Estruturas de Dados

Cristina Gomes Fernandes

ABBs rubro-negras

Uma ABB é rubro-**negra** se

1. todo nó é rubro ou **negro**
2. toda folha (NIL) é **negra**
3. se um nó é rubro, então seus dois filhos são **negros**
4. todo caminho de um nó x até uma folha sua descendente tem o mesmo número de nós **negros**.

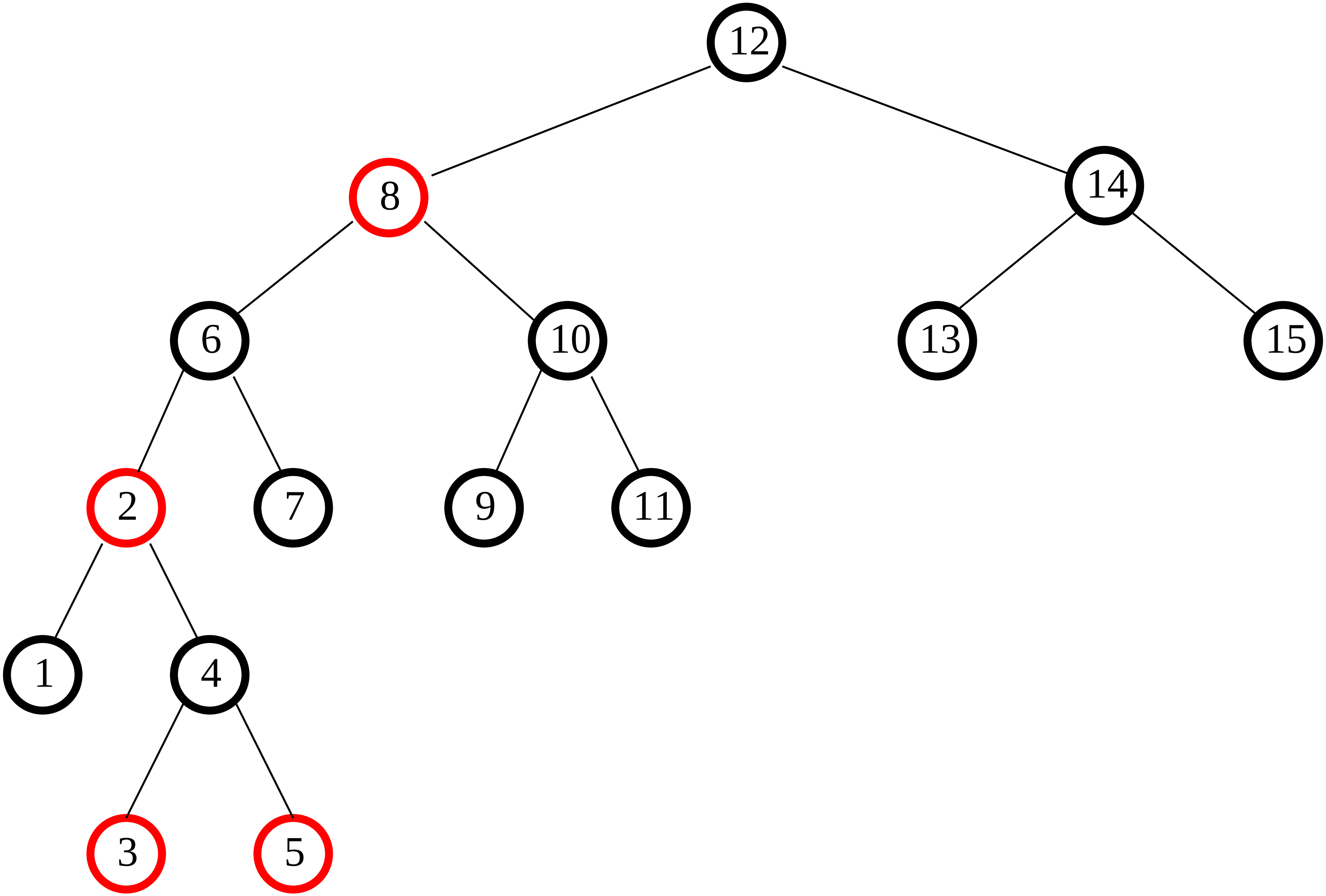
ABBs rubro-negras

Uma ABB é rubro-**negra** se

1. todo nó é rubro ou **negro**
2. toda folha (NIL) é **negra**
3. se um nó é rubro, então seus dois filhos são **negros**
4. todo caminho de um nó x até uma folha sua descendente tem o mesmo número de nós **negros**.

