MODULUL 5 - III - ARBORI ROŞU-NEGRU

1 Noţiuni generale

Definiţie: Un arbore roşu-negru (ARN) este un arbore binar de căutare în care fiecărui nod i se asociază o culoare - roşu sau negru - şi care are următoarele proprietăţi:

1. Fiecare nod este roşu sau negru - are deci un câmp suplimentar *color*
2. Rădăcina este neagră
3. Fiecare frunză este neagră şi NULL
4. Dacă un nod este roşu, ambii fii sunt negri $\Rightarrow$ părintele unui nod roşu este negru.
5. Pentru fiecare nod x, oricare drum de la nod la o frunză NULL se întâlneşte acelaşi număr de noduri negre (inclusiv frunza NULL şi exclusiv nodul de la care se porneşte). Acest număr se numeşte înălţimea neagră a subarborelui de rădăcină x. Notăm înălţimea neagră cu *bh* - *black height*.

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

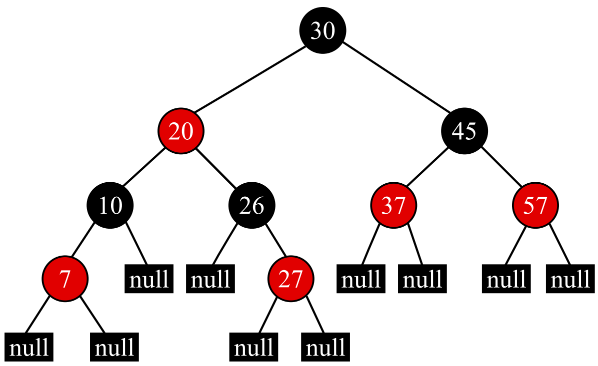

Figure 1: Exemplu de arbore roşu - negru.

Observaţii:

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

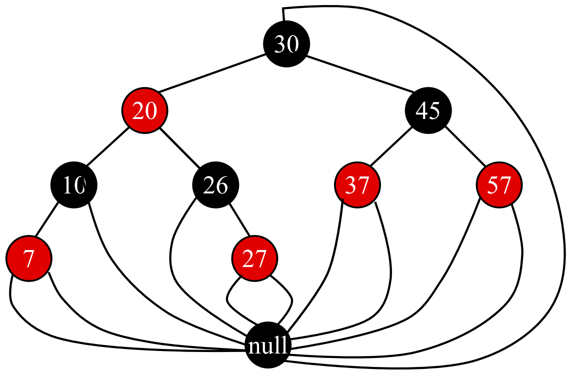

Figure 2: Exemplu de arbore roşu - negru cu santinelă.

- Înălţimea maximă a unui arbore ARN este $2\log_2(n+1)$.
  Se demonstrează prin inducţie că orice subarbore de rădăcină x conţine cel puţin $2^{bh(x)} - 1$ noduri interne. Notând cu h înălţimea arborelui şi cu r rădăcina, din proprietatea 4 se obţine $bh(r) \geq h/2$. 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 NULL 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 descendenţi frunze NULL. Un exemplu de ARN cu santinelă este prezentat în fig. 2.
- Pentru simplitate în continuare vom ignora în desene nodurile NULL.
- Datorită faptului că operaţiile de căutare, maxim, minim, succesor, predecesor depind de înălţimea h a arobrelui, înseamnă că aceste operaţii au complexitatea $O(\log_2 n)$.
- Operaţiile de inserţie şi ştergere sunt ceva mai complicate decât în cazul arborilor binari de căutare simpli, deoarece după inserţie/ştergere trebuie eventual refăcută structura de arbore roşu-negru.

Pentru refacerea proprietăţilor de arbore roşu-negru sunt necesare operaţii de recolorare a nodurilor şi de rotaţie.

1.1 Inserţia într-un arbore roşu-negru

Inserţia propriu-zisă a unui nod într-un arbore roşu-negru se realizează după acelaşi algoritm ca şi inserţia într-un arbore binar de căutare. Practic se porneşte de la rădăcină şi se compară la fiecare pas cheia nodului curent x cu cheia nodului z care se inserează, coborându-se în subarborele stâng dacă $z.info < x.info$ şi în cel drept altfel. Apoi se refac proprietăţile ARN.

