

ESD|2

Estrutura de
Dados

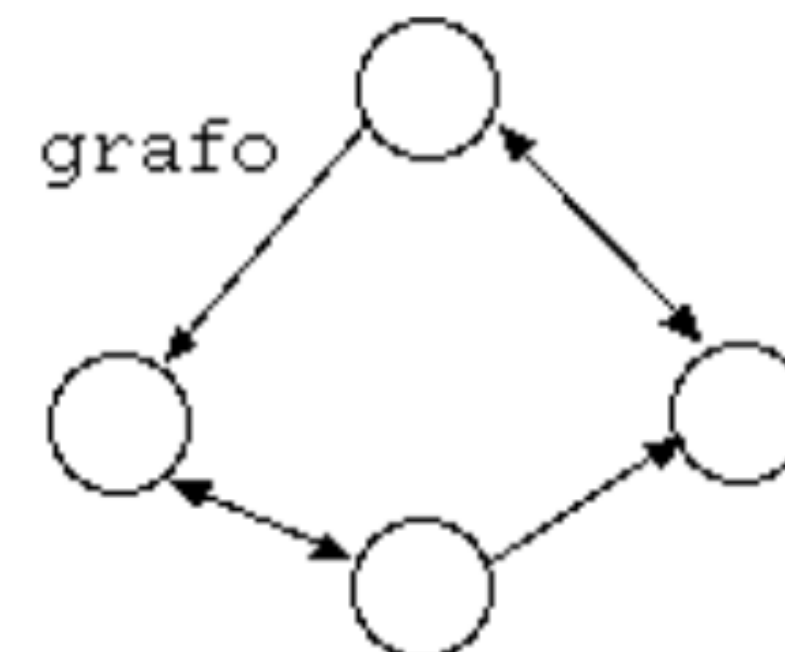
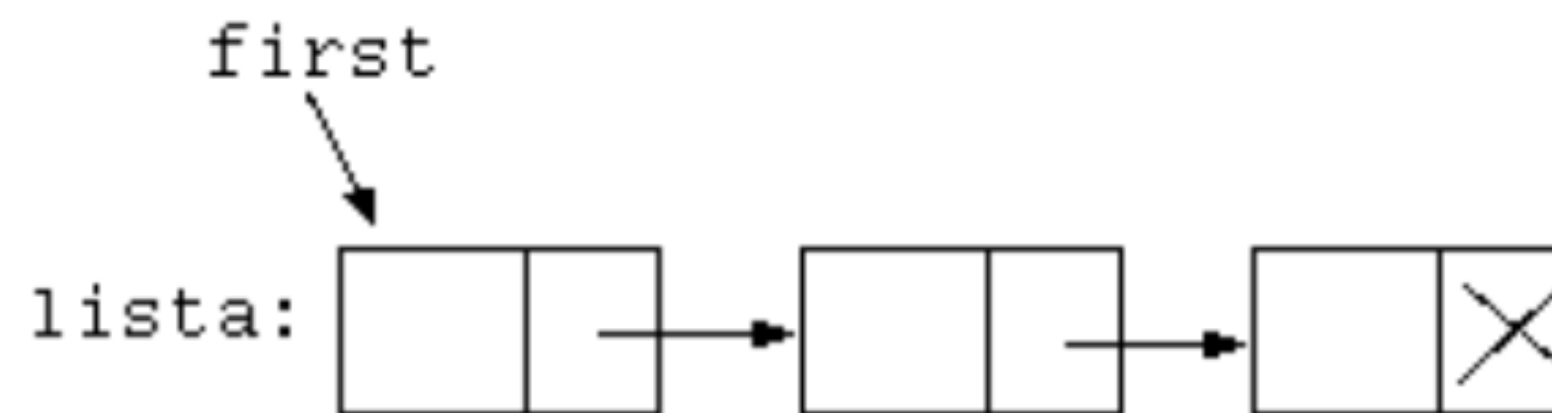
Árvores e Árvore Binária de Busca

Prof. Fernando Sambinelli

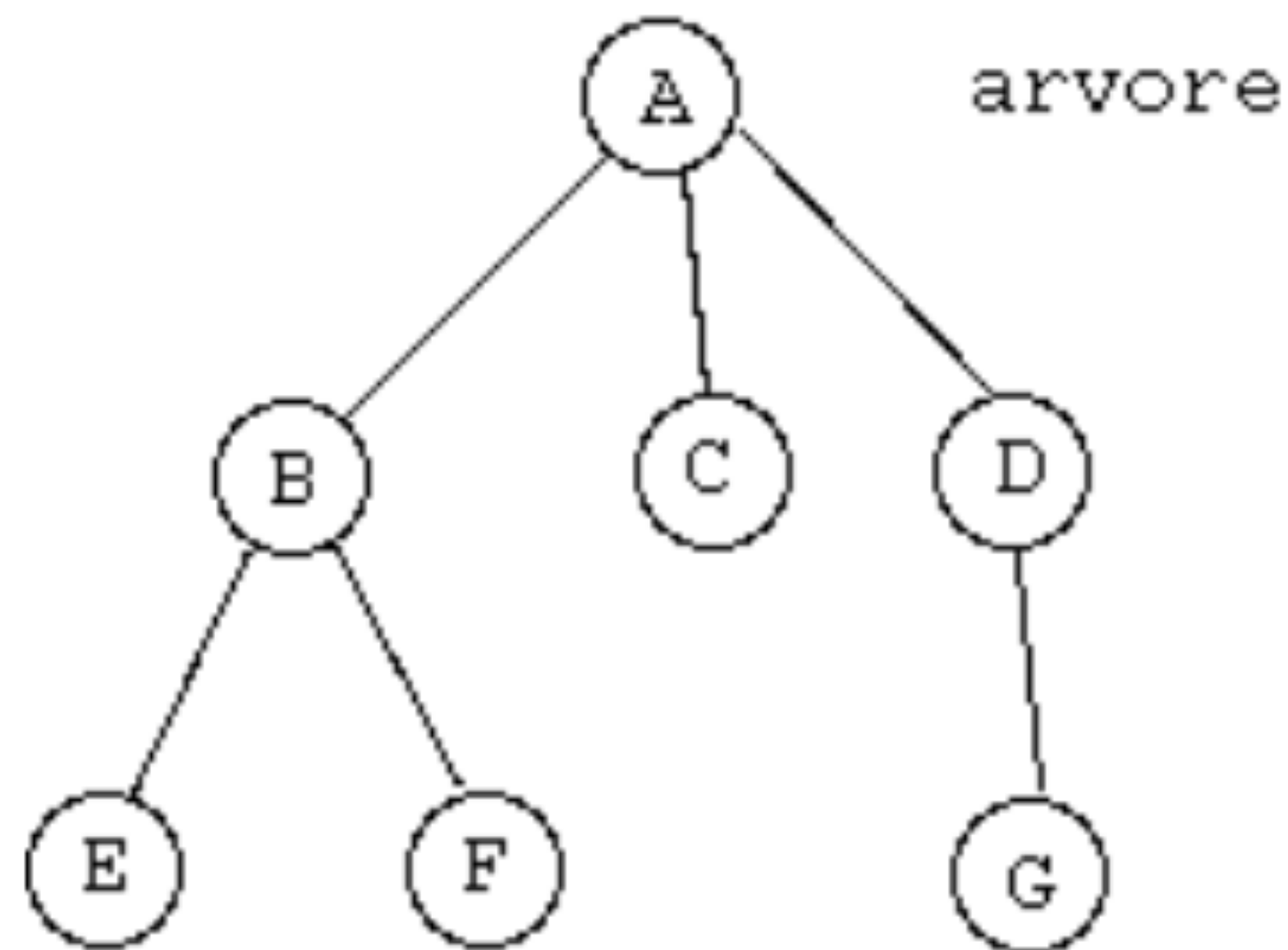
#01 - Árvores

- Definição
- Conceitos e Terminologias
- Representações

- Uma **lista** é um exemplo de uma estrutura de dados **linear**, pois cada elemento tem:
 - um predecessor único, exceto o primeiro elemento da lista
 - tem um sucessor único, exceto o último elemento da lista
- As pilhas e filas são exemplos de estruturas de dados lineares
- Um **grafo** é uma estrutura de dados **não-linear**, pois os seus elementos, designados por nós, podem ter mais de um predecessor ou mais de um sucessor.

