

Cursul nr. 9

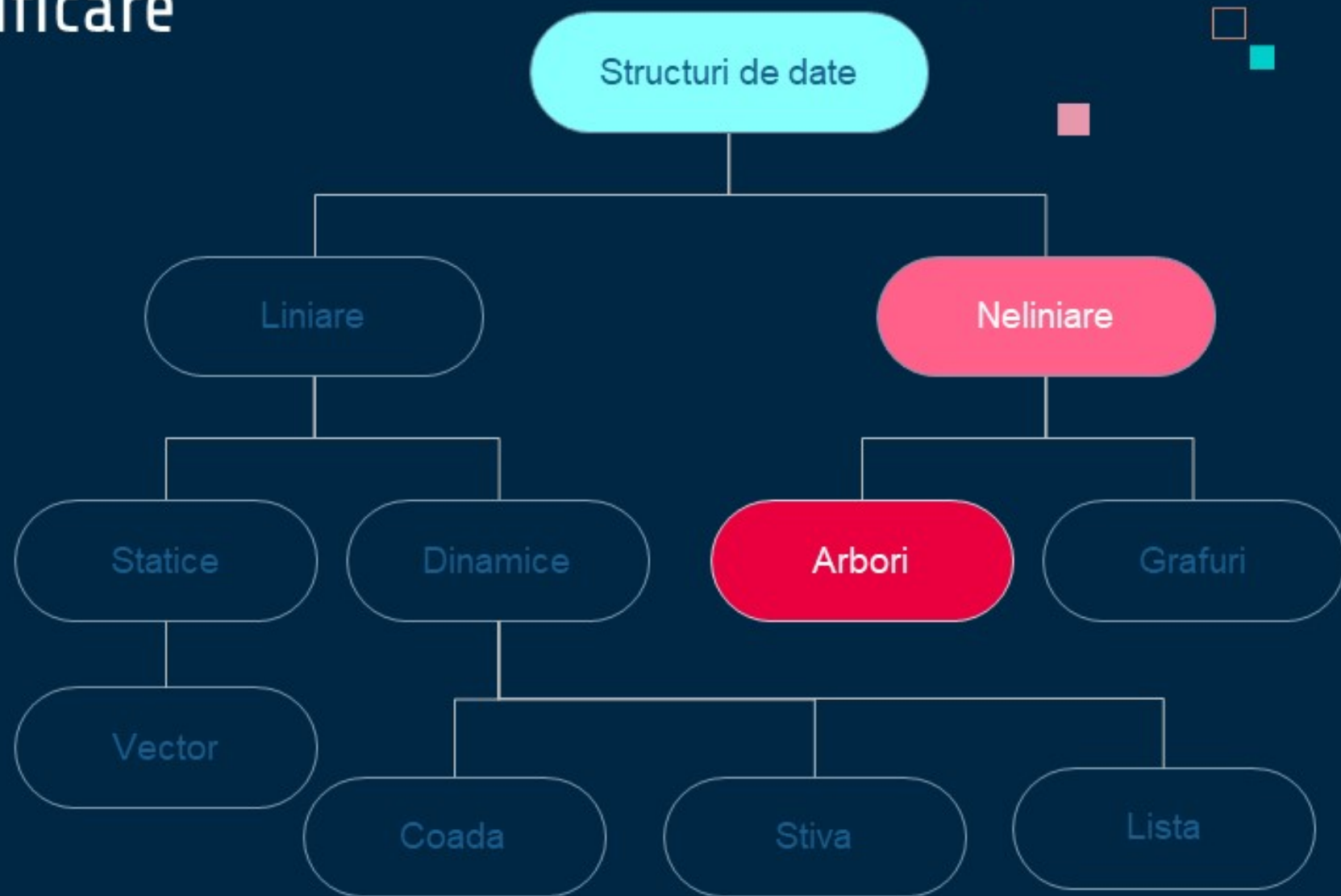
The background of the slide is a dark blue gradient. It is decorated with an abstract pattern of small squares and thin vertical lines. The squares are in various colors: light blue, orange, pink, and teal. Some squares are solid, while others are outlined. The vertical lines are thin and white, extending from the top or bottom of the frame towards the center.

STRUCTURI DE DATE neliniare

Arbore Roșu-Negru - ARN -

Lector dr. Dorin IORDACHE

Clasificare



Agenda



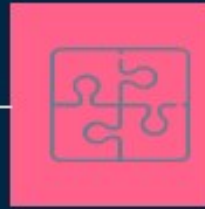
01

Proprietăți



02

Parcurgeri



03

Operații

Proprietăți

01

ARN – definire

ARN = este un ABC echilibrat parțial

în care fiecare nod are un bit suplimentar, iar acel bit este interpretat ca o culoare (roșu sau negru)

ARN îndeplinește următoarele proprietăți:

1. **Proprietate roșu/negru:** fiecare nod este colorat, fie roșu, fie negru. Excepție: nodul rădăcină este negru.
3. **Proprietatea frunzei:** Fiecare frunză (NIL/NULL) este neagră.
4. **Proprietate roșu:** Dacă un nod este roșu, atunci copiii au întodeauna sunt negru.
5. **Proprietatea adâncimii:** pentru fiecare nod, orice cale simplă de la acest nod la oricare dintre frunzele sale descendente are aceeași adâncime neagră (numărul de noduri negre).

ARN – definire

ARN = este un ABC echilibrat parțial

în care fiecare nod are un bit suplimentar, iar acel bit este interpretat ca o culoare (roșu sau negru)

ARN fiecare nod are:

culoare

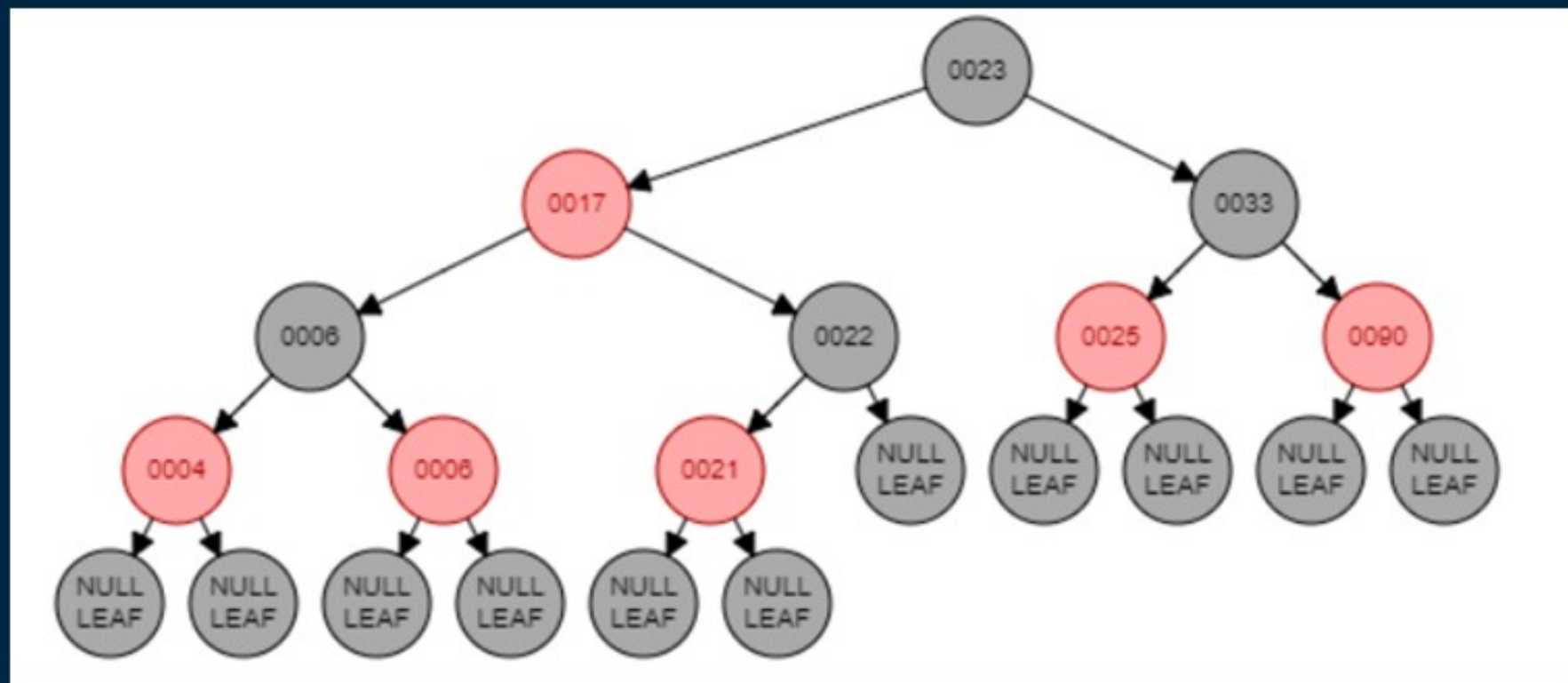
cheie

copilStanga

copilDreapta

parinte (exctie root)

ARN – exemplu



Parcurgeri

02

Parcurgerea ARN

Efectuarea oricărei operații pe un arbore, necesita accesarea unui nod specific.