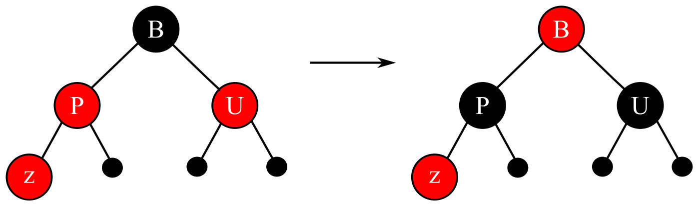

Figure 3: Cazul 1 pentru inserţie.

Observaţii:

- Întotdeauna nodul care se inserează are culoarea roşie
- În arborele T se consideră o santinelă T.Nil

Proprietăţile care pot fi neîndeplinite în cazul inserţiei:

- Proprietăţile 1, 3 şi 5 se păstrează, datorită faptului că se inserează un nod roşu, care are ca descendenţi două frunze NULL, 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 situaţii este suficientă colorarea rădăcinii cu negru.
- Proprietatea 4 poate fi contrazisă, dacă părintele de care s-a legat noul nod are culoarea roşie. În acest caz trebuie refăcută proprietatea. Există în această situaţie 3 cazuri care vor fi discutate în continuare.

Funcţia de refacere a proprietăţilor RN este apelată doar dacă părintele nodului inserat este roşu.

Observaţii:

- Dacă notăm cu z nodul inserat (nodul curent), cu P părintele său, cu U unchiu (fratele părintelui) iar cu B bunicul, atunci z se poate afla la stânga sau la dreapta lui P. Cele două cazuri se tratează în mod similar, prin simetrie. Vom considera în continuare inserţia pe stânga bunicului B.
- Datorită faptului că P are culoarea roşie, deja înainte de inserţie, iar inserţia s-a produs într-un arbore roşu-negru valid, înseamnă că P nu este rădăcină şi deci există nodul $B \neq T.Nil$

Cazurile de refacere a proprietăţii 4 în urma inserţiei:

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

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

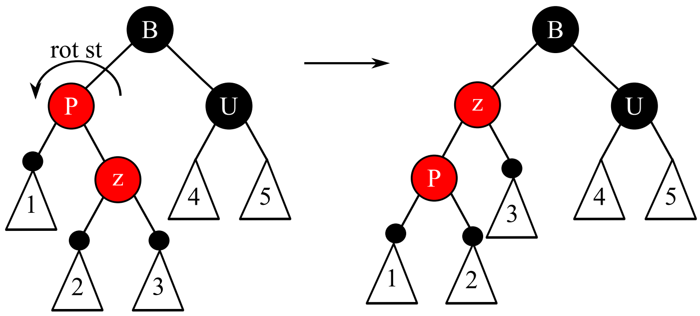

Figure 4: Cazul 2 pentru inserţie.

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

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

Cazul 2: unchiul este negru şi

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

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

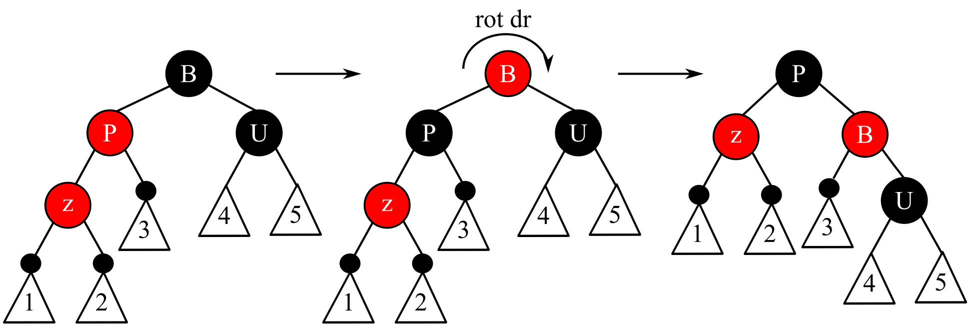

Figure 5: Cazul 3 pentru inserţie.

