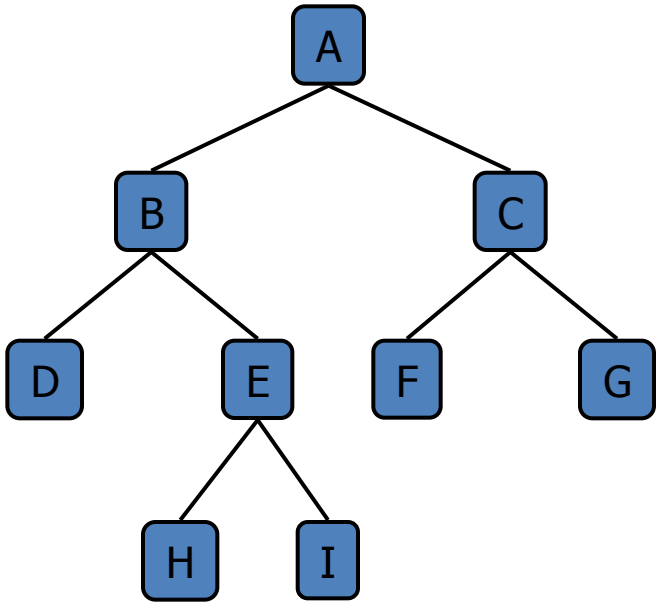
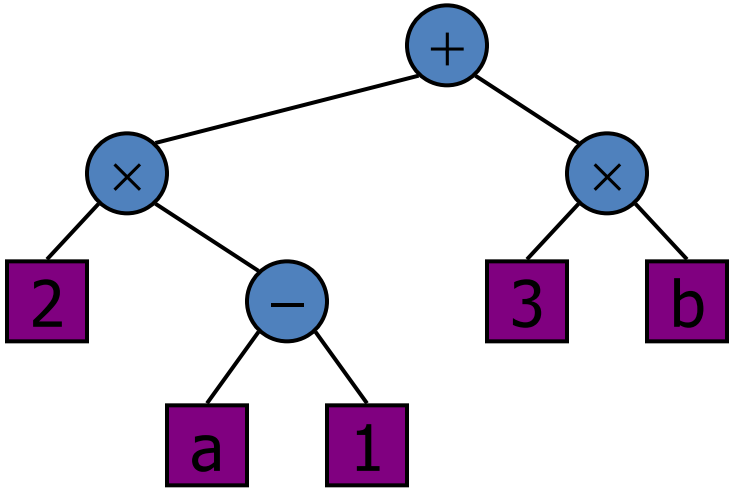
Árvore Binária

- Árvore com as seguintes propriedades:
  - Cada nó interno tem no máximo dois filhos
  - Os filhos de um nó formam um par ordenado (filho da esquerda, filho da direita)
- Árvore binária própria
  - Cada nó tem zero ou dois filhos
- Aplicações
  - Expressões aritméticas
  - Árvores de decisão



