MODULUL 5 - II - ARBORI AVL

1 Noţiuni generale

Arborii AVL sunt arbori binari de căutare aproape balansaţi. Denumirea provine de la autorii acestor arbori, doi matematicieni ruşi G.M. Adelson-Velsky şi E.M. Landis. Considerăm pentru fiecare nod x un *factor de balansare* dat prin

$$fb(x) = h(x.dr) - h(x.st)$$

adică, factorul de balansare al unui nod x este reprezentat de către diferenţa dintre înălţimea subarborelui său drept şi înălţimea subarborelui său stâng. În unele documentaţii diferenţa se realizează între înălţimea subarborelui stâng şi cea a subarborelui drept.

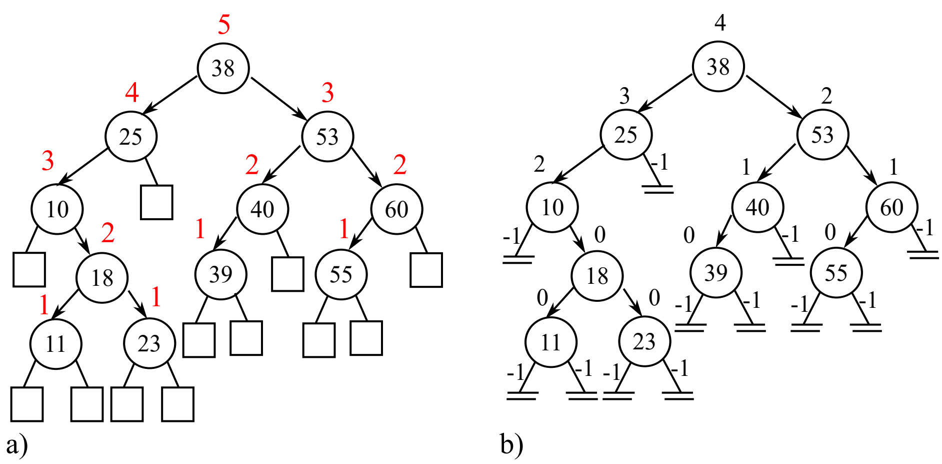

Figura 1: a) Înălţimile nodurilor, considerând frunzele legate de "noduri NULL". b) Înaălţimile nodurilor conform definiţiei uzuale, fără "noduri NULL".

Înălţimea în arborii AVL

Definiţia înălţimii unui nod este lungimea celui mai lung drum de la nod la o frunză, din acest lucru rezultă că înălţimea unei frunze este 0. Pentru calculul corect al factorilor de balansare într-u AVL există două variante:

- se consideră că frunzele, în loc să aibă pe stânga şi pe dreapta NULL, sunt legate de fapt de nişte noduri nule. Astfel înălţimea pentru fiecare nod creşte cu 1 şi deci înălţimea unei frunze va fi 1, iar a unui "nod nul" va fo 0. (fig. 1 a.)
- se consideră în continuare definiţia clasică a înălţmii, iar pentru "NULL" se consideră "înălţimea" -1 (pentru a calcula corect factorul de balansare al unui nod, care are un singur copil - fig1 b.).

În ambele situaţii factorul de balansare obţinut pentru un anumit nod este acelaşi. Precizările de mai sus sunt importante mai ales în contextul implementării unui arbore AVL. În continuarea acestui curs vom considera înălţimea unei frunze ca fiind 0, iar factorul de balansare pentru un nod, care are unul dintre copii nul se va calcula considerând copilul nul cu înălţimea -1, deci ca în figura 1 b.

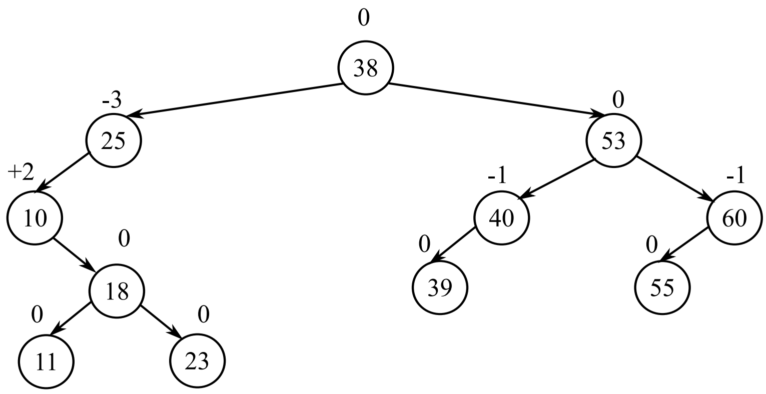

Figura 2: Arbore binar de căutare împreună cu factorii de balansare asociaţi nodurilor. Atenţie: acesta NU este un arbore AVL.

