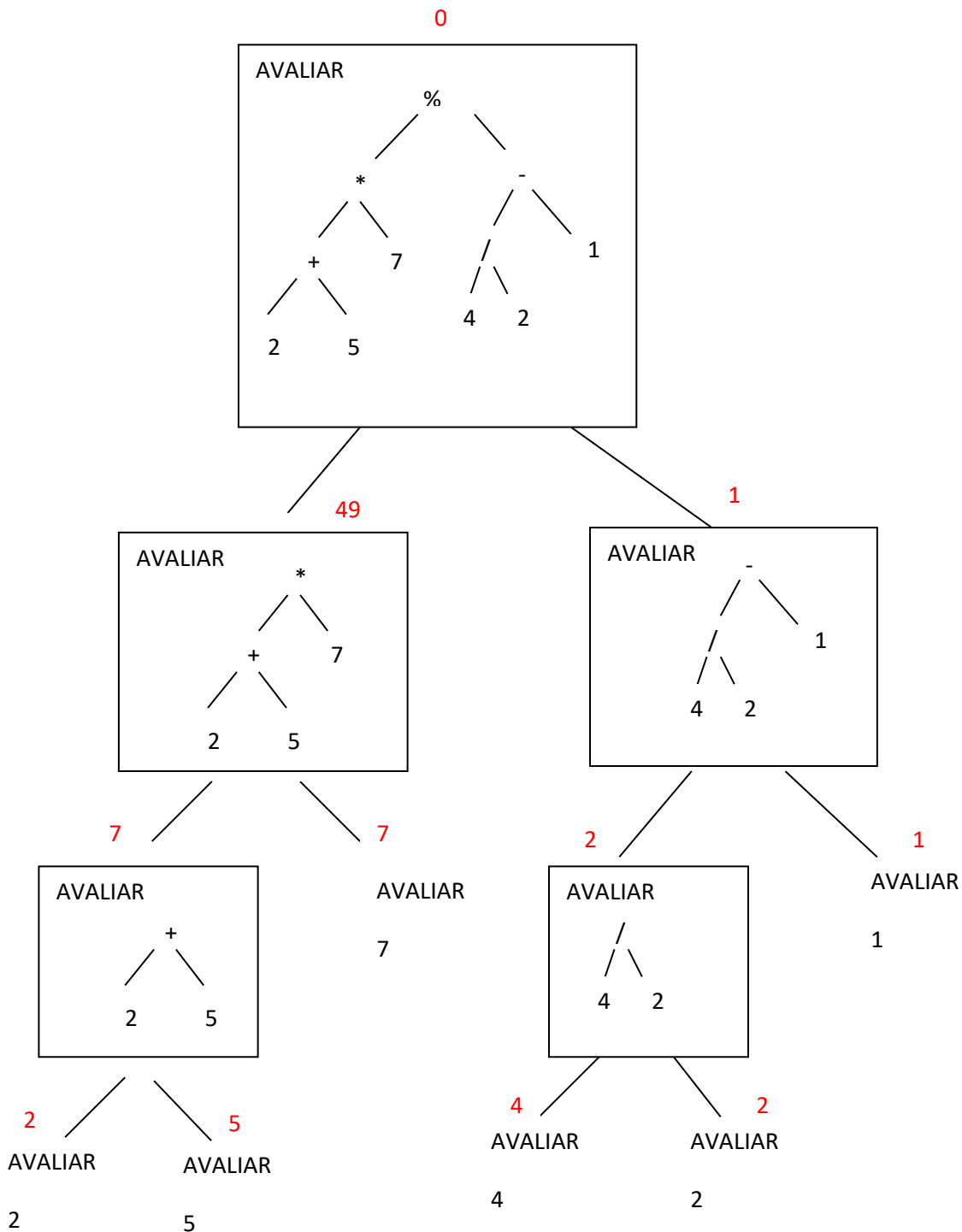


Para representar melhor em termos de classes você pode supor a existência de uma hierarquia de classes, cada uma representando uma expressão aritmética, que por sua vez é um nó da árvore, mas que pode ser avaliado. Nessa hierarquia, qualquer forma de expressão aritmética pode ser avaliada. A classe `ArithmeticExpression.java` tem o propósito de ser a expressão aritmética mais genérica possível. Delas os outros tipos podem ser criados.

A avaliação de uma expressão aritmética representada em uma árvore pode ser feita de forma **recursiva**, onde um método recebe um nó e analisa se ele contém um operador ou um número. No caso de conter um operador o método deve avaliar o filho à esquerda, o filho à direita e em seguida operar o resultado. Por exemplo, avaliar a árvore

54

Produziria apenas o valor 54. Avaliar a árvore B produziria (siga de cima para baixo para ver a quebra recursiva da árvore, depois de baixo para cima nos valores vermelhos que representam as respostas das avaliações das expressões desde as folhas para a raiz):



Baixe os arquivos referente ao roteiro de hoje e implemente sua solução. Procure entender a estrutura das classes através dos comentários e termine a implementação `BTEExpressionImpl`. Note que sua solução pode ou não envolver a extensão do modelo de classes dado inicialmente. Isso depende de seu design.

Voce pode inclusive testar sua implementação instanciando nós (das folhas até a raiz) e fazendo a ligação correta entre eles. Depois disso é só instanciar uma `BTEExpressionImpl` contendo a raiz como root. A avaliação de uma `BTEExpressionImpl` é a avaliação de seu root.

Pense também como você implementaria um método que imprime, por exemplo, a expressão aritmética “soma de A e B” de duas formas:

- Usando notação infixa, ex: $A + B$
- Usando notação prefixa, ex: $+ A B$