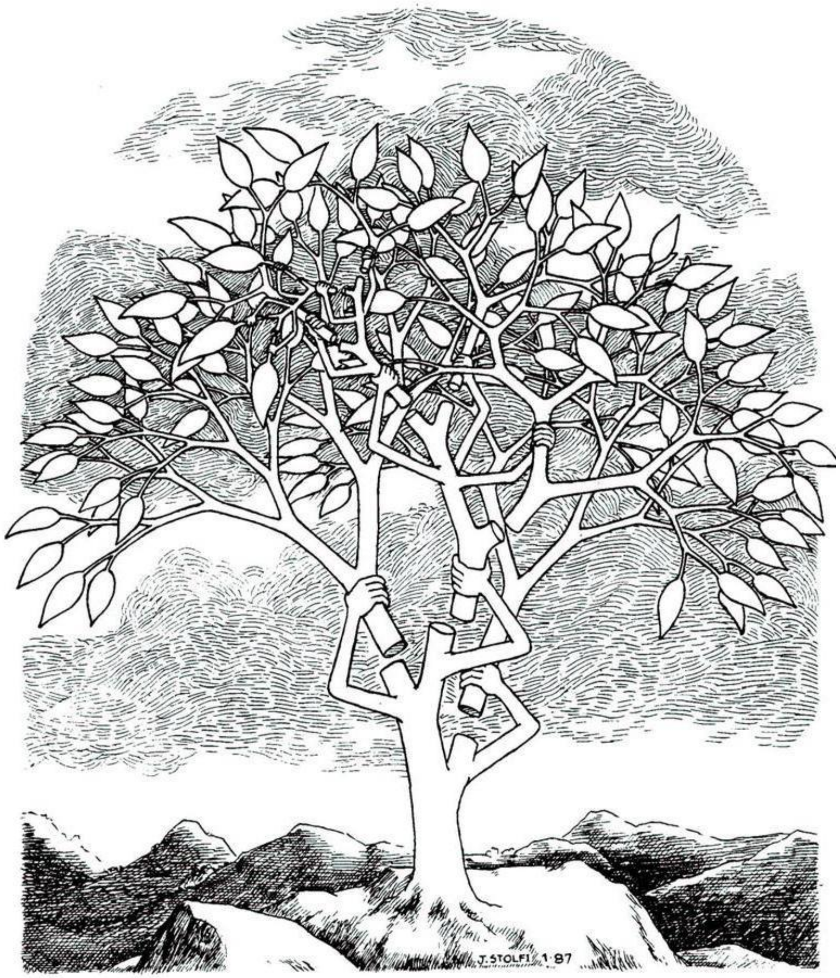


Árvores RN

Árvores Graduadas e Rubro-Negras

Árvores Graduadas e Árvores Rubro-Negras são uma generalização das árvores AVL e preservam a característica do balanceamento.

As árvores graduadas e RN consideram nós externos.

A altura de um nó externo é 0.

Árvores Graduadas

Uma árvore graduada T é uma árvore binária de busca onde a cada nó v pode ser associado um valor inteiro, dito posto(v), satisfazendo:

(i) v é nó externo, então posto(v) = 0

(ii) v é pai de nó externo, então posto(v) = 1

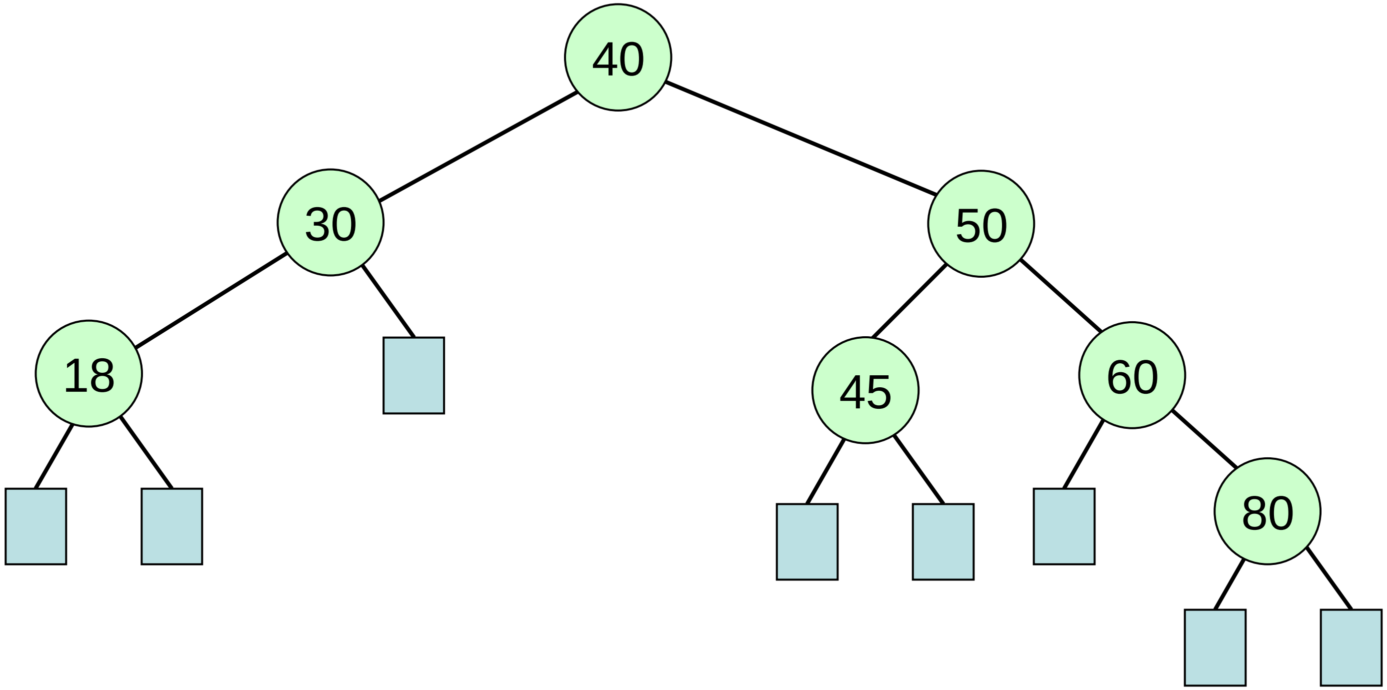
(iii) v possui pai w, então

$$\text{posto}(v) \leq \text{posto}(w) \leq \text{posto}(v) + 1$$

(*iv*) v possui avô w, então posto(v) < posto(w)