Definiţie - nod balansat: Un nod x se numşte balansat, dacă $fb(x) \in \{-1, 0, 1\}$.

În figura 2 este prezentat un exemplu de arbore binar de căutare împreună cu factorii de balansare corespunzători fiecărui nod. Se observă faptul că nodurile cu cheile 10 şi 25 nu sunt balansate.

Definiţie - arbore AVL: Un arbore binar de căutare se numeşte *arbore AVL*, dacă fiecare nod al său este balansat.

Un exemplu de arbore AVL este prezentat în figura 3.

Înălţimea unui arbore AVL

Pentru a determina înălţimea maximă a unui arbore AVL care conţine n noduri se notează cu h înălţimea acestuia şi cu $N(h)$ numărul minim de noduri ale unui arbore AVL de înălţime h. Evident $n \geq N(h)$.

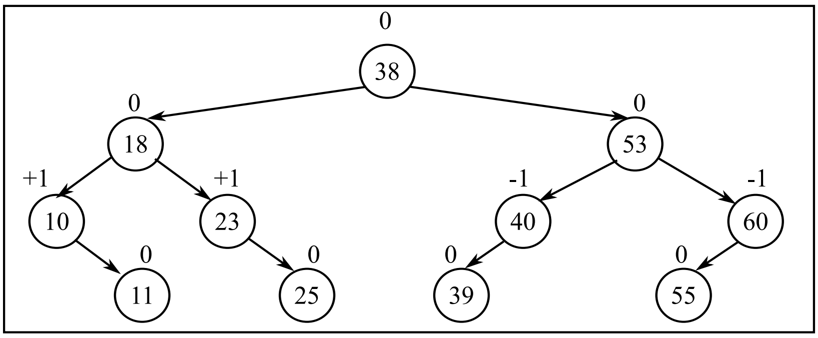

Figura 3: Arbore AVL.

Factorul de balansare al rădăcinii în cazul unui AVL cu număr minim de noduri, de înălţime h este sigur diferit de 0 (altfel s-ar mai putea şterge noduri din subarborele stâng de ex. fără a modifica înălţimea, ceea ce ar contrazice numărul minim de noduri!). Atunci sigur că factorul de balansare al rădăcinii este -1 sau 1, rezultă că pentru un număr minim de noduri, unul dintre subarbori are înălţimea $h-1$ iar celălalt are înălţimea $h-2$.

$$N(h) = N(h-1) + N(h-2) + 1.$$

Cazurile de bază:

$$N(0) = 1$$
$$N(1) = 2$$

Evident: $N(h-1) > N(h-2)$, de unde rezultă $N(h) > 2N(h-2) > 4N(h-4) > \ldots > 2^k N(h-2k)$.
Ajungem la un caz elementar pentru $h-2k=0$ pentru h par - sau $h-2k=1$ pentru h impar. De aici rezultă:
$N(h) > 2^{h/2}$ pentru h par sau $N(h) > 2 * 2^{(h-1)/2} = 2^{(h+1)/2}$ pentru h impar.
Cum $N(h) =$ nr. minim de noduri pentru un AVL de înălţime $h \Rightarrow$ numărul n de noduri dintr-un AVL de înălţime h respectă: $n \geq N(h) > 2^{h/2}$
Deci $h < 2log_2(n)$.

Deoarce complexitatea operaţiilor de căutare, inserţie şi ştergere dintr-un arbore binar de căutare este $O(h)$ rezultă că într-un arbore AVL complexitatea acestor operaţii este $O(log_2 n)$, dacă nu se ţine cont de operaţiile de reechilibrare ale arborelui.

Prin operaţii de inserţie/ştergere se poate produce o debalansare a anumitor noduri. Pentru refacerea proprietăţii de arbore AVL, se utilizează operaţii de **rotaţie**.

2 Operaţii într-un arbore AVL

2.1 Rotaţia

Este o operaţie locală care schimbă structura de pointeri într-un arbore binar, dar păstrează proprietăţile acestuia.

Tipuri de rotaţie:

- Rd = rotaţie spre dreapta în jurul nodului x
- Rs = rotaţie spre stânga în jurul nodului y

Operaţia de rotaţie este ilustrată în figura 4.

