

MODULUL 5 - III - ARBORI ROȘU-NEGRU

1 Noțiuni generale

Definiție: Un arbore roșu-negru (ARN) este un arbore binar de căutare în care fiecărui nod i se asociază o culoare - roșu sau negru - și care are următoarele proprietăți:

1. Fiecare nod este roșu sau negru - are deci un câmp suplimentar *color*
2. Rădăcina este neagră
3. Fiecare frunză este neagră și NIL
4. Dacă un nod este roșu, ambii fii sunt negri \Rightarrow părintele unui nod roșu este negru.
5. Pentru fiecare nod x , oricare drum de la nod la o frunză NIL se întâlnește același număr de noduri negre (inclusiv frunza NIL și exclusiv nodul de la care se pornește). Acest număr se numește înălțimea neagră a subarborelui de rădăcină x . Notăm înălțimea neagră cu *bh* - *black height*.

Un exemplu de arbore roșu-negru este prezentat în figura 1.

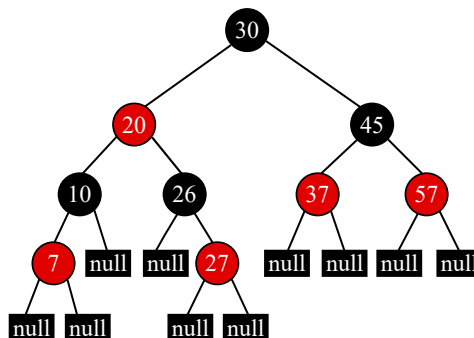


Figura 1: Exemplu de arbore roșu - negru.

Observații:

- Într-un ARN niciun drum de la rădăcină la o frunză nu poate fi mai lung decât dublul unui alt drum la altă frunză. Acest lucru asigură o balansare relativă a arborelui .

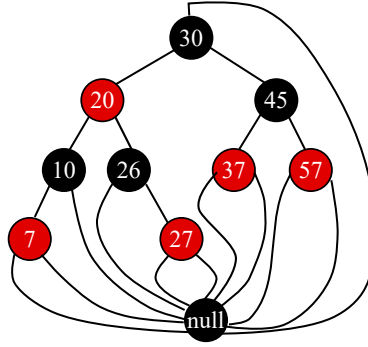


Figura 2: Exemplu de arbore roșu - negru cu santinelă.

- Înălțimea maximă a unui arbore ARN este $2 \log_2(n + 1)$.
Se demonstrează prin inducție că orice subarbore de rădăcină x conține cel puțin $2^{bh(x)} - 1$ noduri interne. Notând cu h înălțimea arborelui și cu r rădăcina, din proprietatea 4 se obține $bh(r) \geq h/2$. Dar $n \geq 2^{bh(r)} - 1 \geq 2^{h/2} - 1$ de unde rezultă $h \leq 2 \log_2(n + 1)$
- Definirea unui nod NIL pentru fiecare frunză presupune un consum inutil de memorie. Din acest motiv se poate considera în locul acestor frunze un singur nod santinelă T.nil către care să indice acele noduri interne care au ca descendenți frunze NIL. Un exemplu de ARN cu santinelă este prezentat în fig. 2.
- Pentru simplitate în continuare vom ignora în desene nodurile NIL.
- Datorită faptului că operațiile de căutare, maxim, minim, succesor, predecesor depind de înălțimea h a arborelui, înseamnă că aceste operații au complexitatea $O(\log_2 n)$.
- Operațiile de inserție și ștergere sunt ceva mai complicate decât în cazul arborilor binari de căutare simpli, deoarece după inserție/ștergere trebuie eventual refăcută structura de arbore roșu-negru.

Prin operații de inserție/ștergere se poate pot strica anumite proprietăți RN. Pentru refacerea acestor proprietăți de arbore ARN, sunt necesare operații de recolorare și de rotație.