Algoritmul de traversare a arborelui – **solutia**

Tipuri de parcurgeri:

- InOrdine
- PreOrdine
- PostOrdine

InOrdine (SND, LNR)

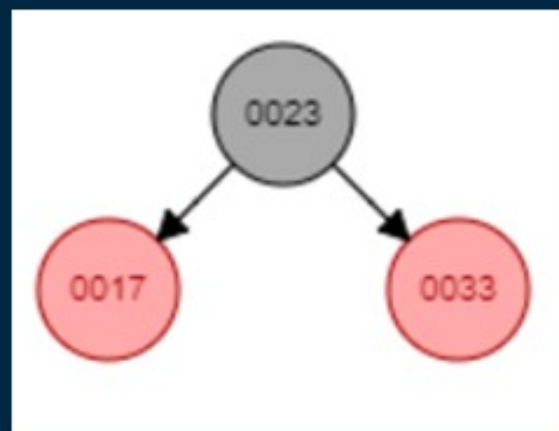
Pas 1: Se parcurg toate nodurile din subarborele din stânga

Pas 2: Apoi nodul rădăcină

Pas 3: Se parcurg toate nodurile din subarborele din dreapta

```
InOrdine(root->stanga)  
afisare(root->data)  
InOrdine(root->dreapta)
```

Nod



Stanga

Dreapta

PreOrdine (NSD, NLR)

Pas 1: Se viziteaza nodul radacina

Pas 2: Se parcurg toate nodurile din subarborele din stânga

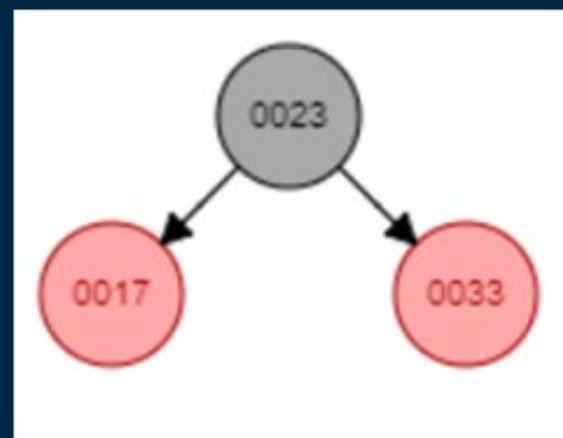
Pas 3: Se parcurg toate nodurile din subarborele din dreapta

```
afisare(root->data)
```

```
PreOrdine(root->stanga)
```

```
PreOrdine(root->dreapta)
```

Nod



Stanga

Dreapta

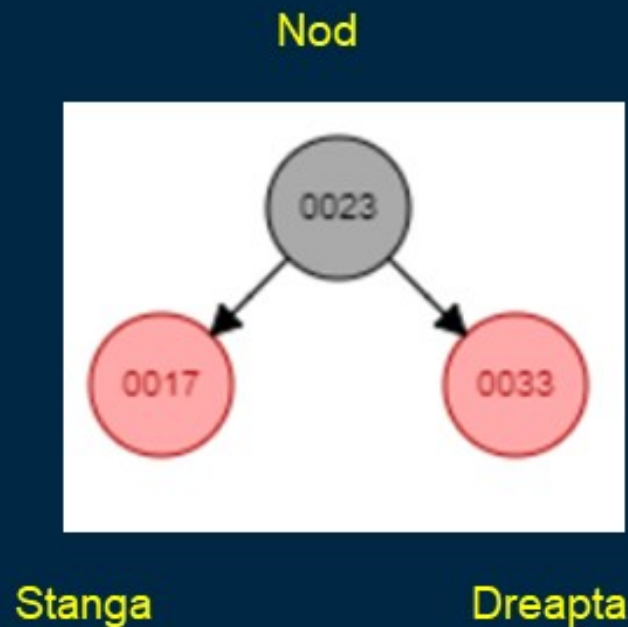
PostOrdine (SDN, LRN)

Pas 1: Se parcurg toate nodurile din subarborele din stânga

Pas 2: Se parcurg toate nodurile din subarborele din dreapta

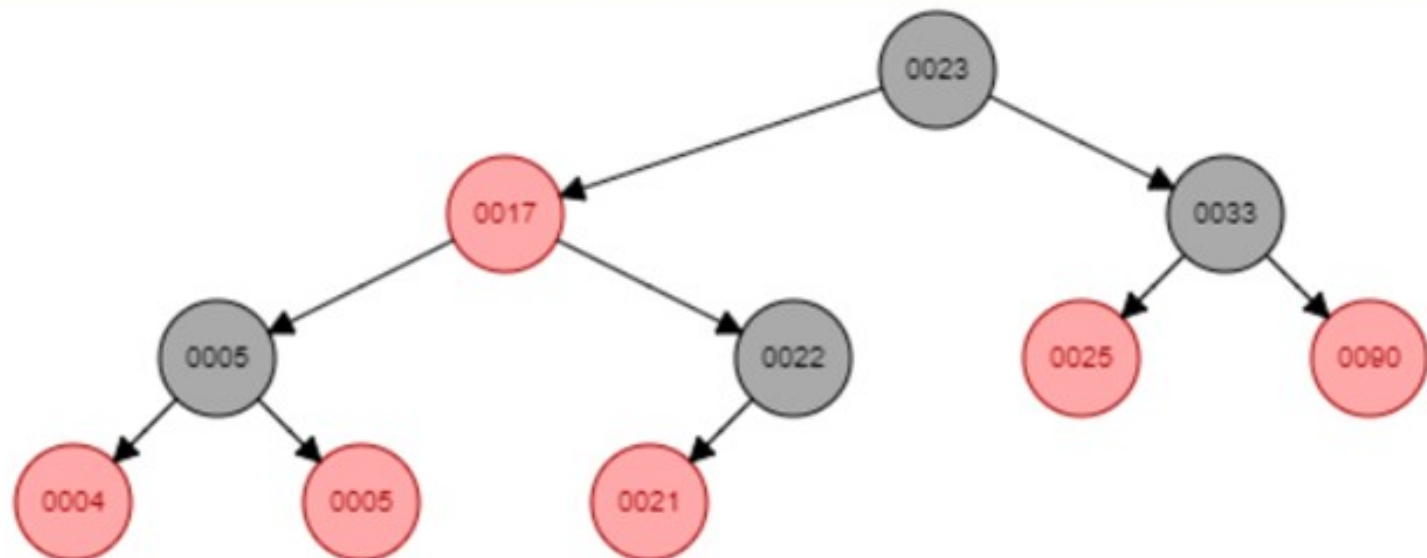
Pas 3: Se viziteaza nodul radacina

```
PostOrdine(root->stanga)  
PostOrdine(root->dreapta)  
afisare(root->data)
```



Parcurgeri - ?

PreOrdine (NSD)



Operații – ARN –

03

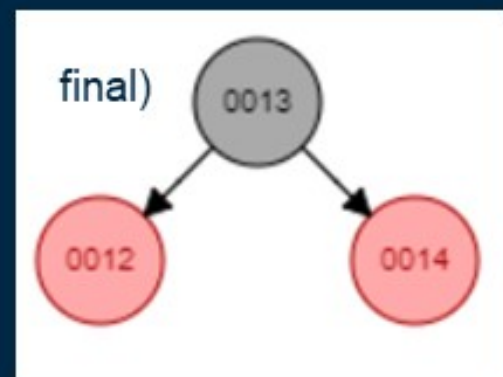
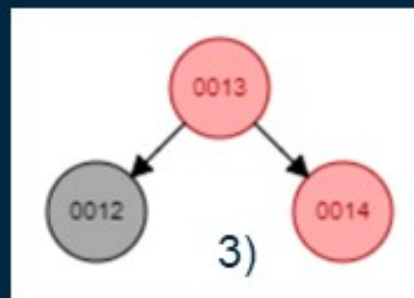
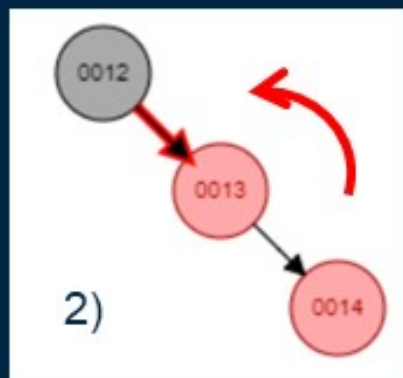
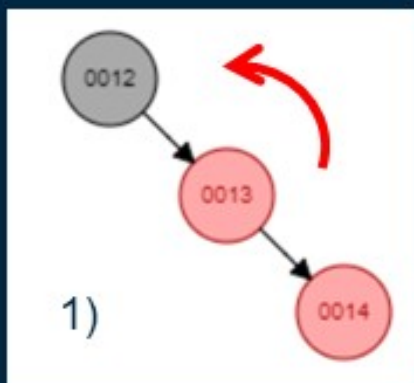
Echilibrare ARN

