ARBORI 2-3-4

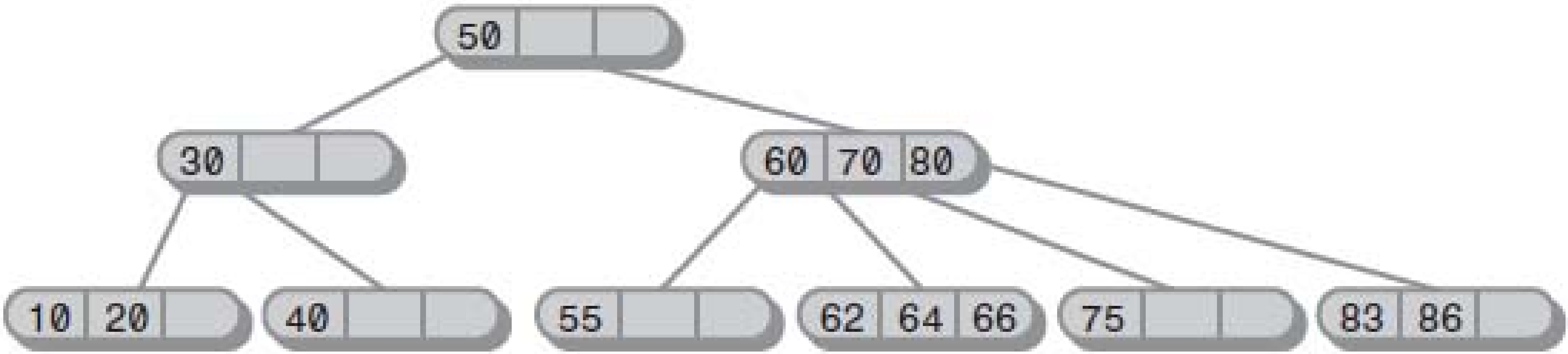
Șl. Dr. Ing. Șerban Radu
Departamentul de Calculatoare
Facultatea de Automatică și Calculatoare

Arbori multicăi

- Într-un arbore binar, fiecare nod conține o anumită informație și poate avea cel mult doi fii
- Dacă permitem existența mai multor informații și a mai multor fii pentru fiecare nod, obținem un **arbore multicăi**

Arbori 2-3-4

- Arborii 2-3-4 sunt arbori multicăi, în care fiecare nod poate avea cel mult patru fii
- Arborii 2-3-4 sunt arbori echilibrați, la fel ca arborii bicolori
- Eficiența lor este ceva mai scăzută, dar sunt mai ușor de programat


50
30
60 70 80
10 20
40
55
62 64 66
75
83 86

Exemplu

- Fiecare **nod** poate conține **unul, două sau trei elemente**
- Primele trei noduri au fii, în timp ce toate cele șase noduri de pe ultimul nivel sunt frunze, care prin definiție nu au fii
- Într-un arbore 2-3-4, **toate frunzele se află întotdeauna pe același nivel**

Explicația numelui

- Cifrele 2,3 și 4 se referă la numărul legăturilor către noduri fii, care pot fi conținute de un nod oarecare
- Pentru nodurile care nu sunt frunze, sunt posibile 3 situații:
  - Un nod cu 1 element are întotdeauna 2 fii
  - Un nod cu 2 elemente are întotdeauna 3 fii
  - Un nod cu 3 elemente are întotdeauna 4 fii

Observații

- Un nod care nu este frunză trebuie să aibă cu un fiu mai mult decât numărul de elemente pe care îl conține
- Dacă numărul legăturilor către fii este **L**, iar numărul elementelor este **D**, avem:
- **L = D + 1**

Observații

- Un nod frunză nu are niciun fiu, dar poate conține una, două sau chiar trei elemente
- Nu se admite existența nodurilor vide
- Din cauză că arborii 2-3-4 pot avea noduri cu până la patru fii, ei se mai numesc **arbori multicăi de ordinul al patrulea**

Observații

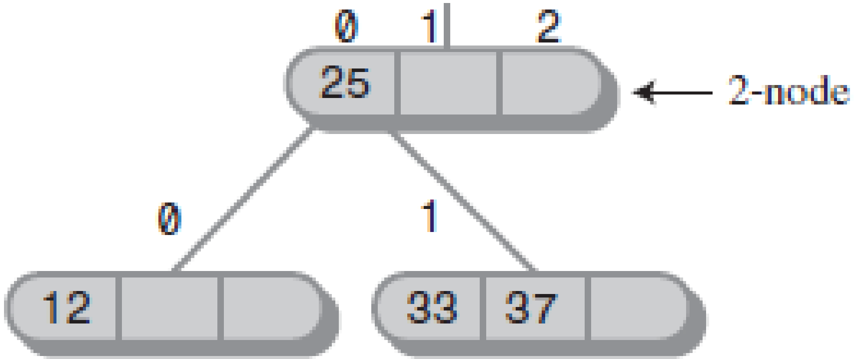
- Un nod nu poate avea un singur fiu, așa cum se întâmplă cu nodurile din arborii binari
- Un arbore binar poate fi gândit ca un caz particular de arbore multicăi de ordinul al doilea, din cauză că fiecare nod poate avea maximum doi fii

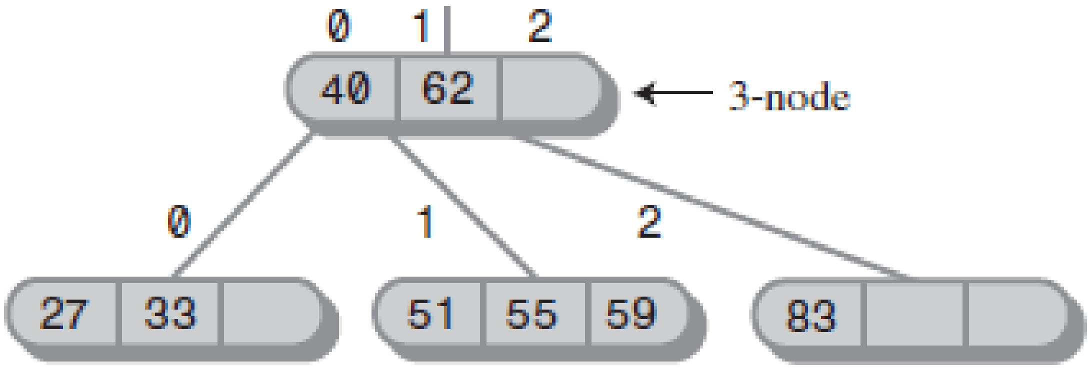
Observații

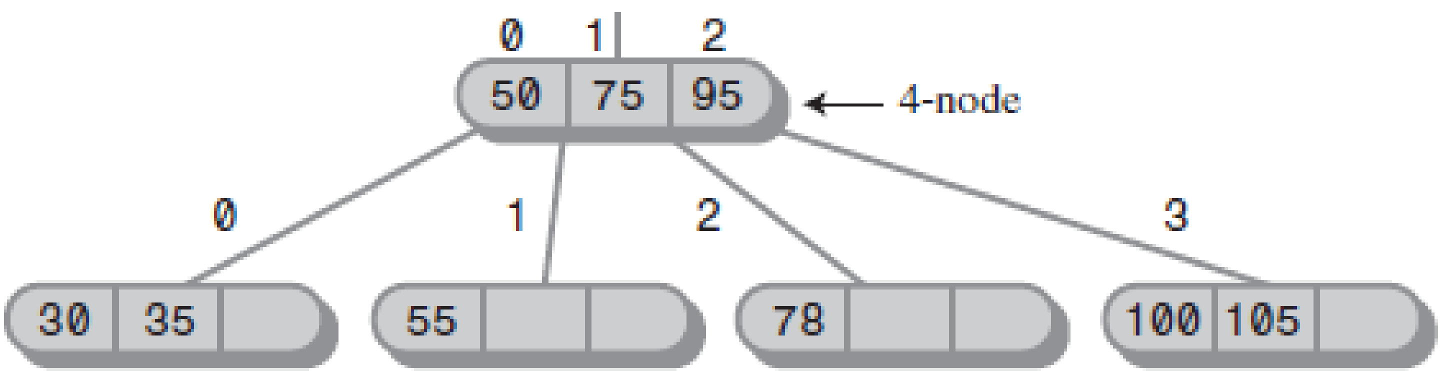
- Într-un arbore binar, un nod are cel mult doi fii
- Situația în care un nod are o singură legătură, către fiul său drept, sau către cel stâng, este permisă
- Cealaltă legatură poate avea valoarea NULL

Observații

- În arborii 2-3-4, existența nodurilor cu un singur fiu nu este permisă
- Un nod cu un singur element trebuie să aibă două legături, cu excepția cazului în care nodul este o frunză și nu are nicio legătură
- Un nod cu 2 legături se numește **2-nod**, cu 3 legături **3-nod**, iar cu 4 legături **4-nod**


0 1 2
25
2-node
0
1
12
33 37


0 1 2
40 62
3-node
0
1
2
27 33
51 55 59
83

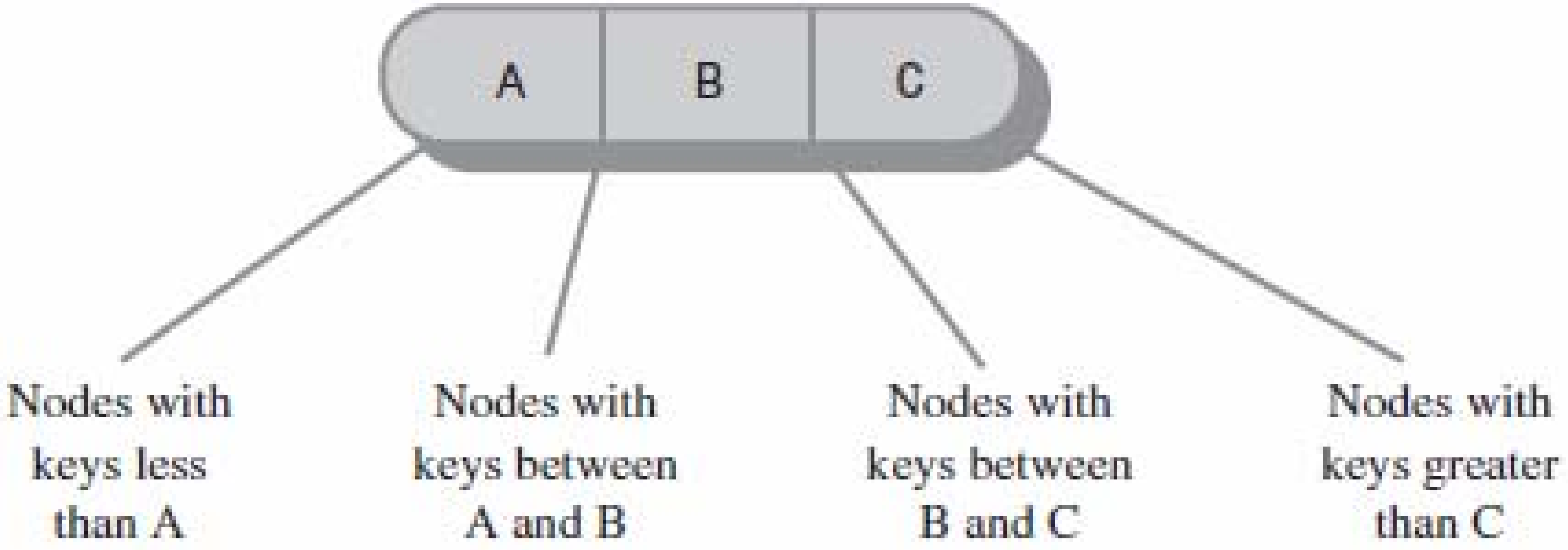

0 1 2
50 75 95
4-node
0
1
2
3
30 35
55
78
100 105

Convenții

- Se numerotează elementele dintr-un nod cu indici de la 0 la 2, iar legăturile către fii, cu indici de la 0 la 3
- Elementele dintr-un nod sunt dispuse în **ordinea crescătoare** a cheilor
- Elementele se scriu de la stânga la dreapta (de la indici mici spre indici mari)