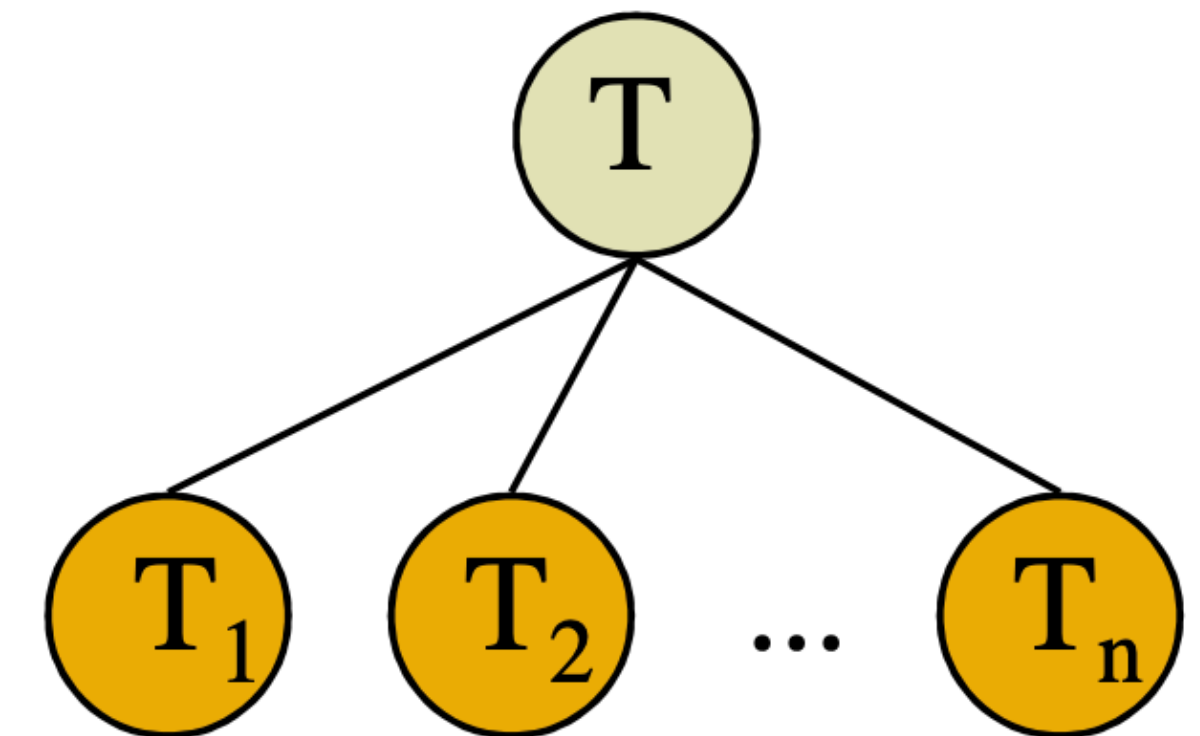


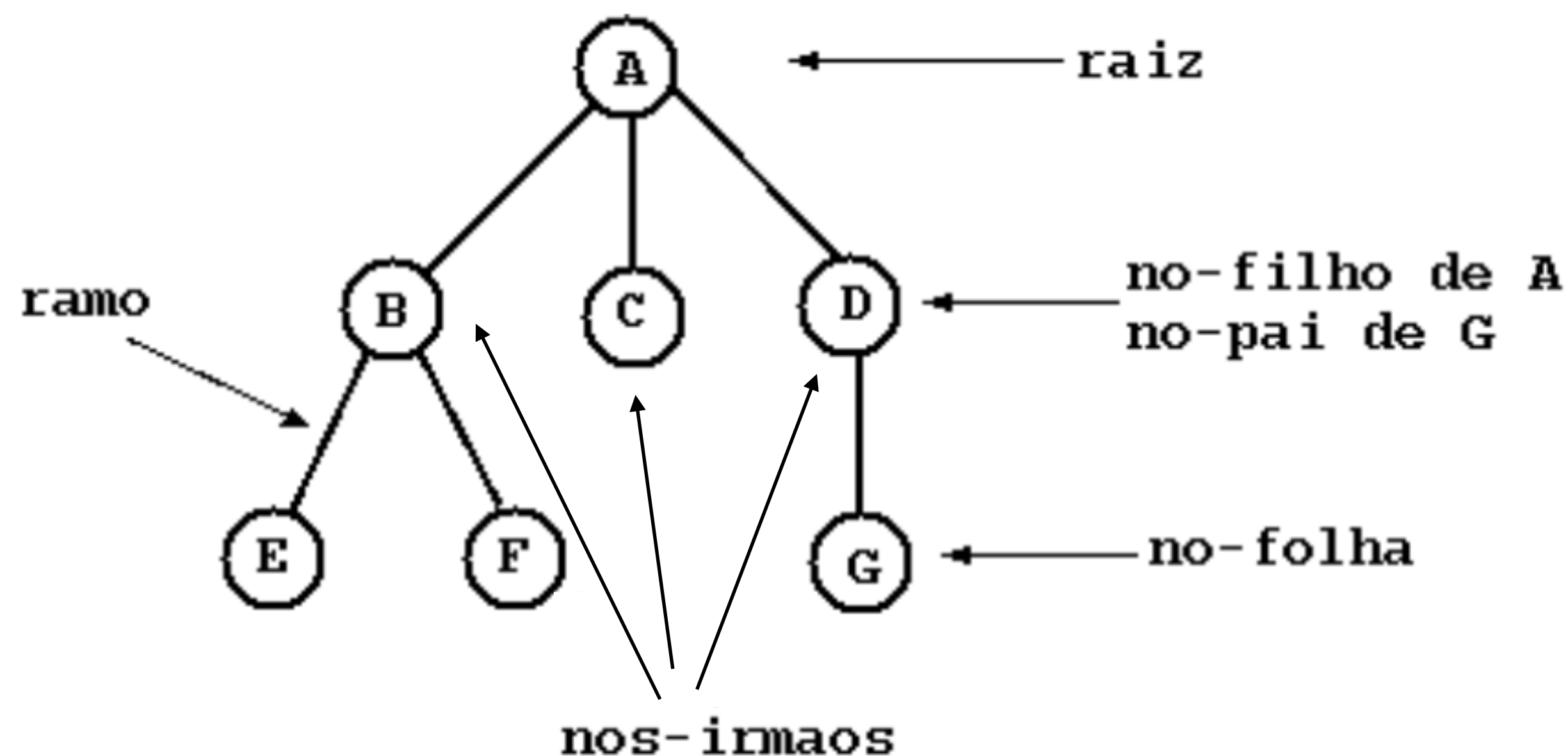
- As **árvores** são um caso especial de grafos, em que cada elemento (nó) tem zero ou mais sucessores, mas tem apenas um predecessor, excepto o primeiro nó (a raiz da árvore)
- São estruturas adequadas para representar informação organizada em hierarquias (ex: estrut. de arquivos, problemas de buscas, problemas de Inteligência Artificial, problemas de compiladores, etc)



Definição Formal de Uma Árvore

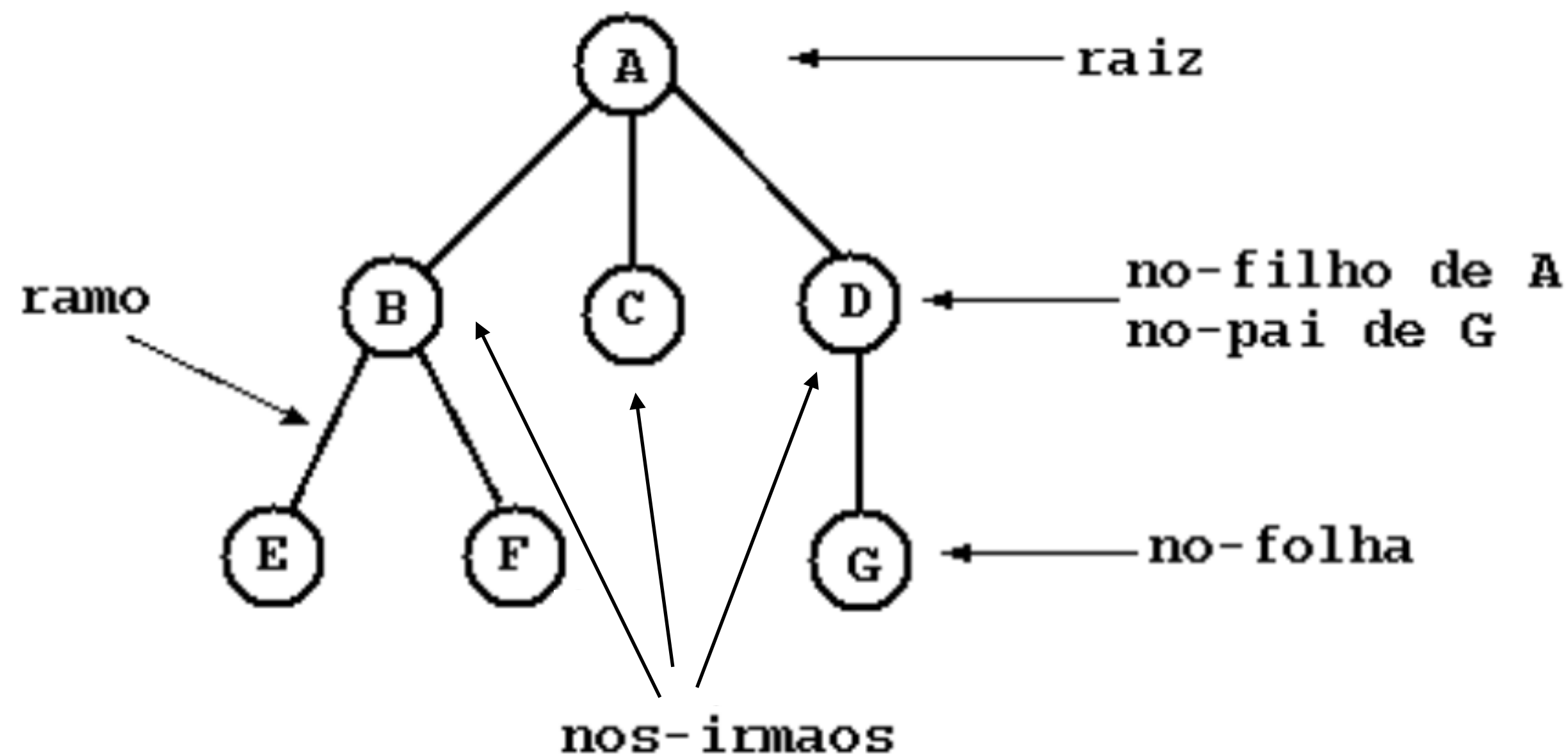
- Árvore T é um conjunto finito de elementos denominados nós ou vértices, tais que:
 - $T = \emptyset$, a árvore é dita vazia;
 - Existe um nó especial, denominado **raiz** de T ; os nós restantes constituem um único conjunto vazio ou são divididos em $m \geq 1$ conjuntos **disjuntos** não vazios (T_1, T_2, \dots, T_n) , cada qual, por sua vez é uma árvore;
 - T_1, T_2, \dots, T_n são chamadas sub-árvores de T ;
 - Um nó sem subárvore é denominado **nó-folha** ou simplesmente **folha**.





- o predecessor (único) de um nó, chama-se **nó-pai**
- os seus sucessores são os **nós-filho**
- o **grau** de um nó é o número sub-árvores (ou nós-filho) que descendem desse nó (A tem grau 3, B tem grau 2)
- um nó-folha não tem filhos, tem grau 0
- o grau da árvore => grau do nó raiz
- um nó-raiz não tem pai
- nós com o mesmo pai são ditos **irmãos**
- os arcos que ligam os nós, chamam-se **ramos**

Conceitos e Terminologias (2)



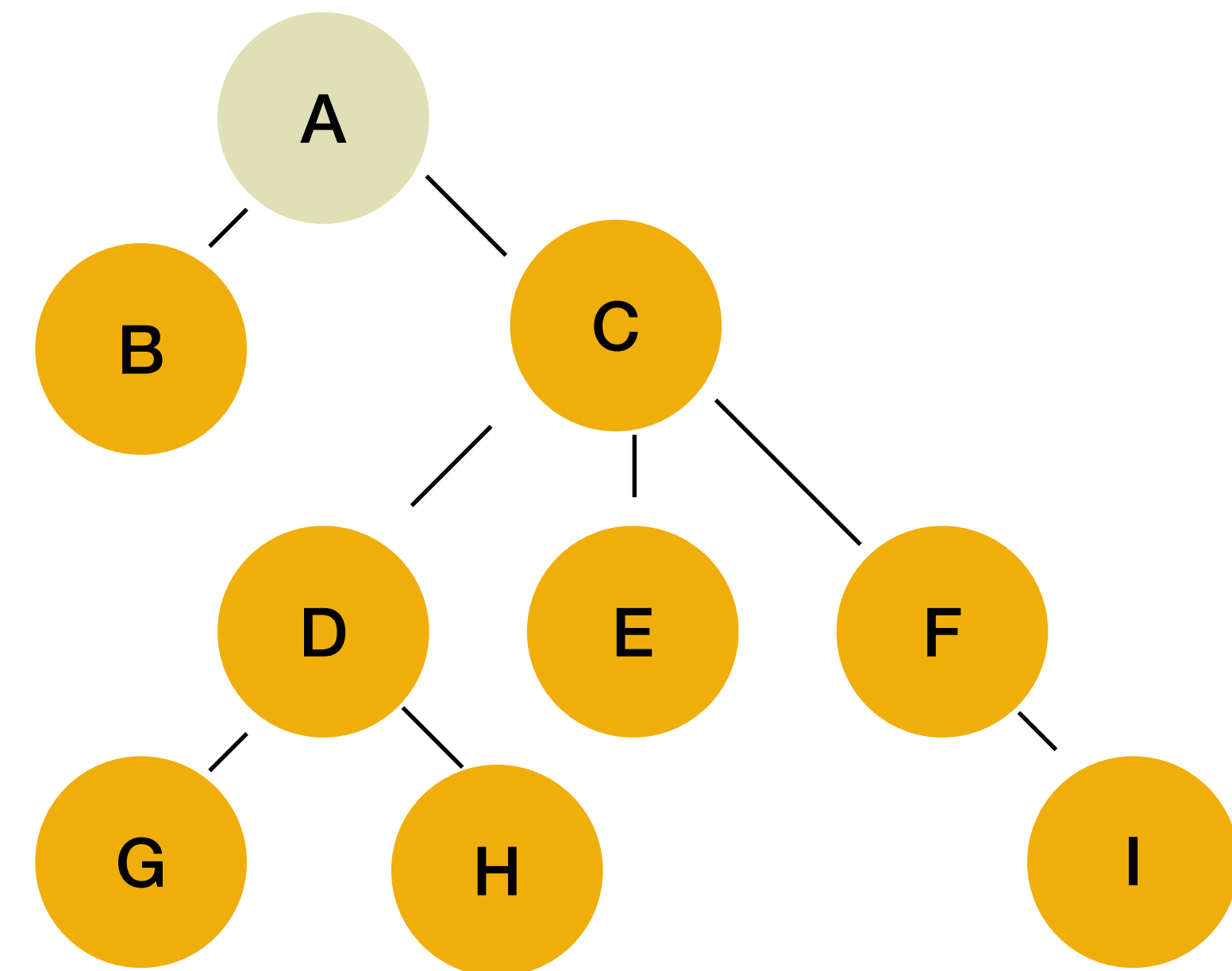
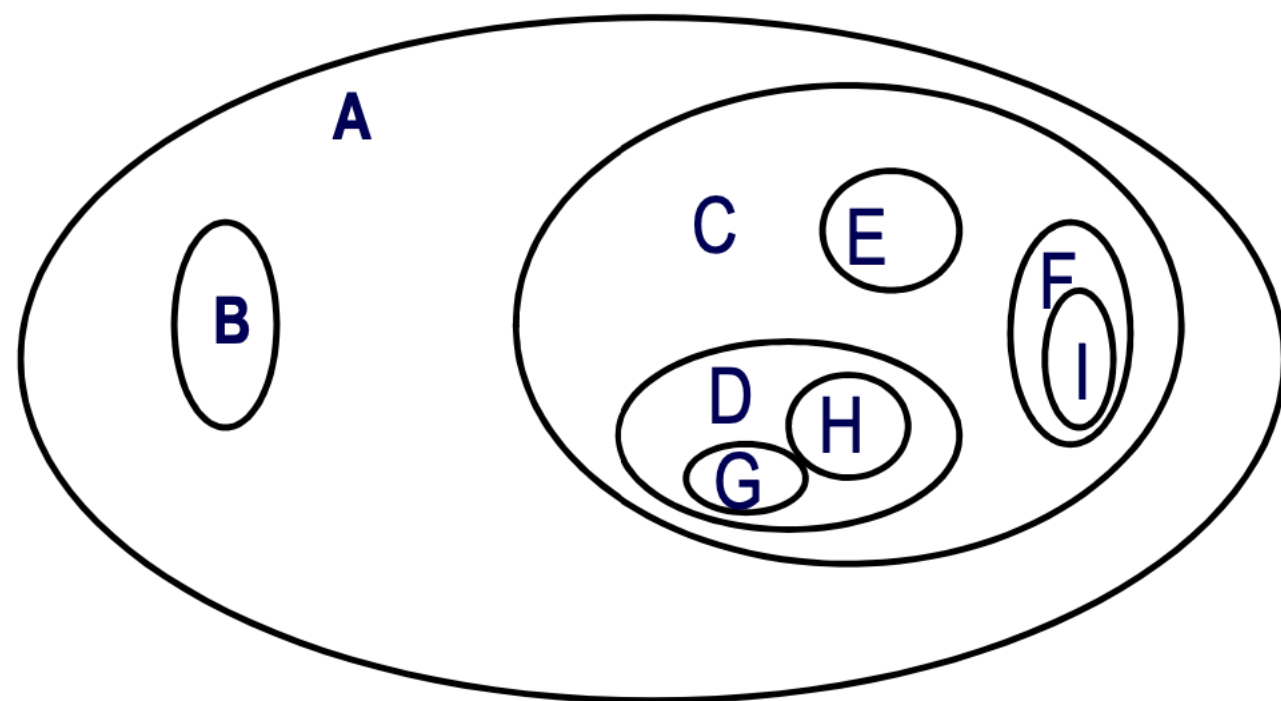
- chama-se **caminho** a uma sequência de ramos entre dois nós:
 - Uma propriedade importante de uma árvore é que existe um e apenas um caminho entre dois quaisquer nós de uma árvore
- o **comprimento** de um caminho é o número de ramos nele contido
- a **profundidade** de um nó n é o comprimento do caminho de n até à raiz; a profundidade da raiz é zero
- a **altura de um nó** é o comprimento do caminho desde esse nó até ao seu nó-folha mais profundo (a altura de um nó folha é zero)
- a **altura de uma árvore** é a altura da raiz (ex: o comprimento do maior caminho de um nó-folha até à raiz).