rotație

- Stânga
- Dreapta
- Stânga-Dreapta
- Dreapta-Stânga

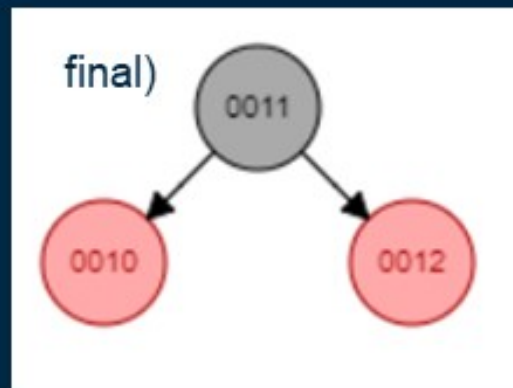
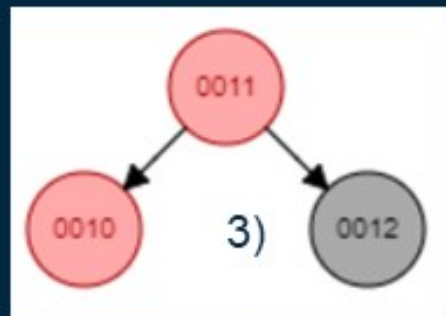
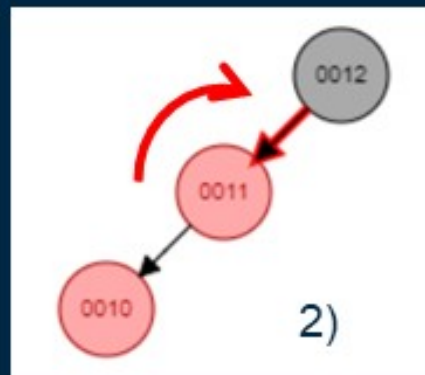
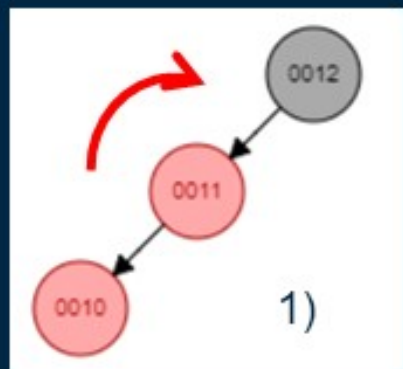
Echilibrare ARN

- rotație
- Stânga



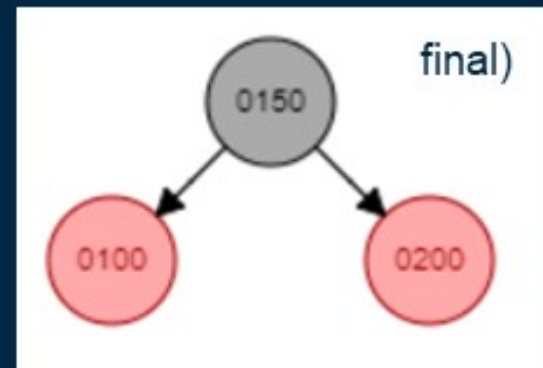
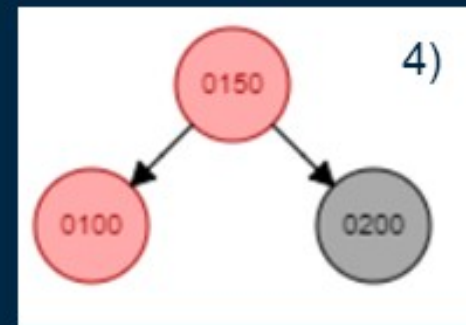
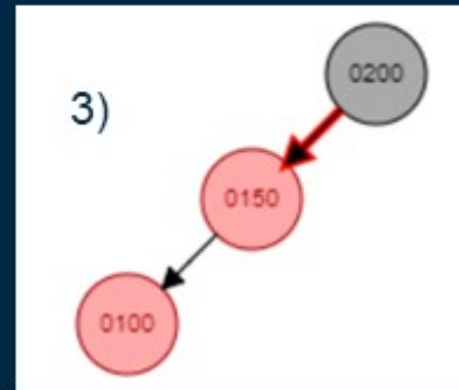
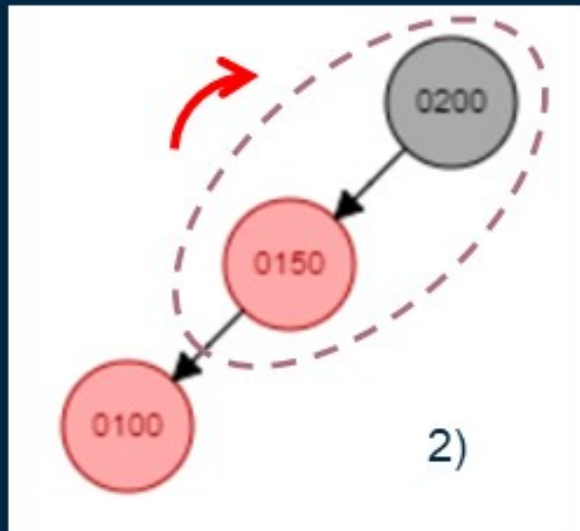
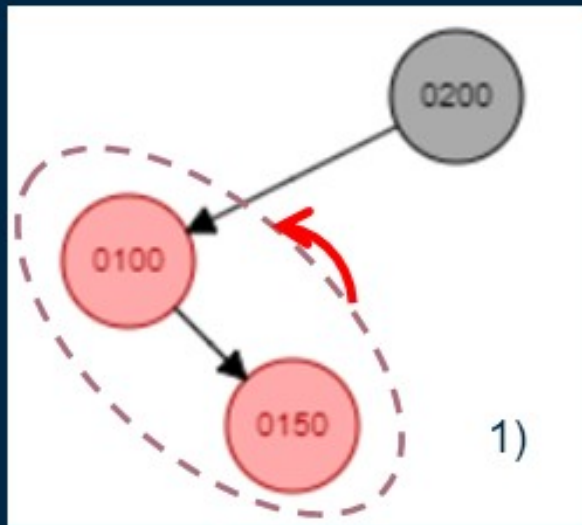
Echilibrare ARN

- rotație
- Dreapta



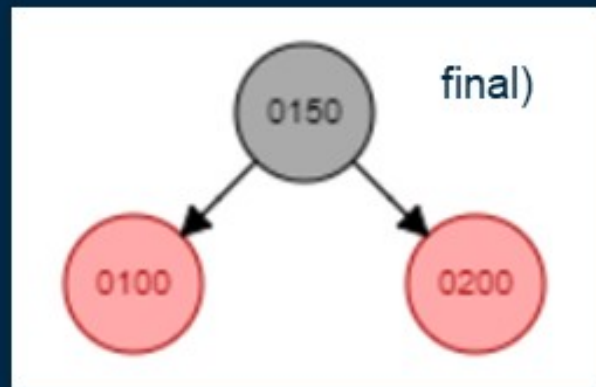
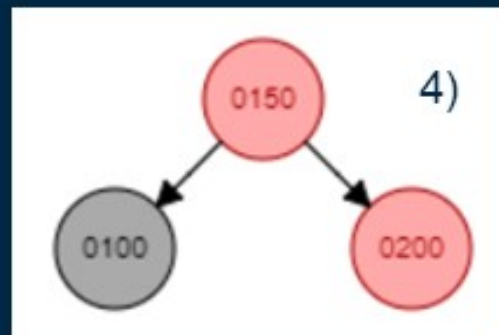
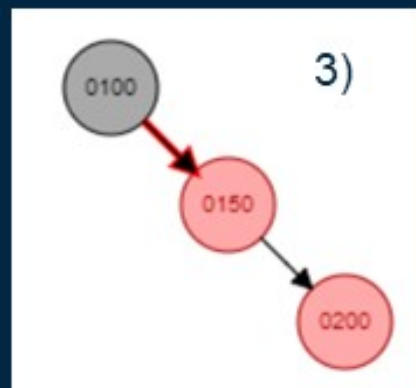
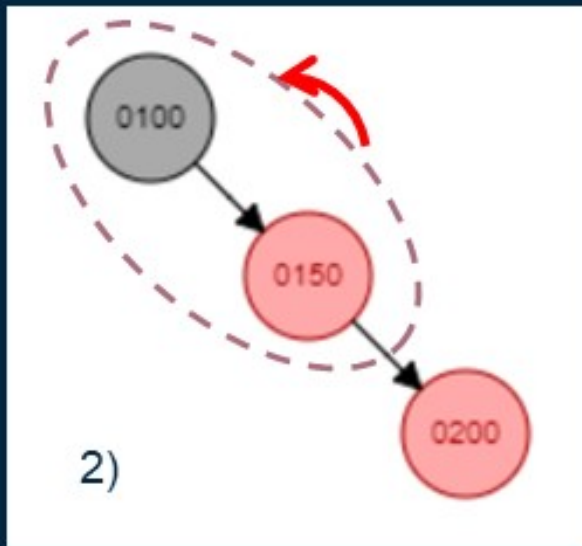
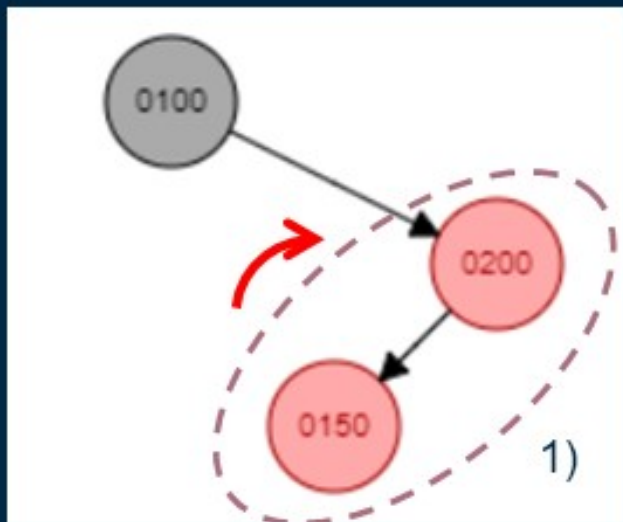
Echilibrare ARN