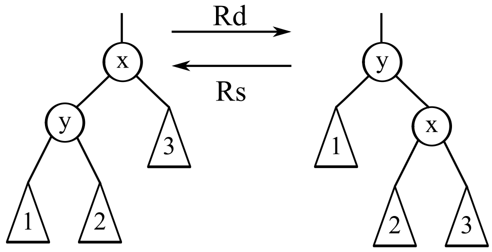

Figura 4: Rotaţii într-un arbore binar de căutare.

În figura 5 este iltustrat grafic modul în care se realizează rotaţia. Iar în figura 6 este prezentată o rotaţie într-un arbore AVL.

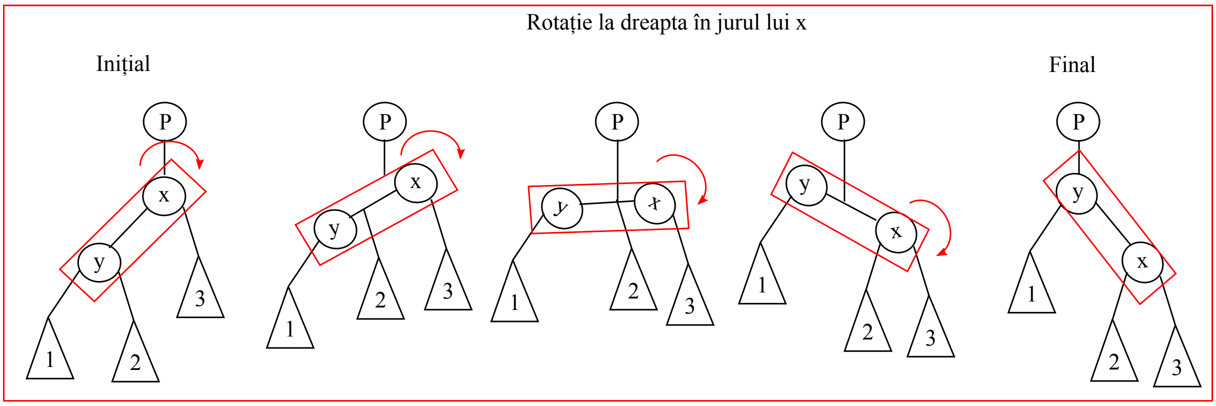

Figura 5: Rotaţie la dreapta în jurul nodului x într-un arbore binar de căutare.

Observaţii:

- pentru a putea efectua o rotaţie spre drapta în jurul nodului x, este necesar ca nodul x să aibă un descendent stâng diferit de NULL.
- pentru a putea efectua o rotaţie spre stânga în jurul nodului y, este necesar ca nodul y să aibă un descendent drept diferit de NULL.

Algoritm pentru rotaţia spre stânga în jurul nodului y

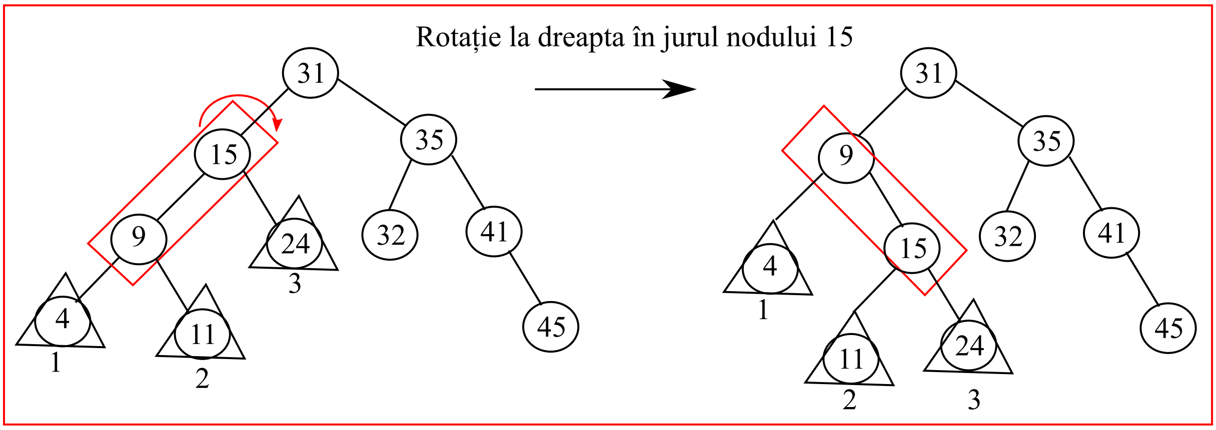

Figura 6: Rotaţie la dreapta în jurul nodului x într-un arbore AVL.