Cazul 3: unchiul este negru şi

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

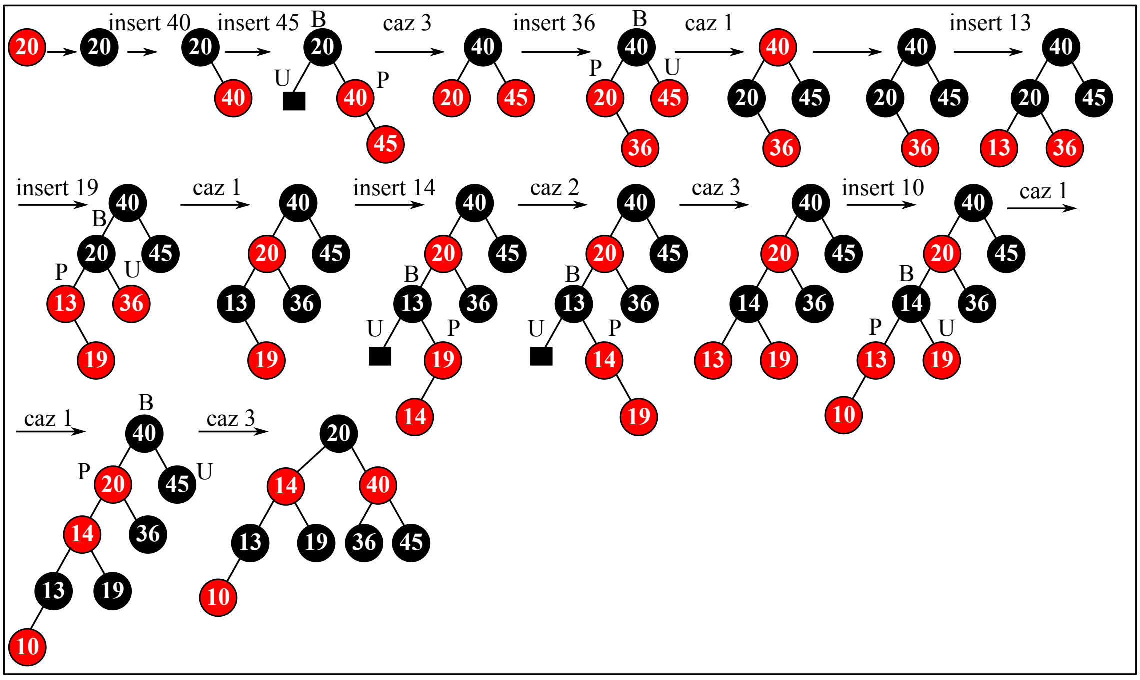

Figure 6: Inserarea cheilor 20, 40, 45, 36, 13, 19, 14, 10 într-un ARN iniţial vid.

b) z se află pe dreapta lui P, iar P se află la dreapta lui B

Soluţionare::

1. Colorare B cu roşu, colorare P cu negru.
2. pentru situaţia a) Rotaţie la dreapta în jurul lui B, pentru situaţia b) Rotaţie la stânga în jurul lui B.

Un exemplu de construcţie a unui ARN prin inserţie succesivă de chei este prezentat în figura 6.

Algoritm: inserarea nodului z în arborele T

```
ARN_INSERT(T,z)
   y = T.Nil
   x = T.rad
   cat timp  x ≠ T.Nil
            y = x
            daca  z.info < x.info  atunci
                 x = x.st
            altfel  x = x.dr
            sfarsit daca
   sfarsit cat timp
   z.p = y
   daca  y = T.Nil  atunci  T.rad = z
```

```
    altfel
            daca z.info < y.info atunci
                  y.st = z
            altfel
                  y.dr = z
            sfarsit daca
    sfarsit daca
    z.st = T.Nil
    z.dr = T.Nil
    z.color = rosu
    ARN_INSERT_REPARA(T,z)
RETURN
```