Uma ABB é rubro-**negra** é caída para a esquerda (*left leaning*) se, para todo nó com um único filho rubro, o filho rubro é o da esquerda.

ABB rubro-negra caída para a esquerda



Árvores 2-3 e 2-3-4 $\times$ rubro-negras

Uma árvore 2-3 é uma B-árvore com $m = 2$.

Uma árvore 2-3-4 é uma B-árvore com $m = 3$.

Árvores 2-3 e 2-3-4 $\times$ rubro-negras

Uma árvore 2-3 é uma B-árvore com $m = 2$.

Uma árvore 2-3-4 é uma B-árvore com $m = 3$.

Um nó de uma árvore 2-3 ou 2-3-4 com
- com três chaves é chamado de 4-nó (árvores 2-3-4),
- duas chaves é chamado de 3-nó, e
- com uma chave é chamado de 2-nó.

Árvores 2-3 e 2-3-4 $\times$ rubro-negras

Uma árvore 2-3 é uma B-árvore com $m = 2$.

Uma árvore 2-3-4 é uma B-árvore com $m = 3$.

Um nó de uma árvore 2-3 ou 2-3-4 com
- com três chaves é chamado de 4-nó (árvores 2-3-4),
- duas chaves é chamado de 3-nó, e
- com uma chave é chamado de 2-nó.

Pode-se representar uma árvore 2-3 por uma rubro-negra em que todo nó negro tem no máximo um filho rubro.

Árvores 2-3 e 2-3-4 $\times$ rubro-negras

Uma árvore 2-3 é uma B-árvore com $m = 2$.

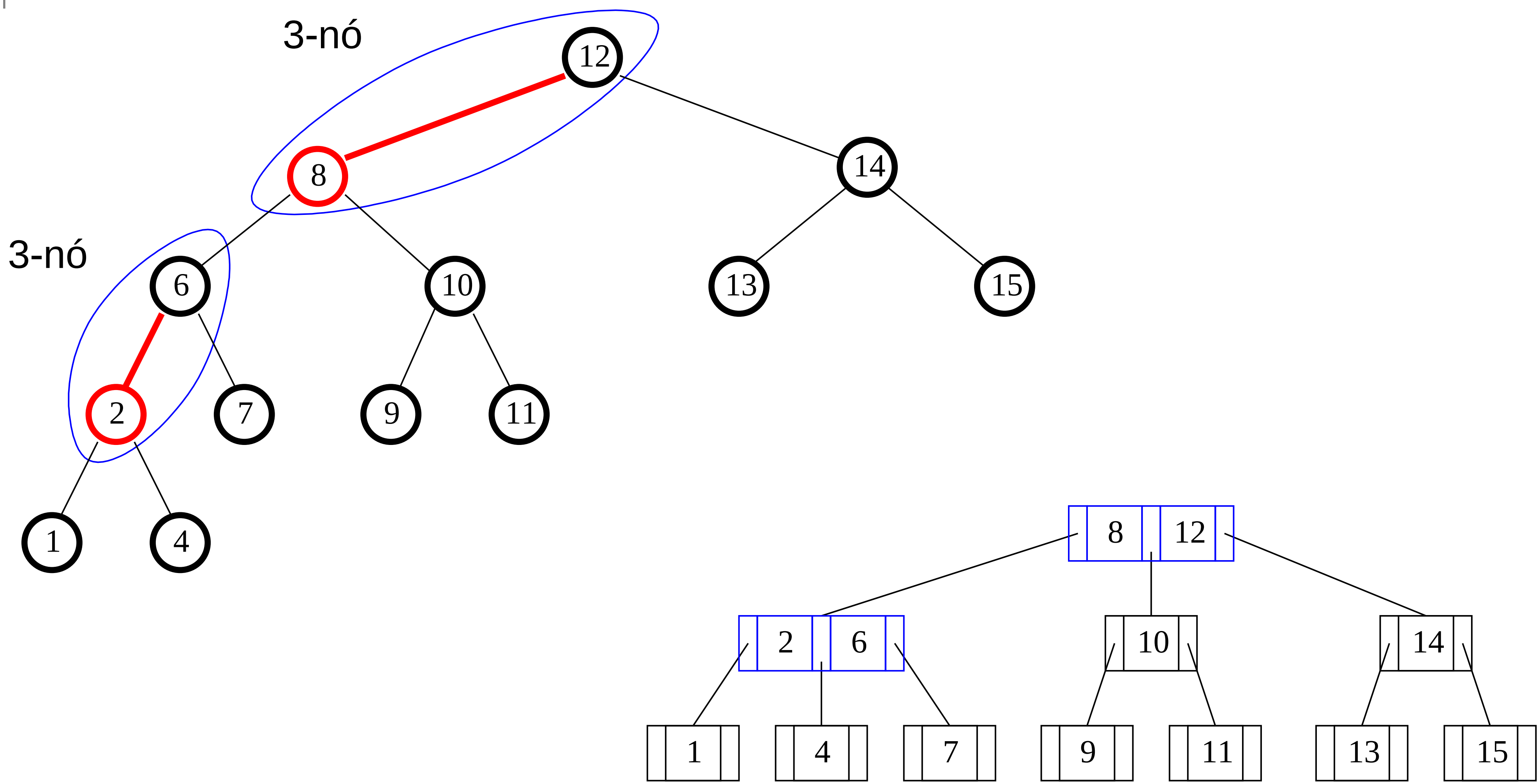
Uma árvore 2-3-4 é uma B-árvore com $m = 3$.