Algoritm 1: ROT_ST(T, y)

$x \leftarrow y.dr$
$y.dr \leftarrow x.st$ //subarb 2 din imag se mută de la x la y
daca $x.st \neq NULL$ **atunci**
  $x.st.p \leftarrow y$//fac legatura subarb. 2 la noul părinte y
sfarsit_daca
$x.p \leftarrow y.p$
daca $y.p = NULL$ **atunci**
  $T.rad \leftarrow x$
sfarsit_daca
altfel
  daca $y = y.p.st$ **atunci**
    //fac legatura de la parintele lui y la x
    $y.p.st \leftarrow x$
  sfarsit_daca
  altfel
    $y.p.dr \leftarrow x$
  sfarsit_daca
sfarsit_daca
$x.st \leftarrow y$ //fac legatura de la x la y
$y.p \leftarrow x$ //fac legatura de y la x

Rotaţia înspre dreapta este simetrică.

Complexitate: $O(1)$

2.2 Inserţia

Prin inserarea unui nod nou x se poate produce o debalansare în arbore. Acest lucru înseamnă că, cel puţin un nod din arbore va avea după recalcularea factorilor de balansare un factor -2 sau 2.

Rebalansare: Nodul nou inserat x are factorul de balansare 0. Recalcularea factorilor

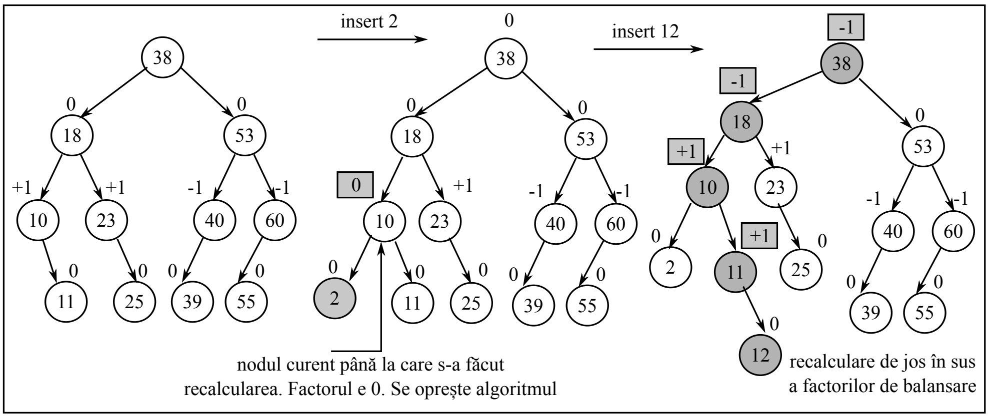

Figura 7: Exemplu de recalculare a factorilor de balansare de la nodul inserat în sus.

de balansare începe de la părintele nodului inserat înspre rădăcină.

Notăm cu x nodul curent. Se recalculează factorul de balansare al lui x.

- Dacă factorul de balansare nou 0, atunci nu a crescut înălţimea subarborelui de rădăcină x, faţă de cât era înainte de inserţie, doar s-a echilibrat. Din acest motiv, nu mai are sens continuarea urcării în arbore. Deci algoritmul se opreşte!
- Dacă factorul de balansare nou este -1 sau $+1$, atunci s-a produs o creştere a înălţimii subarborelui de rădăcină x pe una dintre ramuri (stângă respectiv dreaptă). Acest lucru produce o modificare a factorului de balansare al părintelui mai sus, deci se continuă cu procesul de la nodul curent $x = x.p$.
- Dacă factorul de balansare nou al lui x este -2 sau $+2$, atunci s-a produs o debalansare a subarborelui de rădăcină x şi sunt necesare proceduri de rebalansare, care vor fi prezentate mai jos. În urma acestor proceduri, înălţimea subarborelui curent va reveni la cea de dinainte de inserţie (se va vedea în continuare) şi de aceea nu mai este necesară continuarea rebalansării mai sus, deci algoritmul se opreşte.

Un exemplu de astfel de recalculare este prezentat în figura 7.

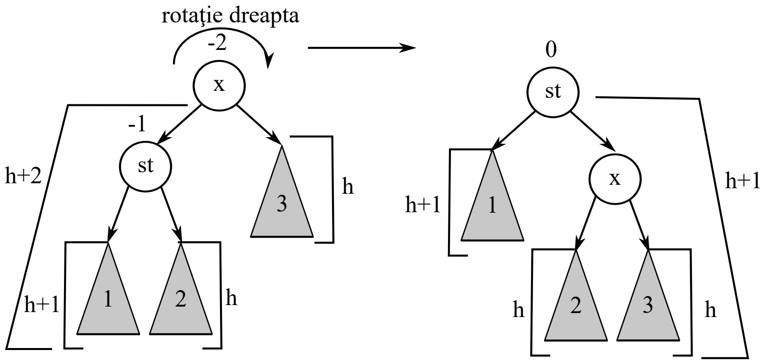

Figura 8: Rebalansarea arbore AVL în cazul 1 a).