1.1 Inserția într-un arbore roșu-negru

Inserția propriu-zisă a unui nod într-un arbore roșu-negru se realizează după același algoritm ca și inserția într-un arbore binar de căutare. Practic se pornește de la rădăcină și se compară la fiecare pas cheia nodului curent x cu cheia nodului z care se inserează, coborându-se în subarborile stâng dacă $z.info < x.info$ și în cel drept altfel. Apoi se refac proprietățile ARN.

Observații:

- Întotdeauna nodul care se inserează are culoarea roșie
- În arborele T se consideră o santinelă $T.nil$

Proprietățile care pot fi neîndeplinite în cazul inserției:

- Proprietățile 1, 3 și 5 se păstrează, datorită faptului că se inserează un nod roșu, care are ca fii două frunze NIL, deci se leagă de nodul santinelă $T.nil$
- Proprietatea 2 poate fi contrazisă dacă nodul inserat este chiar rădăcina sau ca urmare a cazului 1, care va fi discutat în cele ce urmează. Pentru rezolvarea acestei situații este suficientă colorarea rădăcinii cu negru.
- Proprietatea 4 poate fi contrazisă, dacă părintele de care s-a legat noul nod are culoarea roșie. În acest caz trebuie refăcută proprietatea. Există în această situație 3 cazuri care vor fi discutate în continuare.

Funcția de refacere a proprietăților RN este apelată doar dacă părintele nodului inserat este roșu.

Observații:

- Notăm cu z nodul inserat, cu P părintele lui z . Datorită faptului că P are culoarea roșie, deja înainte de inserție, iar inserția se produce într-un arbore roșu-negru valid, înseamnă că P nu este rădăcină și deci există un nod părinte al lui $P \neq T.nil$ pe care îl notăm cu B . Notăm cu U unchiu (fratele părintelui).
- P se poate afla la stânga sau la dreapta lui B . Cele două cazuri se tratează în mod similar, prin simetrie. Vom considera în continuare inserția pe stânga bunicului B .

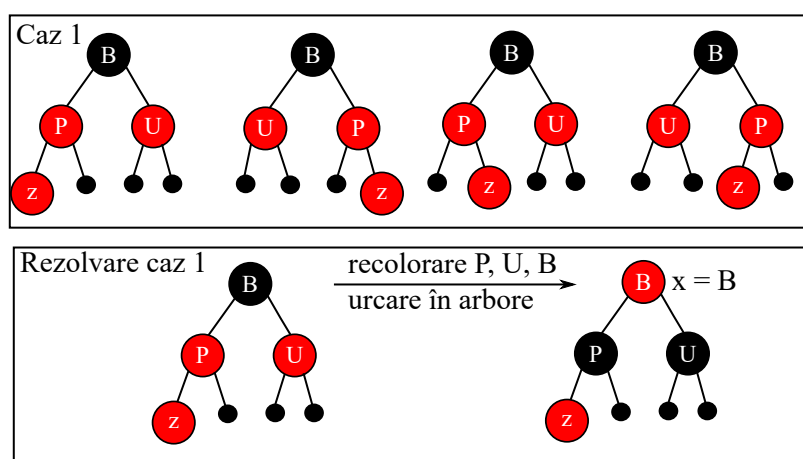


Figura 3: Cazul 1 pentru inserție.

Cazurile de refacere a proprietății 4 în urma inserției:

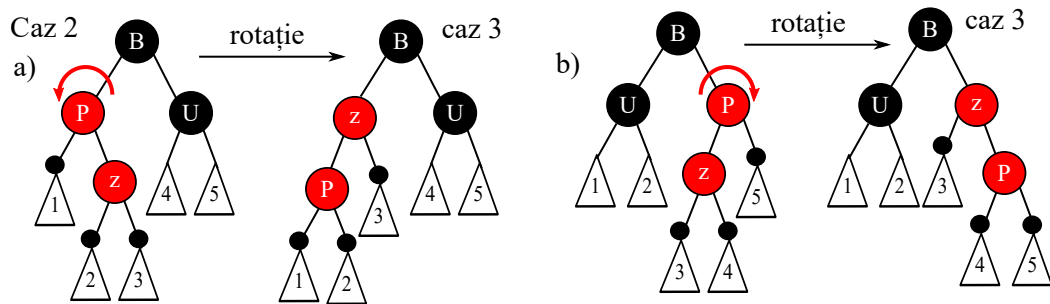


Figura 4: Cazul 2 pentru inserție.