Um nó de uma árvore 2-3 ou 2-3-4 com
- com três chaves é chamado de 4-nó (árvores 2-3-4),
- duas chaves é chamado de 3-nó, e
- com uma chave é chamado de 2-nó.

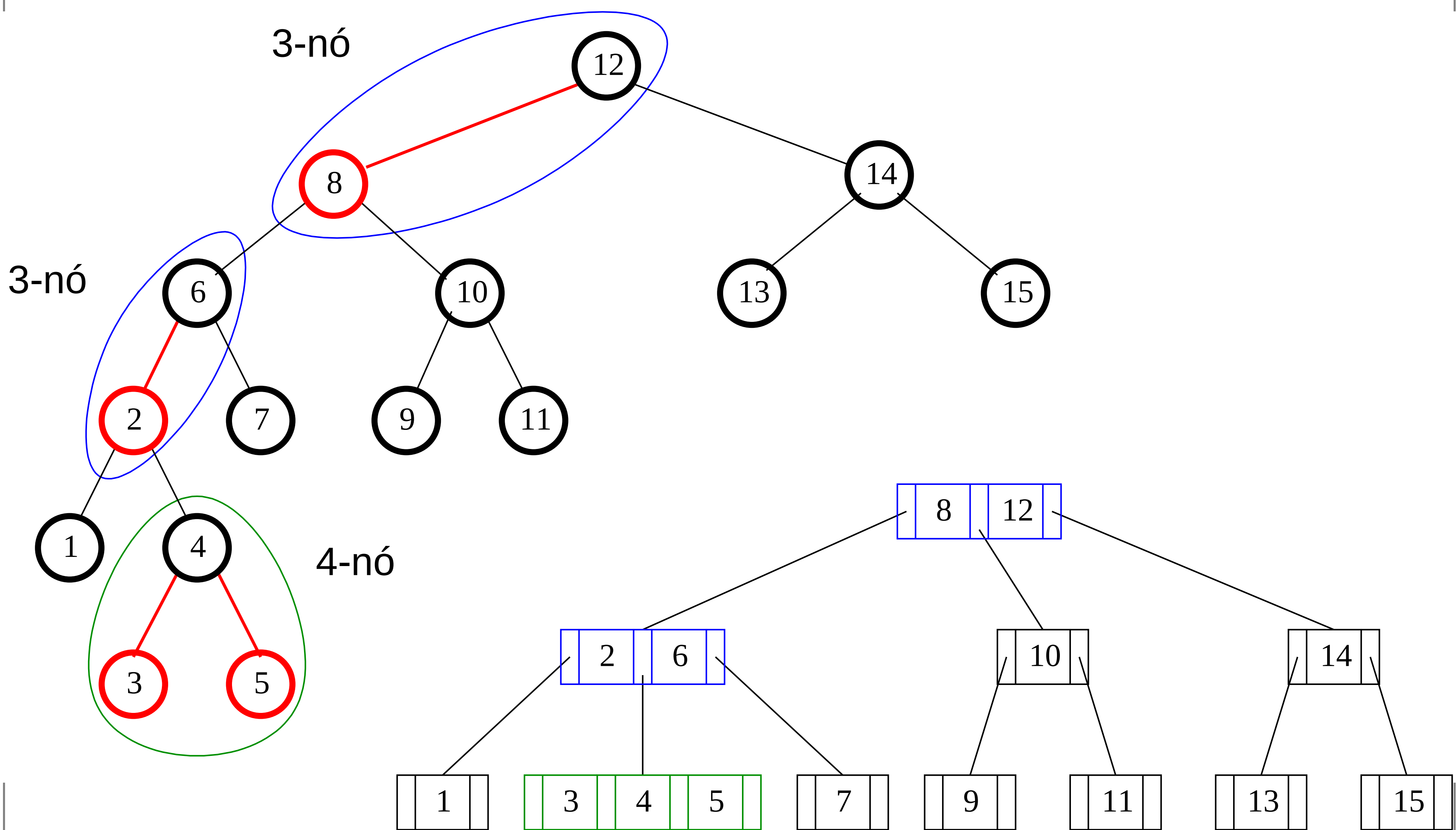
Pode-se representar uma árvore 2-3 por uma rubro-negra em que todo nó negro tem no máximo um filho rubro.