Algoritm: refacerea proprietăţilor de arbore roşu-negru

```
ARN_INSERT_REPARA(T,z)
    cat timp z.p.color = rosu
            daca z.p = z.p.p.st atunci
                  U = z.p.p.dr
__________________________________________________________ caz 1
                  daca U.color = rosu atunci
                        z.p.color = negru
                        U.color = negru
                        z.p.p.color = rosu
                        z = z.p.p
                  altfel
__________________________________________________________ caz 2
                        daca z = z.p.dr atunci
                              z = z.p
                              ROT_ST(T, z)
                        sfarsit daca
__________________________________________________________ caz 3
                        z.p.color = negru
                        z.p.p.color = rosu
                        ROT_DR(T,z.p.p)
                  sfarsit daca
                  altfel
                  //similar dar simetric pentru nodul z
                  //aflat la stânga bunicului
            sfarsit daca
    sfarsit cat timp
    T.rad.color = negru
RETURN
```

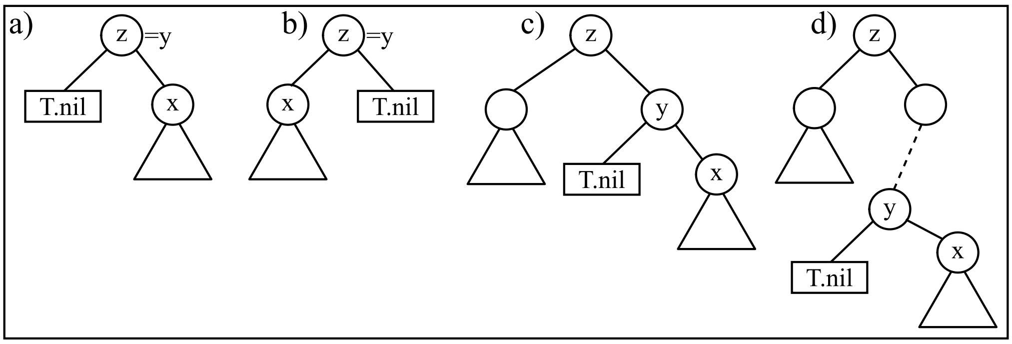

Figure 7: Cazurile pentru ştergerea nodului z. a) z nu are descendent stâng, b) z nu are descendent drept, c) z are 2 descendenţi nenuli şi succesorul lui z este fiu al lui z, d) z are doi descendenţi şi succesorul lui z nu este fiu al lui z.

Complexitate: Inserţia are aceeaşi complexitate ca şi în cazul arborilor binari de căutare simpli, deci $O(h)$. Cum înălţimea este cel mult $2\log_2(n+1)$ rezultă o complexitate $O(\log_2 n)$, iar algoritmul de refacere al proprietăţilor de arbore roşu-negru porneşte de jos înspre rădăcină pe o ramură, deci complexitatea este tot $O(\log_2 n)$. Rezultă complexitatea pentru inserţie este $O(\log_2 n)$.

1.2 Ştergerea dintr-un arbore roşu-negru

Operaţia de ştergere dintr-un arbore roşu-negru este mai complicată decât operaţia de inserţie. În prima etapă, operaţia de ştergere are la bază ştergerea dintr-un arbore binar de căutare, cu câteva modificări, după care trebuie efectuată o operaţie de refacere a proprietăţilor de arbore roşu-negru.

Reamintim faptul că, în cazul ştergerii unui nod z dintr-un arbore binar de căutare T se 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 diferiţi de $T.Nil$ şi $y = sucesor(T, z)$
   a. y descendent direct al lui z - fig.7 c)
   b. y nu este descendent direct al lui z - fig.7 d)

Observaţii:

- Algoritmul de refacere depinde în cazurile 1 şi 2 de culoarea nodului şters z şi de culoarea pe care o are x, iar în cazul 3 de culoarea originală a lui y (după ştergere y preia culoarea lui z) şi de culoarea lui x, unde x este în primele 2 cazuri nodul cu care se înlocuieşte z, iar în al treilea caz este fiul drept al lui y.