Outras Formas de Representação de Árvore

- Representação por parênteses aninhados:

$(A(B)(C(D(G)(H))(E)(F(I))))$

- Representação por diagramas:

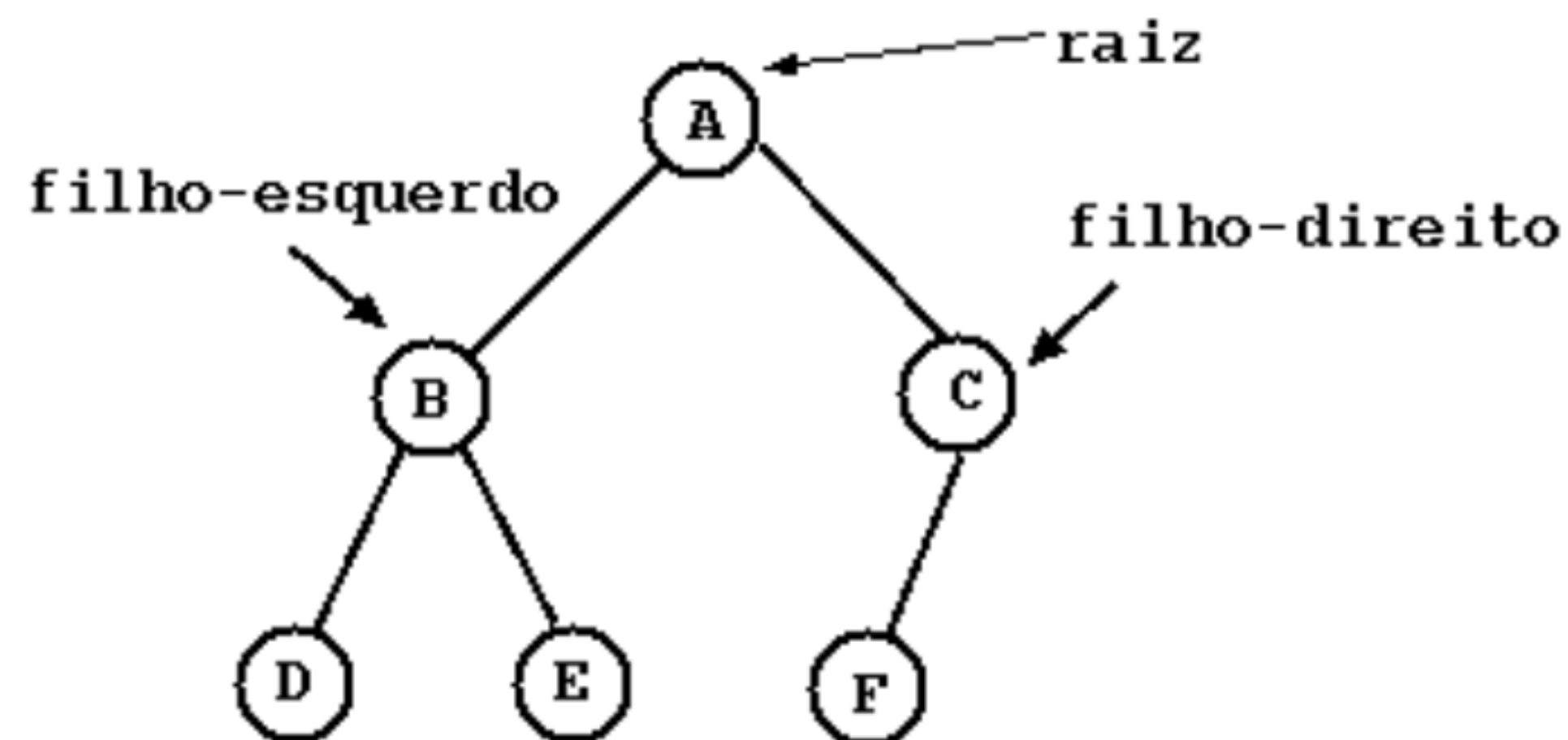


#02 - Árvores Binárias

- Definição
- Árvore Estritamente Binária
- Árvore Binária Completa
- Árvore Binária Cheia
- Árvore Binária Balanceada
- Travessias

Definição Formal de Árvore Binária

- Uma **árvore-binária** é constituída por um conjunto finito de nós. Se o conjunto for vazio, a árvore diz-se **vazia**, caso contrário obedece às seguintes regras:
 - possui um nó especial, a raiz da árvore
 - cada nó possui no máximo dois filhos: filho-esquerdo e/ou filho-direito
 - cada nó, exceto a raiz, possui exatamente um nó-pai.
- Ou dito de forma mais simples: **árvores binárias** são árvores em que cada nó tem 0, 1 ou 2 filhos



- Se r é a raiz da árvore binária T , diz-se que TE e TD são as subárvores esquerda e direita de T , respectivamente.