Pode-se representar uma árvore 2-3-4 por uma árvore rubro-negra.

Árvores 2-3 × rubro-negras



Árvores 2-3-4 × rubro-negras



Rotações

O nó p é tal que $dir(p) \neq \textsc{nil}$ e $cor(dir(p)) = $ RUBRO.

ROTACIONEESQ (p)

1 $q \leftarrow dir(p)$
2 $dir(p) \leftarrow esq(q)$
3 $esq(q) \leftarrow p$
4 $cor(q) \leftarrow cor(p)$
5 $cor(p) \leftarrow$ RUBRO
6 **devolva** q

Rotações

O nó p é tal que $dir(p) \neq \text{NIL}$ e $cor(dir(p)) = $ RUBRO.

ROTACIONEESQ (p)
1 $q \leftarrow dir(p)$
2 $dir(p) \leftarrow esq(q)$
3 $esq(q) \leftarrow p$
4 $cor(q) \leftarrow cor(p)$
5 $cor(p) \leftarrow$ RUBRO
6 **devolva** q

Exercício: Escreva o ROTACIONEDIR.

Ajusta cores

O nó p é interno.

TROQUECORES (p)
1 $cor(p) \leftarrow$ OUTRACOR$(cor(p))$
2 $cor(esq(p)) \leftarrow$ OUTRACOR$(cor(esq(p)))$
3 $cor(dir(p)) \leftarrow$ OUTRACOR$(cor(dir(p)))$

Ajusta cores

O nó p é interno.

TROQUECORES (p)
1 $cor(p) \leftarrow$ OUTRACOR$(cor(p))$
2 $cor(esq(p)) \leftarrow$ OUTRACOR$(cor(esq(p)))$
3 $cor(dir(p)) \leftarrow$ OUTRACOR$(cor(dir(p)))$

OUTRACOR (c)
1 **se** $c =$ RUBRO
2 **então devolva** NEGRO
3 **senão devolva** RUBRO

Inserção em ABB rubro-negra

RUBRO (p)

se $p = \text{NIL}$
  então devolva FALSO
  senão se $cor(p) =$ RUBRO
    então devolva VERDADE
    senão devolva FALSO

NEGRO (p)

devolva não RUBRO(p)

Inserção em ABB rubro-negra

RUBRO (p)
1 **se** $p = \text{NIL}$
2   **então devolva** FALSO
3   **senão se** $cor(p) = $ RUBRO
4   **então devolva** VERDADE
5   **senão devolva** FALSO