Cazu 1: unchiul lui z este roșu (fig. 3).

Deoarece U , P sunt roșii rezultă că B are culoarea neagră, altfel s-ar contrazice proprietatea 4, \Rightarrow este suficientă recolorarea P , U , B , adică P și U devin negri iar B roșu.

În urma acestei modificări poate avea loc o contrazicere a proprietății 4 pentru B și părintele său, deci se reia procedura de refacere a proprietăților roșu-negru, considerând de data aceasta $z = B$. Același procedeu se aplică și în cazul în care z se află pe dreapta lui P .

Atunci când unchiul U este negru se disting cazurile 2 și 3, care diferă prin poziționarea nodului z relativ la P și la B .

Cazul 2: unchiul este negru și

- a) z se află pe dreapta lui P , iar P se află la stânga lui B - fig.4
- b) z se află pe stânga lui P , iar P se află la dreapta lui B

Soluționare: în cazul a) - rotație la stânga după P , în cazul b) - rotație la dreapta după P . Nu se soluționează complet, ci se trece practic în cazul 3, în care refacerea începe de la nodul care prin rotație a coborât, deci de la $z = P$.

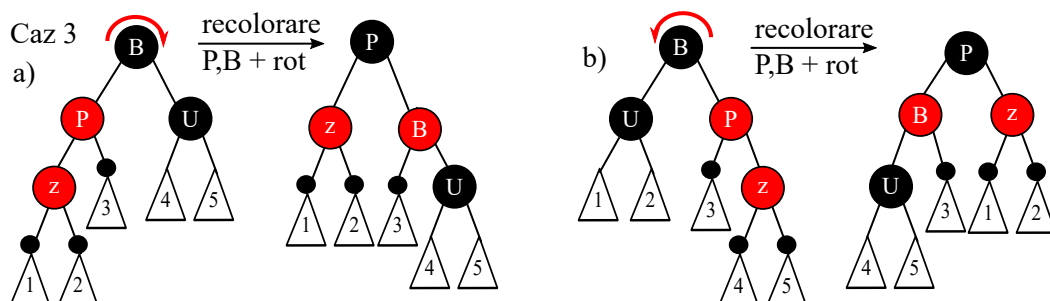


Figura 5: Cazul 3 pentru inserție.