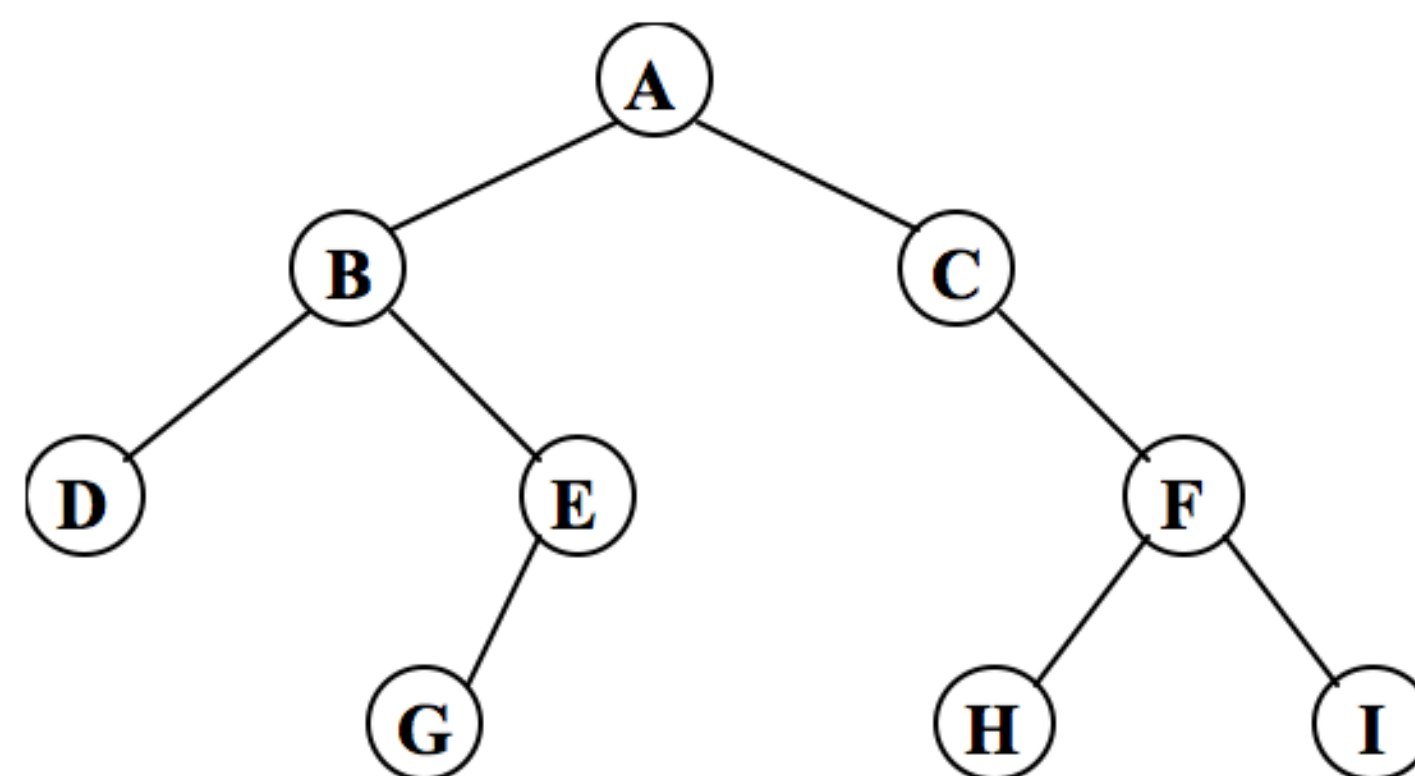


Figura 1. Árvore Binária

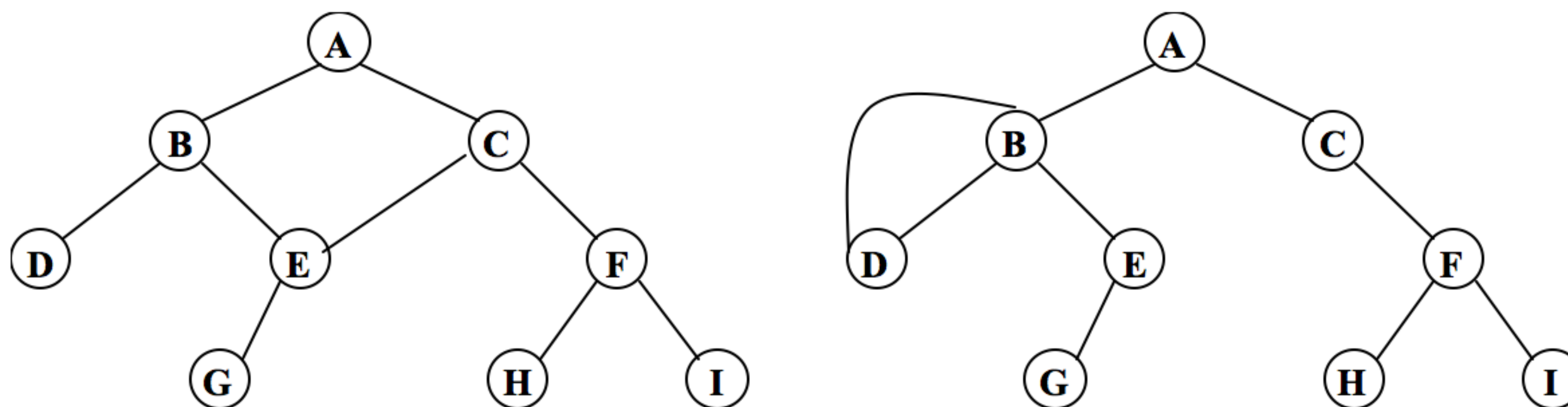
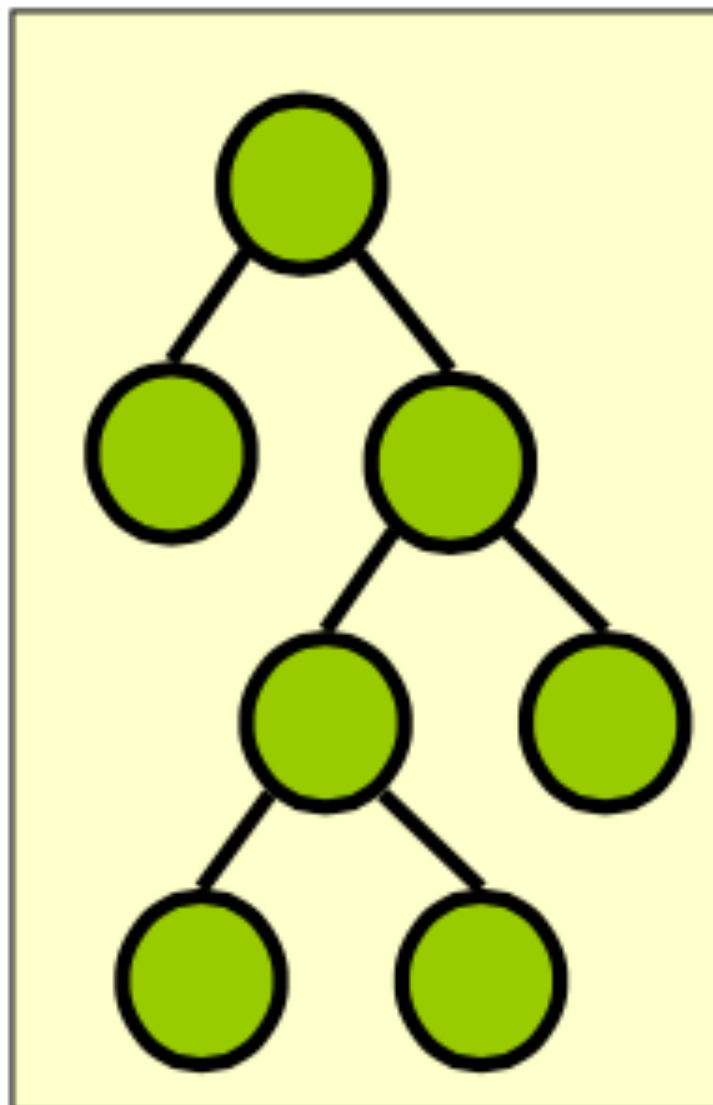


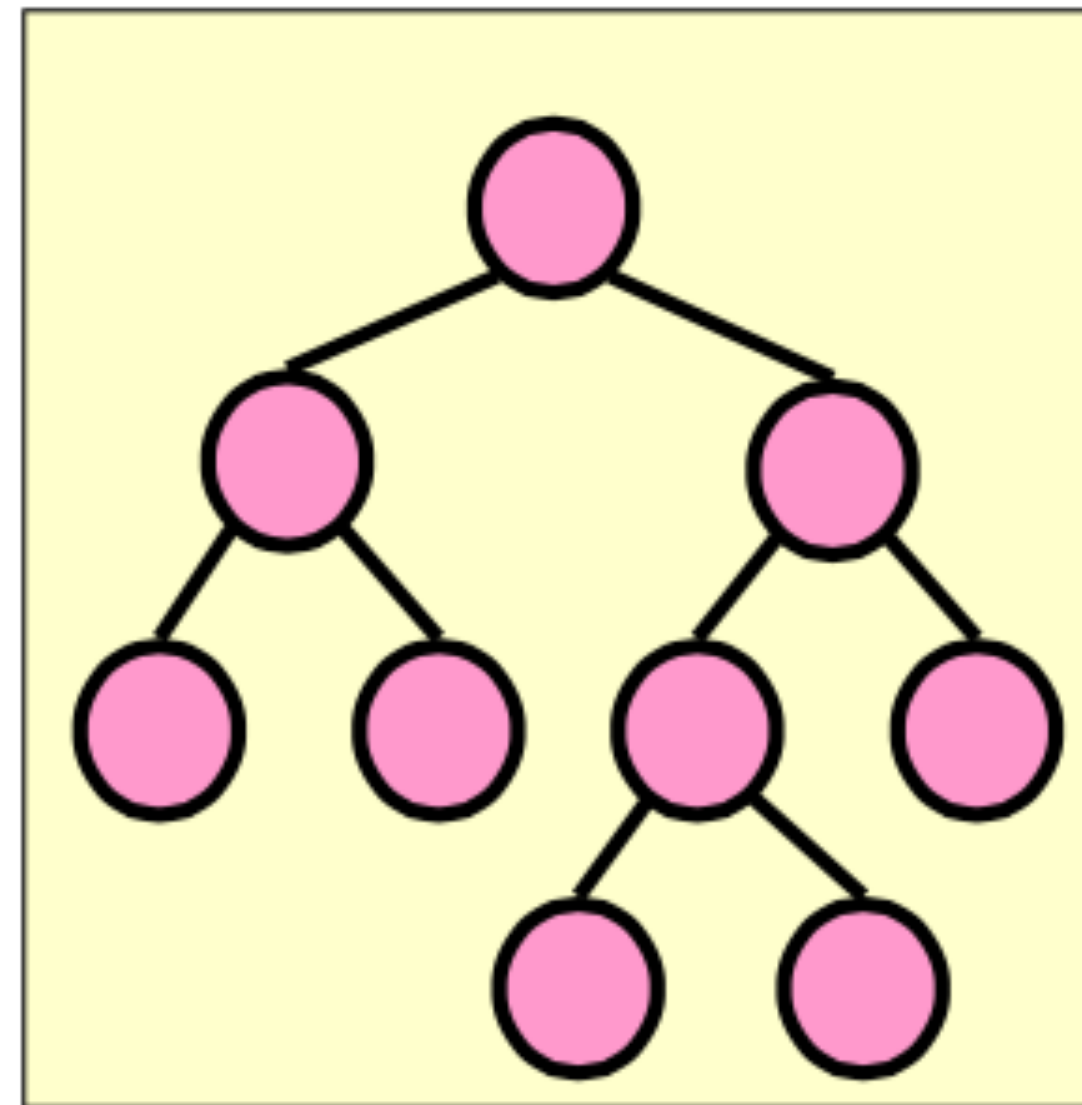
Figura 2. Estruturas que não são Árvores Binárias

