UNIPAC
Barbacena

Centro Universitário Presidente Antônio Carlos
Teoria de Grafos

Remoção de um Vértice (Árvores Binárias)

Felipe Roncalli de Paula Carneiro
felipecarneiro@unipac.br

O que vamos aprender nessa aula

- Recursividade;
- **Tipos de Remoção**
- **Reorganização da Árvore;**

Recursividade

É o mecanismo de programação no qual uma definição de função ou de outro objeto refere-se ao próprio objeto sendo definido. Assim função recursiva é uma função que é definida em termos de si mesma.

Remoção

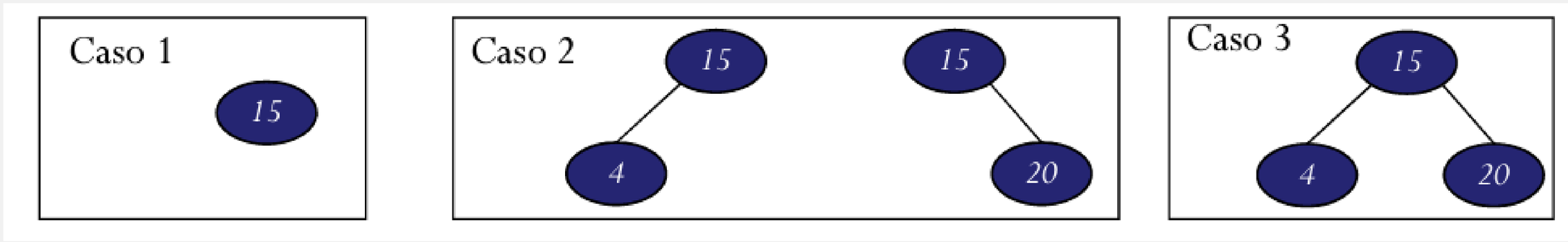
- Na remoção de um nó da árvore binária de busca, o nível de complexidade depende da posição do nó a ser removido da árvore, pois após a remoção a árvore deve preservar sua propriedade fundamental.

- Mais difícil remover nó que tem duas sub-árvores do que remover nó folha.

Remoção

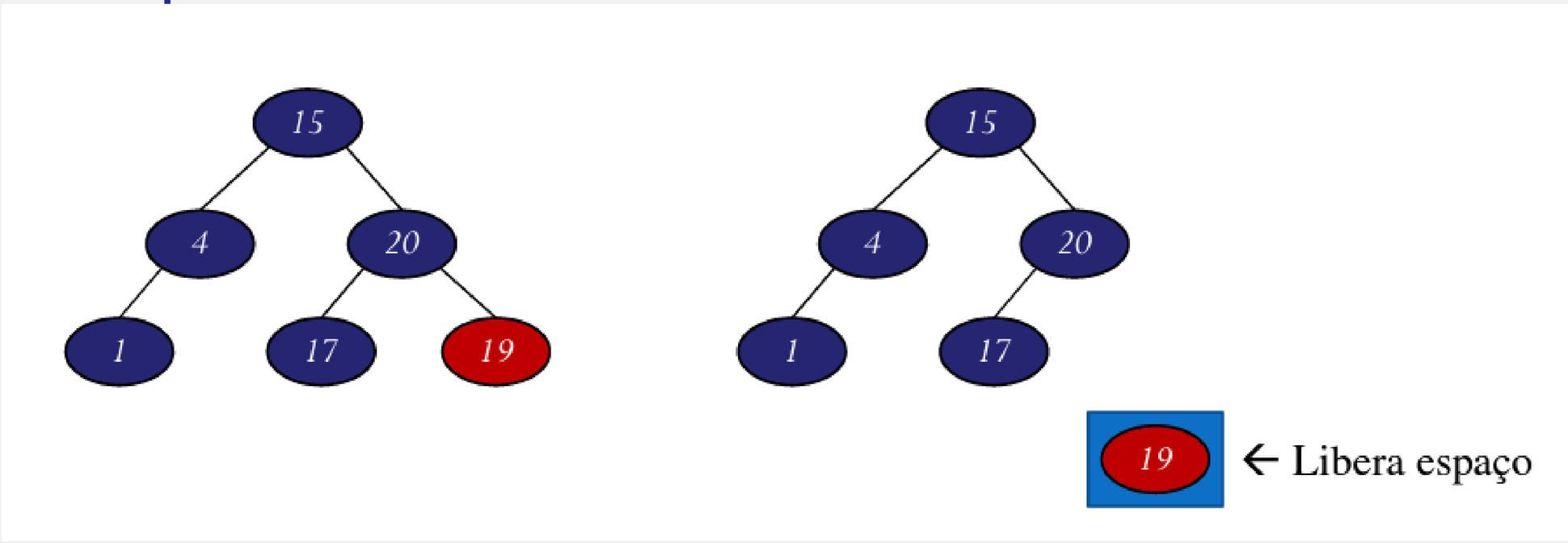
Há três casos possíveis, o:

1) nó é uma folha (não tem filhos)
2) nó tem 1 filho
3) nó tem 2 filhos



Remoção Nó Folha

- Caso mais fácil de tratar.
- O ponteiro apropriado de seu nó pai é ajustado para nil e o nó é removido por desaloque (apagado da memória).
- Exemplo: remover nó com conteúdo 19:



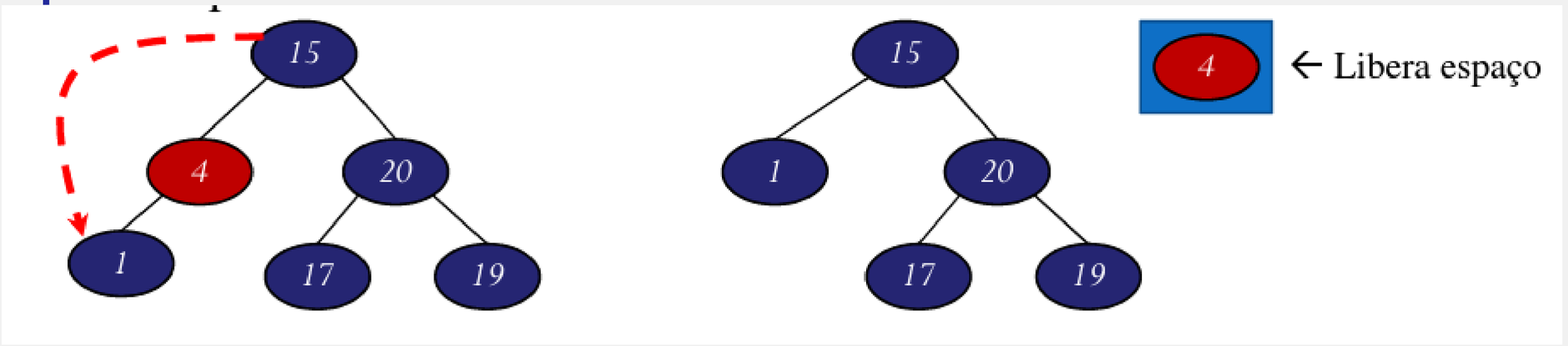
Remoção de nó com 1 filho

- Ponteiro do pai do nó a ser removido é reajustado para apontar para o filho do nó a ser removido.
  - Ou seja, o pai vai apontar para o neto (que passa a ser filho).