NEGRO (p)
1 **devolva** não RUBRO(p)

INSIRA (T, x)
1 $T \leftarrow$ INSIRAREC(T, x)
2 $cor(T) \leftarrow$ NEGRO $\rhd$ a raiz é sempre negra

Inserção em ABB rubro-negra 2-3

Esta é a inserção para árvores 2-3:

INSIRAREC (T, x)
se $T = \text{NIL}$
  então $q \leftarrow$ NOVACÉLULA$(x, \text{NIL}, \text{NIL},$ RUBRO$)$
    devolva q
se $x < \mathit{info}(T)$
  então $\mathit{esq}(T) \leftarrow$ INSIRAREC$(\mathit{esq}(T), x)$
  senão $\mathit{dir}(T) \leftarrow$ INSIRAREC$(\mathit{dir}(T), x)$
se RUBRO$(\mathit{dir}(T))$ **e** NEGRO$(\mathit{esq}(T))$
  então $T \leftarrow$ ROTACIONEESQ(T)
se RUBRO$(\mathit{esq}(T))$ **e** RUBRO$(\mathit{esq}(\mathit{esq}(T)))$
  então $T \leftarrow$ ROTACIONEDIR(T)
se RUBRO$(\mathit{esq}(T))$ e RUBRO$(\mathit{dir}(T))$
  então TROQUECORES(T)
devolva T

Inserção em ABB rubro-negra 2-3

Mostre por indução na altura de T que,
no InsiraRec anterior,
se a árvore T for rubro-negra 2-3 exceto pela raiz,
que pode ser rubra, então a árvore T devolvida satisfaz:

- se a raiz de T era negra, então T é rubro-negra 2-3 exceto pela raiz, que pode ser rubra.
- se a raiz de T era rubra, então T é rubro-negra 2-3 exceto pela raiz, que pode ser rubra, e pode ter o filho esquerdo rubro.

Conclua que, se a T for uma árvore rubro-negra 2-3, então a árvore devolvida por Insira(T, x) é rubro-negra 2-3.

Inserção em ABB rubro-negra

Esta é a inserção para árvores 2-3-4:

InsiraRec (T, x)

se $T = \text{NIL}$
  então $q \leftarrow$ NovaCélula$(x, \text{NIL}, \text{NIL},$ RUBRO$)$
    devolva q
se Rubro$(esq(T))$ e Rubro$(dir(T))$
  então TroqueCores(T)
se $x < info(T)$
  então $esq(T) \leftarrow$ InsiraRec$(esq(T), x)$
  senão $dir(T) \leftarrow$ InsiraRec$(dir(T), x)$
se Rubro$(dir(T))$ **e** Negro$(esq(T))$
  então $T \leftarrow$ RotacioneEsq(T)
se Rubro$(esq(T))$ **e** Rubro$(esq(esq(T)))$
  então $T \leftarrow$ RotacioneDir(T)
devolva T

Inserção em ABB rubro-negra

Faça uma análise da correção desta versão do INSIRAREC.

Conclua da sua análise que, se a T for uma árvore rubro-negra 2-3-4, então a árvore devolvida por INSIRA(T, x) é rubro-negra 2-3-4.

Dica: ajuste as duas condições da prova por indução anterior para árvores rubro-negras 2-3-4, fortalecendo a primeira. Refaça a análise de casos cuidadosamente, para ver se os seus ajustes estão corretos. (Em outras palavras, faça a prova por indução.)

Remoção em ABB

Supõe que x está presente na ABB.

REMOVA (T, x)
$T \leftarrow$ REMOVAREC(T, x)

REMOVAREC (T, x)
se $x < info(T)$
    então $esq(T) \leftarrow$ REMOVAREC$(esq(T), x)$
    senão se $x > info(T)$
        então $dir(T) \leftarrow$ REMOVAREC$(dir(T), x)$
        senão se $dir(T) = \text{NIL}$
            então devolva $esq(T)$
            $p \leftarrow$ MÍNIMO$(dir(T))$
            $info(T) \leftarrow info(p)$
            $dir(T) \leftarrow$ REMOVAMIN$(dir(T))$
devolva T

Remoção em ABB

REMOVAMIN (T)

1 **se** $esq(T) = \text{NIL}$

2 **então devolva** $dir(T)$

3 **senão** $esq(T) \leftarrow$ REMOVAMIN$(esq(T))$

4 **devolva** T

Remoção em ABB

REMOVAMIN (T)

se $esq(T) = \text{NIL}$

  então devolva $dir(T)$

  senão $esq(T) \leftarrow$ REMOVAMIN$(esq(T))$

    devolva T

Exercício: Modifique as rotinas acima para que a remoção funcione mesmo que x não esteja na ABB dada.