Cazurile de rebalansare

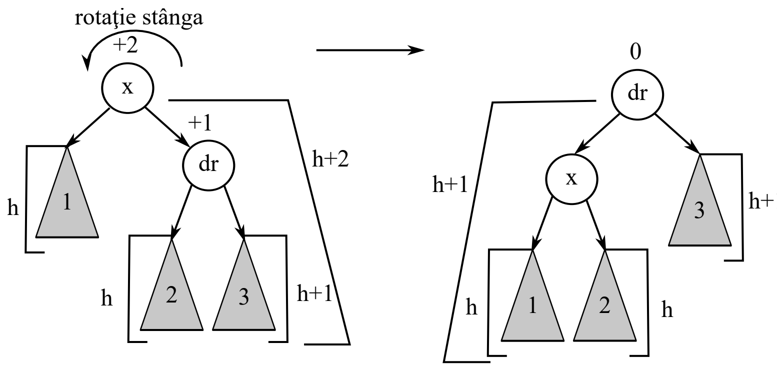

Figura 9: Rebalansarea arbore AVL în cazul 1 b).

Cazul 1

a) $fb(x) = -2$, înseamnă că subarborele stâng este prea înalt. Notăm cu $st = x.st$. Dacă $fb(st) = -1$ atunci rebalansarea se realizează printr-o rotaţie spre dreapta în jurul lui x. Rezultatul acestei operaţii este ilustrat în figura 8. Se observă faptul că noii factori de balansare sunt 0 atât pentru x cât şi pentru st. În plus, înainte de inserţie, subarborele stâng al nodului x avea înălţimea $h + 1$, iar cel drept h. După rebalansare ambii subarbori au înălţimea $h + 1$. Rezultă că înălţimea subarborelui care acum are rădăcina st nu a crescut, faţă de înălţimea înainte de inserţie, când rădăcina era x. Astfel, nu este necesar să continuăm urcarea în arbore, deoarece mai sus nu vor exista modificări în factorul de balansare.

b) $fb(x) = 2$, înseamnă că subarborele drept este prea înalt. Notăm $dr = x.dr$. Dacă $fb(dr) = 1$ atunci se efectuează o rotaţie la stânga în jurul lui x. Rezultatul acestei operaţii este ilustrat în figura 9. Observaţii similare cazului 1.a pot fi făcute şi în acest caz.

Cazul 2:

a) $fb(x) = -2$, înseamnă că subarborele stâng este prea înalt. Notăm cu $st = x.st$. Presupunem că $fb(st) = 1$. Ce se întâmplă dacă se efectuează o rotaţie spre dreapta în jurul lui x? Rezultatul unei astfel de rotaţii este ilustrat în figura 10.

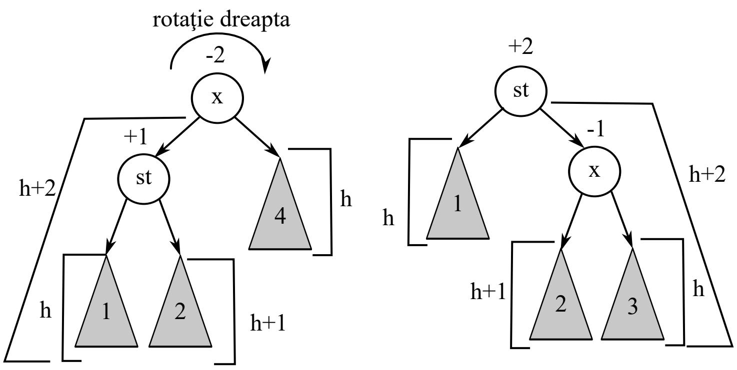

Figura 10: În cazul 2, o rotaţie simplă nu rezolvă problema debalansării arborelui