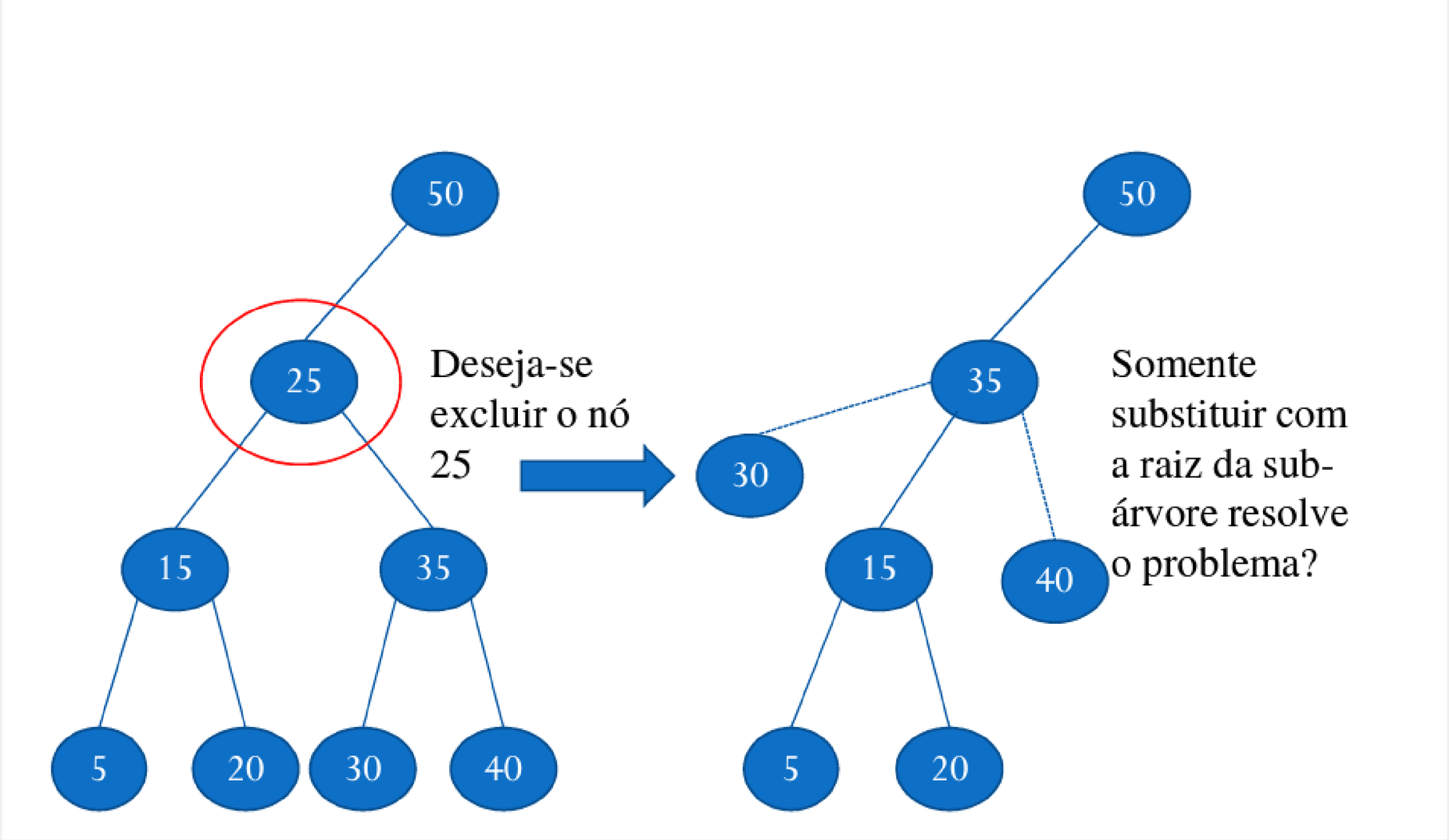
Remoção de nó com 1 filho

- Filhos do nó são elevados em 1 nível;
- Bisnetos perdem um grau de descendência em suas designações de parentesco.