RemovaMin em árvores 2-3

REMOVAMIN (T)

se $esq(T) = \text{NIL}$

    então devolva NIL   ▷ $dir(T) = \text{NIL}$ aqui

se NEGRO$(esq(T))$ **e** NEGRO$(esq(esq(T)))$

    então $T \leftarrow$ MOVARUBROESQ(T)

$esq(T) \leftarrow$ REMOVAMIN$(esq(T))$

▷ Fixup

se RUBRO$(dir(T))$ **e** NEGRO$(esq(T))$

    então $T \leftarrow$ ROTACIONEESQ(T)

se RUBRO$(esq(T))$ **e** RUBRO$(dir(T))$

    então TROQUECORES(T)

devolva T

RemovaMin em árvores 2-3

MOVARUBROESQ (p)

TROQUECORES(p)
se RUBRO$(esq(dir(p)))$
    então $dir(p) \leftarrow$ ROTACIONEDIR$(dir(p))$
        $p \leftarrow$ ROTACIONEESQ(p)
        TROQUECORES(p)
devolva p

RemovaMin em árvores 2-3

MovaRubroEsq (p)

TroqueCores(p)
se Rubro$(esq(dir(p)))$
    então $dir(p) \leftarrow$ RotacioneDir$(dir(p))$
            $p \leftarrow$ RotacioneEsq(p)
            TroqueCores(p)
devolva p

MovaRubroDir (p)

TroqueCores(p)
se Rubro$(esq(esq(p)))$
    então $p \leftarrow$ RotacioneDir(p)
            $dir(p) \leftarrow$ RotacioneEsq$(dir(p))$
            TroqueCores(p)
devolva p

Remoção em ABB rubro-negra 2-3

REMOVA (T, x)
1 $T \leftarrow$ REMOVAREC(T, x)
2 $cor(T) \leftarrow$ NEGRO ▷ a raiz é sempre negra

Remoção em ABB rubro-negra 2-3

REMOVA (T, x)

$T \leftarrow$ REMOVAREC(T, x)
$cor(T) \leftarrow$ NEGRO ▷ a raiz é sempre negra

FIXUP (T)

se RUBRO$(dir(T))$ **e** NEGRO$(esq(T))$
  então ROTACIONEESQ(T)
se RUBRO$(esq(T))$ **e** RUBRO$(dir(T))$
  então TROQUECORES(T)
devolva T

Remoção em ABB rubro-negra 2-3

RemovaRec (T, x)

se $x < info(T)$
  então se Negro$(esq(T))$ **e** Negro$(esq(esq(T)))$
      então $T \leftarrow$ MovaRubroEsq(T)
    $esq(T) \leftarrow$ RemovaRec$(esq(T), x)$
  senão se Rubro$(esq(T))$
      então $T \leftarrow$ RotacioneDir(T)
    se $x = info(T)$ **e** $dir(T) = \text{NIL}$
      então devolva NIL
    se Negro$(dir(T))$ **e** Negro$(esq(dir(T)))$
      então $T \leftarrow$ MovaRubroDir(T)
    se $x = info(T)$
      então $info(T) \leftarrow$ Mínimo$(dir(T))$
          $dir(T) \leftarrow$ RemovaMin$(dir(T))$
      senão $T \leftarrow$ RemovaRec$(dir(T), x)$
FixUp(T)

Remoção em ABB rubro-negra 2-3

Exercício: Mostre que o REMOVAMIN funciona quando T é uma árvore rubro-negra 2-3 exceto pela raiz, que pode ser rubra, ou é uma árvore rubro-negra em que o filho esquerdo da raiz é rubro.