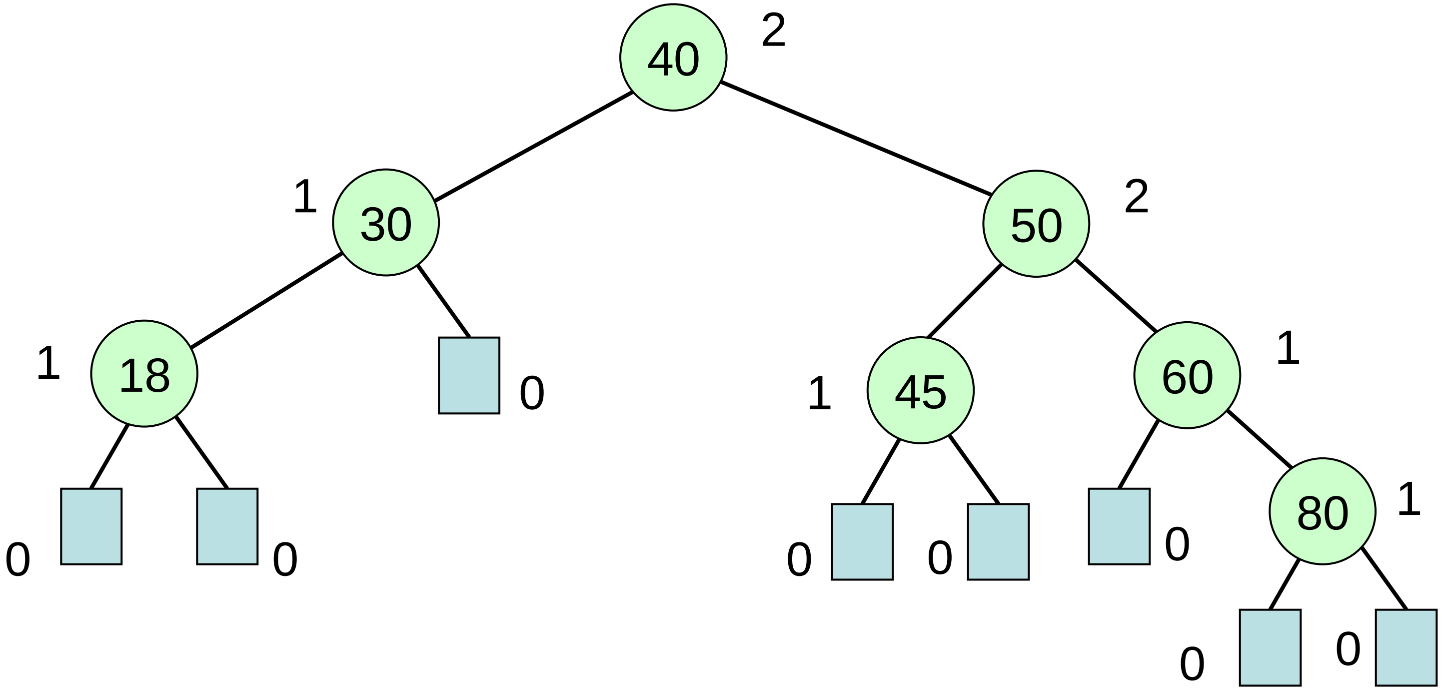
Árvores Graduadas

Essa árvore é graduada?



Árvores Graduadas

Exemplo:



Árvores Graduadas

Definição de ***nó equilibrado***:

Um nó *v* tal que posto(*v*) satisfaz (*i*)-(*iv*) é dito **equilibrado**, caso contrário, é dito **desequilibrado**.

Obs1. T é graduada se e somente se todos os nós de T são equilibrados.

Obs2. Toda árvore AVL é graduada.

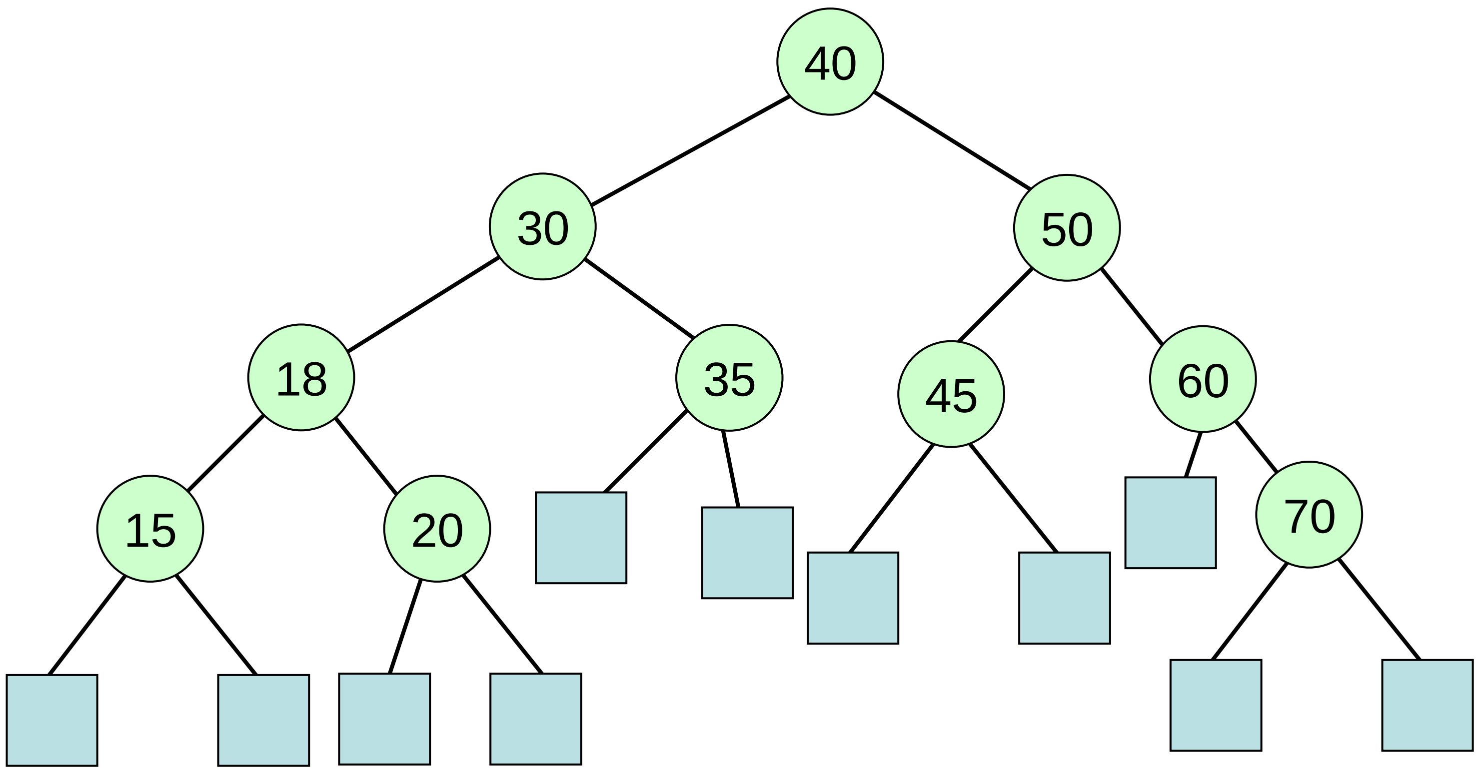
Árvore Rubro-Negra

Uma árvore binária de busca onde a cada nó pode ser atribuído um rótulo (cor) R (rubro) ou N (negro) tal que:

1. Se v é nó externo, então $cor(v) = N$.
2. Todos os caminhos de um nó v a seus descendentes externos possui o mesmo número de nós negros.
3. Se $cor(v) = R$ e $v \neq raiz$, então $cor(pai(v))=N$.