Exemplo: Remover nó 4



Remoção de nó com 2 filhos



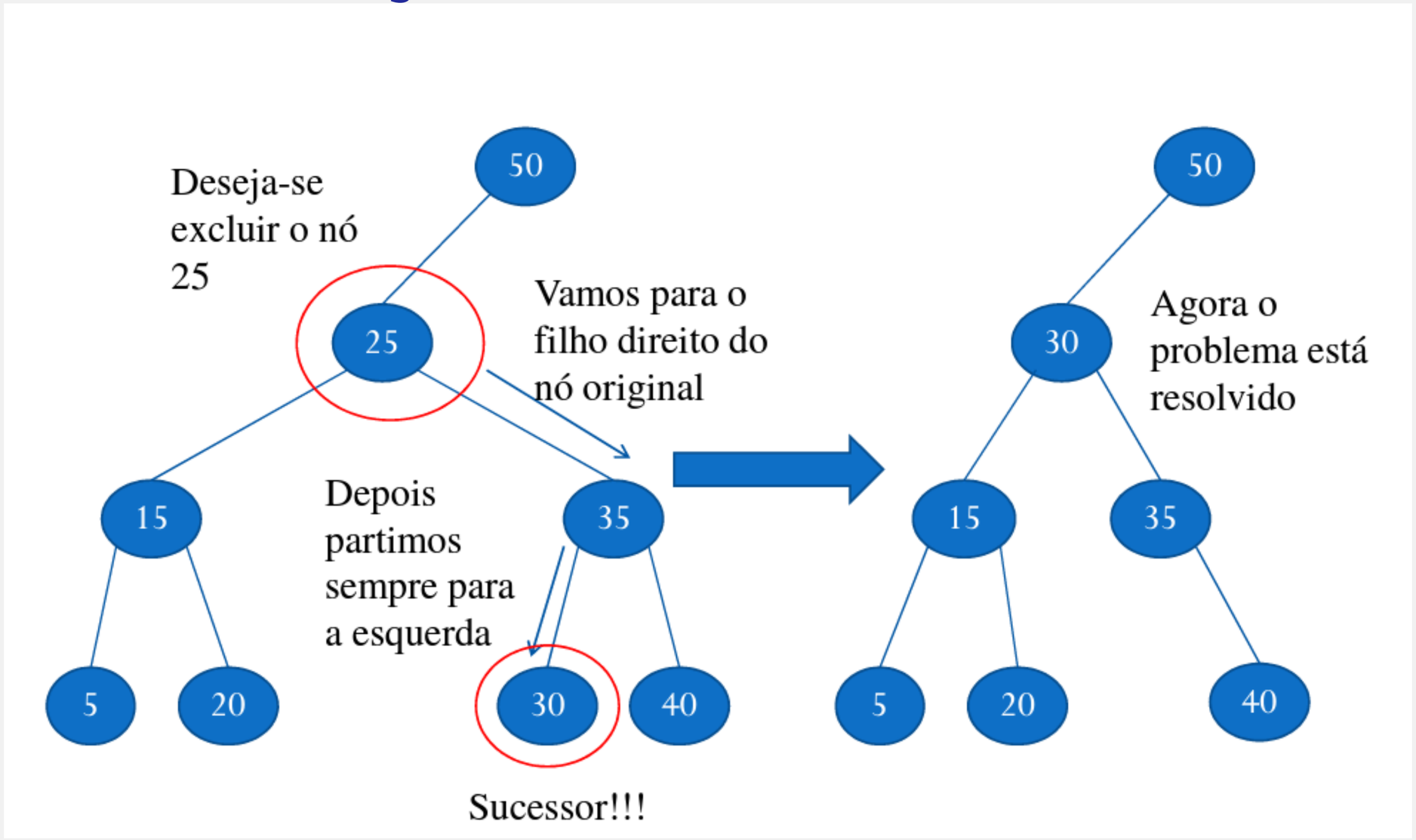
Remoção de nó com 1 filho

- Remoção por cópia (Thomas Hibbard e Donald Knuth)
  - Substituir o nó pelo menor nó à direita e ajustar ponteiros
  - Busca do substituto:
    - ir para direita
    - procurar elemento mais à esquerda (menor filho à esquerda)
    - guardar antecessor, para descobrir pai do substituto
    - Se a direita estiver nula, o substituto será o próximo à esquerda