- Rotație
 - Stânga - Dreapta



Echilibrare ARN

- rotație
 - Dreapta - Stânga



Operații Arbori Rosu Negru

- Căutare
- Adăugare
- Ștergere

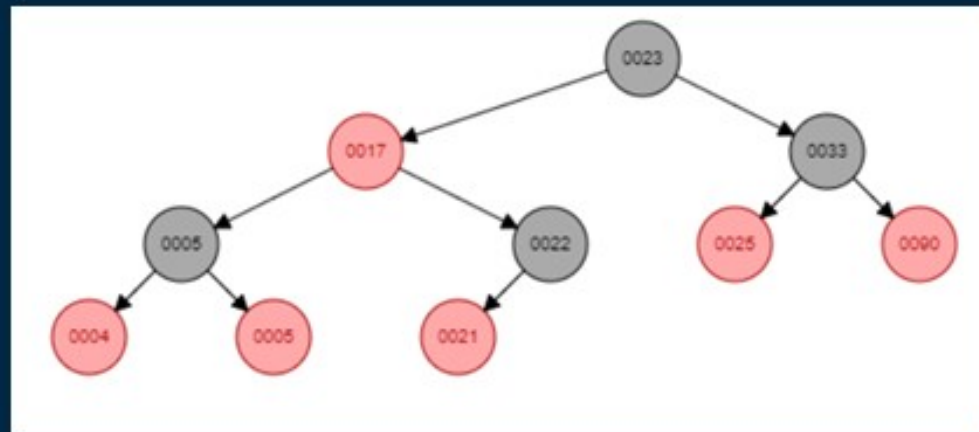
Operații Arbori Roșu Negru

- Căutare

- (1) Pornim de la radacina
- (2) Comparăm valoarea de căutat cu cheia nodului.
Dacă valoarea de căutat este mai mică decât cheia nodului, vom continua căutarea în subarborele stâng, În caz contrar, în subarborele drept
- (3) Se continuă până când găsim valoarea de căutat sau nu mai există subarbori

Care este diferența între ABC și ARN ?

Arbore Roșu Negru



Operații Arbori Roșu Negru

- Căutare

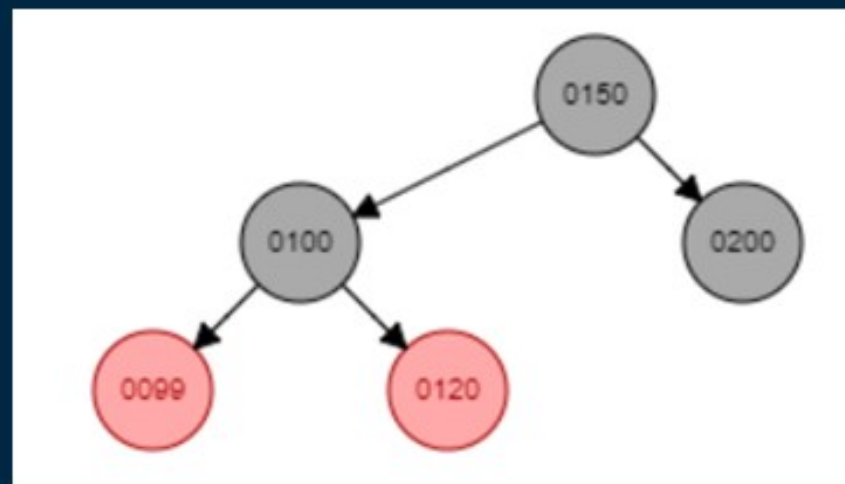
```
begin procedure caut(nod, valoare)
    if nod == NULL then
        return NULL
    endif
    if valoare == nod -> data then
        return nod -> data
    endif
    if valoare < nod -> data then
        return caut(nod -> stânga, valoare)
    else
        return caut(nod -> dreapta, valoare)
    endif
end procedure caut
```

Operații Arbori Roșu Negru

- Adăugare

- (1) Pornim de la radacina
- (2) Comparăm valoarea de adăugat cu cheia nodului.
Dacă valoarea este mai mică decât cheia nodului, vom continua procedeul în subarborele stâng,
În caz contrar, în subarborele drept
- (3) Se continuă până când găsim un nod extern (frunză) și vom adăuga, în subramura stânga, sau dreapta, după caz. Nodul nou introdus are culoarea **Roșu**
- (4) Se verifică îndeplinirea proprietăților. În caz că este încălcată o proprietate se execută operațiile:
 - recolorare
 - rotații

Arbore Roșu Negru



<https://www.programiz.com/dsa/insertion-in-a-red-black-tree>

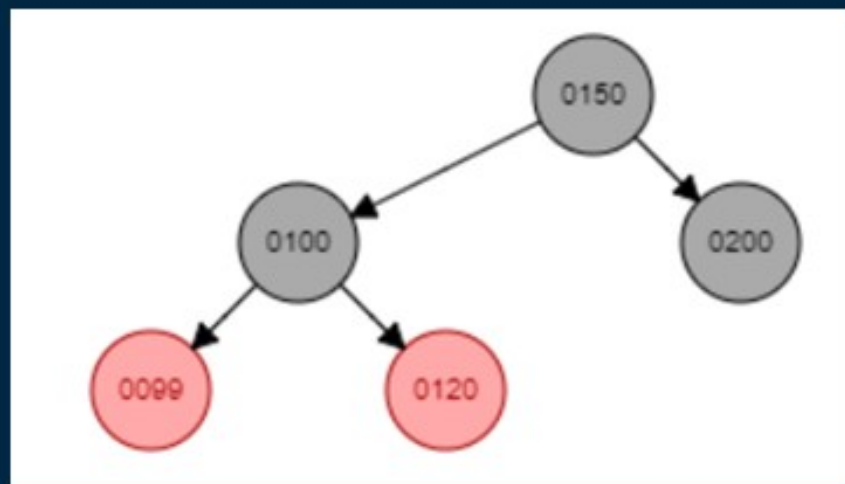
Operații Arbori Roșu Negru

- Adăugare

Exercițiu:

1. adăugăm valoarea 121
2. adăugăm valorile 210, 220

Arbore Roșu Negru



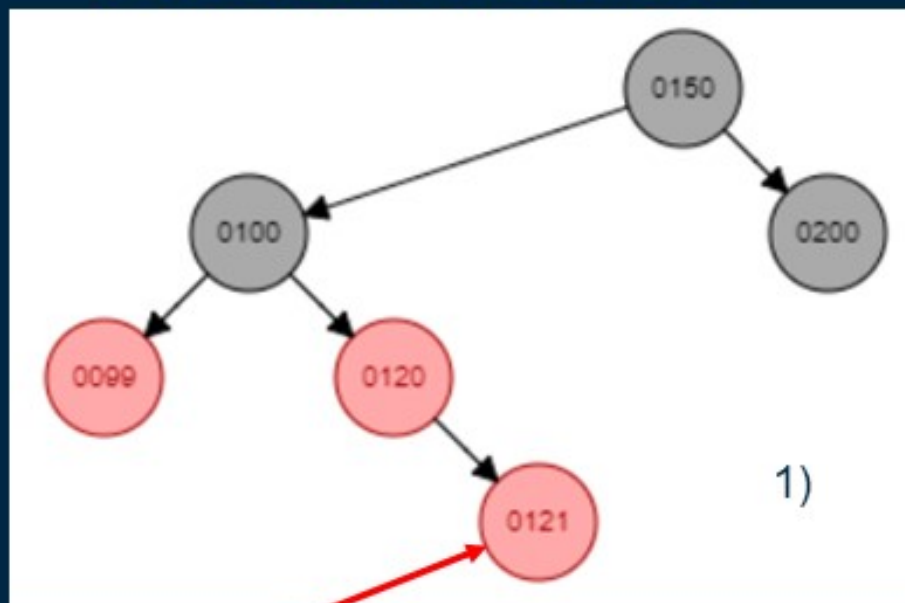
Operații Arbori Roșu Negru

- Adăugare

1. recolorare
2. rotații

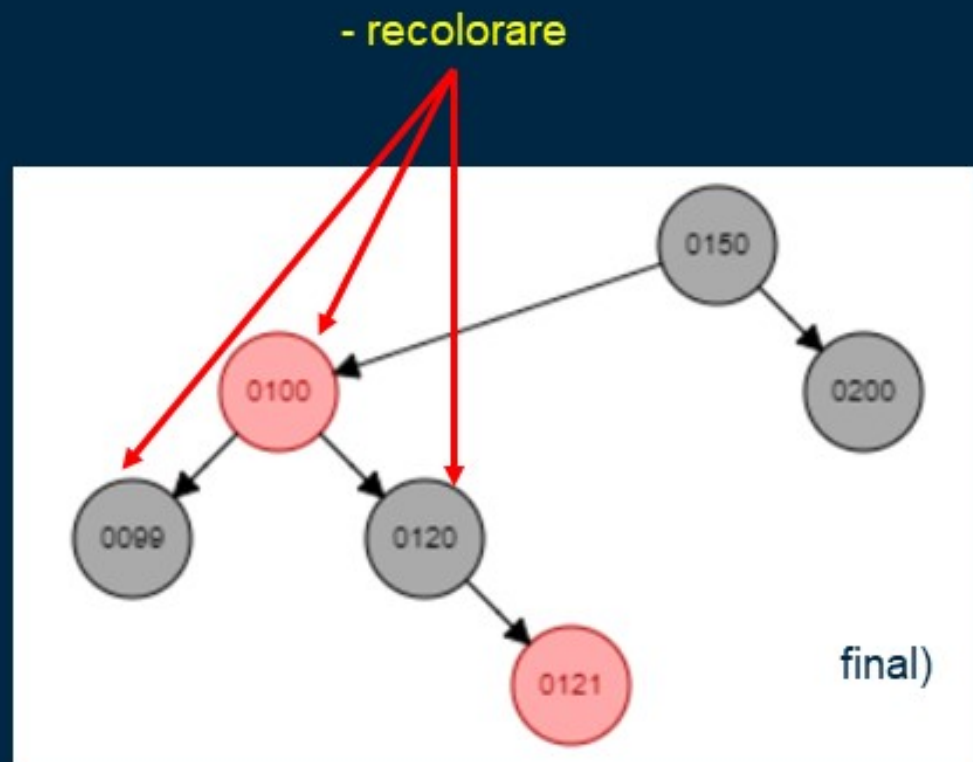
Exercițiu:

1. adăugăm valoarea 121



1)

Nodul nou adăugat
cheia 12



final)

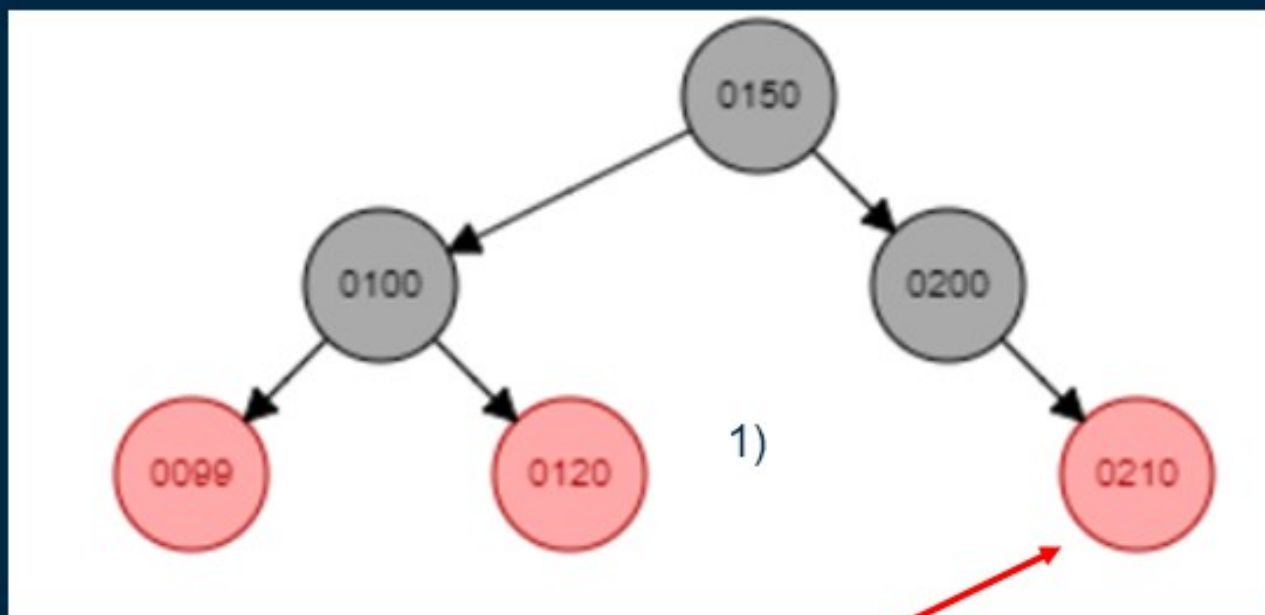
Operații Arbori Roșu Negru

- Adăugare

1. recolorare
2. rotații

Exercițiu:

2. adăugăm valoarea 210



Nodul nou adăugat
cheia 210

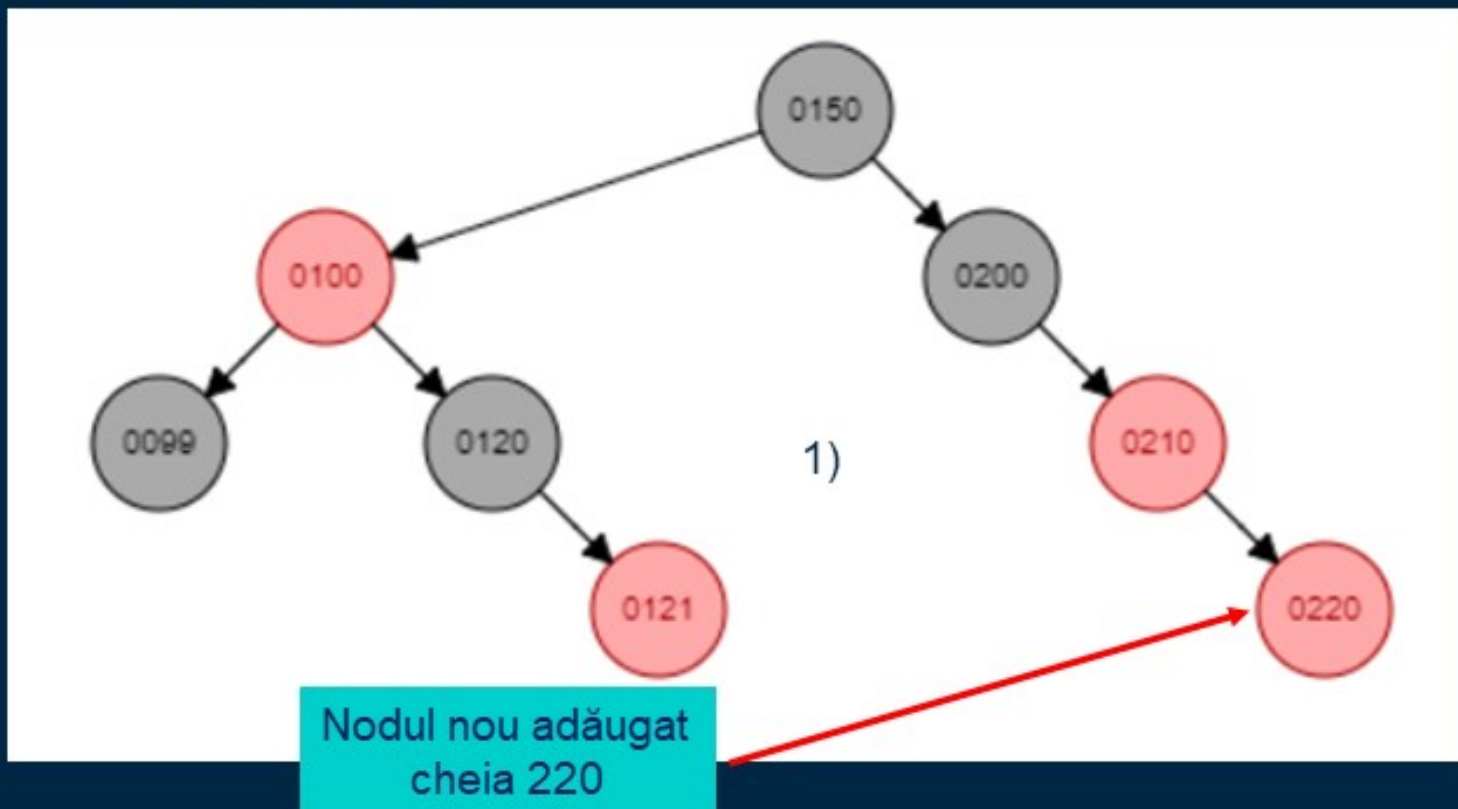
Operații Arbori Roșu Negru

- Adăugare

1. recolorare
2. rotații

Exercițiu:

2. adăugăm valoarea 220



Operații Arbori Roșu Negru

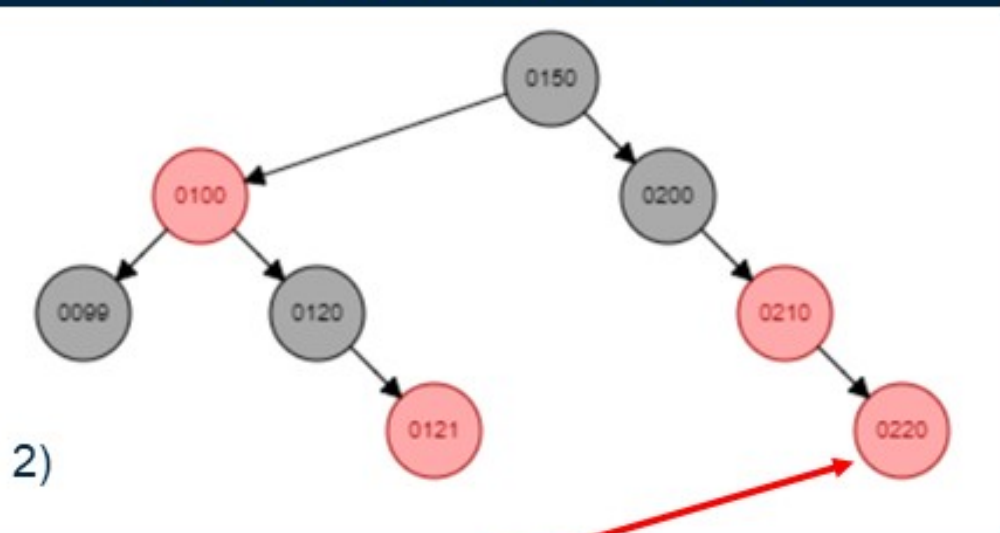
- Adăugare

1. recolorare
2. rotații

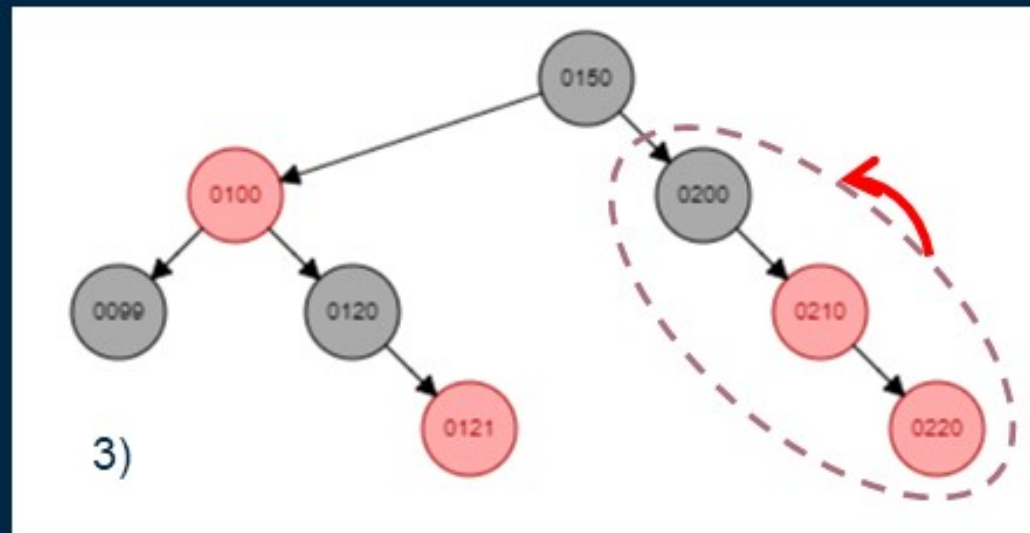
Exercițiu:

2. adăugăm valoarea 220

- rotație stanga



Nodul nou adăugat
cheia 220



Operații Arbori Roșu Negru

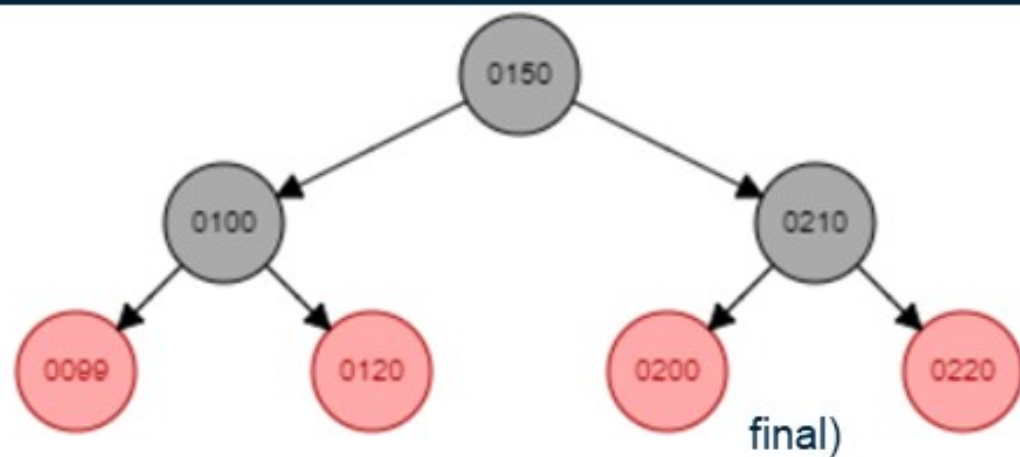
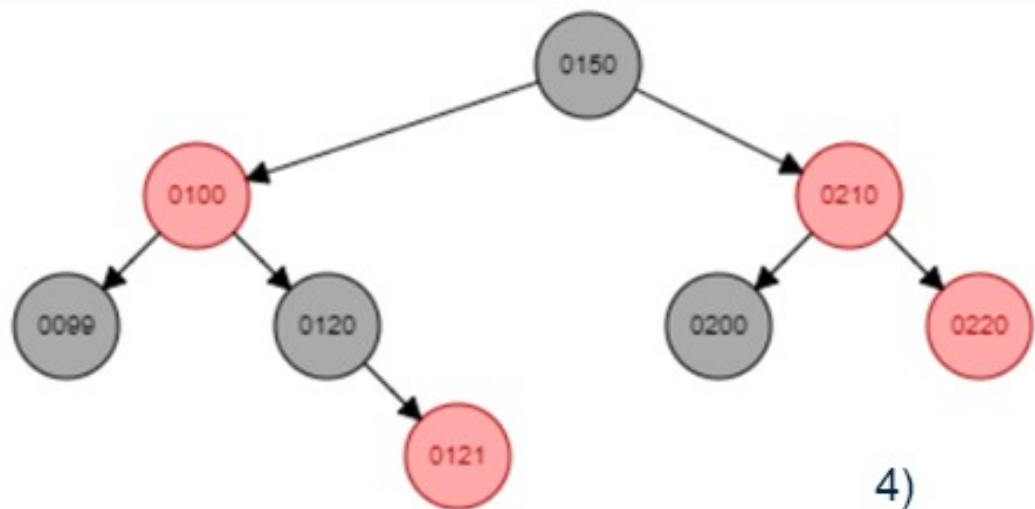
- Adăugare

1. recolorare
2. rotații

Exercițiu:

2. adăugăm valoarea 220

- recolorare

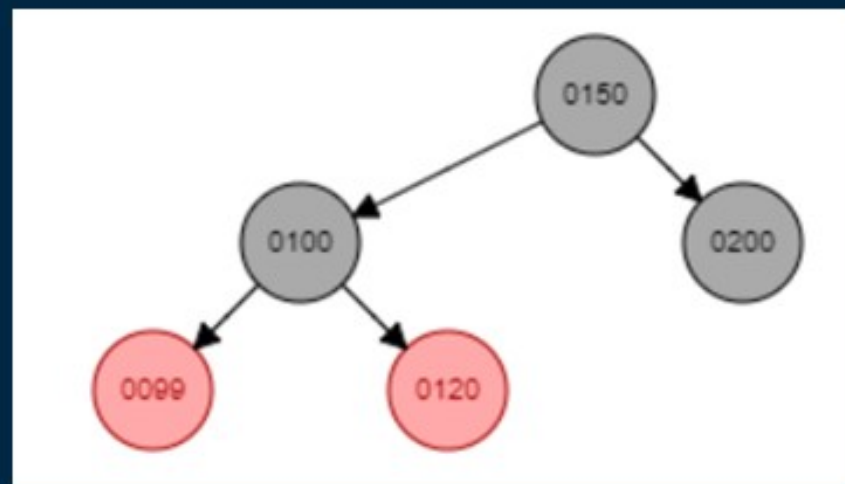


Operații Arbori Roșu Negru

- Ștergere

- (1) Pornim de la radacina
- (2) Comparăm valoarea de șters cu cheia nodului.
Dacă valoarea este mai mică decât cheia nodului, vom continua procedeul în subarborele stâng,
În caz contrar, în subarborele drept
- (3) Se continuă până când găsim nodul, în subramura stânga sau dreapta, care corespunde cheii.
- (4) Se verifică îndeplinirea proprietăților. În caz că este încălcată o proprietate se execută operațiile:
 - recolorare
 - rotații