- În cazurile 1 şi 2, prin ştergerea nodului z şi înlocuirea cu unul dintre descendenţii direcţi, se poate produce o modificare a culorii în poziţia deţinută anterior de z, ceea ce poate duce la violarea proprietăţilor de arbore roşu-negru. La fel în cazul 3, la înlocuirea lui y cu x.
- Nodul x este acela care se deplasează în poziţia deţinută anterior de nodul y. Se observă faptul că, x poate fi santinela $T.Nil$

Refacerea proprietăţilor de arbore roşu-negru:

- Dacă nodul z în cazurile 1 şi 2, respectiv y în cazul 3, a avut culoarea iniţială roşu, atunci se păstrează proprietăţile de arbore RN, deoarece
  1. Nu s-a modificat înălţimea neagră pe nici o ramură prin eliminarea unui nod roşu
  2. Nici un nod roşu nu a căpătat un fiu roşu.
  3. Deoarece nici nodul z, nici nodul y în cazul 3, nu au fost rădăcină (culoarea originală a fost roşie) nu s-a modificat nici culoarea rădăcinii, care a rămas neagră
- Rezultă deci că, proprietăţile 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 micşorează înălţimea neagră pe ramura respectivă
  2. Poate apărea vecinătate între două noduri roşii

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

Notaţii: P=părintele lui x, F=fratele lui x, bh = înălţimea 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 roşie - fig. 8

Facem următoarele observaţii:

- Datorită faptului că înainte de ştergere arborele era RN valid, din F roşu şi F descendent direct al lui P rezultă culoarea lui P este neagră.

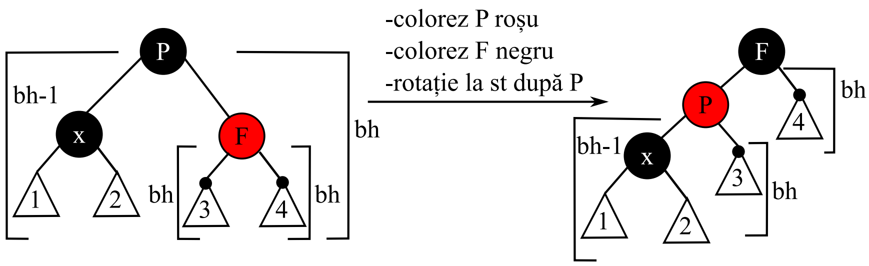

Figure 8: Cazul 1 în ARN

- Copiii lui F sunt ambii negri (eventual T.nil)
- Prin ştergere s-a micşorat înălţimea neagră pe partea stângă a lui P, adică subarborele drept are înălţimea neagră bh, iar subarborele stâng $bh-1$. Cum F este roşu rezultă că ambii săi subarbori au înălţimea bh şi fiii lui F sunt ambii negri.

Refacerea proprietăţilor de arbore RN:

- Colorare P cu roşu şi colorare F cu negru.
- Rotaţie la stânga în jurul lui P. (Dacă x se află pe dreapta lui P, atunci rotaţia este la stânga).

În această situaţie F, acum negru, urcă în locul lui P. La dreapta lui F înălţimea neagră a rămas bh. Înălţimea neagră a subarborelui drept al lui P este bh, iar înălţimea 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 roşu 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 proprietăţilor RN în cazul 1.

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

Observaţii:

- $F \neq T.Nil$, pentru că pe partea stângă a lui P s-a şters un nod negru, iar înainte de ştergere pe ambele părţi ale lui P înălţimea neagră era aceeaşi şi cel puţin 1.
- P poate fi roşu sau negru
- Prin ştergere s-a micşorat înălţimea neagră pe partea stângă a lui P, adică subarborele drept are înălţimea neagră bh, iar subarborele stâng $bh-1$. Cum F este negru rezultă că ambii săi subarbori au înălţimea $bh-1$.