Cazul 3: unchiul este negru și

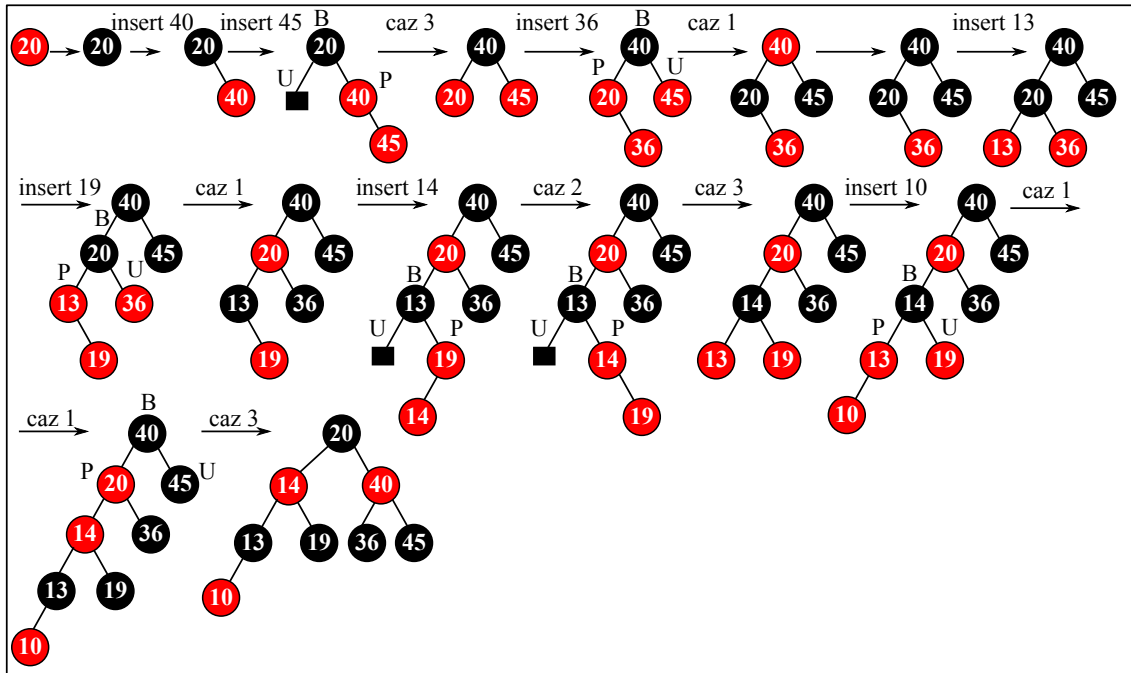


Figura 6: Inserarea cheilor 20, 40, 45, 36, 13, 19, 14, 10 într-un ARN inițial vid.

- z se află pe stânga lui P , iar P se află la stânga lui B - fig.5
- z se află pe dreapta lui P , iar P se află la dreapta lui B

Soluționare::

- Colorare B cu roșu, colorare P cu negru.
- pentru situația a) Rotație la dreapta în jurul lui B , pentru situația b) Rotație la stânga în jurul lui B .

Un exemplu de construcție a unui ARN prin inserție succesivă de chei este prezentat în figura 6.

Algorithm 1: Insert(T, z)

Intrare: Arborele T , nodul z care se inserează

$y \leftarrow T.nil$

$x \leftarrow T.rad$

cat timp $x \neq T.nil$ **executa**

$\bar{y} \leftarrow x$

daca $z.info < x.info$ **atunci**

$x \leftarrow x.st$

sfarsit_daca

altfel

$x \leftarrow x.dr$

sfarsit_daca

sfarsit_cat_timp

$z.p \leftarrow y$

daca $y = T.nil$ **atunci**

$T.rad \leftarrow z$

sfarsit_daca

altfel

daca $z.info < y.info$ **atunci**

$y.st \leftarrow z$

sfarsit_daca

altfel

$y.dr \leftarrow z$

sfarsit_daca

sfarsit_daca

$z.st \leftarrow T.nil$

$z.dr \leftarrow T.nil$

$x.color \leftarrow rosu$

Algoritm 2: ARN-Insert-Repara(T, z)

```
cat_timp  $z.p.color = \text{rosu}$  executa
|  daca  $z.p = z.p.p.st$  atunci
|  |   $U \leftarrow z.p.p.dr$ 
|  |  _____ caz 1
|  |  daca  $U.color = \text{rosu}$  atunci
|  |  |   $z.p.color \leftarrow \text{negru}$ 
|  |  |   $U.color \leftarrow \text{negru}$ 
|  |  |   $z.p.p.color \leftarrow \text{rosu}$ 
|  |  |   $z \leftarrow z.p.p$ 
|  |  sfarsit_daca
|  |  altfel
|  |  _____ caz 2
|  |  |  daca  $z \leftarrow z.p.dr$  atunci
|  |  |  |   $z \leftarrow z.p$ 
|  |  |  |   $ROT\_ST(T, z)$ 
|  |  |  sfarsit_daca
|  |  |  _____ caz 3
|  |  |   $z.p.color \leftarrow \text{negru}$ 
|  |  |   $z.p.p.color \leftarrow \text{rosu}$ 
|  |  |   $ROT\_DR(T, z.p.p)$ 
|  |  sfarsit_daca
|  |  sfarsit_daca
|  |  altfel
|  |  |  //similar dar simetric pentru nodul  $z$ 
|  |  |  //aflat la stanga bunicului
|  |  sfarsit_daca
sfarsit_cat_timp
 $T.rad.color \leftarrow \text{negru}$ 
RETURN
```