Arbore Roșu Negru



Arbori Roșu Negru – complexitate (timp)

Operația	Caz favorabil	Caz mediu	Caz nefavorabil
Căutare	$O(\log n)$	$O(\log n)$	$O(n)$
Adăugare	$O(\log n)$	$O(\log n)$	$O(n)$
Ștergere	$O(\log n)$	$O(\log n)$	$O(n)$

Aplicații ale Arborilor Roșu Negru

- Orice operație de Inserare, Ștergere și Căutare într-un timp eficient
- Aplicații cu hărți vectoriale

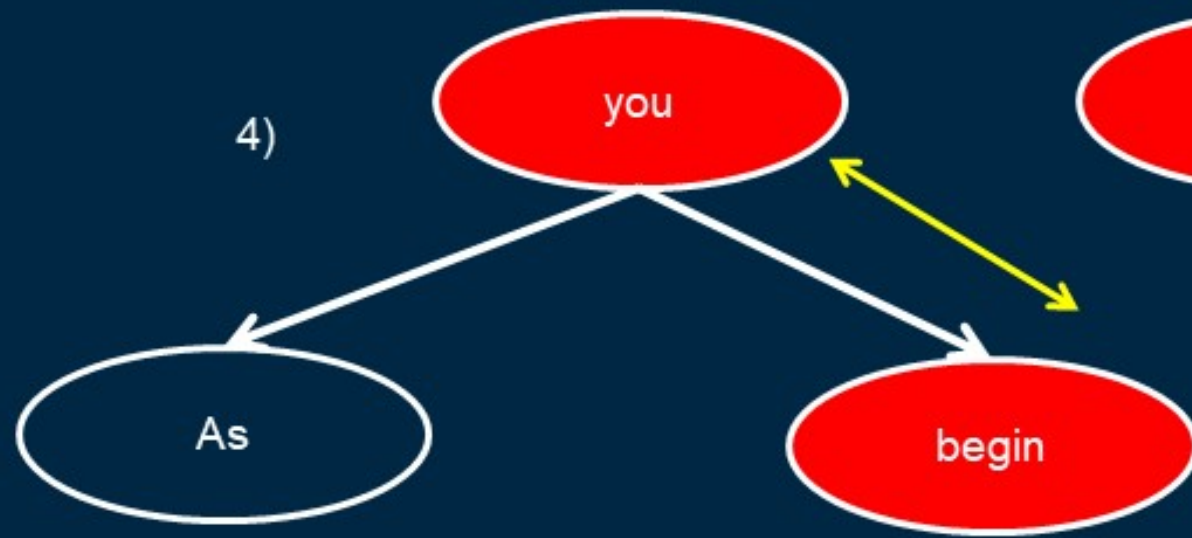
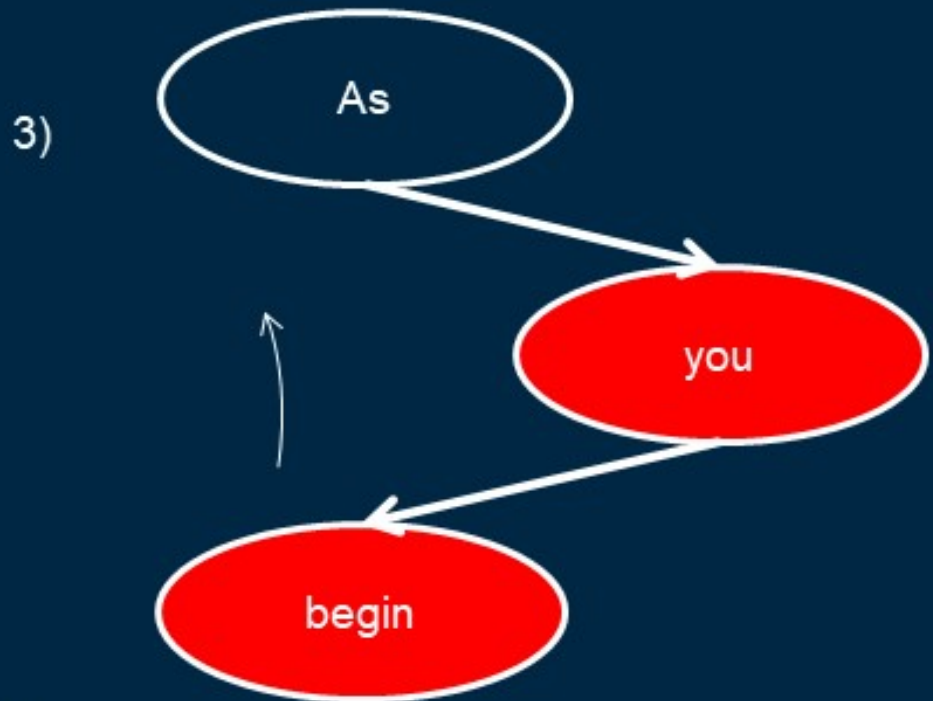
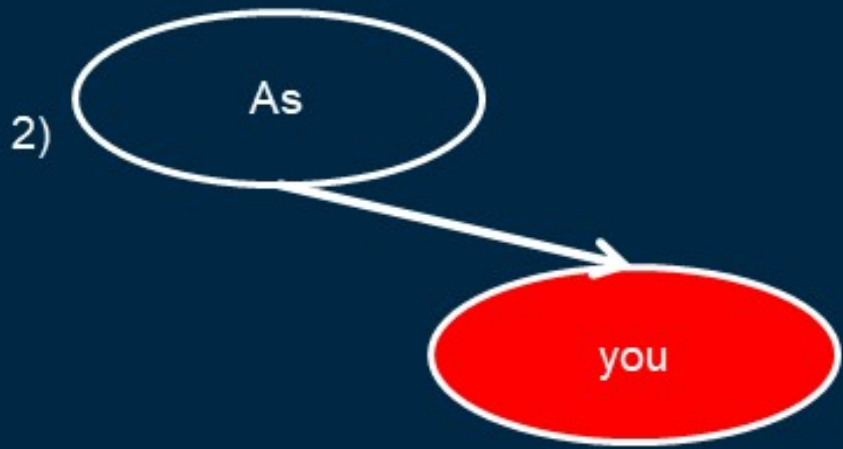
Dezavantaj:

Complicat de implementat

Exemplu generare ARN

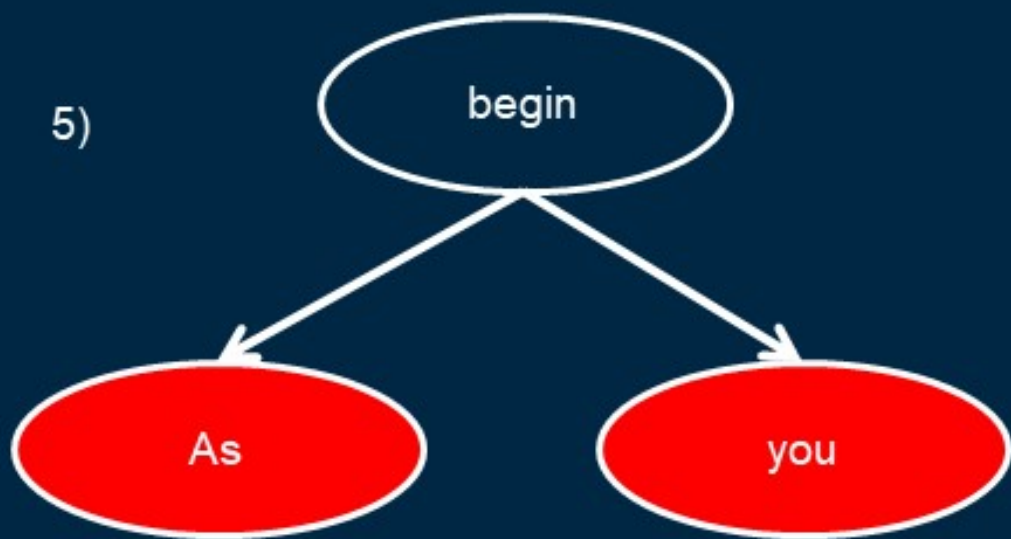
- Generare ARN pentru urmatoarele chei (cuvintele propozitiei):
As you begin to design a document