Classificações Especiais de Árvores Binárias



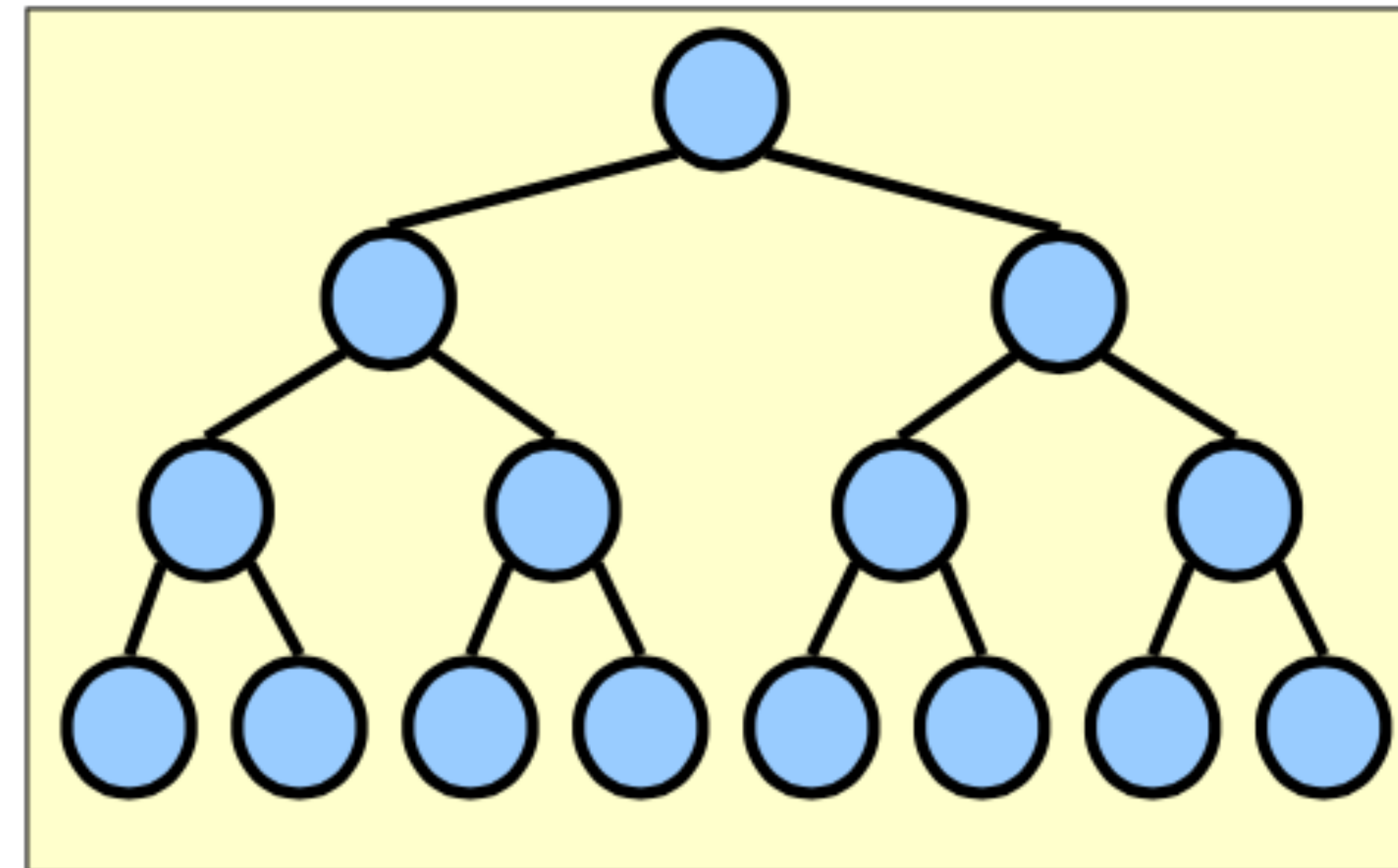
**Estritamente
Binária**

0 ou 2 filhos



**Binária
Completa**

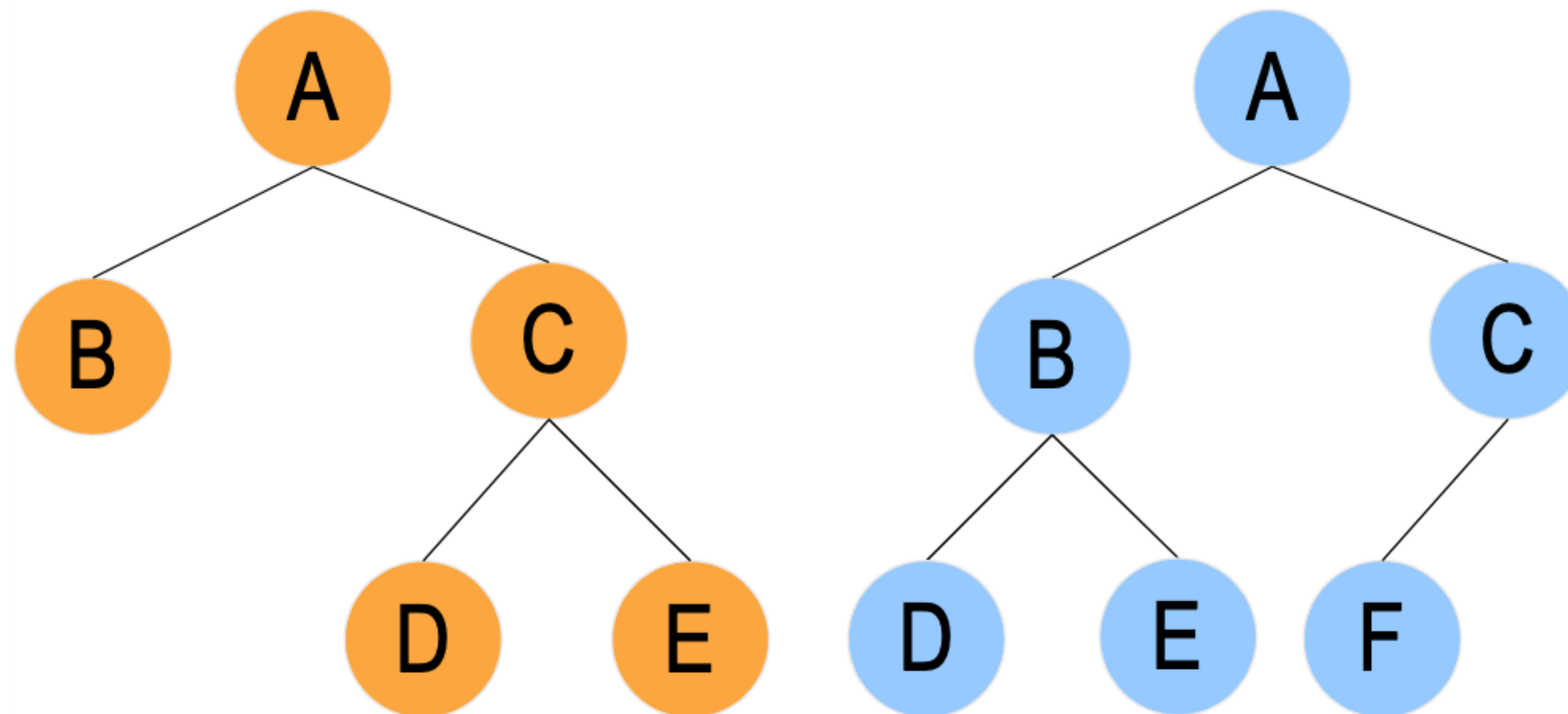
Sub-árvores vazias
no último ou
penúltimo nível



**Binária
Cheia**

Sub-árvores vazias
somente no último nível

- Uma Árvore Binária é dita **Balanceada** se, para cada nó, **as alturas de suas duas sub-árvores** diferem de, no **máximo, 1**.



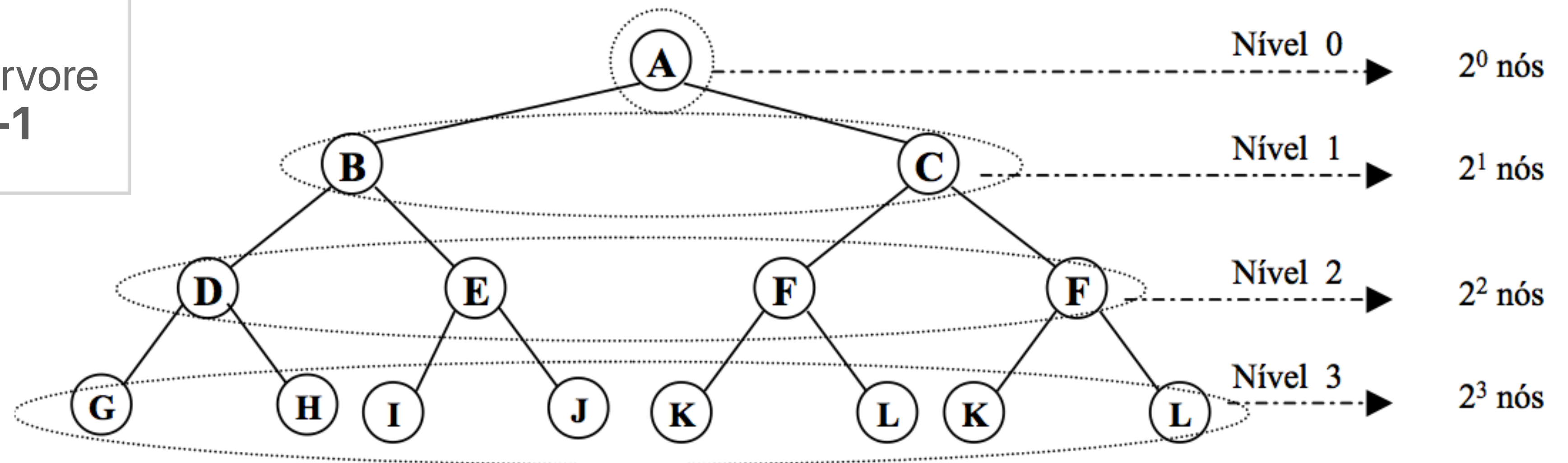
Níveis de uma Árvore Binária

■ O **nível** de um nó X em uma árvore binária é definido como:

- O nó raiz está no nível 0;
- Os outros nós possuem um nível a mais que o nível do seu nó pai.

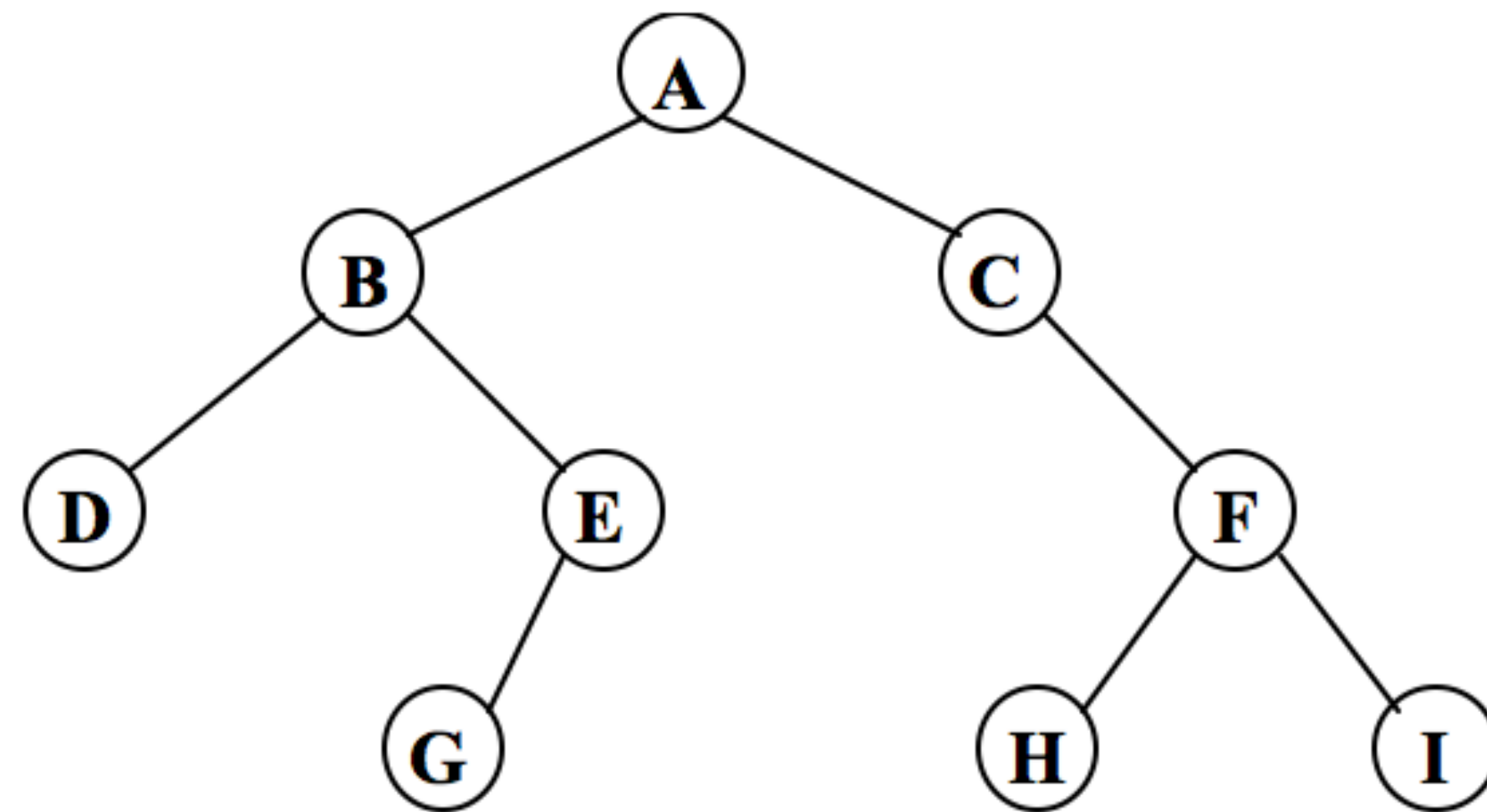
■ A quantidade de nós em uma AB é a soma dos nós dos níveis de 0 a d (profundidade da árvore): $2^{d+1} - 1$

■ Para determinar qual a profundidade d de uma árvore com n nós: $d = \log_2(n+1) - 1$



- Em uma Lista simplesmente encadeada, a única forma de pesquisar seus elementos é percorrer a lista seqüencialmente, em ordem linear
- Porém, em uma árvore existem várias estratégias de pesquisa. Todas as estratégias de pesquisa em uma árvore envolvem a visita ao nó *raiz* e às suas subárvores *esquerda* e *direita* em uma determinada ordem
- Cada estratégia de percurso gera uma sequência linear de nós e podemos então falar em nó predecessor ou sucessor de um nó, segundo um dado percurso
- As três estratégias mais simples de travessia em uma árvore binária são as formas: *pré-ordem*, *in-ordem* e *pós-ordem*.





Pré-ordem: ABDEGCFHI

In-ordem: DBGEACHFI

Pós-ordem: DGEBHIFCA

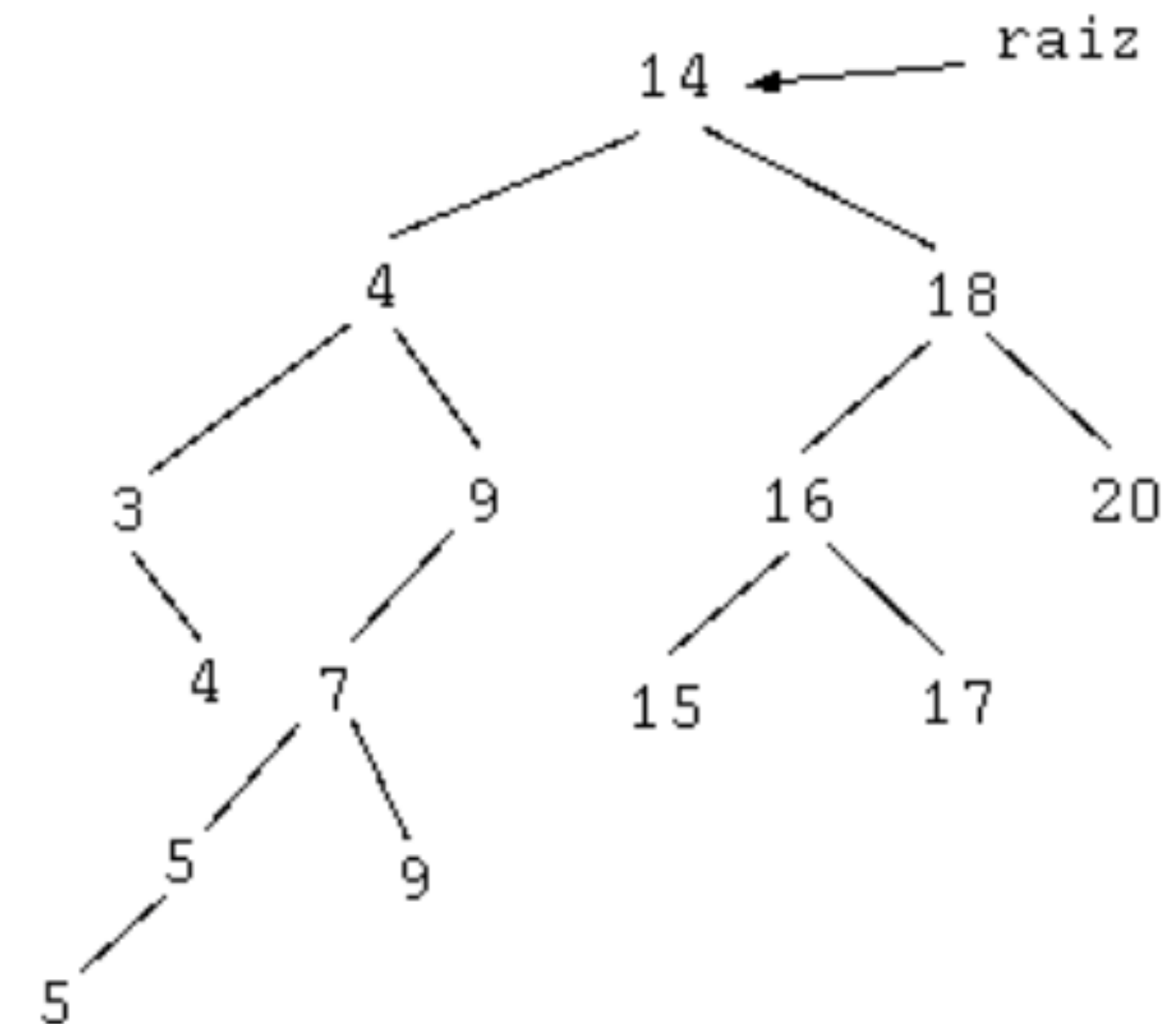
- **pré-ordem:**
 - visite a raiz da árvore;
 - percorra a subárvore esquerda;
 - percorra a subárvore direita.
- **in-ordem:**
 - percorra a subárvore esquerda;
 - visite a raiz da árvore;
 - percorra a subárvore direita.
- **pós-ordem:**
 - percorra a subárvore esquerda;
 - percorra a subárvore direita;
 - visite a raiz da árvore