Reguli

- Toți fiii din subarborele cu rădăcina în fiul 0 au chei mai mici decât elementul 0
- Toți fiii din subarborele cu rădăcina în fiul 1 au chei mai mari decât elementul 0, dar mai mici decât elementul 1
- Toți fiii din subarborele cu rădăcina în fiul 2 au chei mai mari decât elementul 1, dar mai mici decât elementul 2
- Toți fiii din subarborele cu rădăcina în fiul 3 au chei mai mari decât elementul 2


A
B
C
Nodes with keys less than A
Nodes with keys between A and B
Nodes with keys between B and C
Nodes with keys greater than C

Observații

- Echilibrul arborelui se păstrează, chiar dacă se inserează o secvență ordonată crescător (sau descrescător)
- Facilitatea de autoechilibru a arborilor 2-3-4 provine din modul de inserare a noilor elemente

Căutare

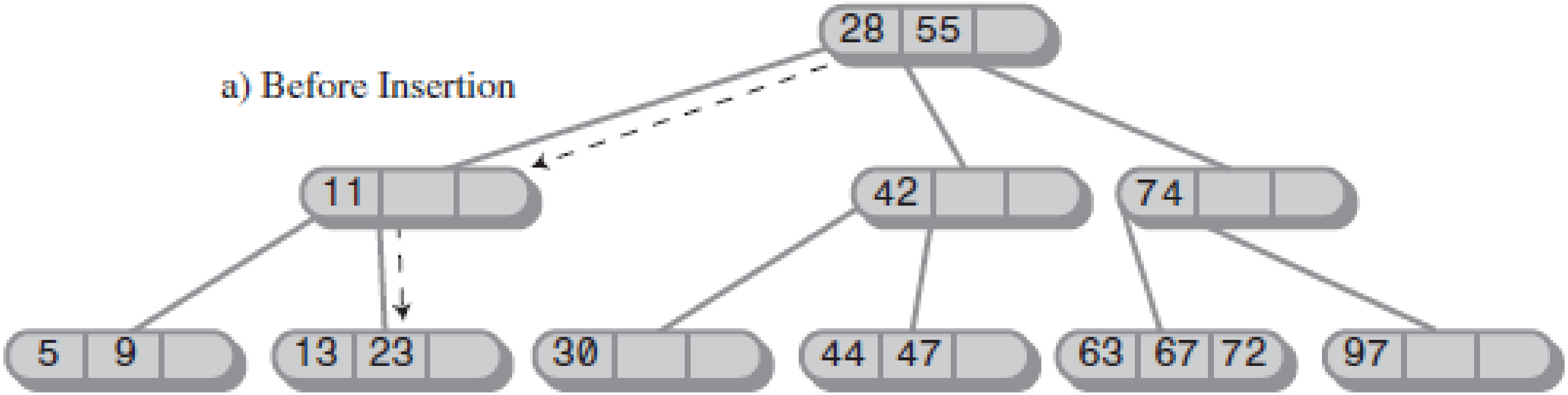
- Căutarea unui element cu o anumită cheie este similară operației dintr-un arbore binar de căutare
- Se pornește de la rădăcină și, dacă nu se găsește cheia căutată, se selectează legătura care conduce la subarborele asociat domeniului corespunzător de valori

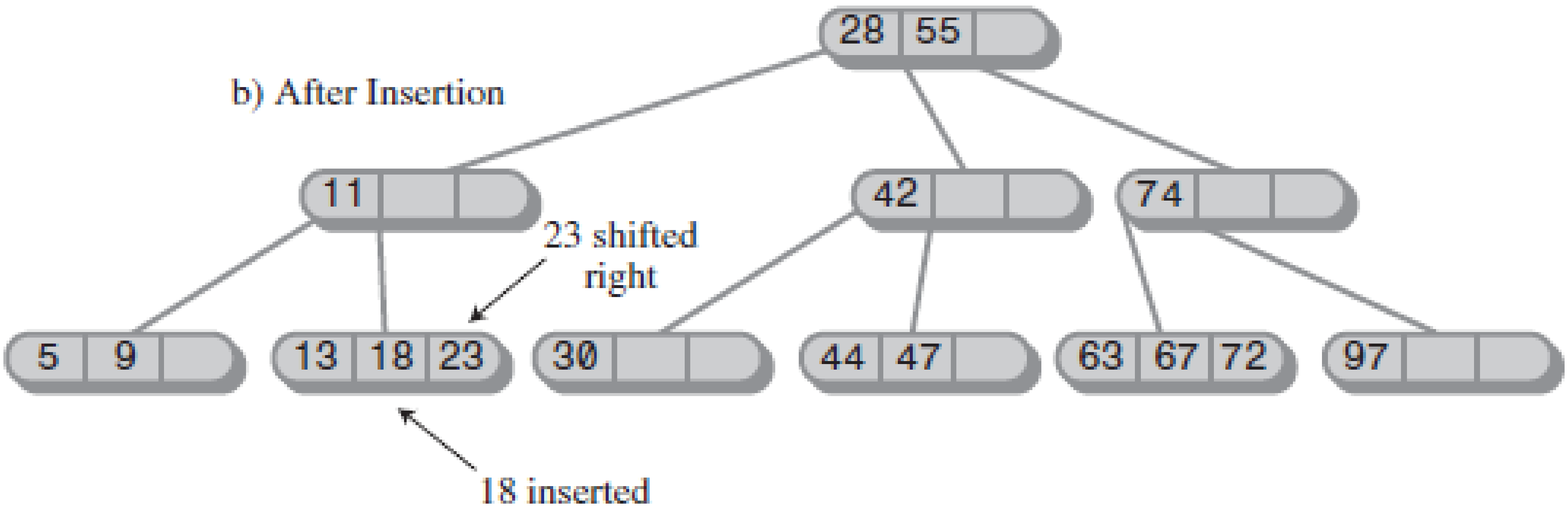
Inserare

- Noile elemente sunt întotdeauna inserate în frunze, care sunt dispuse pe nivelul de la baza arborelui
- Dacă elementele ar fi fost inserate în nodurile care au fii, atunci numărul acestor fii trebuia sa se modifice, pentru a menține structura arborelui

Inserare

- Dacă nu se întâlnesc noduri complete în cursul căutării, inserarea este simplă
- Când se ajunge la nodul frunză potrivit, noul element este inserat în acesta
- Inserarea poate implica deplasarea a unul sau două elemente dintr-un nod, astfel încât cheile să rămână în ordinea corectă și după inserarea noului element


a) Before Insertion
28
55
11
42
74
5
9
13
23
30
44
47
63
67
72
97


b) After Insertion
28
55
11
42
74
23 shifted right
5
9
13
18
23
30
44
47
63
67
72
97
18 inserted

Divizarea nodurilor

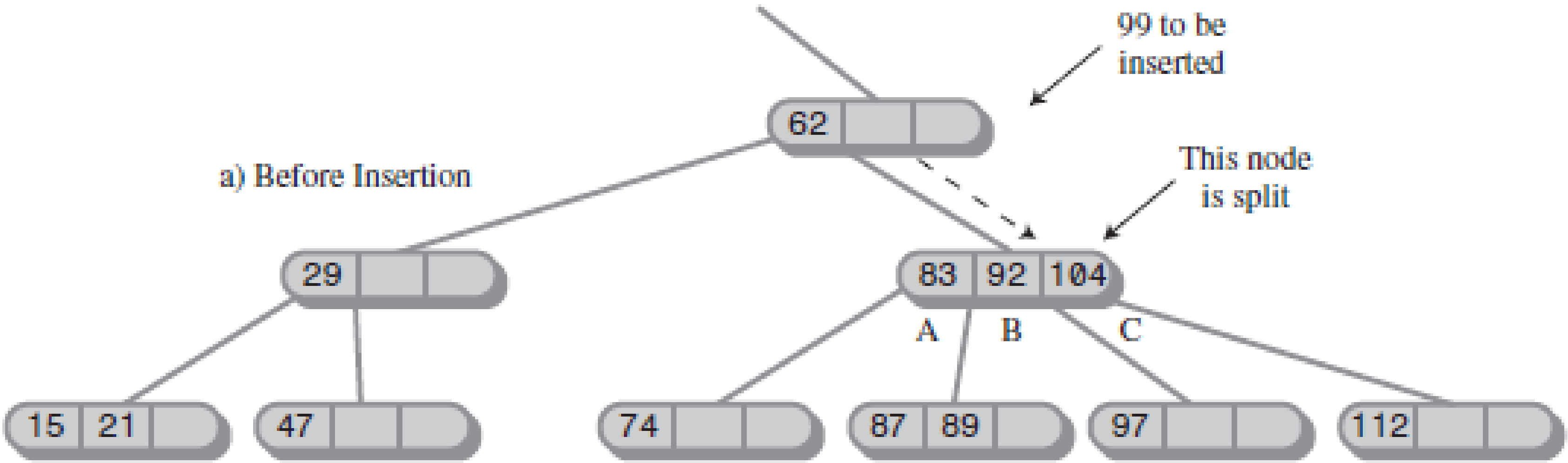
- Inserarea se complică când se întâlnește un nod complet pe calea descendentă către locul de inserare
- Dacă aceasta se întâmplă, nodul va suferi o divizare
- Rolul acestei operații este de a menține echilibrul arborelui

Divizarea nodurilor

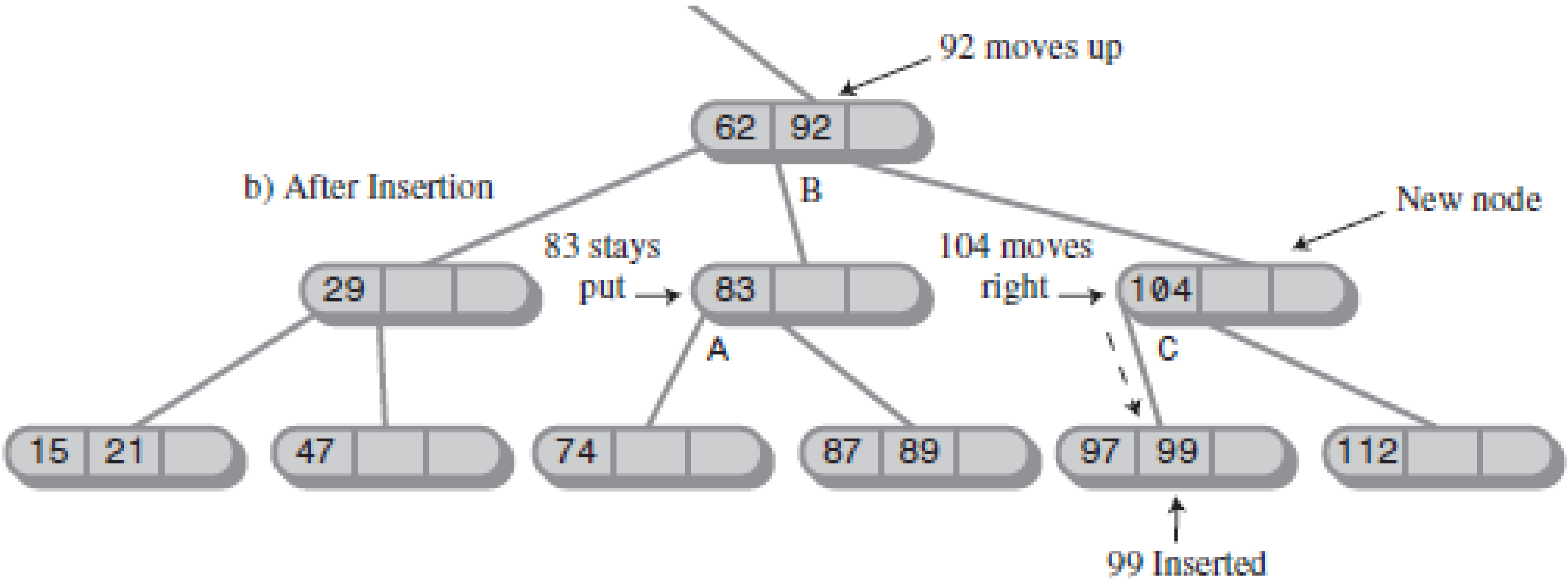
- Se notează elementele din nodul care va fi divizat cu A, B și C
- Se presupune că nodul care va fi divizat nu este rădăcină
- 1. Se creează un nod nou, vid. Acesta este fratele nodului care va fi divizat, fiind plasat la dreapta acestuia