Remoção de nó com 1 filho

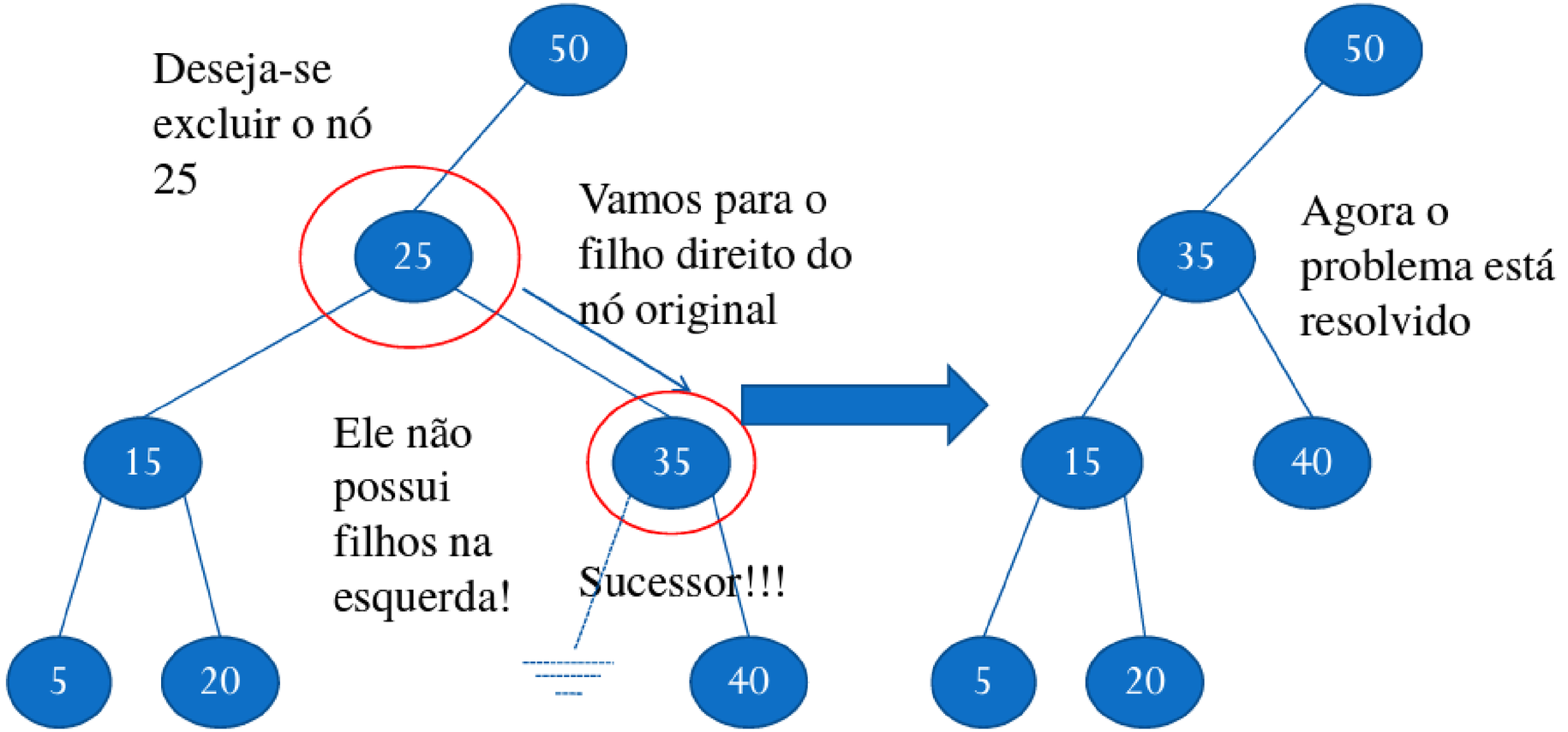
- Ajuste dos ponteiros
  - ajustes no campo ESQ do pai do substituto e no campo DIR do substituto (dispensáveis se nó à direita do removido é exatamente o substituto)
  - campo que referencia nó removido conterá substituto encontrado
  - campo ESQdo substituto aponta para Sub árvore Esquerda do nó removido