Complexitate: Inserția are aceeași complexitate ca și în cazul arborilor binari de căutare simpli, deci $O(h)$. Cum înălțimea este cel mult $2\log_2(n+1)$ rezultă o complexitate $O(\log_2 n)$, iar algoritmul de refacere al proprietăților de arbore roșu-negru pornește de jos înspre rădăcină pe o ramură, deci complexitatea este tot $O(\log_2 n)$. Rezultă complexitatea pentru inserție este $O(\log_2 n)$.

1.2 Ștergerea dintr-un arbore roșu-negru

Operația de ștergere dintr-un arbore roșu-negru este mai complicată decât operația de inserție. În prima etapă, operația de ștergere are la bază ștergerea dintr-un arbore binar de căutare, cu câteva modificări, după care trebuie efectuată o operație de refacere a proprietăților de arbore roșu-negru.

Reamintim faptul că, în cazul ștergerii unui nod z dintr-un arbore binar de căutare T se

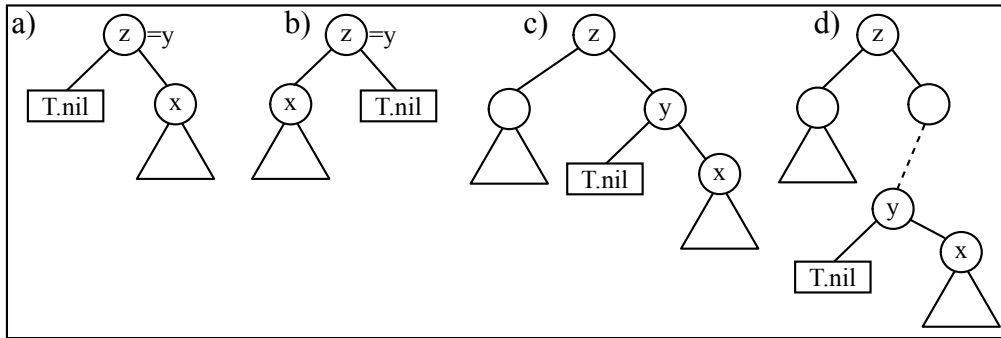


Figura 7: Cazurile pentru ștergerea nodului z . a) z nu are descendent stâng, b) z nu are descendent drept, c) z are 2 descendenți nenuli și succesorul lui z este fiu al lui z , d) z are doi descendenți și succesorul lui z nu este fiu al lui z .

iau în considerare cazurile:

1. $z.st = T.Nil$ - fig.7 a)
2. $z.dr = T.Nil$ - fig.7 b)
3. z are doi fii diferiți de $T.Nil$ și $y = \text{succesor}(T, z)$
 - a. y descendent direct al lui z - fig.7 c)
 - b. y nu este descendent direct al lui z - fig.7 d)

Observații:

- Algoritmul de refacere depinde în cazurile 1 și 2 de culoarea nodului șters z și de culoarea pe care o are x , iar în cazul 3 de culoarea originală a lui y (după ștergere y preia culoarea lui z) și de culoarea lui x , unde x este în primele 2 cazuri nodul cu care se înlocuiește z , iar în al treilea caz este fiul drept al lui y .
- În cazurile 1 și 2, prin ștergerea nodului z și înlocuirea cu unul dintre descendenții direcți, se poate produce o modificare a culorii în poziția deținută anterior de z , ceea ce poate duce la violarea proprietăților de arbore roșu-negru. La fel în cazul 3, la înlocuirea lui y cu x .
- Nodul x este acela care se deplasează în poziția deținută anterior de nodul y . Se observă faptul că, x poate fi santinela $T.Nil$

Refacerea proprietăților de arbore roșu-negru:

- Dacă nodul z în cazurile 1 și 2, respectiv y în cazul 3, a avut culoarea inițială roșu, atunci se păstrează proprietățile de arbore RN, deoarece
 1. Nu s-a modificat înălțimea neagră pe nici o ramură prin eliminarea unui nod roșu
 2. Nici un nod roșu nu a căpătat un fiu roșu.