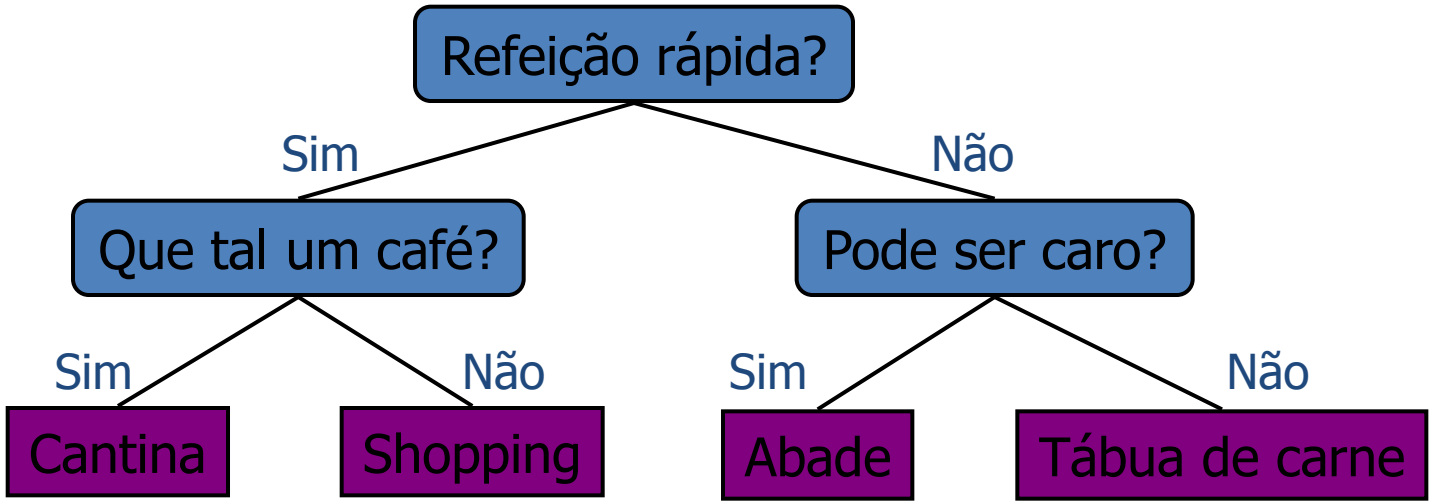
Árvore de Expressões Aritméticas

- Árvore associada a uma expressão aritmética
  - Nós internos: operadores
  - Nós externos: operandos
- Exemplo: (2 x (a–1) + (3 x b))