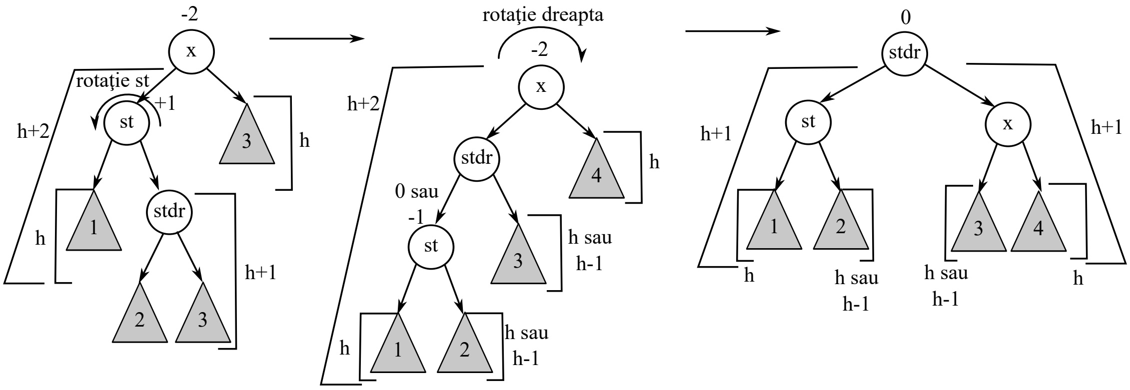

Figura 11: Rebalansarea arborelui în cazul 2.

Deci nu se rezolvă debalansarea, ci se mută pe cealaltă parte a arborelui. Soluţia este următoarea:

- întâi rotaţie la stânga în jurul lui st
- apoi rotaţie la dreapta în jurul lui x.

Se obţine rezultatul din fig. 11.

b) $fb(x) = 2$, înseamnă că subarborele dreapta este prea înalt. Notăm cu $dr = x.dr.$. Presupunem că $fb(dr) = -1$. Dacă efectuăm o rotaţie la stânga în jurul lui x se obţine un efect similar ca la punctul a. Soluţia este deci:

- întâi rotaţie la dreapta în jurul lui dr
- apoi rotaţie la stânga în jurul lui x

Un exemplu de inserţie într-un arbore AVL este prezentat în figura 12.

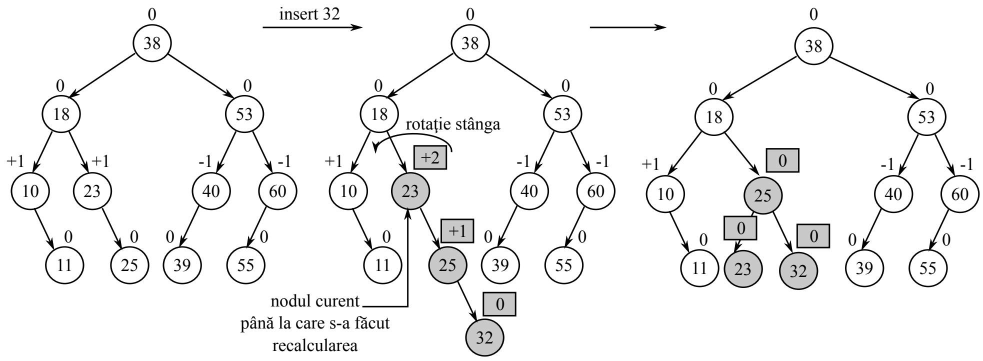

Figura 12: Inserţia nodului cu cheia 32 în arborele AVL din figură.

2.2.1 Ştergerea unui nod

Ştergerea unui nod se realizează la fel ca pentru arborii binari de căutare obişnuiţi. Dacă notăm cu z nodul care trebuie şters atunci:

- dacă z are cel mult un descendent, se porneşte recalcularea factorilor de balansare de la părintele lui z
- dacă z are doi descendenţi, atunci se determină y, succesorulnodului z, se înlocuieşte nodul y cu descendentul său drept, iar nodul z cu nodul y. În acest caz recalcularea factorilor de balansare începe de la părintele succesorului nodului şters.