#03 - Árvores Binárias de Busca (ABB)

- Definição
- Operações em ABB

Definição Formal de Árvore Binária de Busca (ABB)

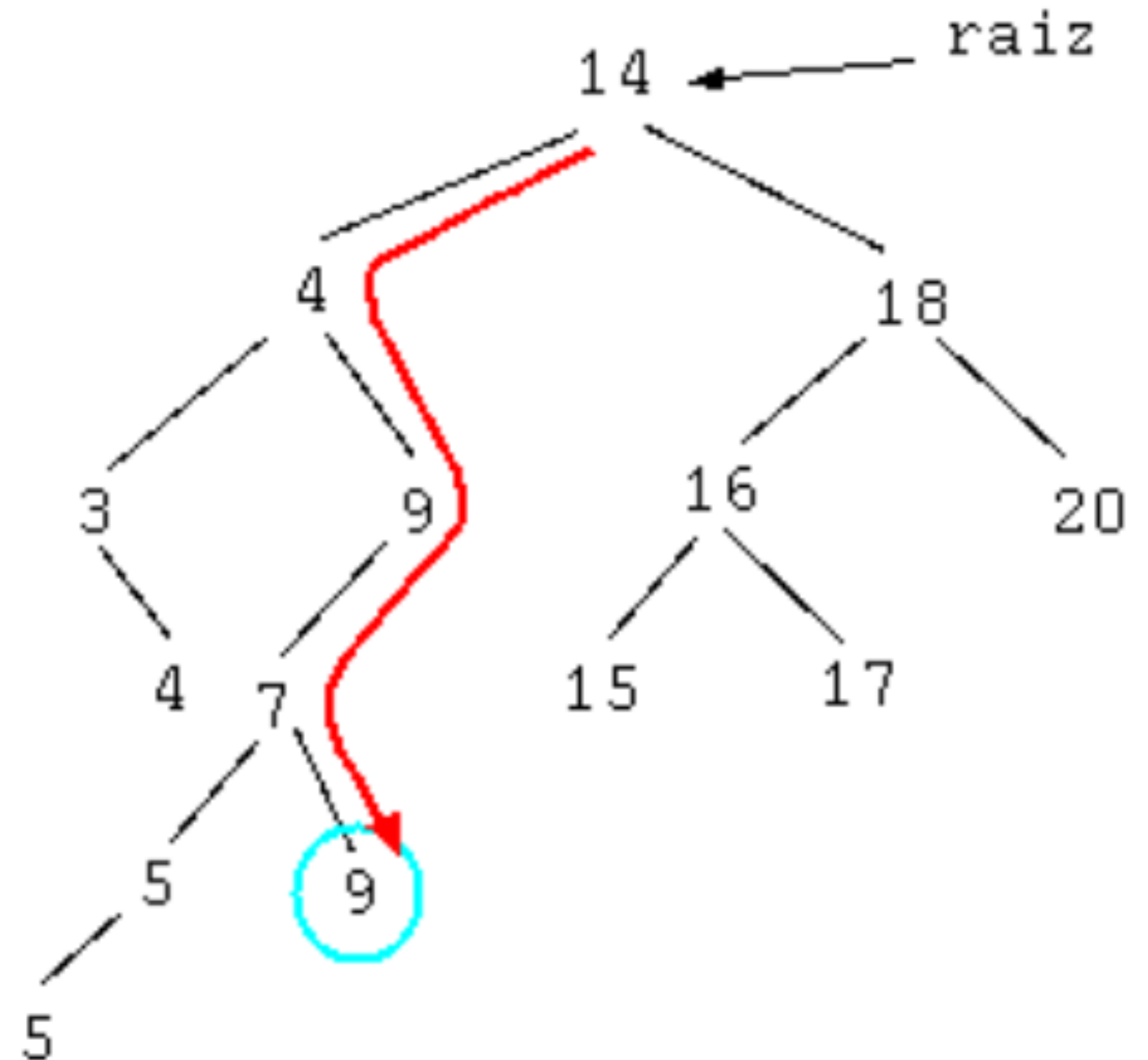
- Uma árvore binária (T) diz-se de busca se :
 - $T = \emptyset$, a árvore for vazia;
 - Cada nó de T contém uma chave que satisfaz as condições a seguir:
 - Todas as chaves (se existirem) na sub-árvore esquerda de cada nó são menores (ou iguais) que a chave do nó;
 - Todas as chaves (se existirem) na sub-árvore direita de cada nó são maiores que a chave do nó;
 - As sub-árvores esquerda e direita de cada nó também são árvores binárias de busca.



Operação de Inserção na ABB

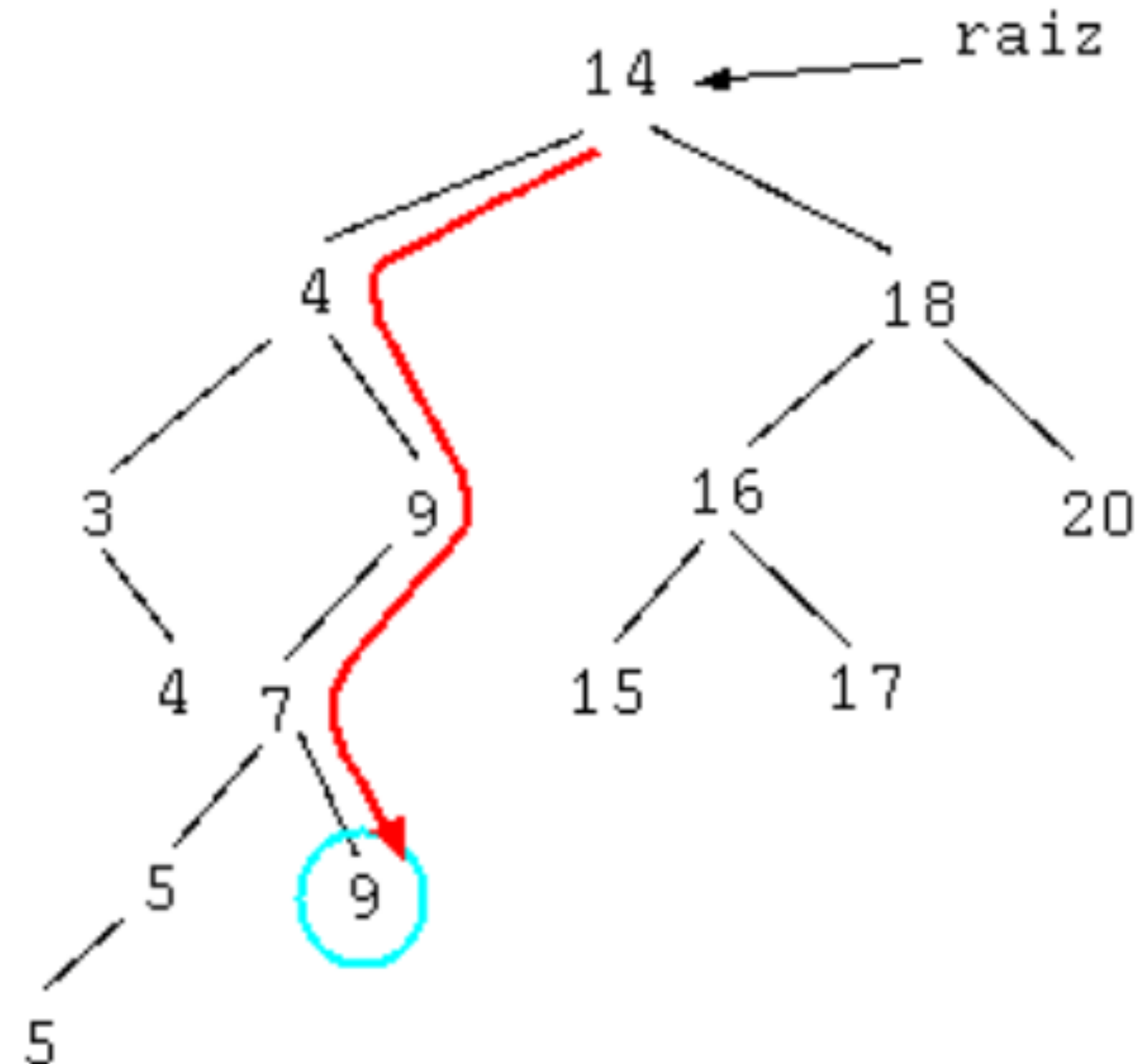
- Ilustração do percurso na árvore quando se pretende inserir uma nova chave

seq: 14, 18, 4, 9, 7, 16, 3, 5, 4
17, 20, 15, 9, 5



Operação de Pesquisa em ABB

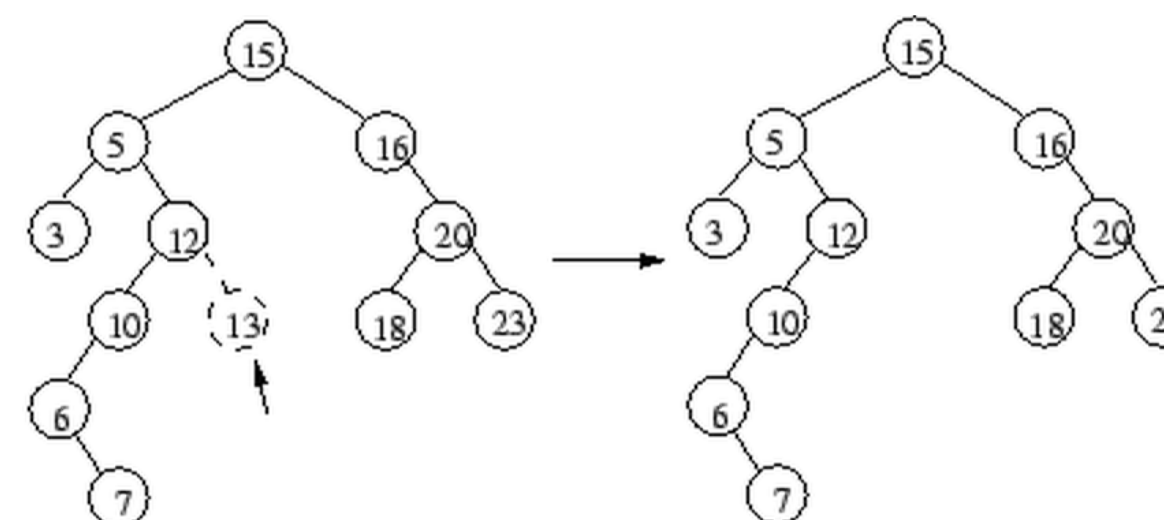
- Dada uma árvore binária de busca T e um valor x, pretende-se verificar se T contém x
- Quando se pensa em “percorrer” uma árvore, deve pensar-se numa solução recursiva em que temos de lidar com três casos:
 - Árvore vazia ou percorreu-se até um nó folha de forma ordenada e não se achou x;
 - Nó corrente contém x;
 - Continuar a procura numa das sub-árvores, de acordo com a relação de ordem entre x e o valor no nó corrente