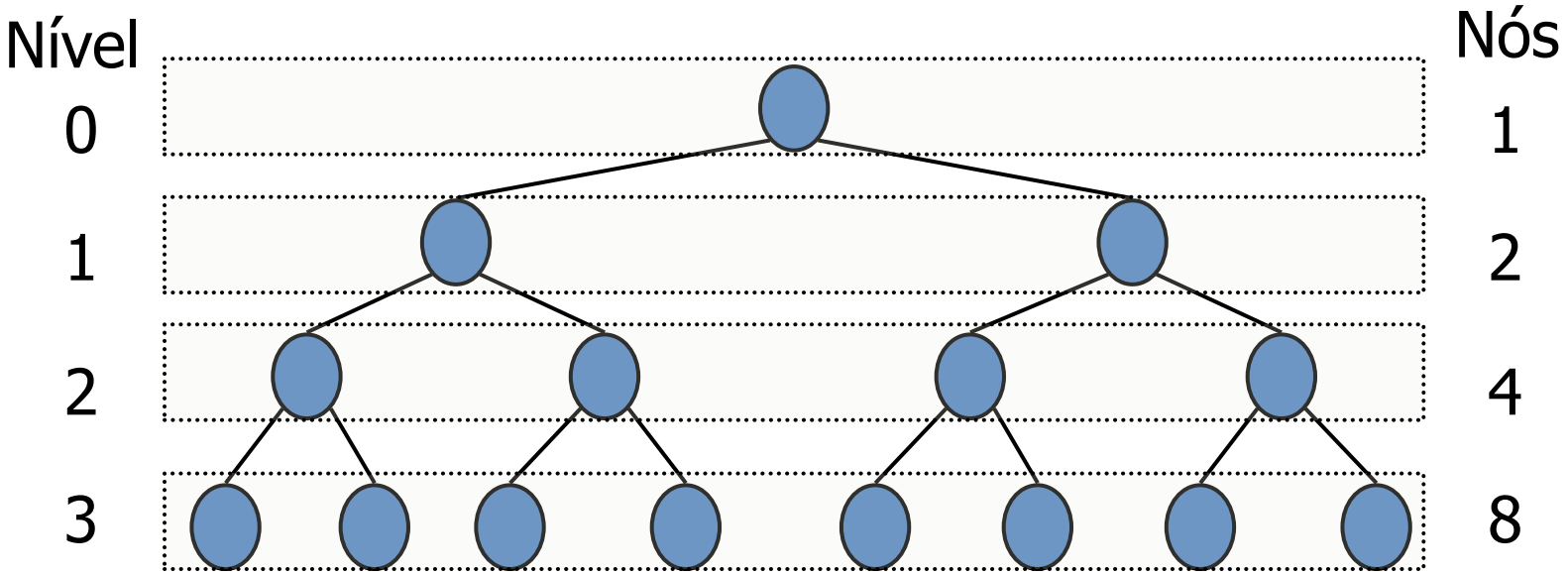


Árvore de Decisão

- Árvore associada a um processo de decisão
  - Nós internos: questões com respostas sim/não
  - Nós externos: decisões
- Exemplo: Onde jantar?



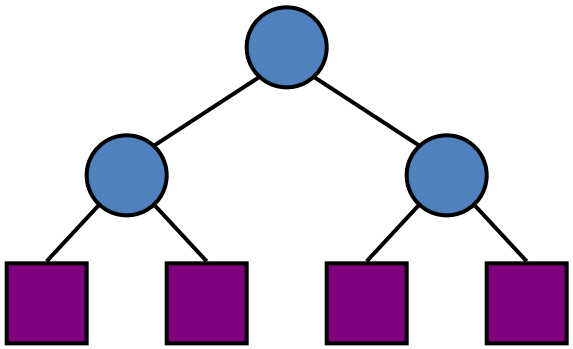
Propriedades da Árvore Binária

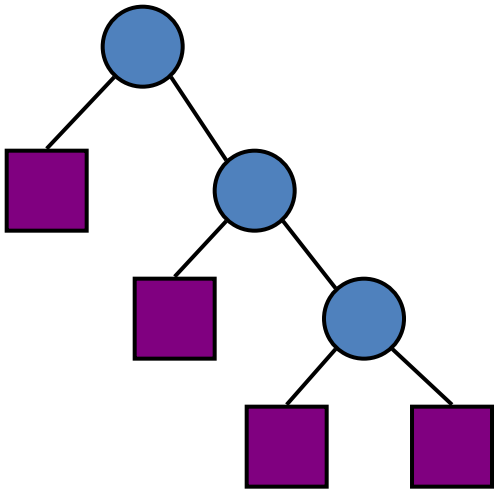


- Número máximo de nós em um nível h é 2^h
- Número total de nós é, no máximo, $2^{h+1} -1$

Propriedades da Árvore Binária Própria

- Notação
  - n = número de nós
  - e = nós externos
  - i = nós internos
  - h = altura (height)





- Propriedades
  - $e = i + 1$
  - $n = 2e - 1$
  - $h + 1 \leq e \leq 2^h$
  - $h \leq i \leq 2^h - 1$
  - $h \geq \log_2 (n + 1) - 1$
  - $h \leq (n - 1)/2$

AB – Operações de Acesso

- Estende o TAD Árvore
- Métodos de acesso
  - Nó leftChild(Nó v)
    - retorna o filho esquerdo de um nó
  - Nó rightChild(Nó v)
    - retorna o filho direito de um nó
  - Nó sibling(Nó v)
    - Retorna o irmão de um nó

AB – Caminhamentos

- Prefixado

```
Algoritmo binaryPreOrder(T, v)
    execute a ação para o nó v
    se T.isInternal(v) então
        binaryPreOrder(T, T.leftChild(v))
        binaryPreOrder(T, T.rightChild(v))
```

- Pós-fixado

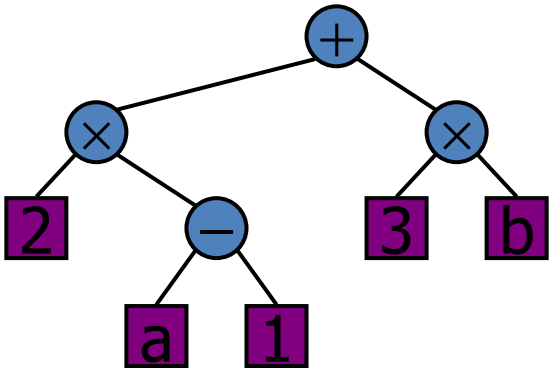
```
Algoritmo binaryPostOrder(T, v)
    se T.isInternal(v) então
        binaryPostOrder(T, T.leftChild(v))
        binaryPostOrder(T, T.rightChild(v))
    execute a ação para o nó v
```