Divizarea nodurilor

- 2. Elementul C este deplasat în noul nod
- 3. Elementul B este deplasat în părintele nodului divizat
- 4. Elementul A rămâne în locul în care este
- 5. Cei doi fii situați cel mai la dreapta sunt deconectați de la nodul divizat și conectați la noul nod



99 to be inserted
62
a) Before Insertion
This node is split
29
83
92
104
A
B
C
15
21
47
74
87
89
97
112


92 moves up
62
92
b) After Insertion
B
New node
83 stays
put
104 moves
right
29
83
104
A
C
15
21
47
74
87
89
97
99
112
99 Inserted

Observații

- Operația de diviziune poate fi descrisă prin aceea că un 4-nod este transformat în două 2-noduri
- Efectul divizării nodului este deplasarea elementelor în sus (spre nodul părinte) și spre dreapta
- Această rearanjare menține echilibrul în arbore

Observații

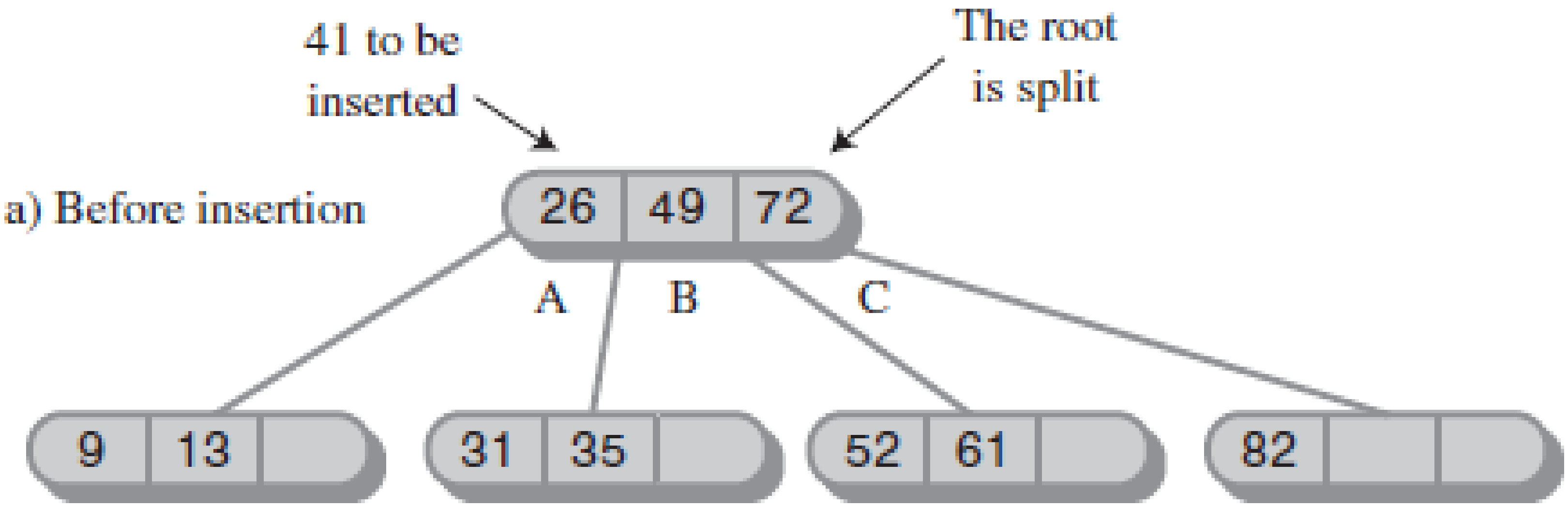
- Este posibil să existe mai multe noduri complete pe calea de la rădăcină către locul de inserare
- În acest caz, se vor efectua diviziuni multiple

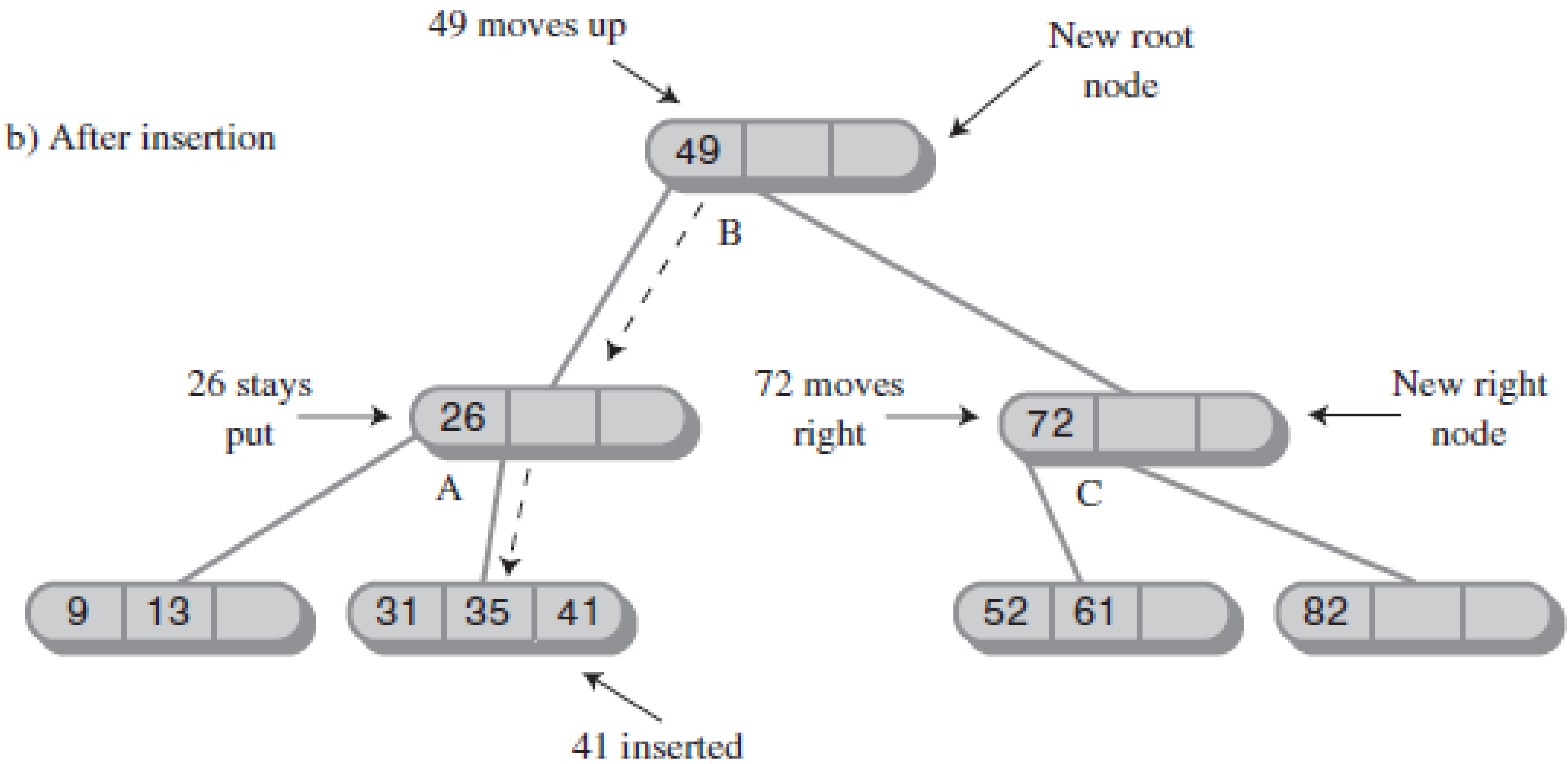
Divizarea rădăcinii arborelui

- Dacă nodul **rădăcină** întâlnit la începutul căutării locului de inserare este **complet**, procesul de divizare implică următorii pași:
- 1. Se creează un nod nou, care va deveni noua rădăcină și părintele nodului divizat
- 2. Se creează un al doilea nod, care va deveni fratele nodului divizat

Divizarea rădăcinii arborelui

- 3. Elementul C este deplasat în noul frate
- 4. Elementul B este deplasat în noua rădăcină
- 5. Elementul A rămâne pe loc
- 6. Cei doi fii ai nodului divizat aflați cel mai la dreapta sunt deconectați de la acel nod și conectați la noul nod aflat în partea dreaptă


41 to be inserted
The root is split
a) Before insertion
26
49
72
A
B
C
9
13
31
35
52
61
82



b) After insertion
49 moves up
New root node
49
B
26 stays put
26
72 moves right
72
New right node
A
C
9
13
31
35
41
52
61
82
41 inserted

Observații

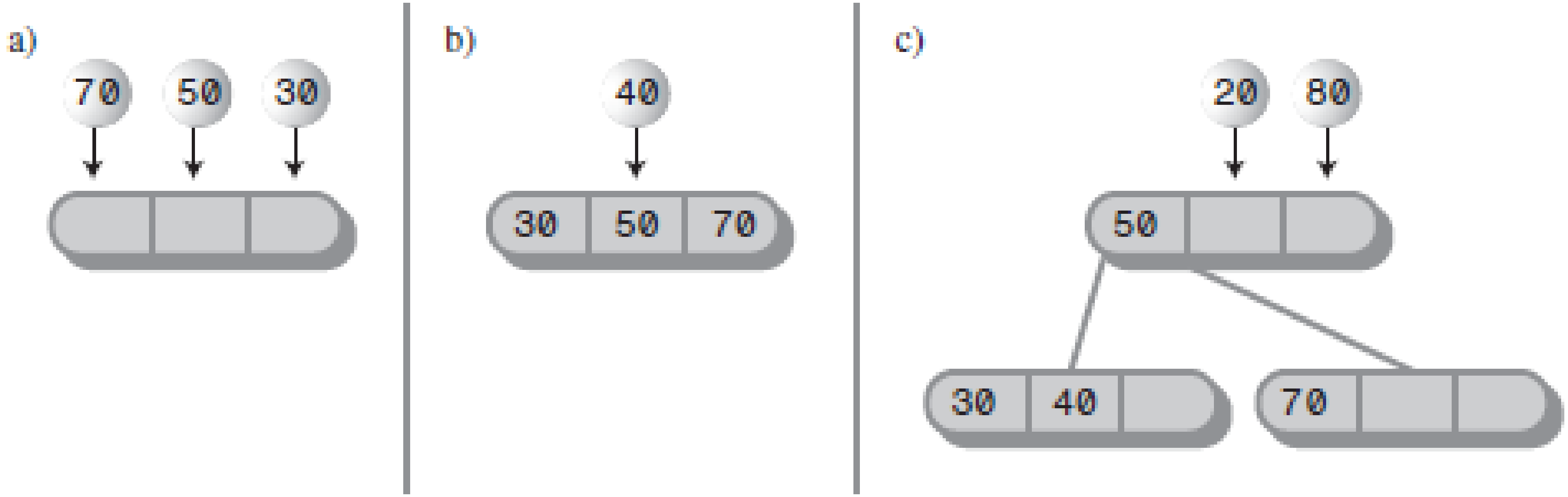
- Operația creează o nouă rădăcină, situată cu un nivel mai sus față de vechea rădăcină
- Înălțimea totală a arborelui crește cu o unitate
- Se poate afirma că un 4-nod a fost divizat în trei 2-noduri
- După efectuarea diviziunii, căutarea locului de inserare continuă, parcurgând arborele în jos