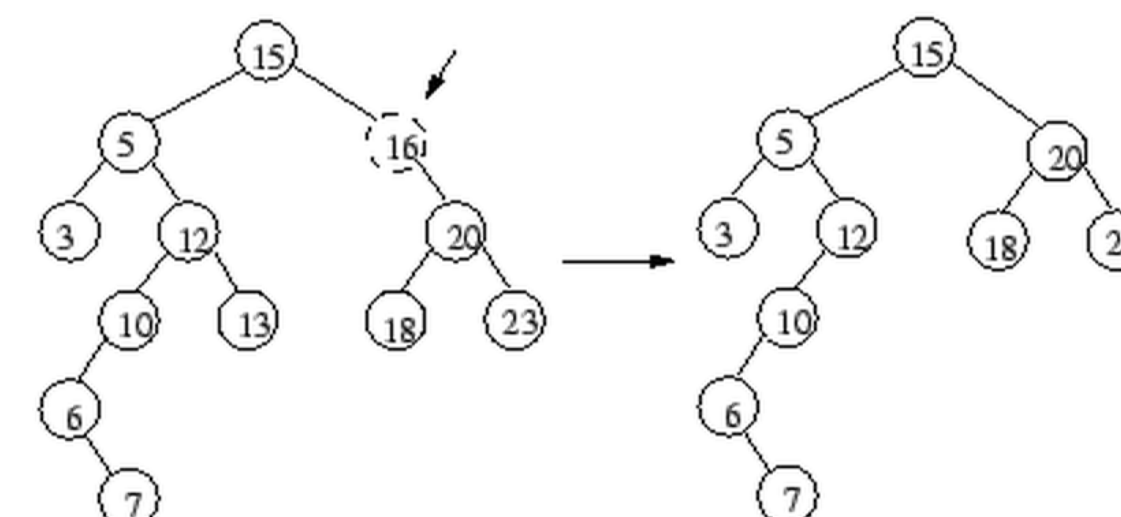
Operação de Exclusão em ABB

- Há três casos especiais para se remover um nó em ABB:

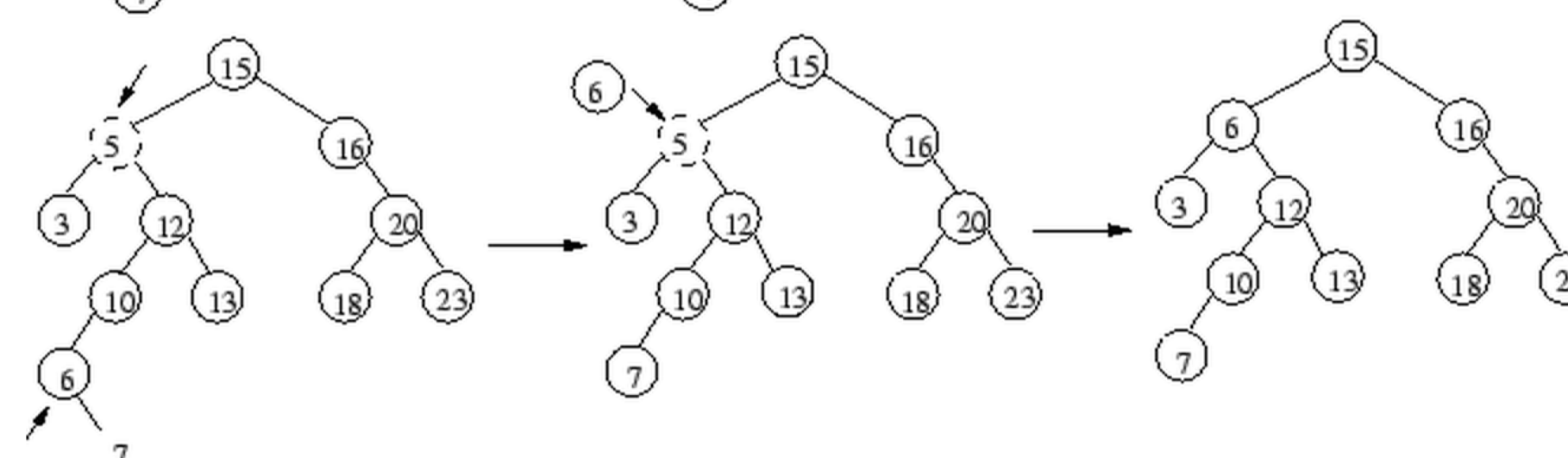
- Nó folha (ex: remover o nó 13)



- Nó-interior com apenas um filho (ex: remover o nó 16)

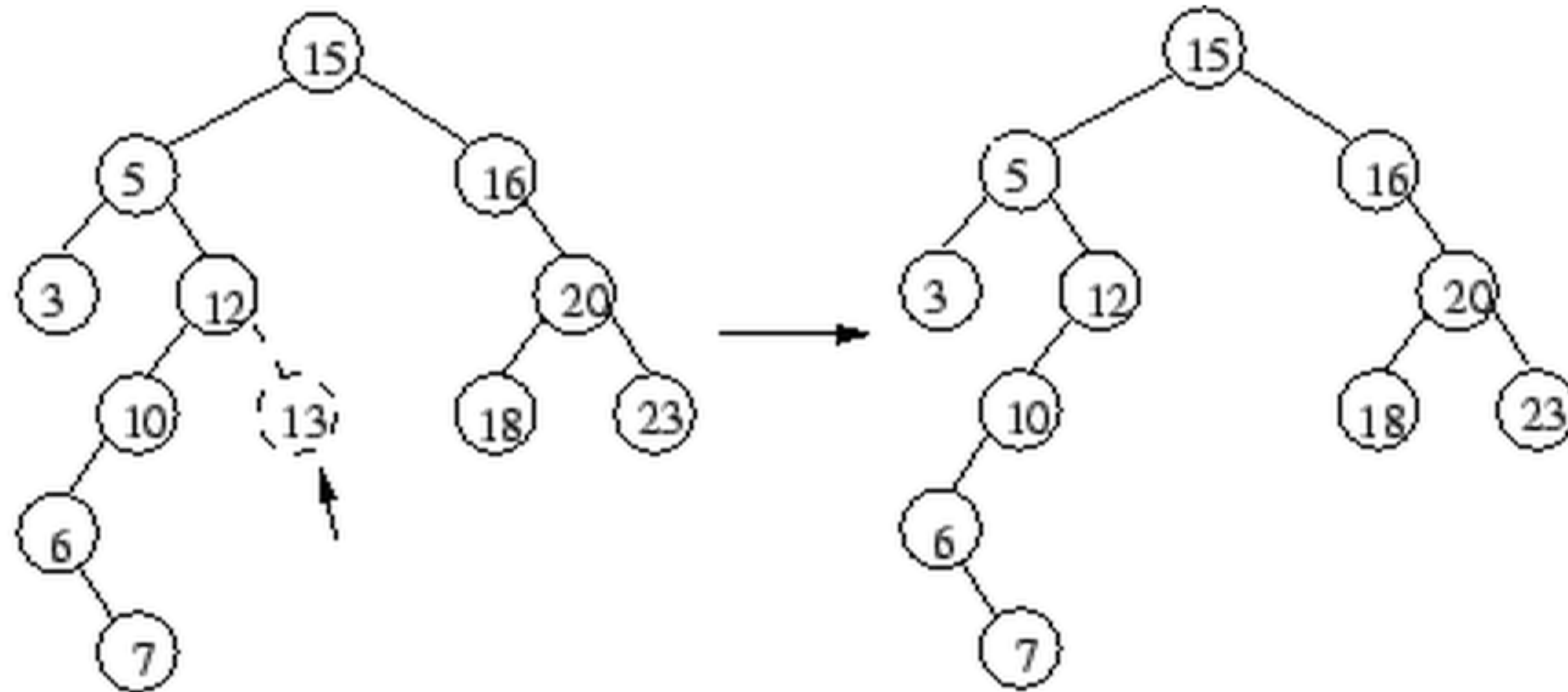


- Nó-interior com dois filhos (ex: remover o nó 5)



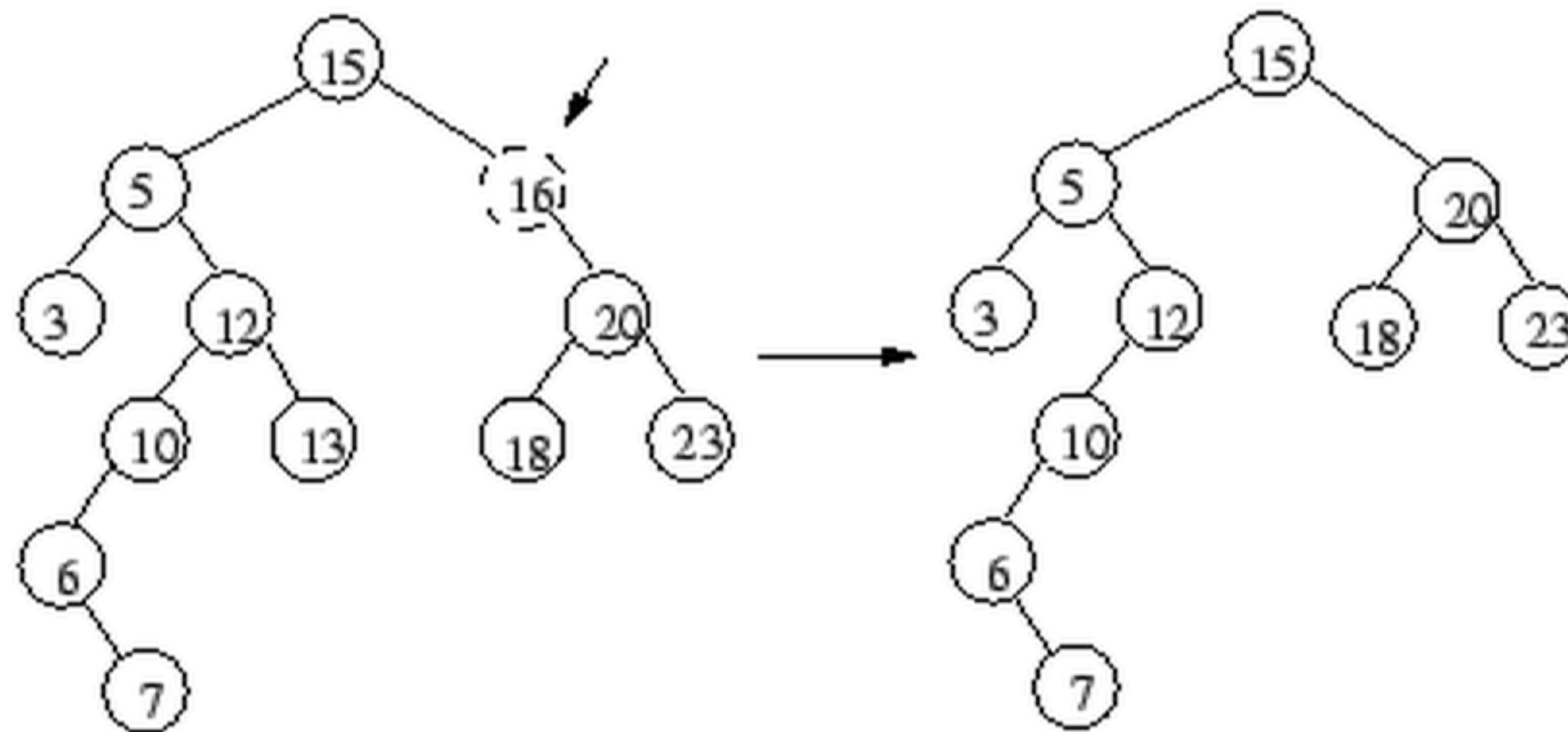
Operação de Exclusão em ABB

- Nó folha (ex: remover o nó 13)