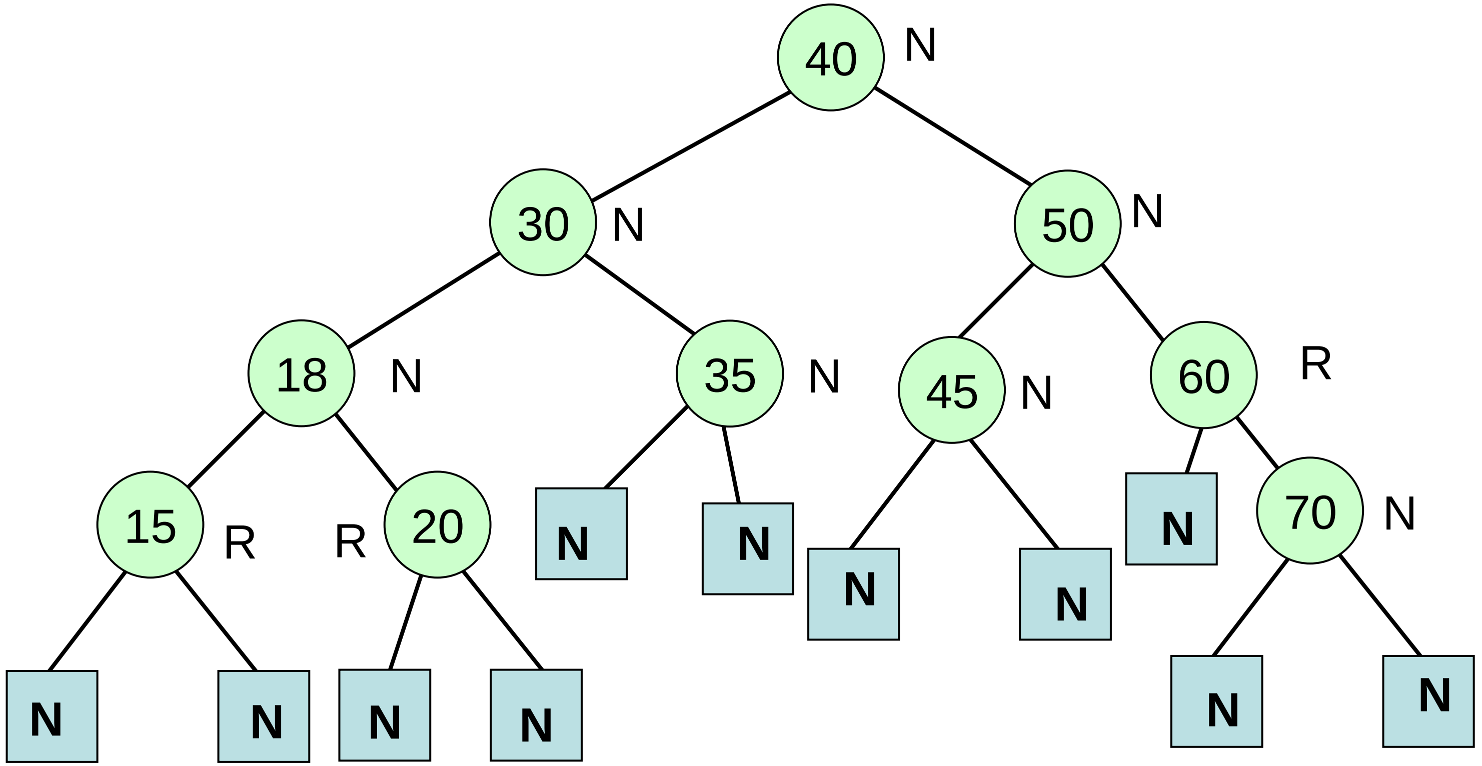
Árvore Rubro-Negra

Essa árvore é rubro-negra?