Divizarea la parcurgerea descendentă

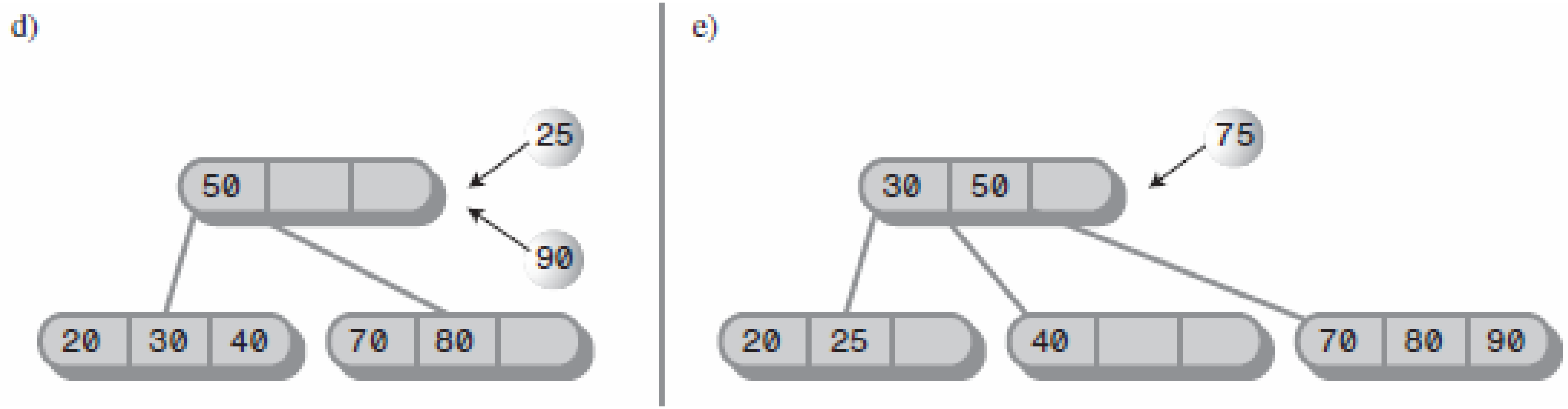
- Din cauză că divizarea tuturor nodurilor complete are loc la parcurgerea descendentă a arborelui, efectul unei operații de divizare nu se poate propaga înapoi spre rădăcină
- Părintele unui nod care urmează să fie divizat este, cu siguranță, incomplet

Divizarea la parcurgerea descendentă

- Elementul B din nodul divizat va putea fi inserat în nodul părinte, fără a provoca divizarea acestuia
- Dacă nodul părinte are deja doi fii înaintea divizării unuia dintre aceștia, va deveni un nod complet
- Divizarea nodului se va efectua la următoarea operație de inserare în care nodul va fi implicat

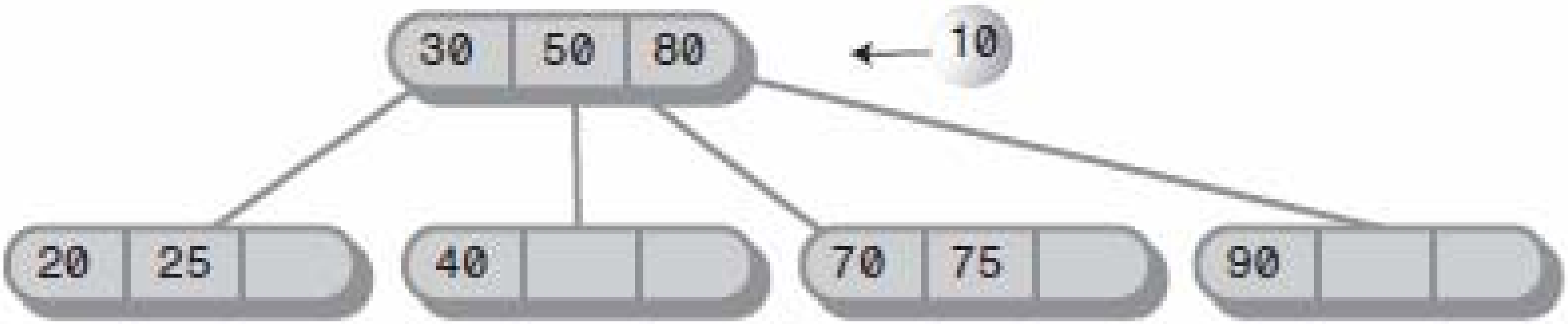


a)
70
50
30
b)
40
30
50
70
c)
20
80
50
30
40
70

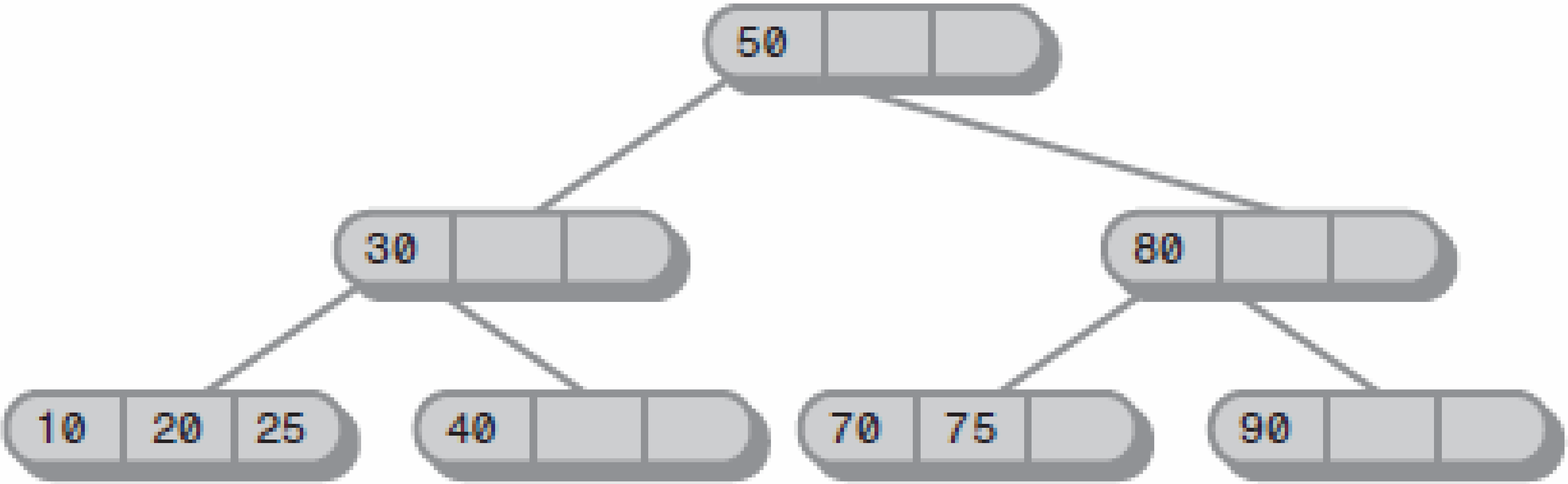


d)
50
25
90
20 30 40
70 80
e)
30 50
75
20 25
40
70 80 90

f)



g)



Arbori 2-3-4 și arbori bicolori

- Deși arborii 2-3-4 și arborii bicolori par entități distincte, cele două tipuri de arbori sunt echivalente
- Arborii de un anumit tip se pot transforma în arbori de celălat tip, apicând câteva reguli simple
- Operațiile necesare pentru menținerea echilibrării lor sunt echivalente

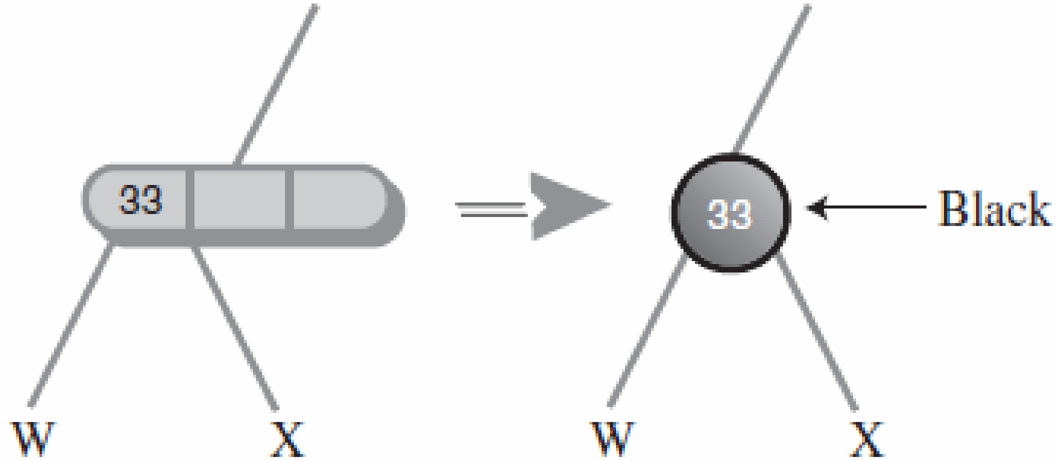
Arbori 2-3-4 și arbori bicolori

- Cele două tipuri de arbori sunt **izomorfe**
- Echivalența celor două structuri este utilă la analiza eficienței lor
- Arborii 2-3-4 au fost descoperiți primii
- Arborii bicolori au evoluat pornind de la arborii 2-3-4

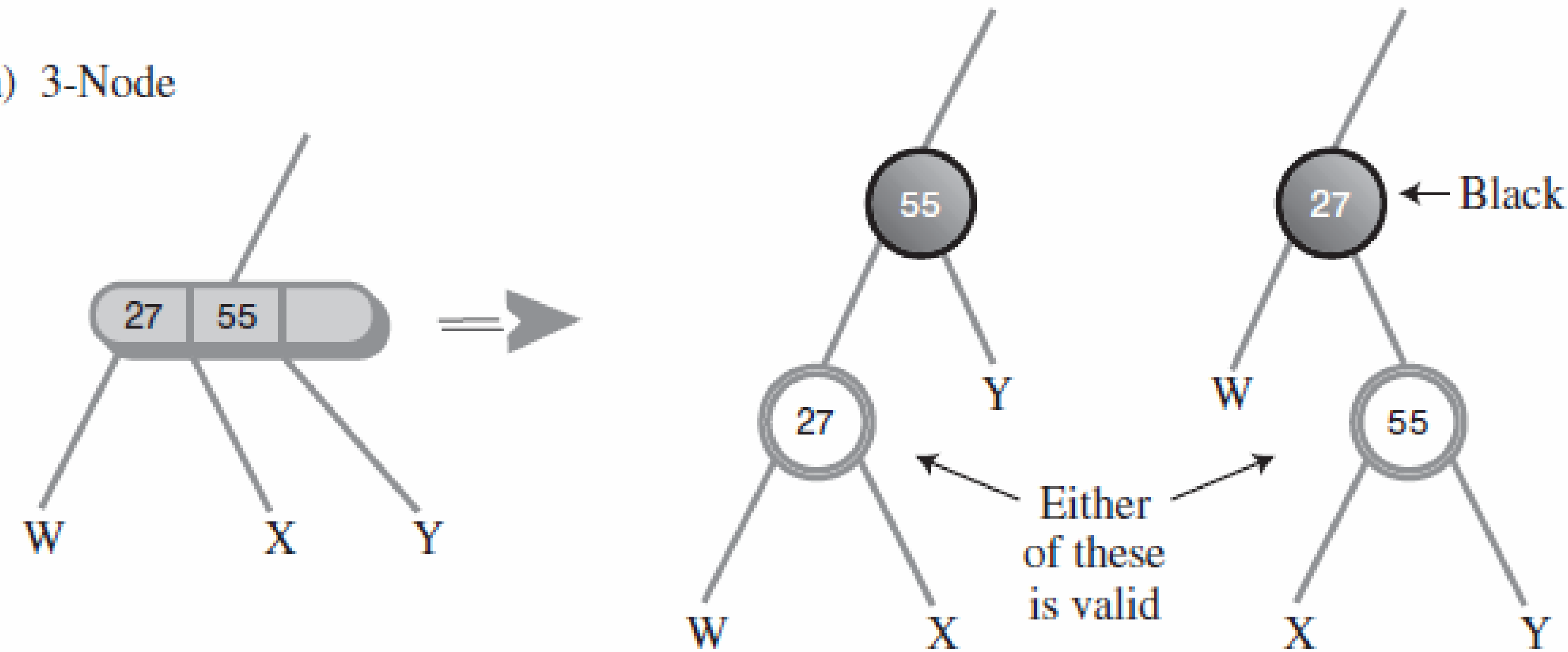
Transformarea unui arbore 2-3-4 în arbore bicolor