Remoção de nó com 2 filhos



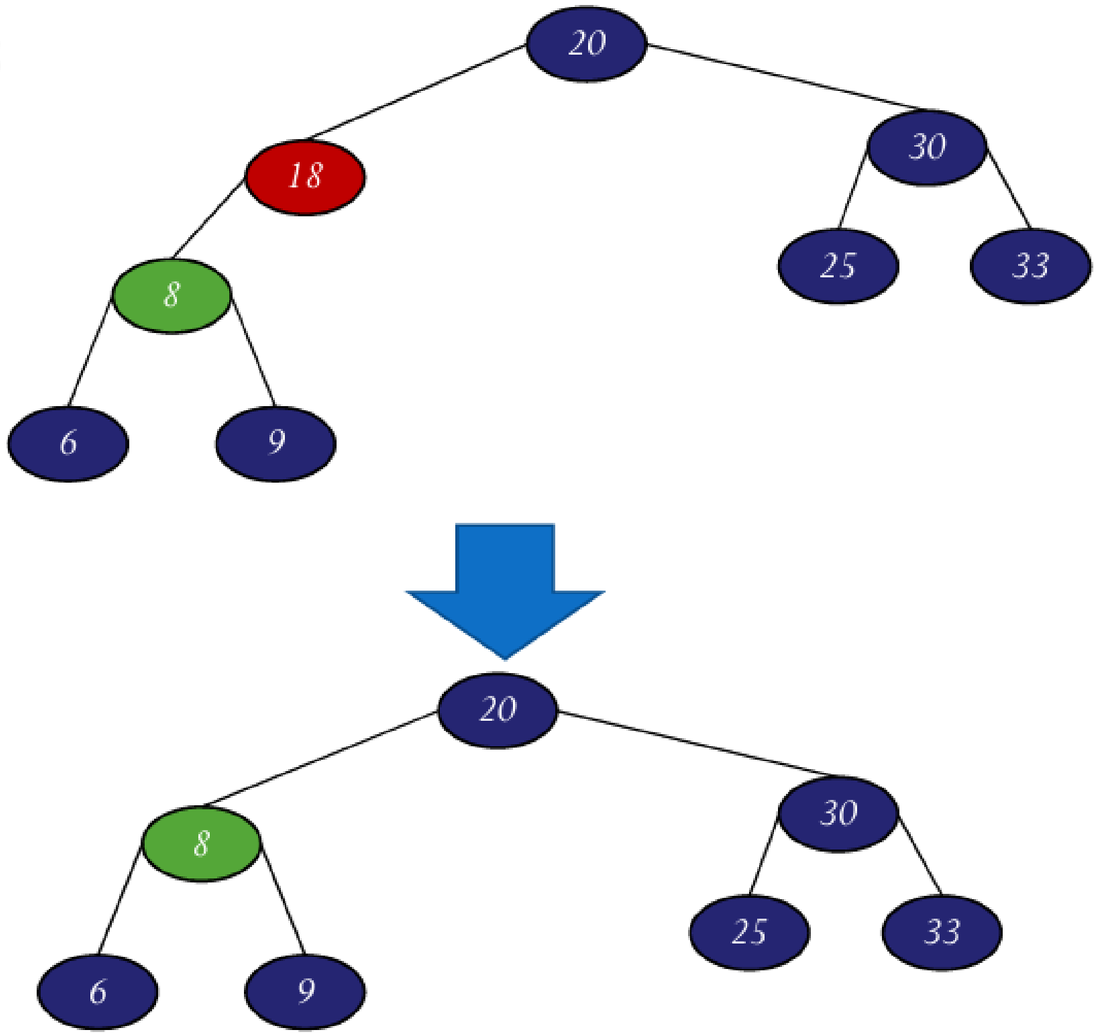
Remoção de nó com 2 filhos

- E quando chegarmos ao filho direito do nó original e ele não tiver filhos do lado esquerdo?