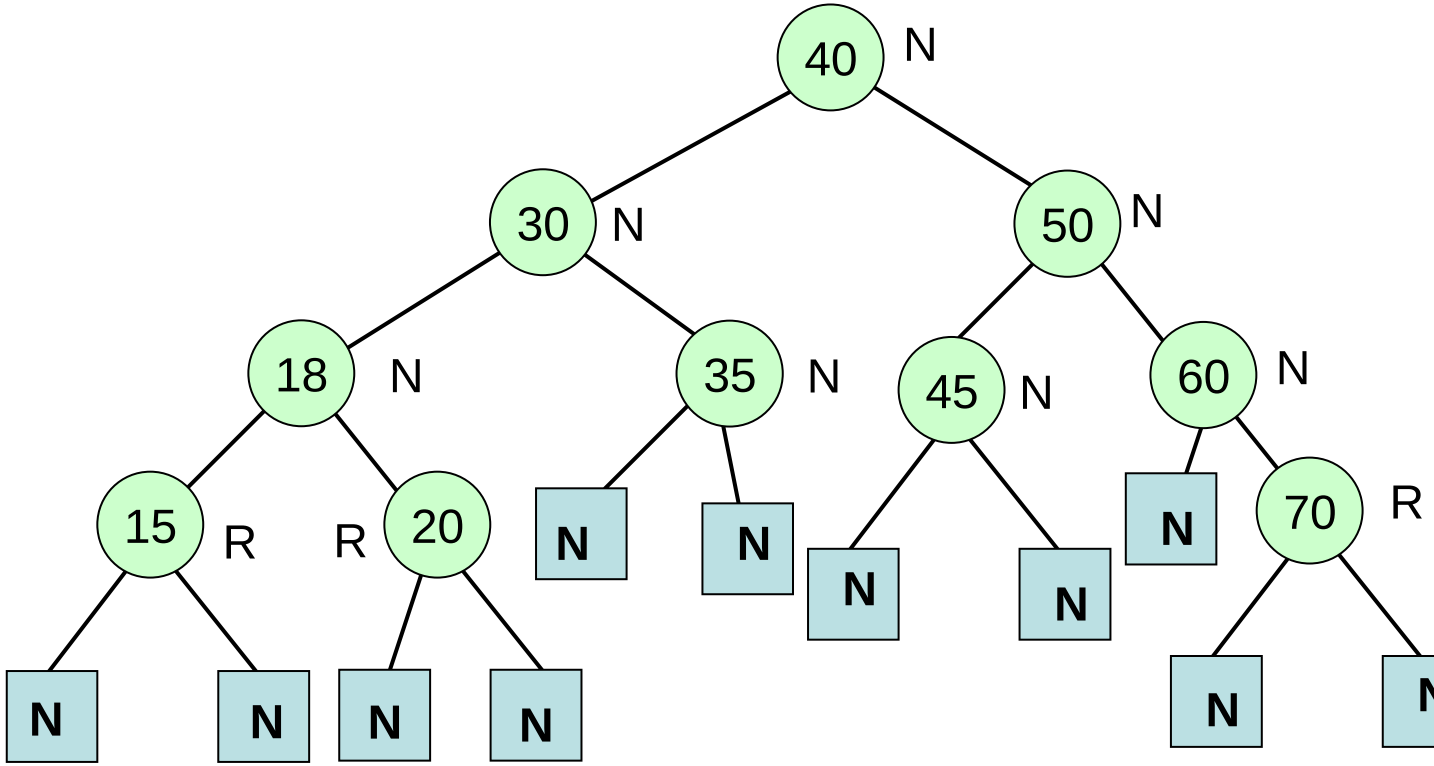
Árvore Rubro-Negra

Esta rotulação está correta?



Árvore Rubro-Negra

Rotulação correta



Árvore Rubro-Negra

APLICAÇÃO

CFS (Completely Fair Scheduler) é o algoritmo que faz o agendamento de processos no Linux.

Ele gerencia a alocação de recursos para execução de processos e tem como objetivo maximizar a utilização geral da CPU, também maximizando o desempenho interativo.

Referência: Tong Li, Dan Baumberger, Scott Hahn (2009) “Efficient and Scalable Multiprocessor Fair Scheduling Using Distributed Weighted Round-Robin”, Proceedings of the 14th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, 65-74.

Árvores Graduadas e Rubro-Negras

Teorema:

Uma árvore é graduada se e somente se é rubro-negra

Para mostrar isso veremos as Conversões A e B.

Árvores Graduadas e Rubro-Negras

Conversão A: Graduada $\rightarrow$ Rubro-negra

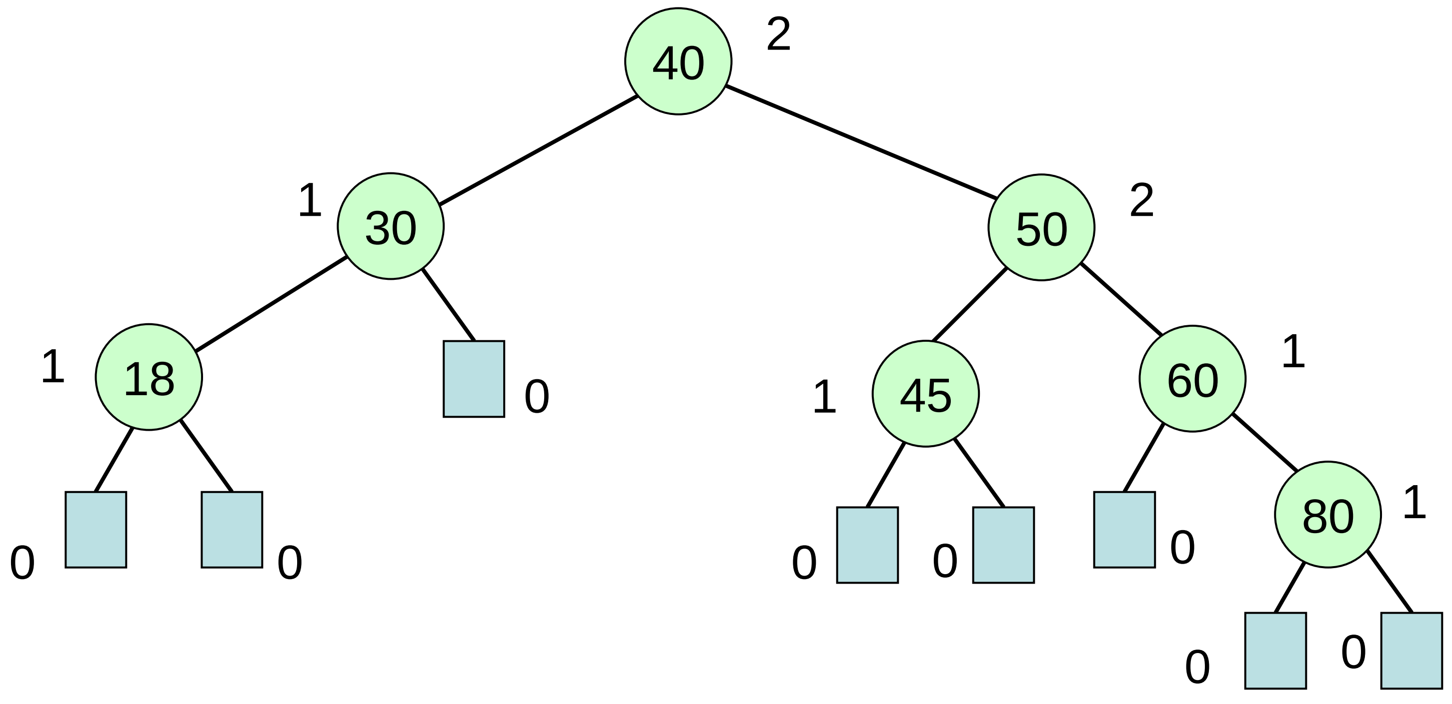
Seja T uma árvore graduada. Atribui-se uma cor R ou N a cada nó de T, percorrendo-se a árvore de cima para baixo

a. Atribui-se cor N a raiz de T

b. Seja *v* um nó ≠ raiz e *w* pai de *v*
   cor(v) = R, se posto(v) = posto(w)
   cor(v) = N, caso contrário