- 1. Se transformă toate 2-nodurile din arborii 2-3-4 în noduri negre
- 2. Se transformă orice 3-nod într-un nod fiu C (cu doi fii la rândul său) și un părinte P (având ca fii pe C și încă un nod)
- Obs: Nu contează care element devine fiul și care element devine părintele; C va fi roșu, iar P va fi negru

a) 2-Node



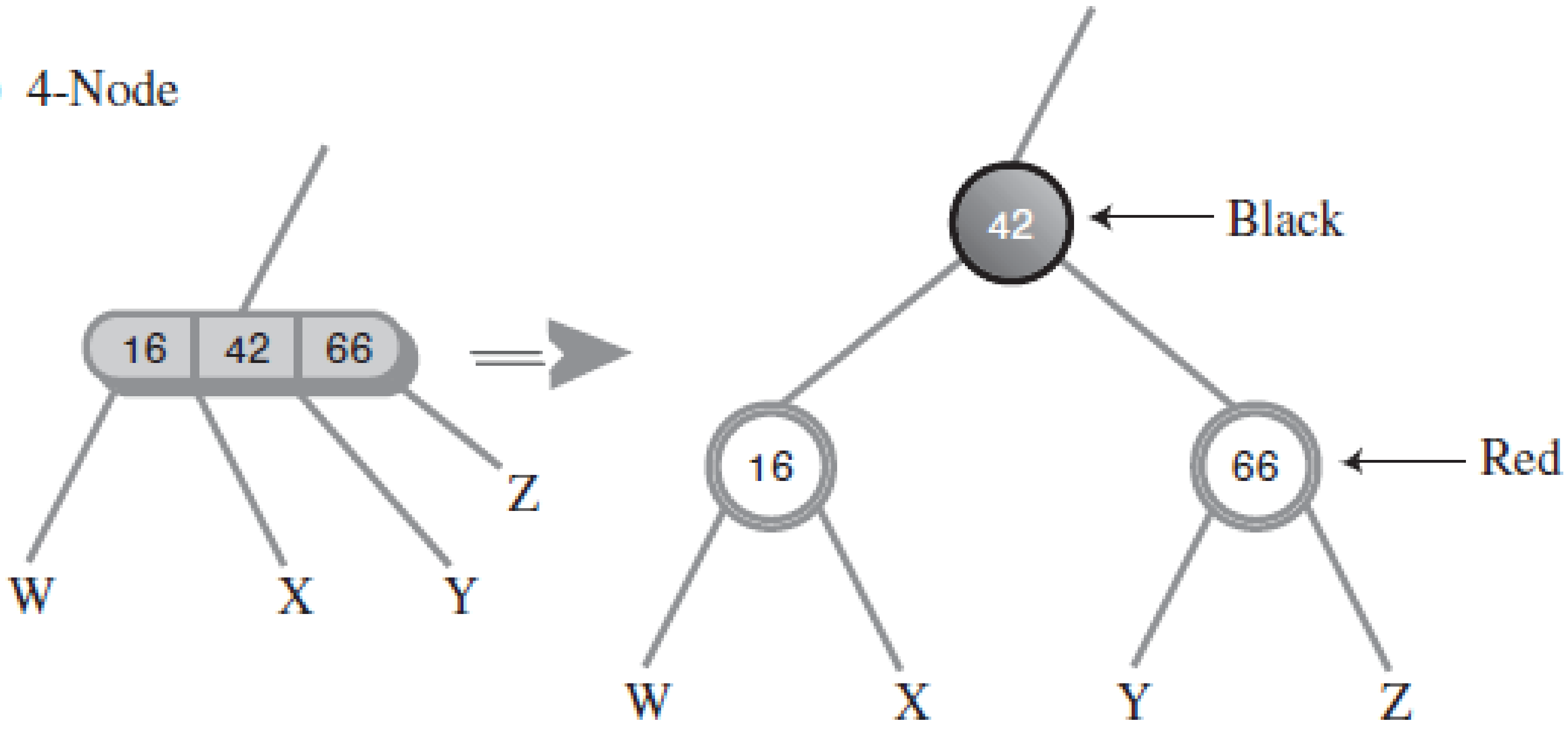
a) 3-Node



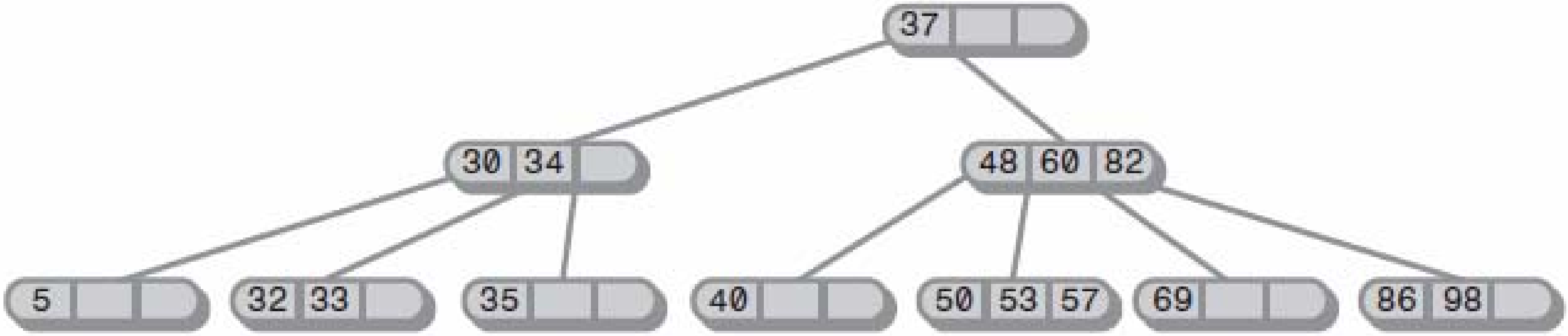
Transformarea unui arbore 2-3-4 în arbore bicolor

- 3. Se transformă toate 4-nodurile într-un părinte P cu doi fii C1 și C2, ambii cu câte doi fii
- Obs: C1 și C2 sunt roșii, în timp ce P este negru

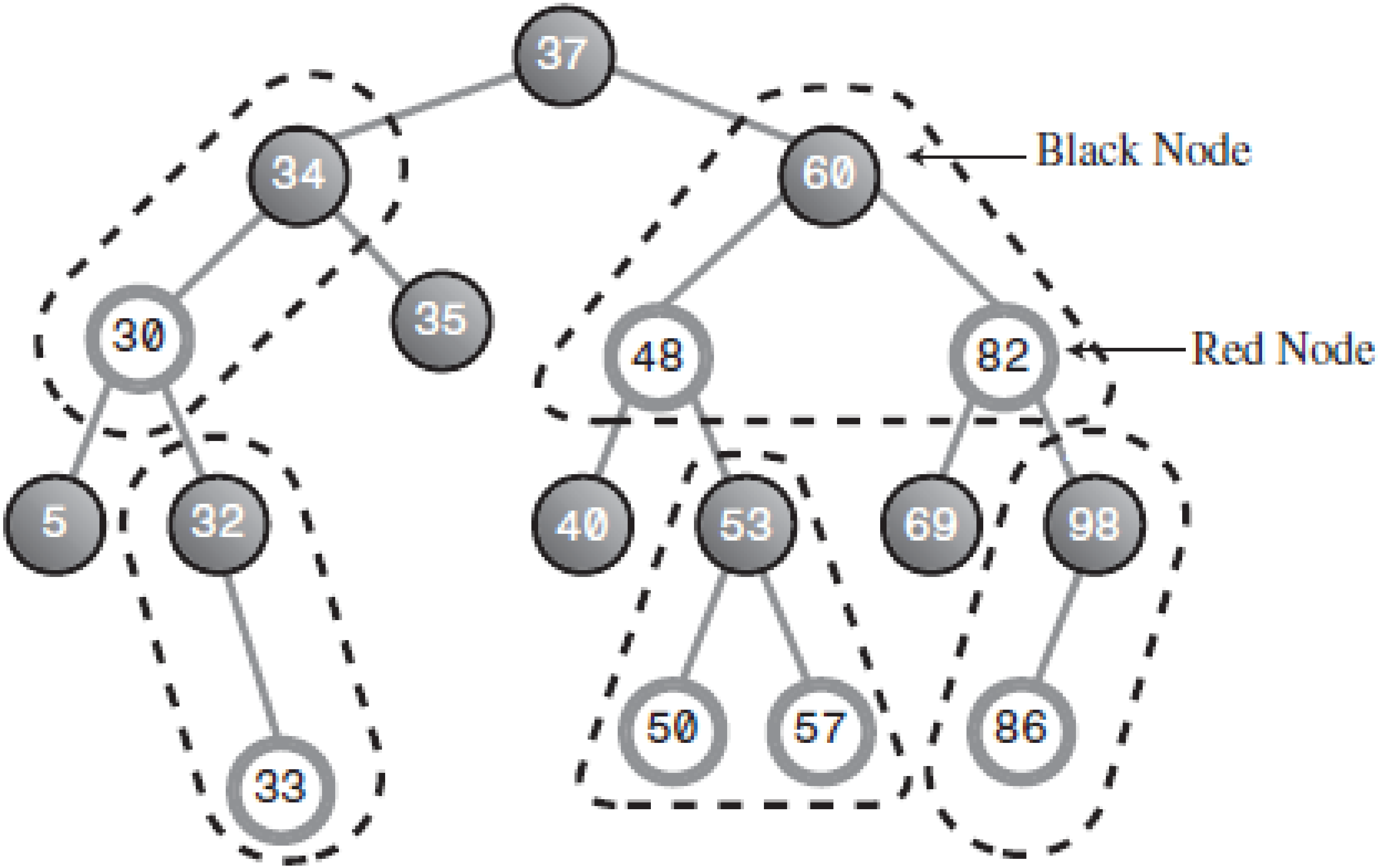
a) 4-Node



a) 2-3-4 tree



b) Red-black tree



Exemplu de transformare

- Liniile punctate înconjoară subarborii rezultați din 3-noduri și 4-noduri
- Regulile de colorare sunt automat satisfăcute, prin transformările efectuate:
  - Două noduri roșii nu vor fi niciodată conectate
  - Pe fiecare cale de la rădăcină spre frunze (sau noduri inexistente) există același număr de noduri negre

Observații

- Un 3-nod dintr-un arbore 2-3-4 este echivalent cu un părinte cu un fiu roșu dintr-un arbore bicolor
- Un 4-nod este echivalent cu un părinte cu doi fii roșii