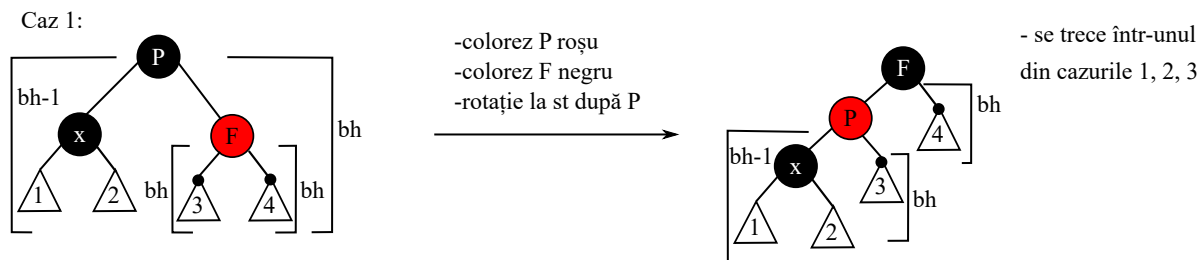


Figura 8: Cazul 1 în ARN

3. Deoarece nici nodul z , nici nodul y în cazul 3, nu au fost rădăcină (culoarea originală a fost roșie) nu s-a modificat nici culoarea rădăcinii, care a rămas neagră
- Rezultă deci că, proprietățile de arbore RN sunt violate doar dacă a fost eliminat un nod negru. Există mai multe cazuri care trebuie tratate.
 - Prin eliminarea unui nod negru din arbore
 1. În primul rând se micșorează înălțimea neagră pe ramura respectivă
 2. Poate apărea vecinătate între două noduri roșii

Cazul 0: culoarea nodului x cu care s-a făcut înlocuirea este roșie. În acest caz singurul lucru care trebuie făcut este recolorarea lui x în negru. Acest lucru rezolvă atât problema (1) cât și problema (2).

Notății: P =părintele lui x , F =fratele lui x , bh = înălțimea neagră originală a subarborelui cu rădăcina P (exclusiv nodul P).

În continuare vom considera că nodul x se află la stânga nodului P . Pentru x la dreapta lui P modul de rezolvare este simetric.

Cazul 1: F are culoarea roșie - fig. 8

Facem următoarele observații:

- Datorită faptului că înainte de ștergere arborele era RN valid, din F roșu și F descendent direct al lui P rezultă culoarea lui P este neagră.
- Copiii lui F sunt ambii negri (eventual T.nil)
- Prin ștergere s-a micșorat înălțimea neagră pe partea stângă a lui P , adică subarboarele drept are înălțimea neagră bh , iar subarboarele stâng $bh - 1$. Cum F este roșu rezultă că ambii săi subarbori au înălțimea bh și fiii lui F sunt ambii negri.

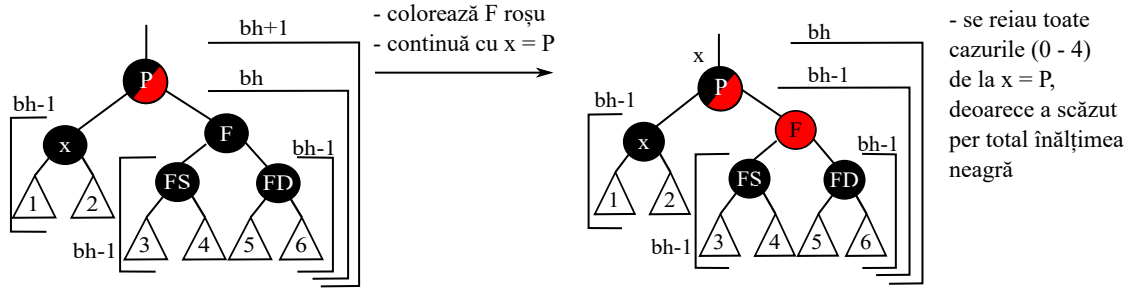


Figura 9: Cazul 2 în ARN

Refacerea proprietăților de arbore RN:

- Colorare P cu roșu și colorare F cu negru.
- Rotație la stânga în jurul lui P . (Dacă x se află pe dreapta lui P , atunci rotația este la dreapta).

În această situație F , acum negru, urcă în locul lui P . La dreapta lui F înălțimea neagră a rămas bh . Înălțimea neagră a subarborelui drept al lui P este bh , iar înălțimea neagră subarborelui stâng al lui P este $bh - 1$. Deci problema în nodul P a rămas, dar fiul drept al lui P nu mai este roșu ci negru, ceea ce conduce la unul dintre cazurile 2, 3 sau 4.

Deci din cazul 1 se trece la cazul 2, 3 sau 4!

În figura 8 este ilustrată procedura de refacere a proprietăților RN în cazul 1.

Cazul 2: F este negru și ambii fii ai săi sunt negri (eventual T.Nil) - fig. 9.

Observații:

- $F \neq T.Nil$, pentru că pe partea stângă a lui P s-a șters un nod negru, iar înainte de ștergere pe ambele părți ale lui P înălțimea neagră era aceeași și cel puțin 1.
- P poate fi roșu sau negru
- Prin ștergere s-a micșorat înălțimea neagră pe partea stângă a lui P , adică subarboarele drept are înălțimea neagră bh , iar subarboarele stâng $bh - 1$. Cum F este negru rezultă că ambii săi subarbori au înălțimea $bh - 1$.

Refacerea proprietăților de arbore RN:

- Se colorează F roșu \Rightarrow înălțimea neagră a subarborelui drept al lui P este $bh - 1$, deci egală cu cea a subarborelui stâng, dar înălțimea neagră a arborelui P este mai mică decât înainte de ștergere, deci cu o unitate mai mică decât a fratelui său (dacă există). Această problemă se rezolvă astfel:
 - Dacă P este roșu atunci este suficient să colorăm P cu negru - fig.9 a).

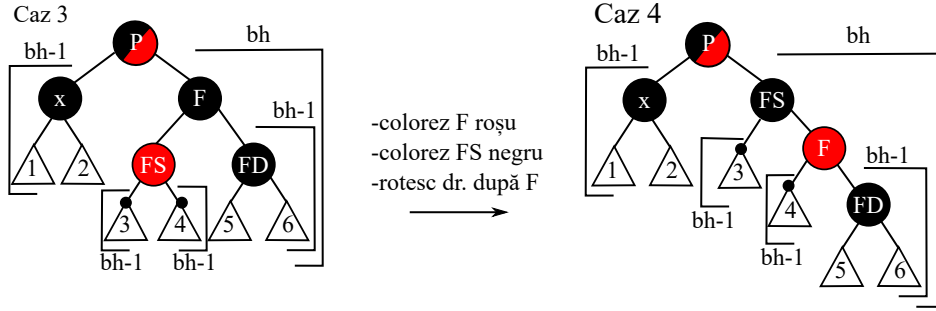


Figura 10: Cazul 3 în ARN. P poate avea culoarea roșie sau culoarea neagră.

- Altfel problema de refacere a proprietăților RN se reia pentru subarboarele care are ca rădăcină părintele lui P . Se observă din figura 9 b) că, în acest caz înălțimea neagră a întregului arbore care are rădăcina P a scăzut cu 1 și trebuie reechilibrat de la acest nod în sus.