Refacerea proprietăţilor de arbore RN:

- Se colorează F roşu $\Rightarrow$ înălţimea neagră a subarborelui drept al lui P este $bh-1$, deci egală cu cea a subarborelui stâng, dar înălţimea neagră a arborelui P este mai

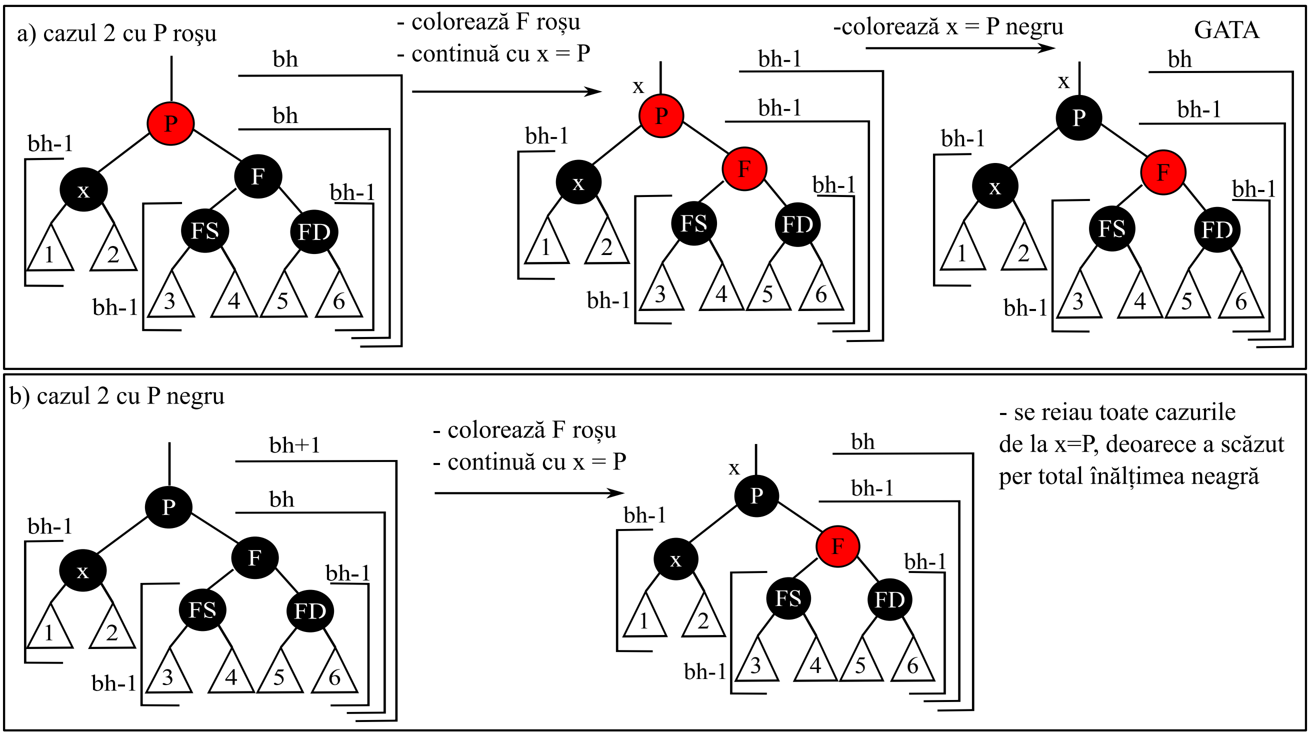

Figure 9: Cazul 2 în ARN

mică decât înainte de ştergere, deci cu o unitate mai mică decât a fratelui său (dacă există). Această problemă se rezolvă astfel:

- Dacă P este roşu atunci este suficient să colorăm P cu negru - fig.9 a).
- Altfel problema de refacere a proprietăţilor RN se reia pentru subarborele care are ca rădăcină părintele lui P. Se observă din figura 9 b) că, în acest caz înălţimea neagră a întregului arbore care are rădăcina P a scăzut cu 1 şi trebuie reechilibrat de la acest nod în sus.