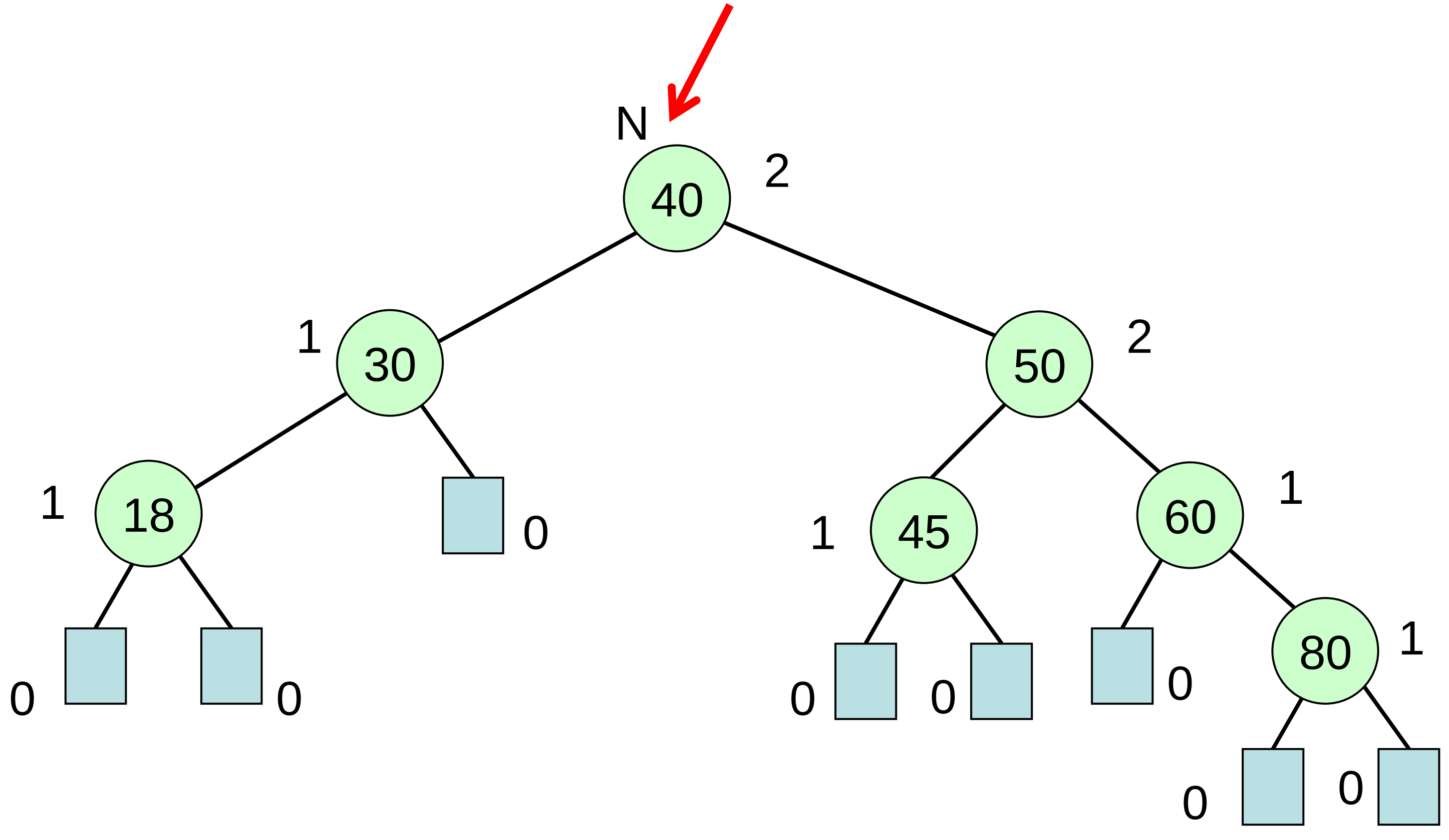
Árvores Graduadas e Rubro-Negras

Exercício: Aplicar a conversão A



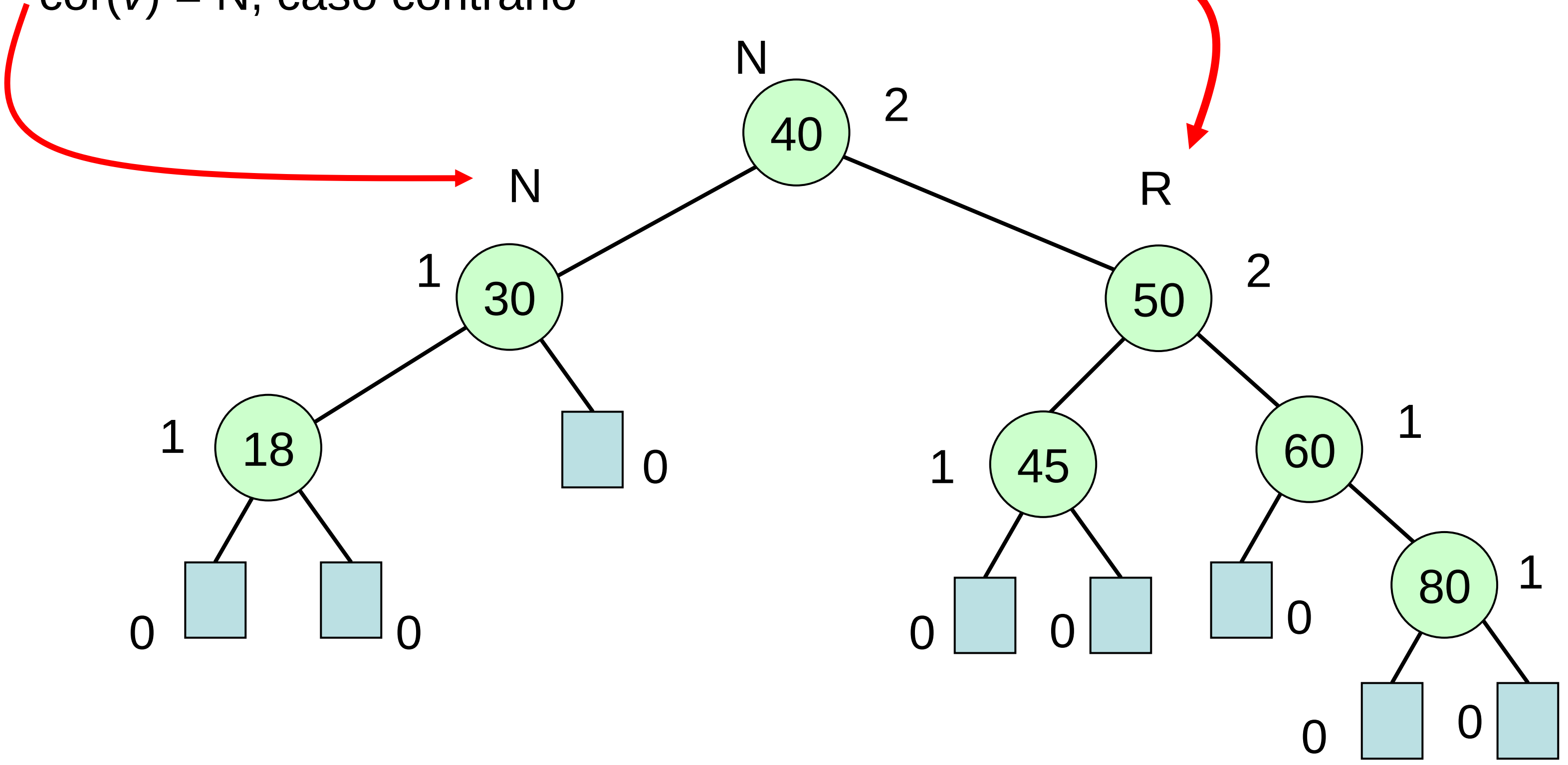
Árvores Graduadas e Rubro-Negras

a. Atribui-se cor N a raiz de T



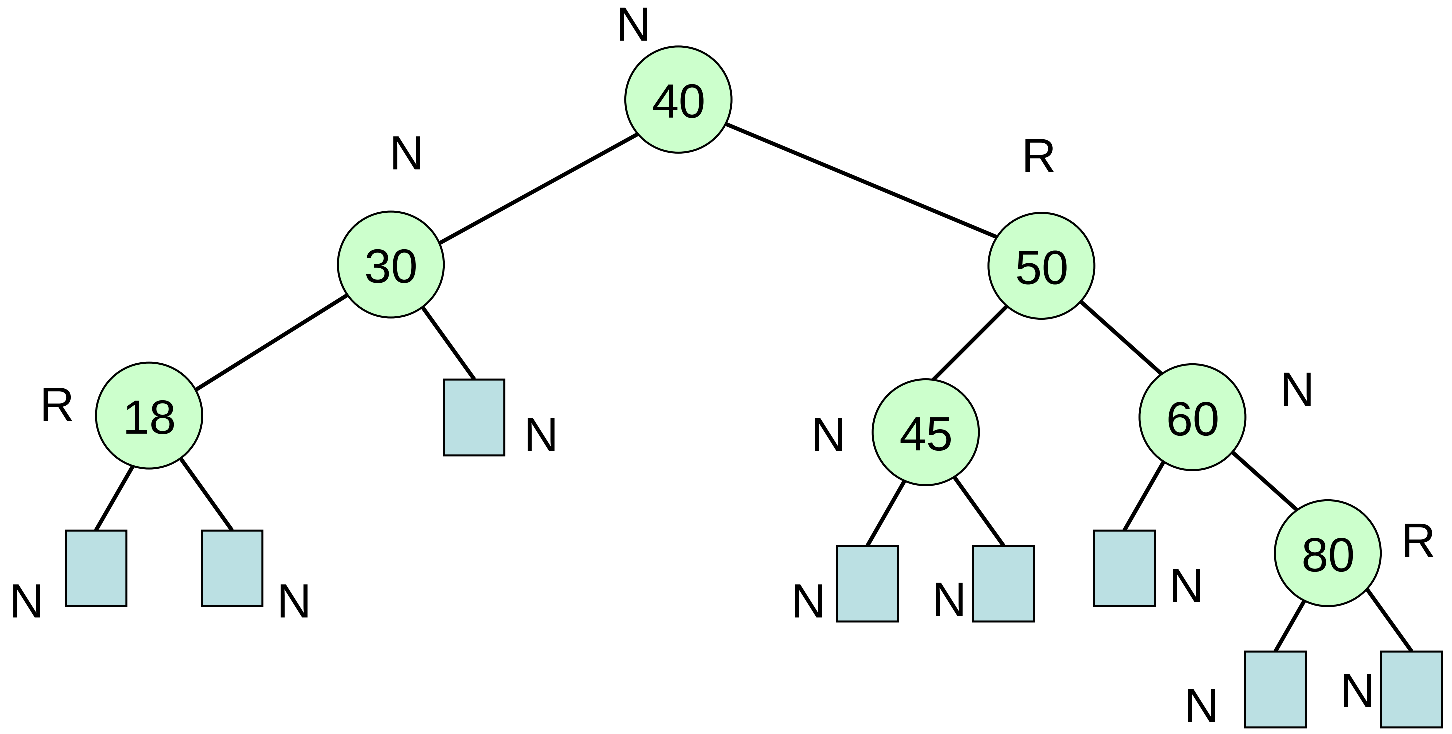
Árvores Graduadas e Rubro-Negras

b. Seja *v* um nó ≠ raiz e *w* pai de *v*
   cor(*v*) = R, se posto(*v*) = posto(*w*)
   cor(*v*) = N, caso contrário



Árvores Graduadas e Rubro-Negras

b. Seja *v* um nó ≠ raiz e *w* pai de *v*
cor(*v*) = R, se posto(*v*) = posto(*w*)
cor(*v*) = N, caso contrário