Cazul 3: a) F are culoarea neagră, fiul stâng al lui F notat cu FS este roșu, iar cel drept notat cu FD este negru. (fig. 10). Respectiv cazul 3b) simetric, atunci când F este pe stânga lui P și x pe dreapta, F este negru, FD este roșu și FS este negru. Cazul 3b) se soluționează simetric cu 3a) prezentat mai jos.

Observații:

- Culoarea lui P poate fi roșie sau neagră.
- F sigur este diferit de $T.Nil$, pentru că pe partea stângă a lui P s-a șters un nod negru, iar înainte de ștergere pe ambele părți ale lui P înălțimea neagră era aceeași și cel puțin 1.
- Înălțimea neagră a subarboarei stâng al lui P este $bh - 1$, înălțimea neagră a subarboarei cu rădăcina F este bh . Înălțimile negre ale subarborilor FS și FD sunt $bh - 1$.

Refacerea proprietăților de arbore RN: - fig. 10

- Recolorare F și $FS \Rightarrow F$ devine roșu și FS devine negru \Rightarrow înălțimea neagră a lui FS devine bh și înălțimea neagră a lui FD devine $bh - 1$.
- Rotație la dreapta în jurul lui $F \Rightarrow FS$ urcă în locul lui F , înălțimea neagră a lui FS devine bh , înălțimea neagră a lui F devine $bh - 1 =$ înălțimea neagră a fiului stâng al lui FS .

Problema la care am ajuns în continuare este refacerea proprietăților RN pentru cazul 4.

Cazul 4: F are culoarea neagră iar fiul drept notat cu FD este roșu. Fiul stâng, FS , poate avea oricare dintre cele două culori. (fig. 11). În cazul simetric, atunci când x este

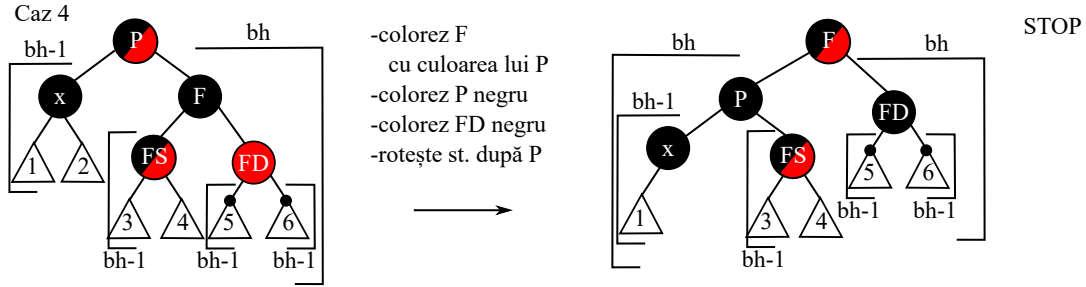


Figura 11: Cazul 4 în ARN

pe dreapta lui P , FS este roșu.

Observații:

- Culoarea lui P poate fi roșu sau negru.
- F sigur este diferit de $T.Nil$, pentru că pe partea stângă a lui P s-a șters un nod negru, iar înainte de ștergere pe ambele părți ale lui P înălțimea neagră era aceeași și cel puțin 1.
- Înălțimea neagră a subarborelui stâng al lui P este $bh - 1$, înălțimea neagră a subarborelui cu rădăcina F este bh . Înălțimile negre ale lui FS și FD sunt $bh - 1$.

Refacerea proprietăților de arbore RN: - fig. 11.

- Colorare F cu culoarea lui P
- Colorare P cu negru
- Colorare FD cu negru
- Rotație la stânga în jurul lui P (respectiv dreapta în cazul simetric).

Prin aceste operații rezultă: creșterea cu o unitate a înălțimii negre a subarborelui F , adică $bh(F) = bh + 1$. Dar $bh(x) = bh - 1$. Prin rotația în jurul lui P , P fiind acum negru, se rebalansează arborele.

Pseudocodul pentru ștergere poate fi găsit în bibliografia recomandată.