AB – Caminhamentos

- Interfixado

  Expressão Aritmética:

  ((2 × (a - 1)) + (3 × b))

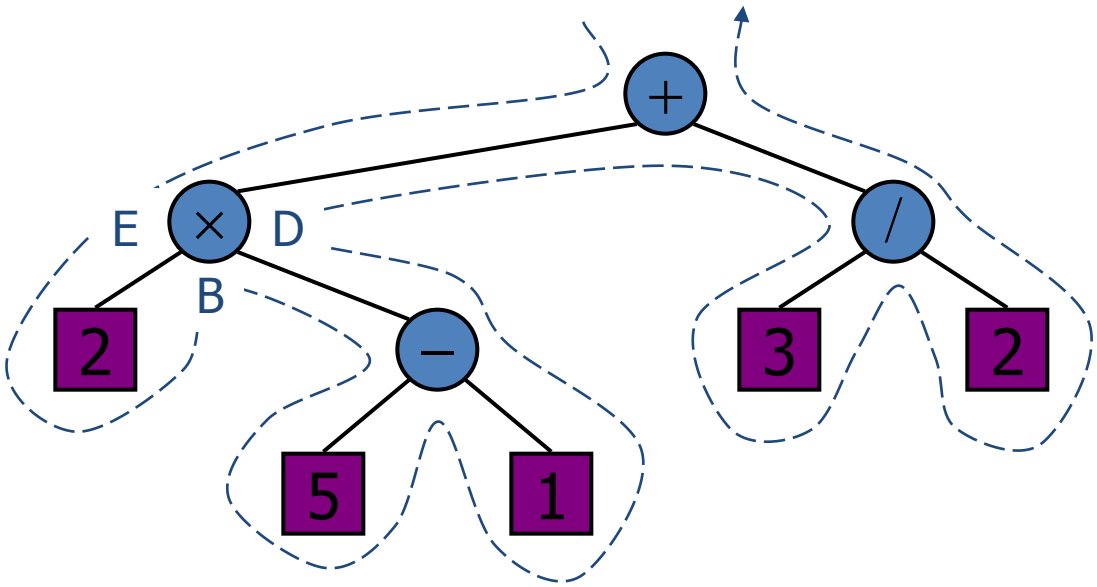


Algoritmo inOrder(T, v)

se T.isInternal(v) então inOrder(T, T.leftChild(v))
execute a ação para o nó v
se T.isInternal(v) então inOrder(T, T.rightChild(v))

AB – Caminhamento de Euler

- Caminho que visita cada aresta exactamente uma vez
  - pela esquerda (prefixado – pré ordem)
  - por baixo (interfixado – em ordem)
  - pela direita (pós-fixado – pós ordem)



AB – Caminhamento de Euler

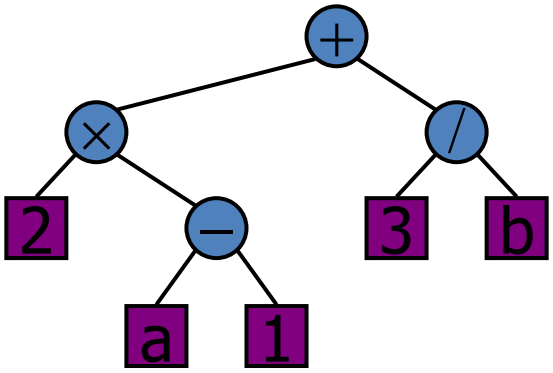
- Cada nó *v* da árvore é “visitado” três vezes pelo caminhamento de Euler. Cada “visita” pode corresponder a uma “ação” a ser tomada pelo algoritmo, como segue:
  - “ação pela esquerda” (antes do caminhamento sobre a subárvore da esquerda de *v*);
  - “ação por baixo” (entre o caminhamento entre as duas subárvores de *v*); e
  - “ação pela direita” (depois do caminhamento sobre a subárvore da direita de *v*).