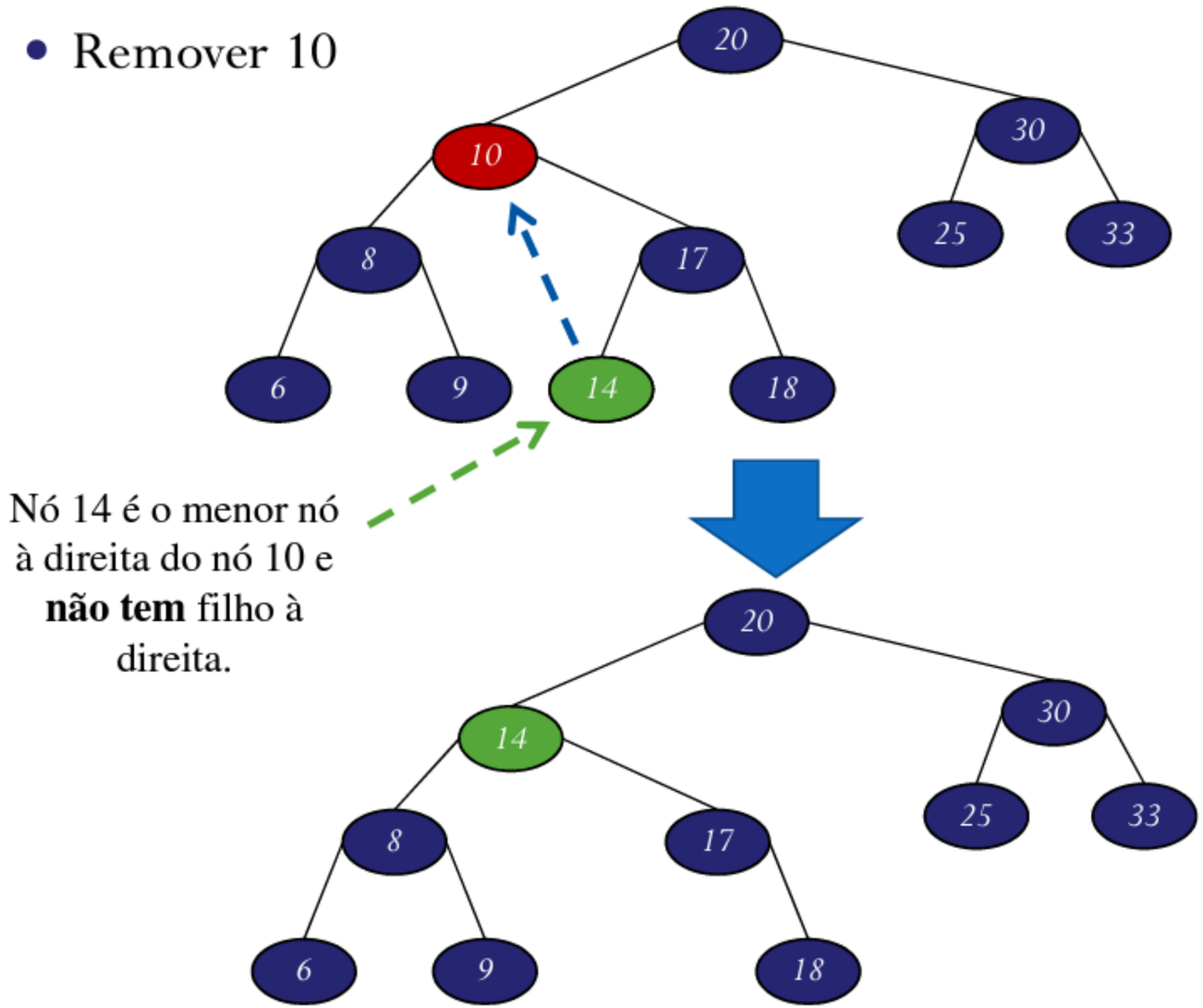
Exemplo - Remoção

- Remover 18



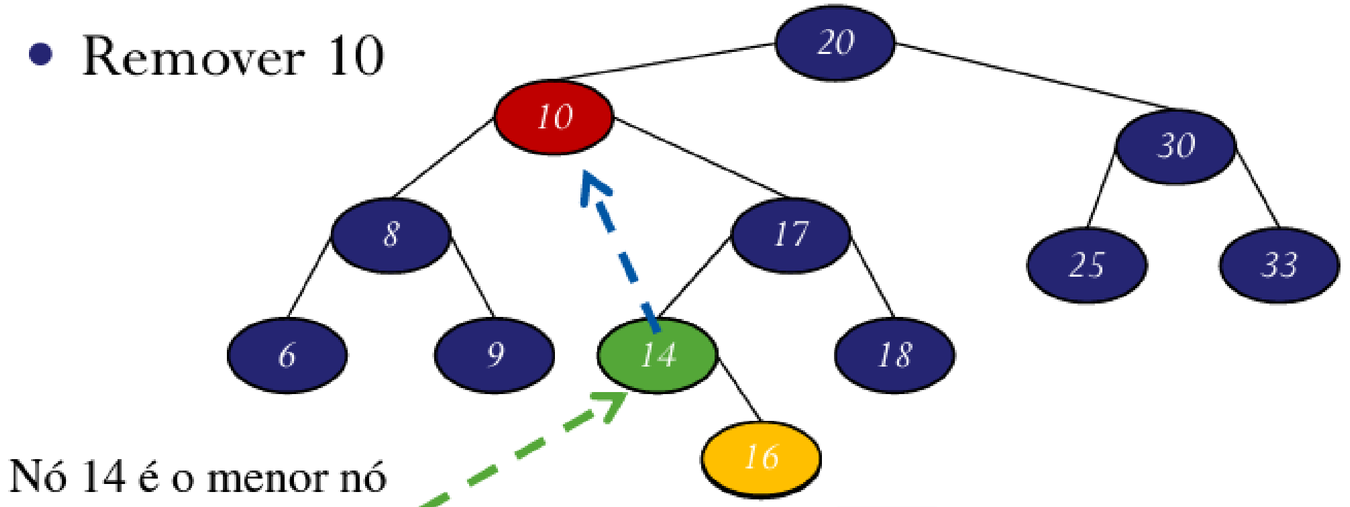
Exemplo - Remoção

- Remover 10

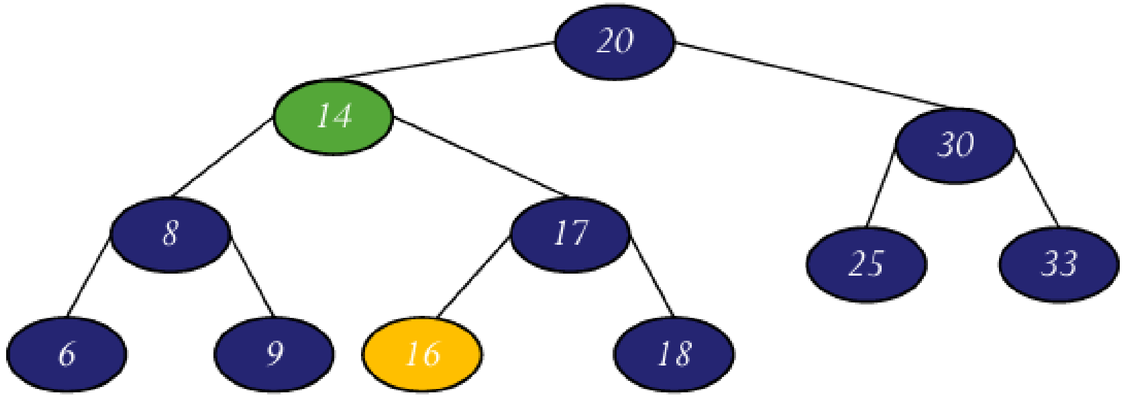


Exemplo - Remoção

- Remover 10



Nó 14 é o menor nó à direita do nó 10 e **tem** filho à direita.



Exercício

A partir da árvore binária de busca abaixo, contendo as chaves 40, 30, 15, 50, 45, 13, 80, 71, 20, executar as operações. Apresente a árvore resultante após a execução de cada operação.

1. Remover 50
2. Remover 30
3. Remover 13
4. Inserir 50
5. Remover 71

Exercício

A partir da árvore binária de busca abaixo, contendo as chaves 50,40,80,35,45,10,38,60,90,55,70,85,100,81,88,42, executar as operações. Apresente a árvore resultante após a execução de cada operação.

1. Remover 80
2. Remover 60
3. Remover 90
4. Remover 100
5. Remover 45

Exercício

Implemente o código para remover nós a partir do código passado na aula anterior. (Temos que implementar uma solução que atenda os 3 casos)

Dúvidas??