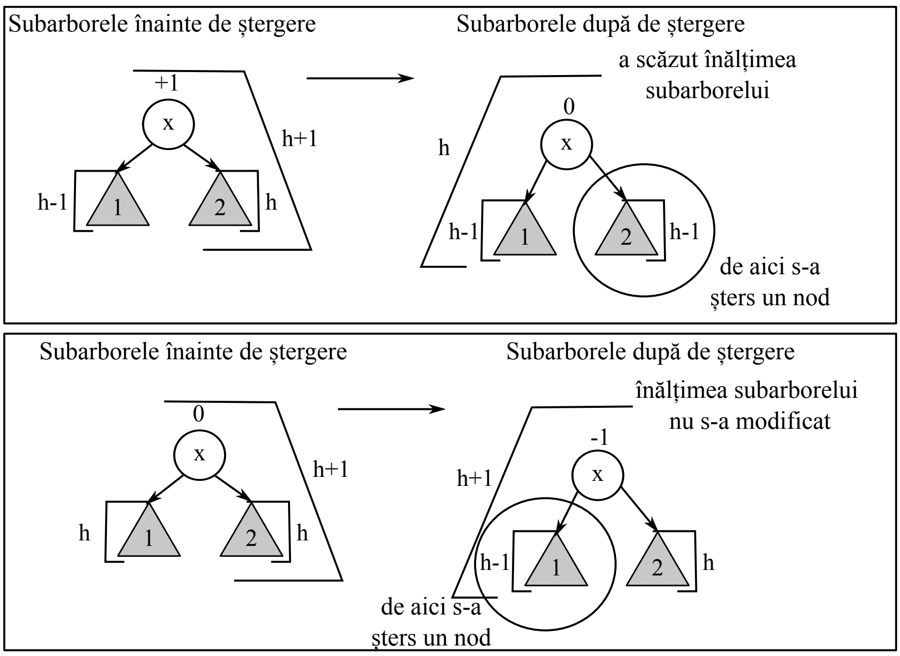

Figura 13: Modificarea înălţimii subarborelui curent după şteregerea unui nod.

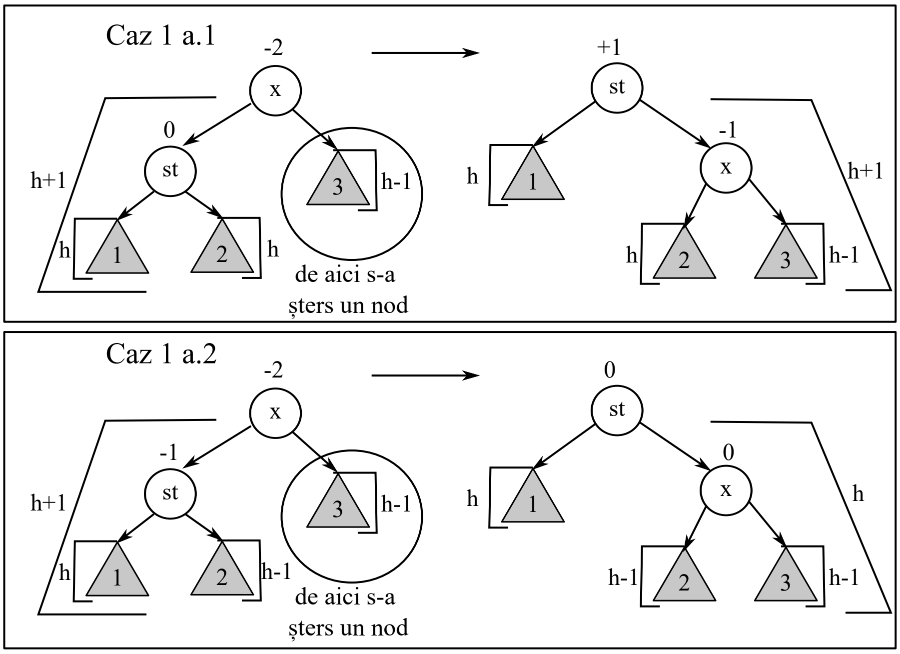

Figura 14: Rebalansarea unui arborele AVL după ştergerea unui nod, cazul 1 a.

Rebalansarea arborelui:

Notăm cu x nodul curent, pentru care se recalculează factorul de blanasare. Spre deosebire de inserţie, algoritmul nu se opreşte atunci când factorul de balansare recalculat al nodului curent x este 0, deoarece în acest caz a scăzut înălţimea subarborelui de rădăcină x. Acest lucru, ilustrat în figura 13, produce modificări ale factorului de balansare şi la părintele său şi deci continuă urcarea în arbore.

În schimb, atunci când factorul de balansare al nodului x devine -1 sau $+1$, înseamnă că doar s-a produs o scădere a înălţimii pe una dintre ramurile lui x, dar nu şi a subarborelui de rădăcină x (fig. 13), deci algoritmul de rebalansare se poate opri.

De asemenea, algoritmul nu se opreşte însă după prima rebalansare a unui nod cu factorul -2 sau $+2$, ci în cele mai multe cazuri trebuie să continue de la nodul curent la părinte până la rădăcină.