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

Observaţii:

- Culoarea lui P poate fi roşie sau neagră.
- F sigur este diferit de $T.Nil$, pentru că pe partea stângă a lui P s-a şters un nod negru, iar înainte de ştergere pe ambele părţi ale lui P înălţimea neagră era aceeaşi şi cel puţin 1.
- Înălţimea neagră a subarborelui stâng al lui P este $bh - 1$, înălţimea neagră a subarborelui cu rădăcina F este bh. Înălţimile negre ale subarborilor FS şi FD

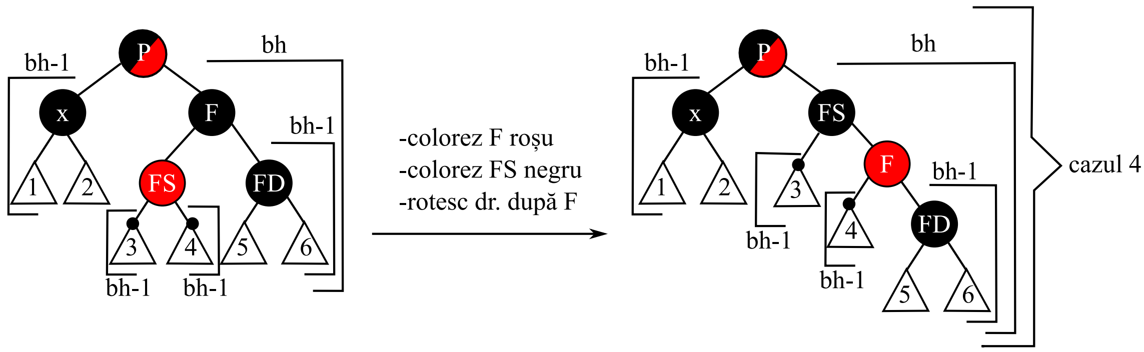

Figure 10: Cazul 3 în ARN. P poate avea culoarea roşie sau culoarea neagră.

sunt $bh - 1$.

Refacerea proprietăţilor de arbore RN: - fig. 10

- Recolorare F şi $FS \Rightarrow F$ devine roşu şi FS devine negru $\Rightarrow$ înălţimea neagră a lui FS devine bh şi înălţimea neagră a lui FD devine $bh - 1$.
- Rotaţie la dreapta în jurul lui $F \Rightarrow FS$ urcă în locul lui F, înălţimea neagră a lui FS devine bh, înălţimea neagră a lui F devine $bh - 1 =$ înălţimea neagră a fiului stâng al lui FS.

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

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

Observaţii:

- Culoarea lui P poate fi roşu sau negru.
- F sigur este diferit de $T.Nil$, pentru că pe partea stângă a lui P s-a şters un nod negru, iar înainte de ştergere pe ambele părţi ale lui P înălţimea neagră era aceaşi şi cel puţin 1.
- Înălţimea neagră a subarborelui stâng al lui P este $bh - 1$, înălţimea neagră a subarborelui cu rădăcina F este bh. Înălţimile negre ale lui FS şi FD sunt $bh - 1$.

Refacerea proprietăţilor de arbore RN: - fig. 11.

- Colorare F cu culoarea lui P
- Colorare P cu negru
- Colorare FD cu negru

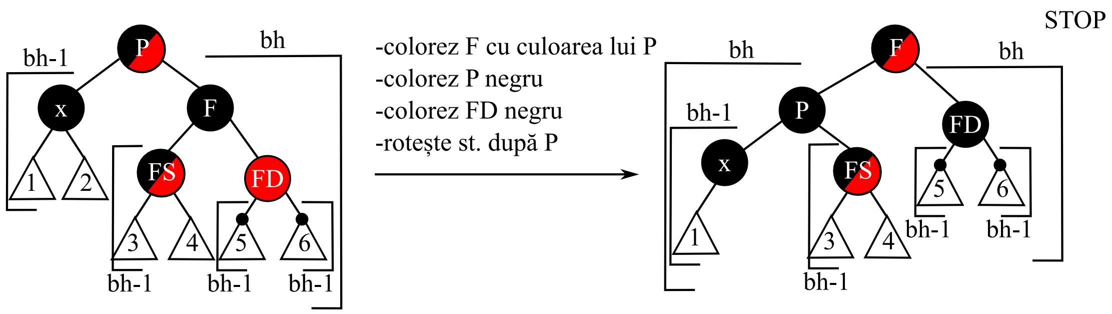

Figure 11: Cazul 4 în ARN

- Rotaţie la stânga în jurul lui P (respectiv dreapta în cazul simetric).

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

Pseudocodul pentru ştergere poate fi găsit în bibliografia recomandată.