Exercício: Mostre que o REMOVAREC funciona quando T é uma árvore rubro-negra 2-3 exceto pela raiz, que pode ser rubra, ou é uma árvore rubro-negra em que o filho esquerdo da raiz é rubro.

RemovaMin

REMOVAMIN (T)

se $esq(T) = \text{NIL}$
    então devolva NIL $\triangleright$ $dir(T) = \text{NIL}$ aqui
se NEGRO$(esq(T))$ **e** NEGRO$(esq(esq(T)))$
    então $T \leftarrow$ MOVARUBROESQ(T)
$esq(T) \leftarrow$ REMOVAMIN$(esq(T))$
FIXUP T

RemovaMin

REMOVAMIN (T)
1 **se** $esq(T) = \text{NIL}$
2 **então devolva** NIL ▷ $dir(T) = \text{NIL}$ aqui
3 **se** NEGRO$(esq(T))$ **e** NEGRO$(esq(esq(T)))$
4 **então** $T \leftarrow$ MOVARUBROESQ(T)
5 $esq(T) \leftarrow$ REMOVAMIN$(esq(T))$
6 FIXUP T

REMOVA (T, x)
1 $T \leftarrow$ REMOVAREC(T, x)
2 $cor(T) \leftarrow$ NEGRO ▷ a raiz é sempre negra

Remoção em ABB rubro-negra

$\textsc{RemovaRec}\ (T, x)$

se $x < info(T)$
  então se $\textsc{Negro}(esq(T))$ **e** $\textsc{Negro}(esq(esq(T)))$
      então $T \leftarrow \textsc{MovaRubroEsq}(T)$
    $esq(T) \leftarrow \textsc{RemovaRec}(esq(T), x)$
  senão se $\textsc{Rubro}(esq(T))$
      então $T \leftarrow \textsc{RotacioneDir}(T)$
    se $x = info(T)$ **e** $dir(T) = \textsc{nil}$
      então devolva $\textsc{nil}$
    se $\textsc{Negro}(dir(T))$ **e** $\textsc{Negro}(esq(dir(T)))$
      então $T \leftarrow \textsc{MovaRubroDir}(T)$
    se $x = info(T)$
      então $info(T) \leftarrow \textsc{Mínimo}(dir(T))$
          $dir(T) \leftarrow \textsc{RemovaMin}(dir(T))$
      senão $dir(T) \leftarrow \textsc{RemovaRec}(dir(T), x)$
$\textsc{FixUp}(T)$

MovaRubroEsq para rubro-negra 2-3-4

MovaRubroEsq (p)

TroqueCores(p)
se Rubro$(esq(dir(p)))$
então $dir(p) \leftarrow$ RotacioneDir$(dir(p))$
$p \leftarrow$ RotacioneEsq(p)
TroqueCores(p)
se Rubro$(dir(dir(p)))$
então $dir(p) \leftarrow$ RotacioneEsq$(dir(p))$
devolva p

MovaRubroDir para rubro-negra 2-3-4

MovaRubroDir (p)

1 TroqueCores(p)
2 **se** Rubro$(esq(esq(p)))$
3 **então** $p \leftarrow$ RotacioneDir(p)
4 TroqueCores(p)
5 **se** Negro$(esq(dir(p)))$
6 **então** $dir(p) \leftarrow$ RotacioneEsq$(dir(p))$
7 **devolva** p

FixUp para ABB rubro-negra 2-3-4

FIXUP (T)

```
se RUBRO(dir(T))
   então T ← ROTACIONEESQ(T)
se RUBRO(esq(T)) e RUBRO(esq(esq(T)))
   então T ← ROTACIONEDIR(T)
se RUBRO(esq(T)) e RUBRO(dir(T))
   então TROQUECORES(T)
devolva T
```

Mais exercícios e referências

Exercício: Mostre que o REMOVAREC funciona quando T é uma árvore rubro-negra 2-3-4 exceto pela raiz, que pode ser rubra, ou é uma árvore rubro-negra em que o filho esquerdo da raiz é rubro ou os dois filhos da raiz são rubros.

Referências:

Artigo:

`http://www.cs.princeton.edu/~rs/talks/LLRB/LLRB.pdf`

Implementação:

`http://www.cs.princeton.edu/~rs/talks/LLRB/Java/`