Observații

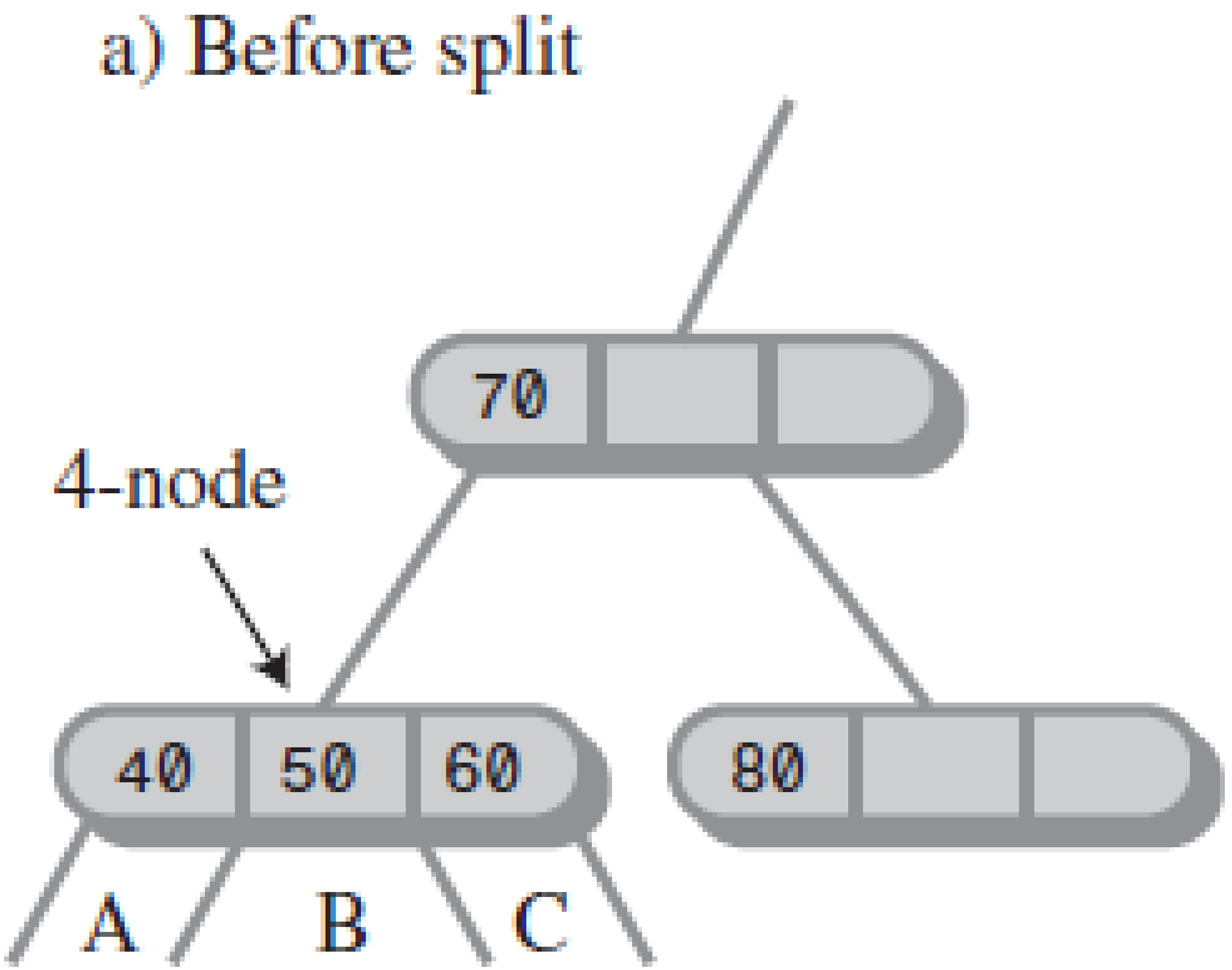
- Un nod părinte negru, cu un fiu tot negru, dintr-un arbore bicolor, nu poate reprezenta un 3-nod dintr-un arbore 2-3-4
- Acesta va reprezenta un 2-nod, cu un fiu care este tot un 2-nod
- În mod similar, un părinte negru cu doi fii negri, nu poate reprezenta un 4-nod

Echivalența operațională

- Nu numai structura de arbore bicolor este similară cu cea de arbore 2-3-4
- Operațiile permise asupra celor două tipuri de arbori sunt echivalente
- Într-un arbore 2-3-4, echilibrul se menține utilizând divizarea nodurilor
- În arborii bicolori, echilibrul se realizează cu ajutorul inversiunilor de culori și al rotațiilor

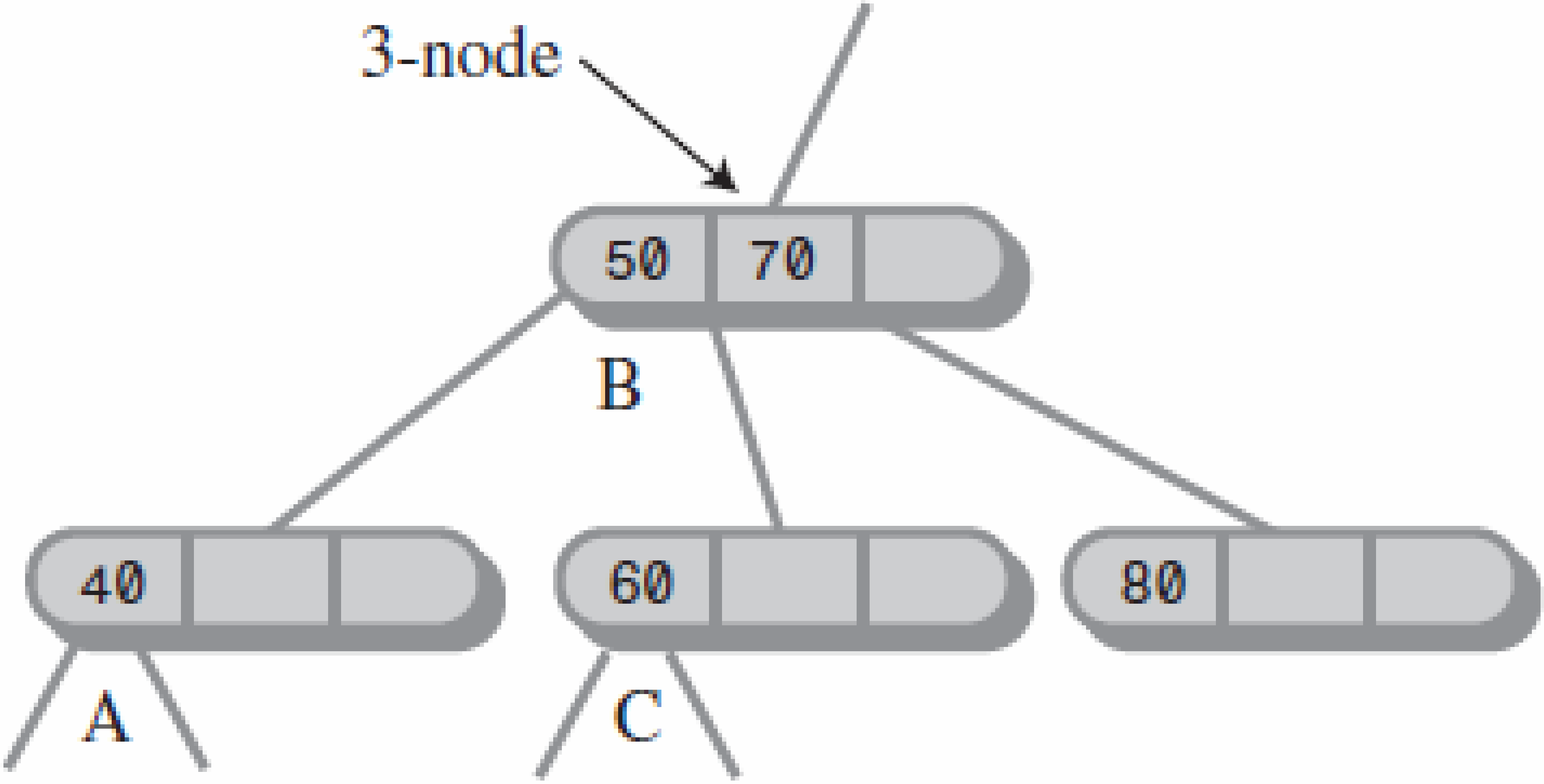
Divizarea 4-nodurilor și inversările de culori

- Pe măsură ce coborâm într-un arbore 2-3-4, în căutarea locului de inserare pentru un nod nou, fiecare 4-nod este divizat în două 2-noduri
- Într-un arbore bicolor, operația echivalentă este inversarea culorilor