Árvores Graduadas e Rubro-Negras

Conversão B: Rubro-negra $\rightarrow$ Graduada

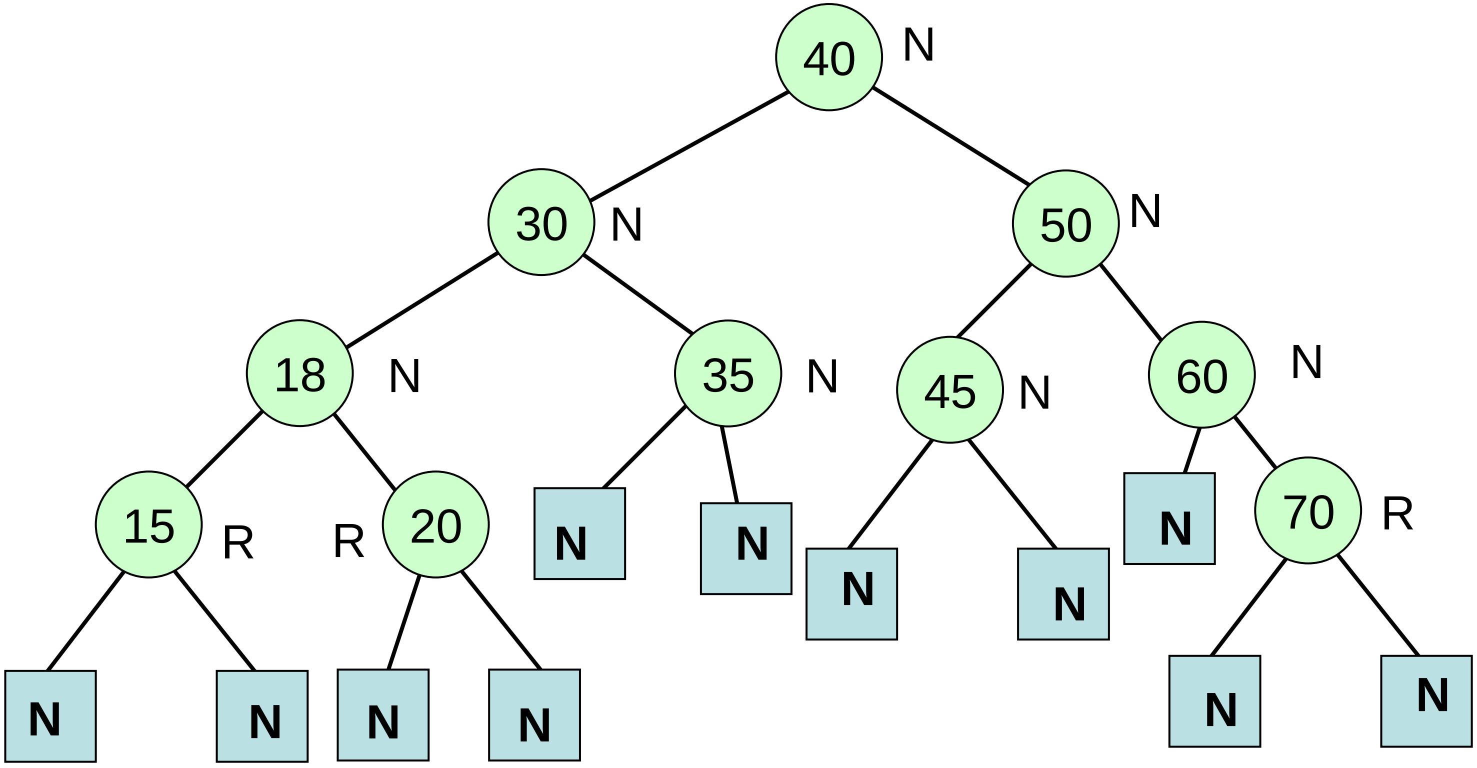
Seja T uma árvore RN. Atribui-se um valor posto(*v*) a cada nó *v* de T, percorrendo-se a árvore de baixo para cima.

a. Atribui-se posto(*v*) = 0 para todos os nós externos

b. Seja *v* um nó ≠ externo e *q* filho de *v*
   Se cor(*q*) = R, então posto(*v*) = posto(*q*)
   Se cor(*q*) = N, então posto(*v*) = posto(*q*) + 1

Árvores Graduadas e Rubro-Negras

Exercício: Aplicar a conversão B



Árvores Graduadas e Rubro-Negras

Lema 5.1: Seja T uma árvore RN na qual foi aplicada a conversão B, então:

(i) $v \neq$ nó externo, posto(v) independe do filho considerado pelo algoritmo

(ii) posto(v) = número de nós negros no caminho de um filho de v até qualquer descendente externo.

Árvores Graduadas e Rubro-Negras

Prova: Indução em *h*

Obs: Se v é nó externo, $h(v) = 0$.
Se v é folha, $h(v) = 1$.

(i) $v \neq$ nó externo, posto(v) independe do filho considerado pelo algoritmo

(ii) posto(v) = número de nós negros no caminho de um filho de v até qualquer descendente externo.

Base da Indução:
Se $h(v) = 0$, v é nó externo
(*i*) vale trivialmente
(*ii*) vale pois v não tem filhos e posto(v) = 0.

Hipótese da Indução: Considere o lema válido para todo nó com altura < $h(v)$.

Árvores Graduadas e Rubro-Negras

Passo da Indução:

Seja NN o número de nós negros no caminho de um filho de v até qualquer descendente externo.

Se $h(v) > 0$, então v possui filhos w_1 e w_2.

Seja w_i o filho de maior altura, assim $h(v) = h(w_i) + 1$.

Pela hipótese de indução o lema vale para w_i.

Logo, posto(w_i) = $NN(w_i)$.

(i) $v \neq$ nó externo, posto(v) independe do filho considerado pelo algoritmo

(ii) posto(v) = número de nós negros no caminho de um filho de v até qualquer descendente externo.

Árvores Graduadas e Rubro-Negras

(i) $v \neq$ nó externo, posto(v) independe do filho considerado pelo algoritmo

(ii) posto(v) = número de nós negros no caminho de um filho de v até qualquer descendente externo.

Caso 1. v é R

Se v é R, então seus filhos são N.

$NN(v) = NN(w_1) + 1$, $NN(w_1) = NN(v) - 1$
$NN(v) = NN(w_2) + 1$, $NN(w_2) = NN(v) - 1$

$$NN(w_1) = NN(w_2)$$

Por indução:

posto(w_1) = $NN(w_1)$ e posto(w_2) = $NN(w_2)$
posto(w_1) = posto(w_2) (*i*) – Provado

Pela conversão:

posto(v) = posto(w_1) + 1
posto(v) = $NN(w_1) + 1 = NN(v)$ (*ii*) - Provado

Árvores Graduadas e Rubro-Negras

(i) $v \neq$ nó externo, posto(v) independe do filho considerado pelo algoritmo

(ii) posto(v) = número de nós negros no caminho de um filho de v até qualquer descendente externo.

Caso 2. v é N

Supor w_i é N

$$NN(v) = NN(w_i) + 1,\ \text{posto}(w_i) = NN(w_i)$$

$$NN(v) = \underbrace{\text{posto}(w_i)}_{NN(w_i)} + \underbrace{1}_{w_i} = \text{posto}(v)$$

Supor w_i é R

$$NN(v) = NN(w_i) = \text{posto}(w_i) \text{ e}$$

$$NN(v) = \text{posto}(v)$$

Árvores Graduadas e Rubro-Negras

Portanto, a escolha do filho é indiferente para a determinação do posto(v). ■

Árvores Graduadas e Rubro-Negras

Lema 5.2: Seja T uma árvore graduada na qual foi aplicada a conversão A, então o número de nós negros no caminho de um filho do nó *v* até seus descendentes externos é posto(*v*).

Prova: A conversão é feita de cima para baixo atribuindo cor **N** a raiz. De (*iii*) sabe-se que o decremento do posto de pai para filho é no máximo 1.

Árvores Graduadas e Rubro-Negras

Esse decremento corresponde a um nó com a cor N pela conversão A. Como todo nó externo tem posto 0, por (*i*) o número de nós negros é o mesmo para qualquer caminho a partir de um filho do nó dado e equivale ao seu posto.