AB – Caminhamento de Euler

- Interfixado

  Expressão Aritmética:

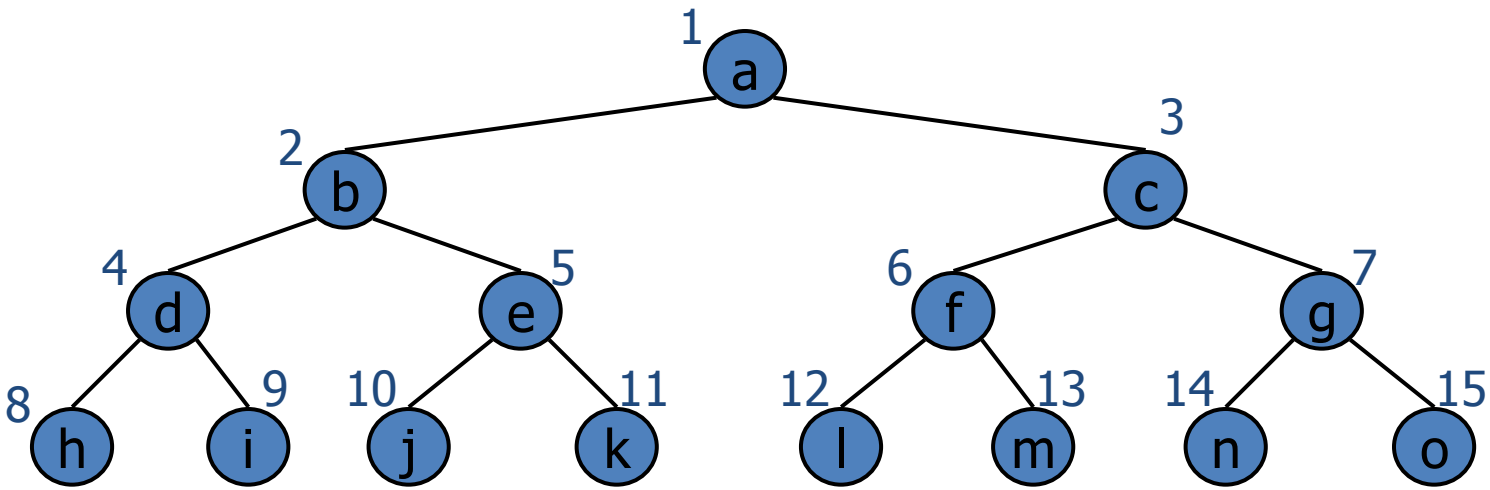
  ((2 × (a - 1)) + (3 / b))

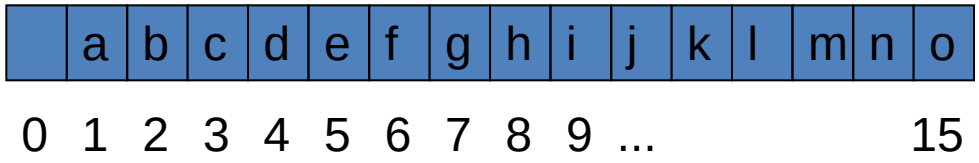



```
Algoritmo printExpression(T, v)
    se T.isInternal(v) então //ação "pela esquerda"
       print ( "(" );
    se T.hasLeft(v) então
         printExpression(T, T.leftChild(v));
    print ( v.element() ); //ação "por baixo"
    se T.hasRight(v) então
         printExpression(T, T.rightChild(v));
    se T.isInternal(v) então
         print ( ")" ); //ação "pela direita"
```

Estruturas de Dados para Árvores

- Árvore binária baseada em arranjo





Estrutura Encadeada para Árvores Binárias

- Um nó (BTNode) armazena referências para:
  - Nó pai
  - Elemento
  - Filho da esquerda
  - Filho da direita