a) Before split
70
4-node
40
50
60
80
A
B
C

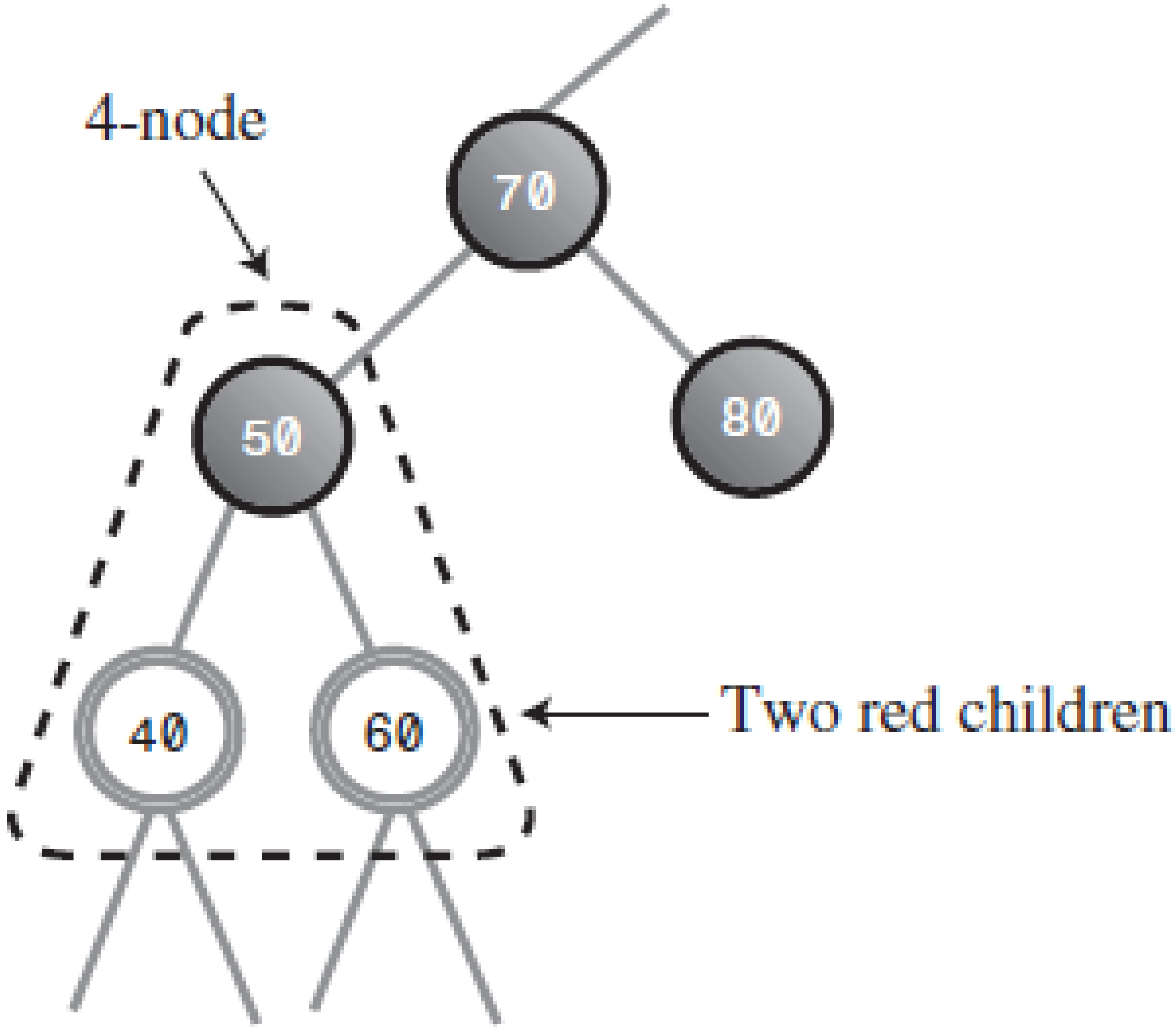
b) After split



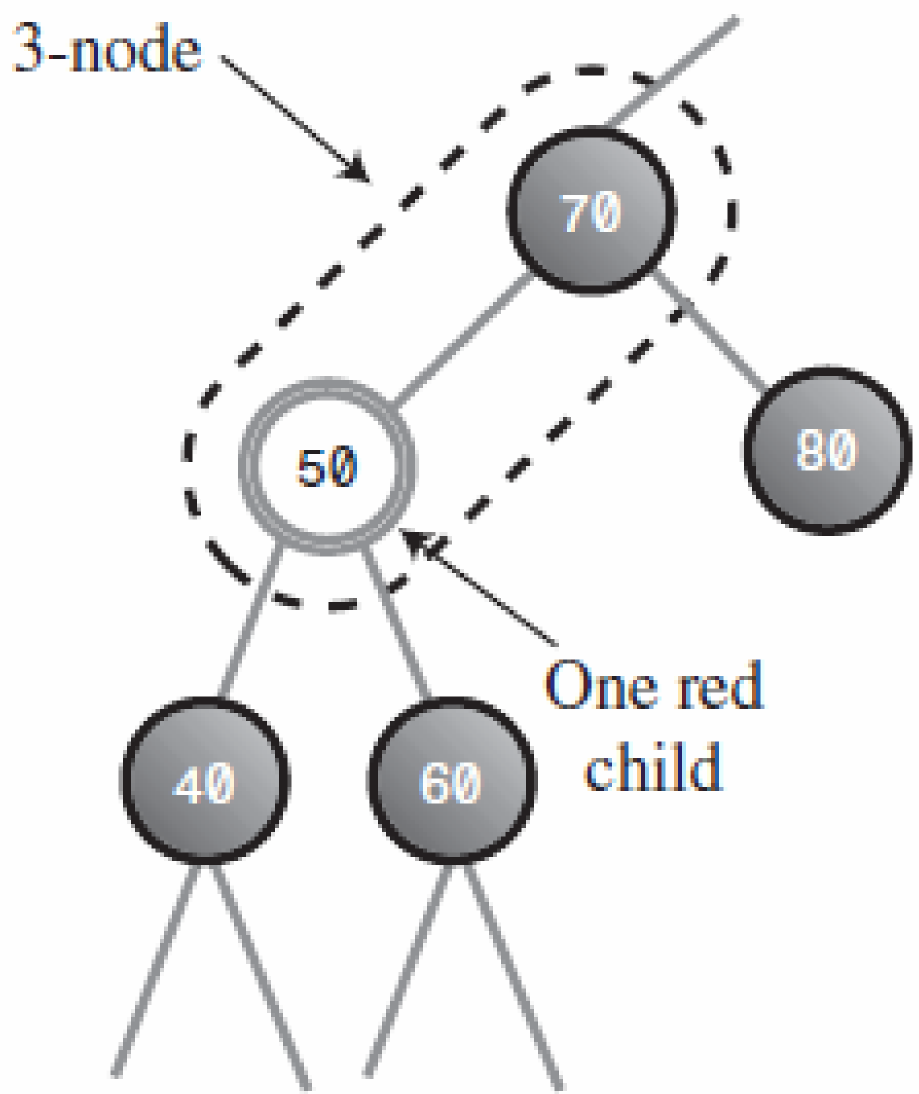
Exemplu

- Un 4-nod dintr-un arbore 2-3-4 este divizat în două 2-noduri, iar 2-nodul care era părintele 4-nodului va deveni 3-nod

c) Before color flip



d) After color flip



Exemplu

- În arborele bicolor echivalent cu arborele 2-3-4, linia punctată înconjoară mulțimea de noduri echivalentă cu 4-nodul
- Prin efectuarea unei inversări de culoare, nodurile 40 și 60 devin negre, iar 50 devine roșu

Observații

- Nodul 50, împreună cu părintele său, formează echivalentul unui 3-nod
- **Divizarea unui 4-nod**, în operația de inserare într-un **arbore 2-3-4**, este **echivalentă** cu **inversarea culorilor**, de la inserarea într-un **arbore bicolor**

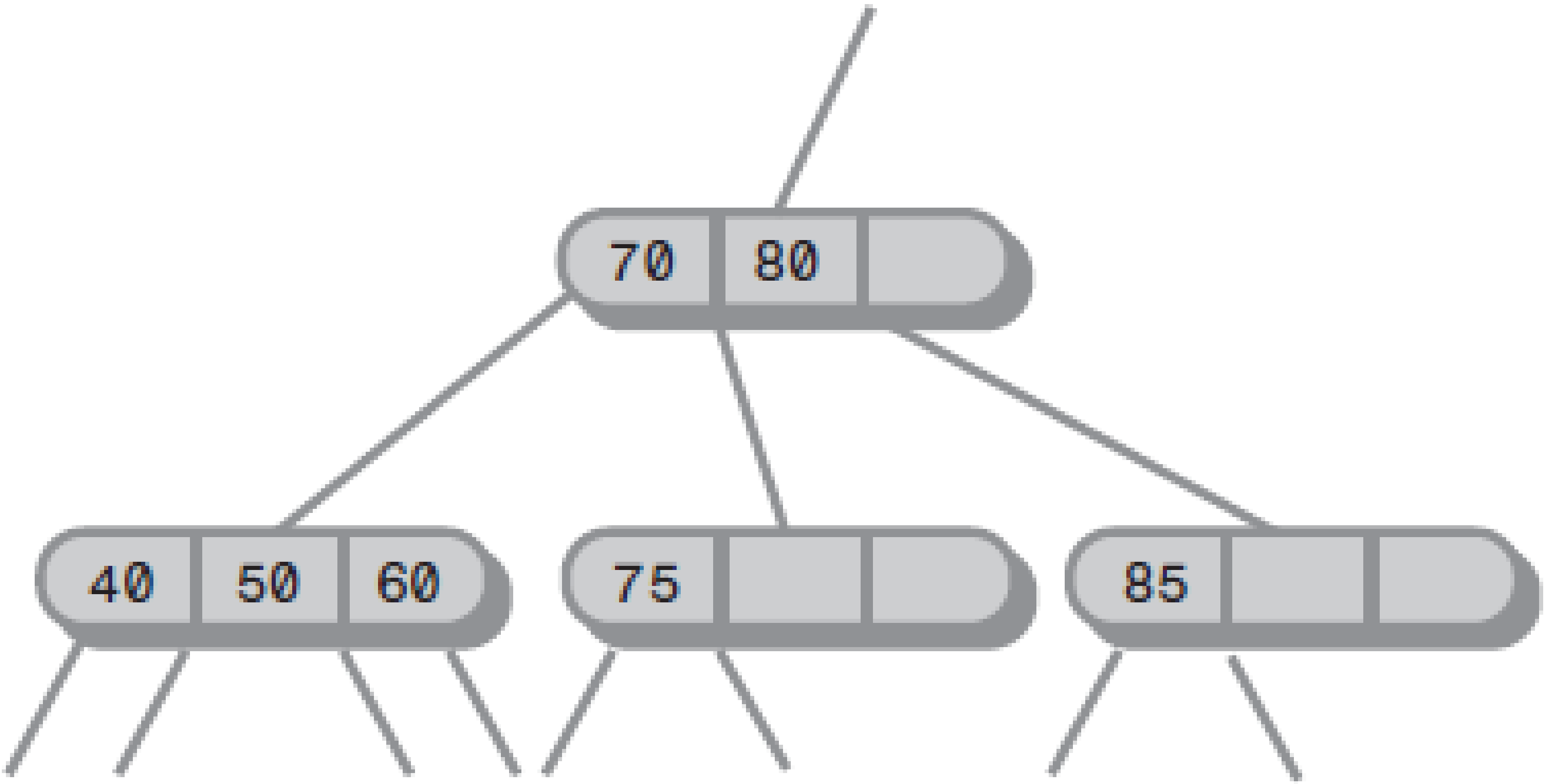
Divizarea 3-nodurilor și rotațiile

- La transformarea unui 3-nod dintr-un arbore 2-3-4 în echivalentul său bicolor, sunt posibile două situații
- Oricare dintre cele două elemente poate deveni nod părinte
- În funcție de care dintre ele este ales, nodul fiu va fi situat în dreapta sau în stânga, iar înclinarea muchiei dintre părinte și fiu va fi spre dreapta sau spre stânga

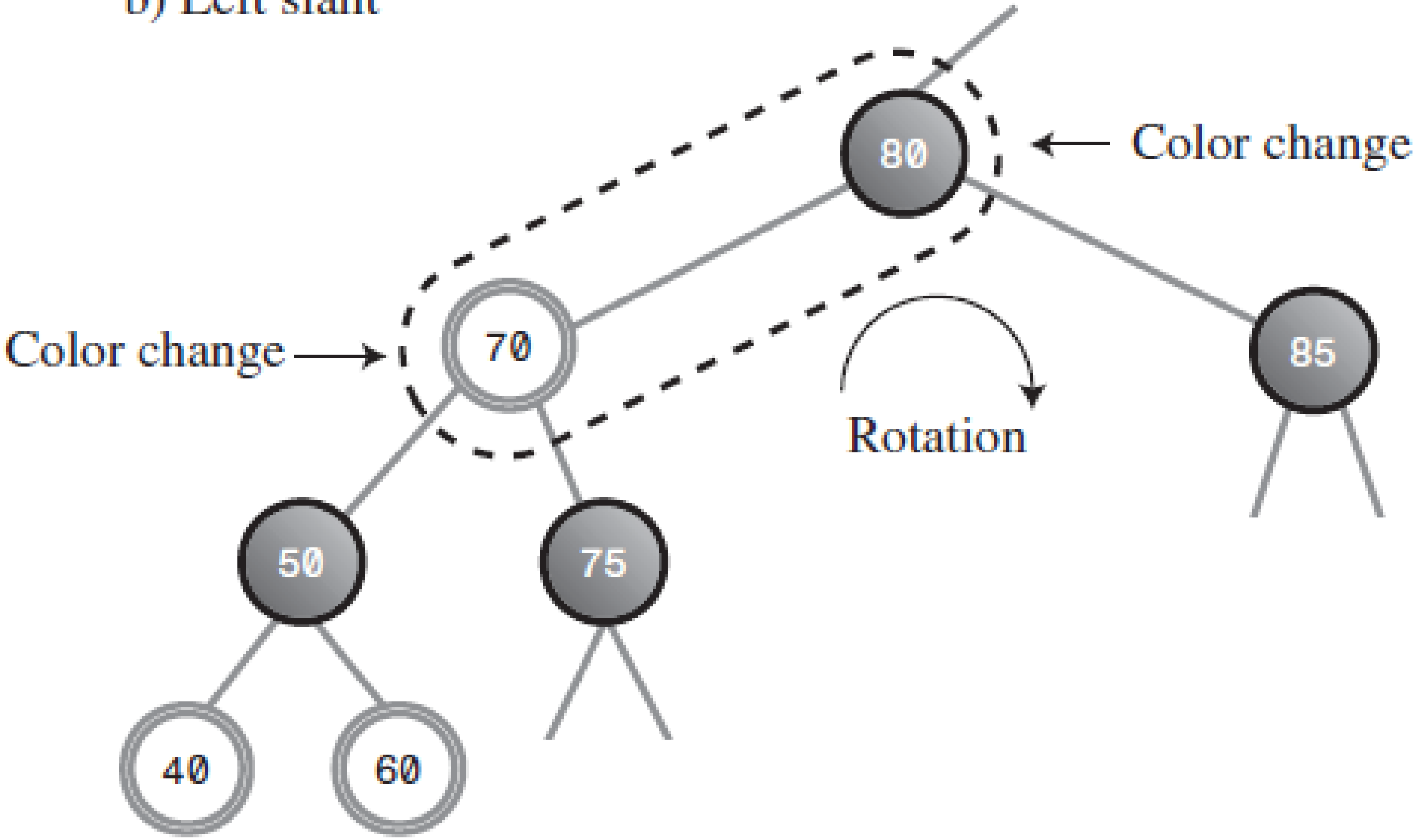
Observații

- Ambele poziții sunt corecte, dar este posibil să nu fie la fel de utile în acțiunea de echilibrare a arborilor