Considerând nodul curent x, rebalansarea are următoarele cazuri:

Cazul 1

a. Dacă x are factorul de balansare nou -2 şi factorul lui $st = x.st$ este -1 sau 0, atunci rotaţie la dreapta în jurul lui x. În figura 14 este ilustrat modul de recalculare a factorilor de balansare pentru nodurile x şi st. Se observă şi următorul fapt important. Subarborele de rădăcină x a avut înainte de ştergere înălţimea $h + 2$. În cazul în 1.a.1 din fig. 14, în care y a avut factorul de balansare 0 înainte de rebalansare, după rebalansare se observă că nu s-a modificat înălţimea subarborelui, care acum are rădăcina st, şi deci nu mai este necesară continuarea urcării în arbore.

În schimb, daca st a avut factorul de balansare -1, cazul 1.a.2, se observă din figura 14, că după rebalansare, înălţimea subarborelui care acum are rădăcina st, a scăzut cu 1, ceea ce produce eventuale debalansări mai sus în arbore, deci trebuie continuat la părintele lui x.

b. Dacă x are factorul de balansare nou 2 şi factorul de balansare al lui $dr = x.dr$ este 0 sau 1 atunci rotaţie la stânga în jururl lui x. În figura 15 este ilustrat modul de recalculare a factorilor de balansare pentru nodurile x şi dr. La fel ca în cazul 1.

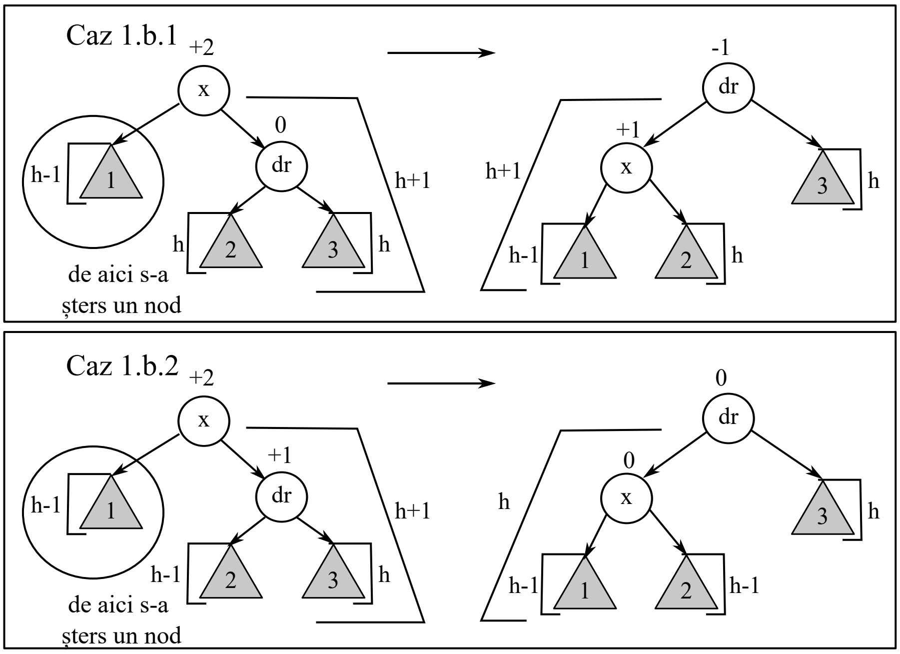

Figura 15: Rebalansarea unui arborele AVL după ştergerea unui nod, cazul 1 b.

a, se observă şi următoarele: Subarborele de rădăcină x a avut înainte de ştergere înălţimea $h+2$. În cazul în 1.b.1, în care dr a avut factorul de balansare 0 înainte de rebalansare, după rebalansare se observă că nu s-a modificat înălţimea subarborelui, care acum are rădăcina dr, şi deci nu mai este necesară continuarea urcării în arbore.

În schimb, daca dr a avut factorul de balansare $+1$, cazul 1.b.2, se observă din fig. 15, că după rebalansare, înălţimea subarborelui care acum are rădăcina dr, a scăzut cu 1, ceea ce produce eventuale debalansări mai sus în arbore, deci trebuie continuat la părintele lui x.

Cazul 2

a. Dacă x are factorul de balansare nou -2 şi nodul $y = x.st$ are factorul de balansare 1, atunci pentru rebalansare se efectuează:

- Întâi rotaţie la stânga în jurul lui y
- Apoi rotaţie la dreapta în jurul lui x.

b. Dacă x are factorul de balansare nou 2 şi $y = x.dr$ are factorul de balansare -1, atunci pentru rebalansare se efectuează:

   – Întâi rotaţie la dreapta în jurul lui y

   – Apoi rotaţie la stânga în jurul lui x.

În acest caz, după rebalansare scade înălţimea subarborelui curent, comparativ cu înălţimea avută înainte de ştergere. Este deci necesară continuarea algoritmului de la părintele nodului curent x.