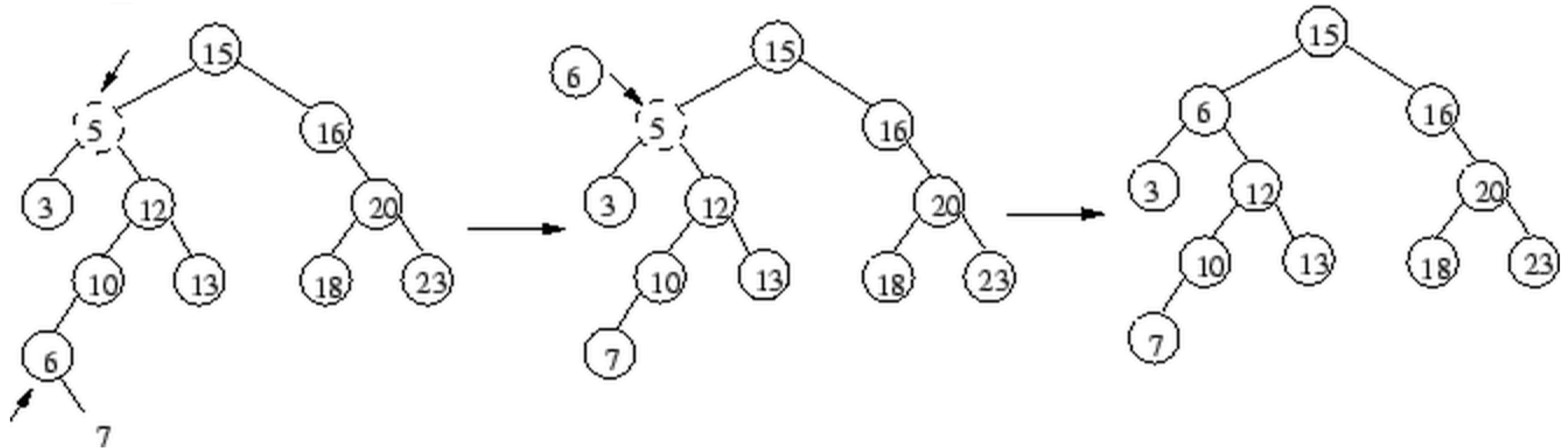
Operação de Exclusão em ABB

- Nó-interior com apenas um filho
(ex: remover o nó 16)



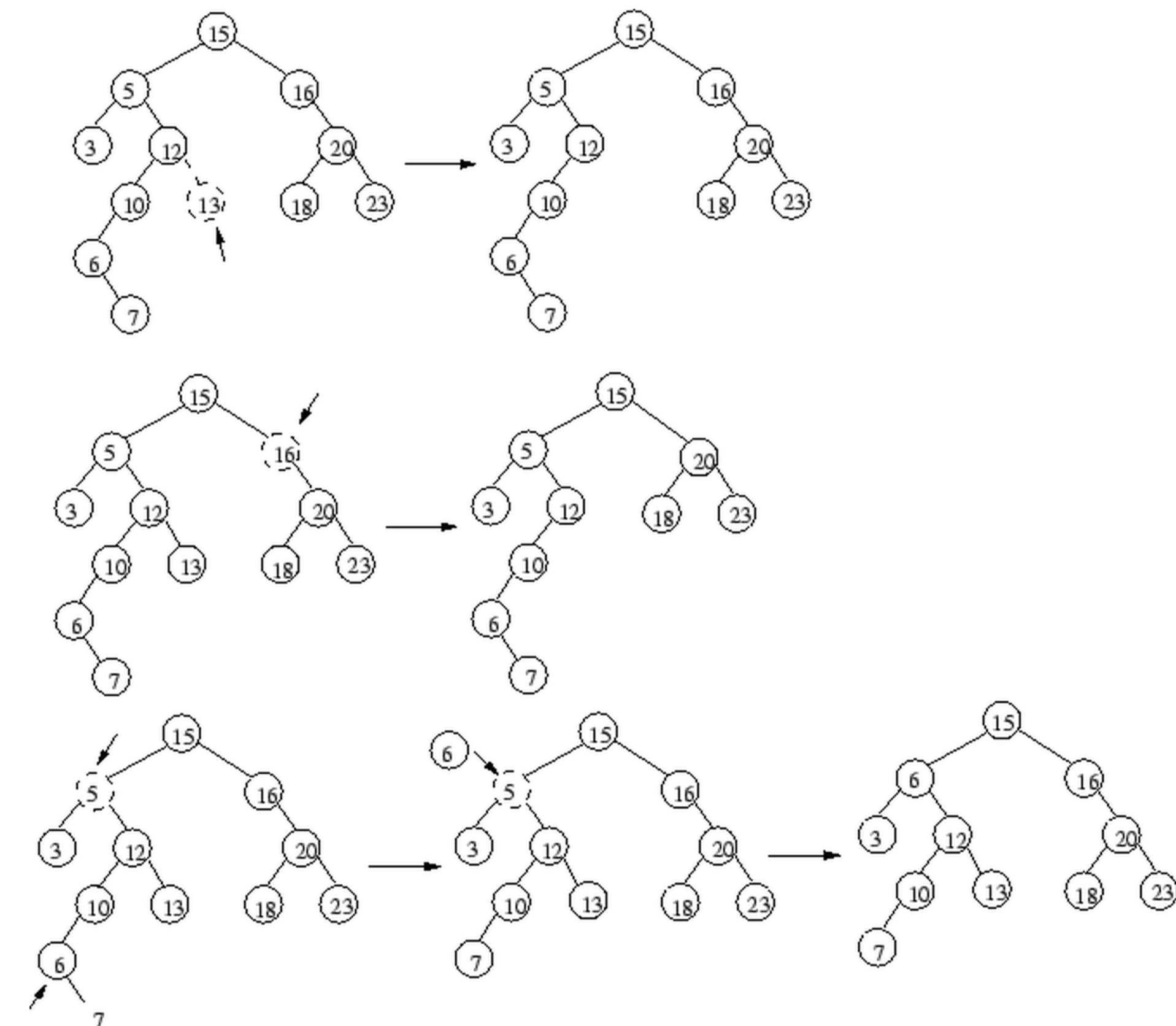
Operação de Exclusão em ABB

- Nó-interior com dois filhos (ex: remover o nó 5)



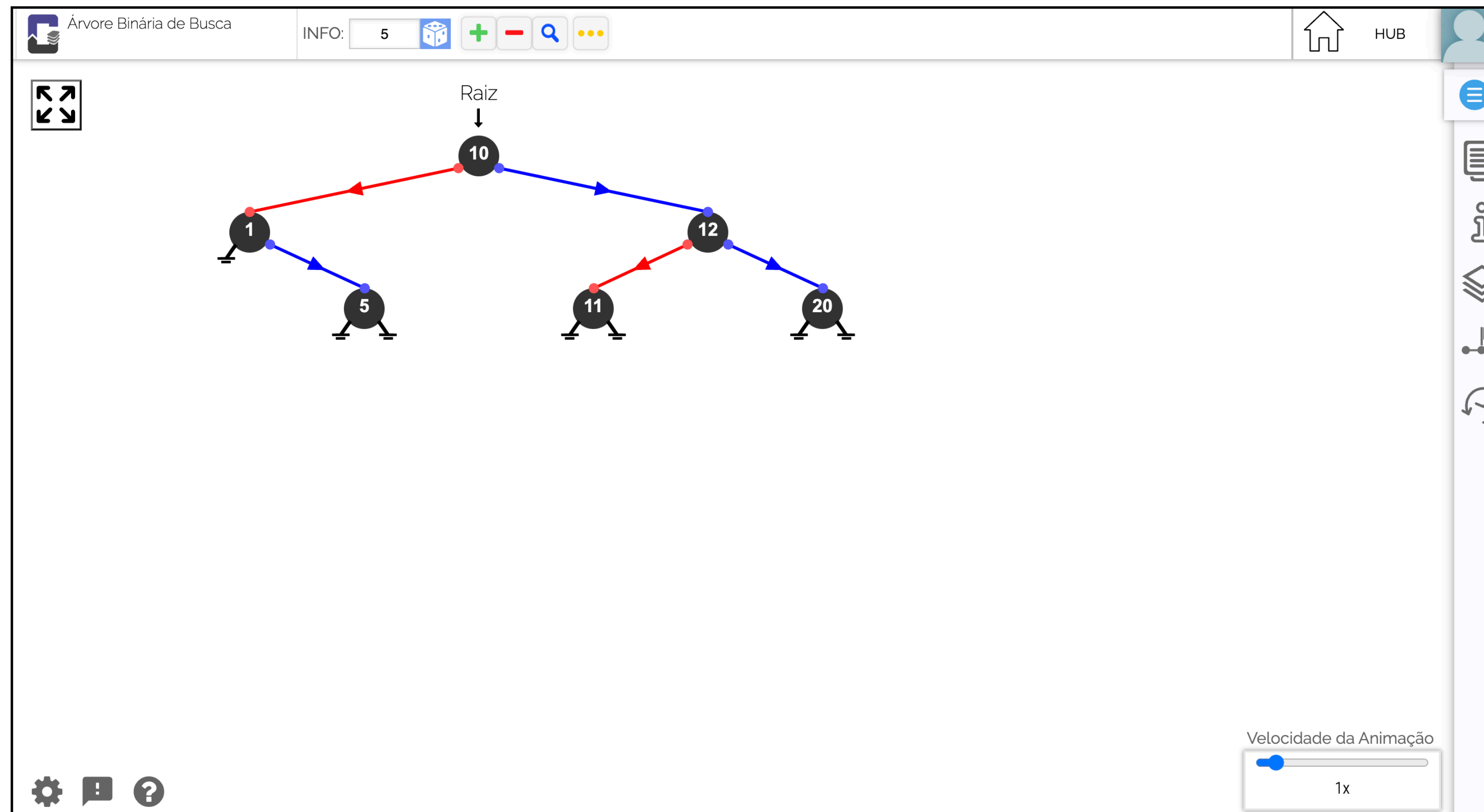
Algoritmo de Exclusão em ABB

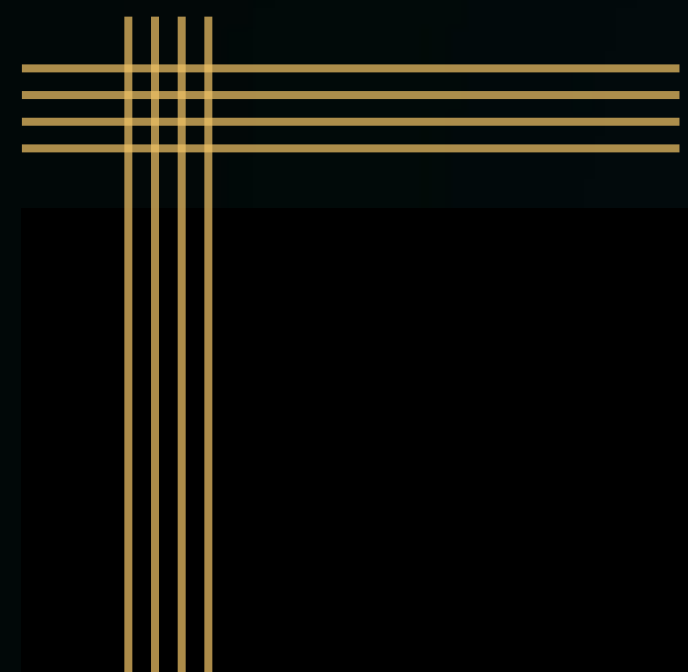
- A operação de remover um nó que contém um valor x de uma árvore T , caso tal nó exista, é um pouco mais complexa e requer:
 - localizar o nó com o valor x , seja nx esse nó;
 - se nx é um nó-folha, simplesmente remove-se o nó;
 - se nx é um nó-interior, é necessário mais cuidado para não ficarmos com 2 árvores desconexas:
 - se nx só tiver um filho, a sub-árvore pendurada nesse nó toma o lugar de nx ;
 - se nx tiver dois filhos, então devemos procurar o nó com menor valor entre os descendentes do filho-direito (ou o maior dos descendentes do filho-esquerdo) para tomar o lugar de nx .



Simulador de Árvore Binária

- https://portaldoprofessor.fct.unesp.br/projetos/cadilag/apps/structs/arv_binaria.php





ESD|2

Estrutura de
Dados

Prof. Fernando Sambinelli

sambinelli@ifsp.edu.br



**INSTITUTO
FEDERAL**
São Paulo