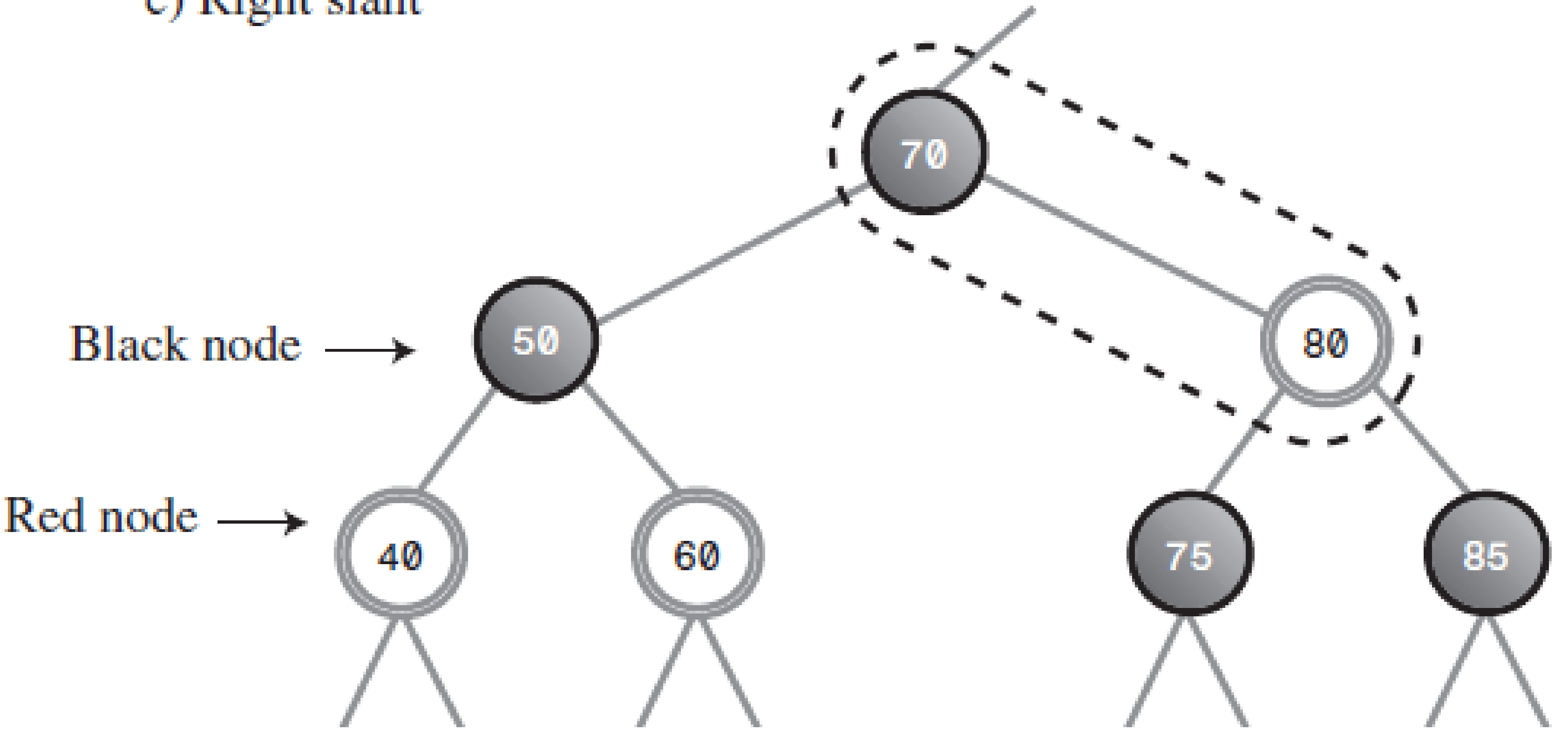
a) 2-3-4 tree



b) Left slant



c) Right slant



Exemplu

- Un arbore 2-3-4 poate fi transformat în doi arbori bicolori echivalenți, derivați din arborele 2-3-4 inițial, prin aplicarea regulilor de transformare
- Diferența dintre cei doi arbori constă în alegerea elementului din cadrul 3-nodului, care va fi părintele:
  - Părintele poate fi 80
  - Părintele poate fi 70

Observații

- Deși aceste poziții sunt ambele valide, se observă că primul arbore bicolor nu este echilibrat, iar al doilea este echilibrat
- Pentru a echilibra primul arbore bicolor, se aplică o rotație spre dreapta și două inversări de culori
- Se generează al doilea arbore bicolor, care este echilibrat

Observații

- Există o **echivalență** între **rotațiile** din **arborii bicolori** și **alegerea nodului părinte** din operația de transformare a unui arbore 2-3-4 în arbore bicolor
- O echivalență similară există și pentru dubla rotație, necesară în cazul nodurilor care sunt nepoți interiori

Eficiența arborilor 2-3-4

- Analiza eficienței arborilor 2-3-4 este mai dificilă decât analiza eficienței arborilor bicolori, dar echivalența dintre cele două categorii de arbori oferă un punct de pornire

Viteza

- Într-un arbore bicolor, fiecare operație de căutare conduce la vizitarea câte unui nod de pe fiecare nivel, indiferent dacă se caută un element existent sau locul de inserare pentru un nou element
- Numărul de niveluri dintr-un arbore bicolor (care este un arbore binar echilibrat) este aproximativ $\log_2(N+1)$, deci duratele de căutare vor fi proporționale cu această valoare

Viteza

- În arborii 2-3-4, trebuie să vizităm câte un singur nod de pe fiecare nivel, dar acești arbori au mai puține niveluri, în comparație cu arborii bicolori cu același număr de elemente – vezi exemplele precedente

Observații

- În arborii 2-3-4, pot exista până la patru fii pentru fiecare nod
- Dacă fiecare nod este complet, înălțimea arborelui este direct proporțională cu $\log_4 N$
- Înălțimea unui arbore 2-3-4 reprezintă jumătate din cea a unui arbore bicolor cu același număr de noduri, dacă nodurile din arborele 2-3-4 sunt toate complete

Observații

- Din cauză că, în general, nu toate nodurile sunt complete, înățimea unui arbore 2-3-4 variază între limitele $\log_2(N+1)/2$ și $\log_2(N+1)$
- Înățimea redusă a arborilor 2-3-4 determină diminuarea corespunzătoare a timpului de căutare, în comparație cu arborii bicolori

Obervații

- Fiecare nod conține mai multe elemente care trebuie examinate, ceea ce determină o creștere a timpului de căutare
- Din cauză că elementele dintr-un nod sunt examinate utilizând o căutare liniară, timpul de căutare va crește de M ori, M reprezentând numărul mediu de elemente din fiecare nod
- Timpul de căutare este proporțional cu $M*\log_4 N$

Observații

- Unele noduri conțin un singur element, altele 2 sau 3
- Dacă estimăm că media numărului de elemente este 2, rezultă un timp de căutare proporțional cu $2*\log_4 N = \log_2 N$

Observații

- Pentru arborii 2-3-4, numărul mai mare de elemente din fiecare nod tinde să anuleze avantajul adus de înălțimea redusă a arborilor
- Timpii de căutare pentru un arbore 2-3-4 și pentru un arbore bicolor sunt aproximativi egali, fiind de ordinul O(logN)

Memoria necesară

- Fiecare nod dintr-un arbore 2-3-4 necesită memorarea pointerilor către trei elemente și, respectiv, către cei patru fii
- Acest spațiu poate fi alocat sub formă de tablouri sau de variabile individuale
- Nu toată memoria alocată este utilizată efectiv

Memoria necesară

- Un nod cu un singur element va irosi două treimi din spațiul alocat pentru elemente și jumătate din cel alocat pentru fii
- Un nod cu două elemente va irosi o treime din spațiul alocat pentru elemente și un sfert din cel alocat pentru fii, deci va utiliza efectiv numai 5/7 din spațiul alocat

Observații

- Presupunând că, în medie, nodurile au câte două elemente, rezultă că aproximativ 2/7 din spațiul de memorare alocat este neutilizat
- Datorită echilibrării lor, arborii bicolori conțin mai puține noduri cu un singur fiu, deci aproape toată memoria alocată este utilizată efectiv

Observații

- Arborii bicolori sunt mai eficienți decât arborii 2-3-4, din punct de vedere al utilizării memoriei
- Din punct de vedere al programării, arborii 2-3-4 sunt mai ușori de implementat

Concluzii

- Un arbore multicăi are mai multe chei și mai mulți fii decât un arbore binar
- Arborii 2-3-4 sunt arbori multicăi, în care fiecare nod are cel mult trei elemente și cel mult patru fii
- Într-un arbore multicăi, cheile din fiecare nod sunt ordonate crescător

Concluzii

- În arborii 2-3-4, toate inserările se efectuează în nodurile frunză, care se află toate pe același nivel
- Arborii 2-3-4 conțin trei tipuri de noduri:
  - 2-noduri, care au o cheie și doi fii
  - 3-noduri, care au două chei și trei fii
  - 4-noduri, care au trei chei și patru fii
- Arborii 2-3-4 nu pot conține 1-noduri
- Căutarea într-un arbore 2-3-4 presupune examinarea cheilor din fiecare nod

Concluzii

- Dacă valoarea căutată nu se găsește în nodul curent, următorul nod ales în procesul de căutare va fi:
  - Fiul numerotat cu 0, când valoarea este mai mică decât cheia 0
  - Fiul 1, când valoarea este între cheia 0 și cheia 1
  - Fiul 2, când valoarea este între cheia 1 și cheia 2
  - Fiul 3, când valoarea este mai mare decât cheia 2

Concluzii

- Inserarea într-un arbore 2-3-4 presupune divizarea fiecărui nod complet, la parcurgerea descendentă a arborelui, în căutarea locului de inserare
- Divizarea rădăcinii creează două noduri noi în arbore, în timp ce divizarea oricărui alt nod creează un singur nod nou

Concluzii

- Înălțimea unui arbore 2-3-4 poate crește numai în urma divizării rădăcinii
- Există o corespondență biunivocă între arborii 2-3-4 și arborii bicolori

Concluzii

- Pentru a transforma un arbore 2-3-4 într-un arbore bicolor:
  - Fiecare 2-nod va deveni un nod negru
  - Fiecare 3-nod va deveni un nod părinte negru cu un fiu roșu
  - Fiecare 4-nod va deveni un nod părinte negru cu doi fii roșii

Concluzii

- La transformarea unui 3-nod într-un nod părinte și un nod fiu, oricare dintre cele două chei se poate insera în nodul părinte
- Divizarea unui nod într-un arbore 2-3-4 este analoagă operației de inversare a culorilor dintr-un arbore bicolor

Concluzii

- O rotație într-un arbore bicolor corespunde cu schimbarea orientării între cele două orientări posibile la transformarea unui 3-nod
- Înălțimea unui arbore 2-3-4 este cel mult egală cu $\log_2 N$
- Arborii 2-3-4 utilizează ineficient memoria, deoarece majoritatea nodurilor nu sunt nici măcar pe jumătate complete