Árvores Graduadas e Rubro-Negras

Teorema: Uma árvore é graduada
se e somente se
é rubro-negra

Prova

1. Se T é graduada então T é rubro-negra.

Considere v e w nós de T, w pai de v.

Pela conversão A, sabe-se que:
cor(v) = R se posto(w) = posto(v)
Caso contrário, cor(v) = N

Árvores Graduadas e Rubro-Negras

Por (*i*) e (*ii*) sabe-se que se *v* é nó externo, então posto(*v*) ≠ posto(*w*)

Então, cor(*v*) = N ■

(a) está satisfeito.

(b) é satisfeito no Lema 5.2.

a. Se *v* é nó externo, então cor(*v*) = N.
b. Todos os caminhos de um nó *v* a seus descendentes externos possui o mesmo número de nós negros.
c.Se cor(*v*) = R e *v* ≠ raiz, então cor(pai(*v*))=N.

Considere que cor(*v*) = R e *w* pai de *v*.

Se *w* é raiz da árvore, então cor(*w*) = N, por construção

Árvores Graduadas e Rubro-Negras

Se w não é raiz, considere u avô de v. Como cor(v) = R, então posto(v) = posto(w).

Por (iv) posto(u) ≠ posto(w) e cor(w) = N.
(c) está satisfeito. ■

Logo, T é rubro-negra.

Árvores Graduadas e Rubro-Negras

Prova

2. Se T é rubro-negra então T é graduada.

Considere a conversão B em T.

Pelo Lema 5.1, sabe-se que o posto(*v*) é calculado a partir de qualquer um de seus filhos.

Se *v* é nó externo, então é N e posto(*v*) = 0

Para *w* pai de *v* tem-se posto(*w*)=posto(*v*) + 1

(*i*) e (*ii*) estão satisfeitas.

Árvores Graduadas e Rubro-Negras

- Considere o nó v, seu pai w, seu avô u (se existir)

Se cor(v) = R, então posto(w0 = posto(v)
Senão,
posto(w) = posto(v) + 1

(iii) é satisfeita

- Tem-se ou cor(v) = N ou cor(w) = N, ou ambas. Logo, posto(u) > posto(v)

(iv) é satisfeita

Logo T é graduada

Balanceamento de Árvores Rubro-Negras

Lema 5.3: Seja T uma árvore graduada com n nós internos e altura h. Então

$$h(v) \leq 2 \times \text{posto}(v)$$

Prova: Sejam atribuídos rótulos R e N conforme a conversão A.

A prova é por indução em h.

Caso Base: $h(v) = 0$, v é nó externo e posto(v) = 0.
Vale o lema ♦

Balanceamento de Árvores Rubro-Negras

Quando $h(v) > 0$, supor o lema verdadeiro para todos os nós w tais que $h(w) < h(v)$.

Sejam w_1 e w_2 filhos de v e w_i o filho de maior altura.

Sabe-se que $h(w_i) < h(v)$. Então o lema vale para os filhos de v, logo $h(w_i) \leq 2 \times \text{posto}(w_i)$.

Dois casos podem ocorrer:

1. $\text{posto}(w_i) = \text{posto}(v) - 1$
2. $\text{posto}(w_i) = \text{posto}(v)$

Balanceamento de Árvores Rubro-Negras

Caso 1. posto(w_i) = posto(v) -1

$h(w_i) \leq 2 \times \text{posto}(w_i) < 2 \times \text{posto}(v)$

Como $h(v) = h(w_i) + 1$, então $h(v) \leq 2 \times \text{posto}(v)$ ♦

Caso 2. posto(w_i) = posto(v)
Sabe-se que w_i é R e v é N.

Do lema 5.2, tem-se que posto(w_i) = NN(w_i).
Chamando de NR(v) o maior número de nós rubros existentes entre v e um descendente, tem-se:

Balanceamento de Árvores Rubro-Negras

$h(w_i) = \text{NN}(w_i) + \text{NR}(w_i)$

Como, $\text{NR}(w_i) < \text{NN}(w_i)$

(w_i é R e não é contado, o nó externo é obrigatoriamente N, não existem 2 nós R consecutivos)

Então, $\text{NR}(w_i) \leq \text{NN}(w_i) - 1$

$$h(w_i) \leq \text{NN}(w_i) + \text{NN}(w_i) - 1 = 2 \times \text{posto}(w_i)$$

$$= 2 \times \text{posto}(v) - 1$$

Como, $h(v) = h(w_i) + 1$

$$h(v) \leq 2 \times \text{posto}(v) \;\blacklozenge$$

Balanceamento de Árvores Rubro-Negras

Seja T uma árvore binária de busca formada de nós externos e internos e v um nó de T.

A subárvore interna de T com raiz v é obtida de $T(v)$ pela exclusão dos nós externos. $T_i(v)$

O lema 5.4 estabelece um limite inferior para o número de nós de uma subárvore interna de uma árvore graduada.

Balanceamento de Árvores Rubro-Negras

Lema 5.4. $|T_l(v)| \geq 2^{\text{posto}(v)} - 1$

Prova. A subárvore $T_l(v)$ possui cardinalidade mínima quando T satisfaz a seguinte condição: todo nó w de pai z possui posto(w) = posto(z) – 1.

Se isto não ocorre, então posto(w) = posto(z). Neste caso, substituir a árvore $T_l(z)$ por $T_l(w)$. A nova árvore é também graduada e $T_l(v)$ possui menos vértices que no caso anterior.

Balanceamento de Árvores Rubro-Negras

Contudo, se posto(w) = posto(z) – 1 para todo nó w de pai z, então T é uma árvore completa, com todas as folhas no mesmo nível (árvore cheia).

Além disso posto(v) $\leq h(v)$ para todo nó v.

Como $2^{h(v)} = |T_l(v)| + 1$ na árvore de tamanho mínimo, então de um modo geral vale

$$|T_l(v)| \geq 2^{\text{posto}(v)} - 1$$

♦

Balanceamento de Árvores Rubro-Negras

Teorema 5.2. Seja T uma árvore graduada com n nós internos e altura h. Então,

$$h(T) \leq 2\lfloor \log(n+1) \rfloor$$

Prova. Seja r a raiz de T.

Pelo lema 5.4, $|T_l(r)| \geq 2^{\text{posto}(r)} - 1$.

Pelo lema 5.3, $h(r) \leq 2 \times \text{posto}(r)$.

$|T_l(r)| = n$ e $h(r) = h(T)$

Então $\log(n+1) \geq \text{posto}(r)$.

Como posto(r) é inteiro, vale $\lfloor \log(n+1) \rfloor \geq \text{posto}(r)$

Logo, $h(T) \leq 2\lfloor \log(n+1) \rfloor$

♦