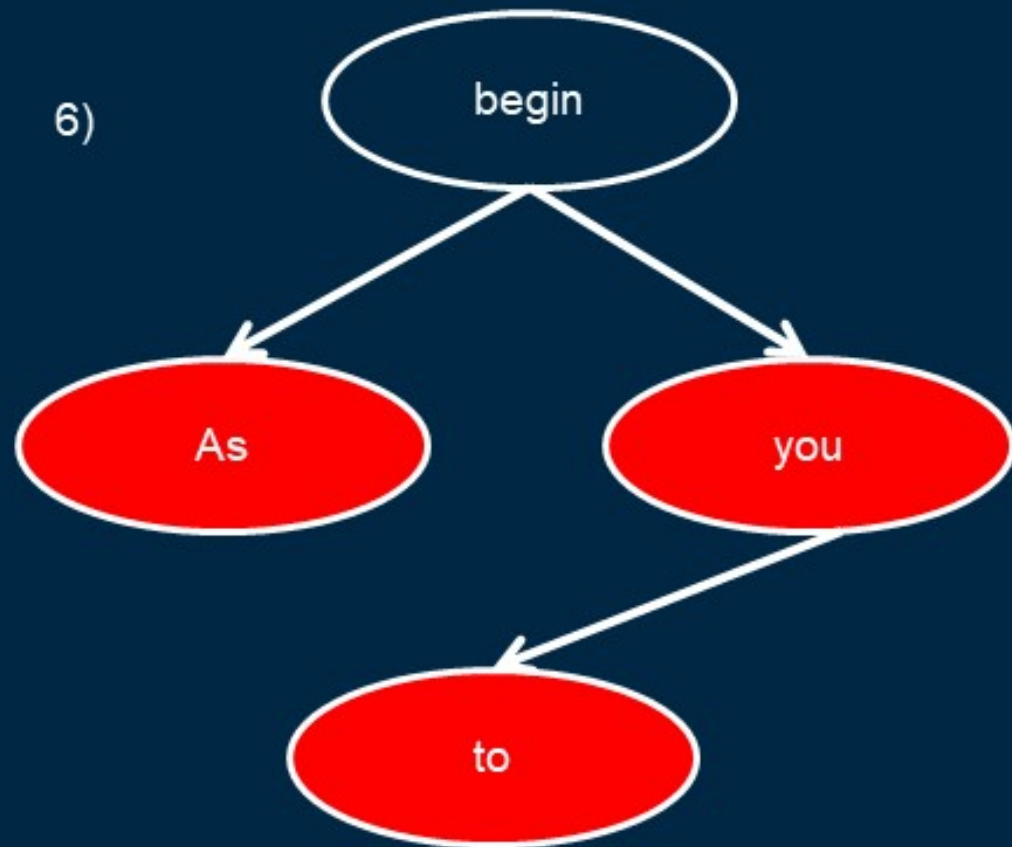


Rotatie cu interschimbare

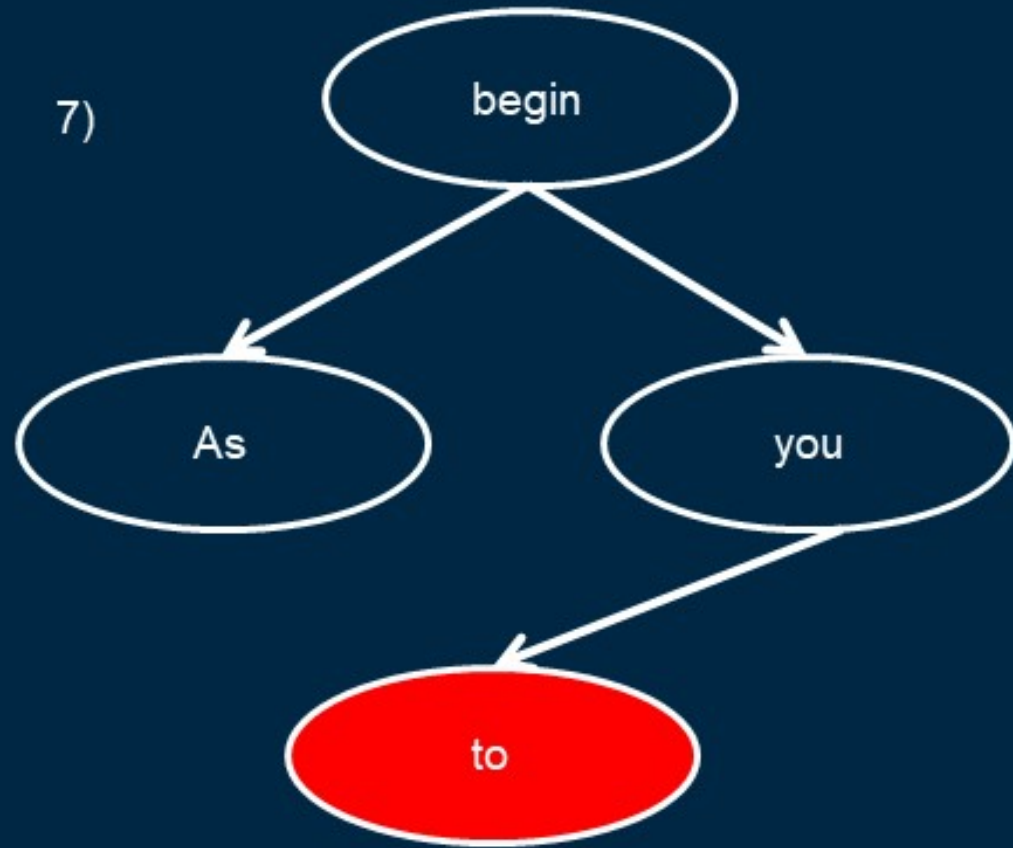
5)



6)

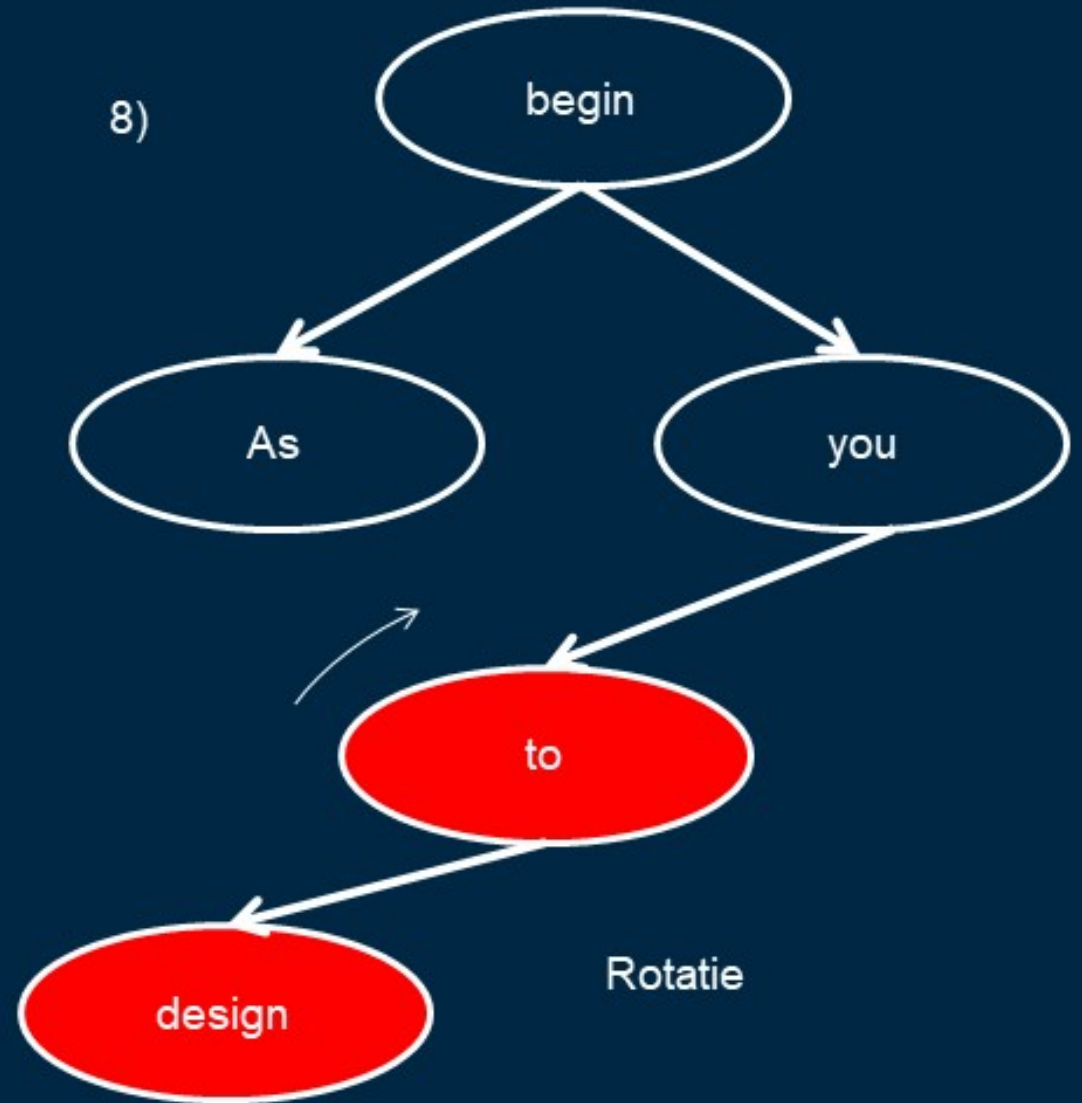


7)



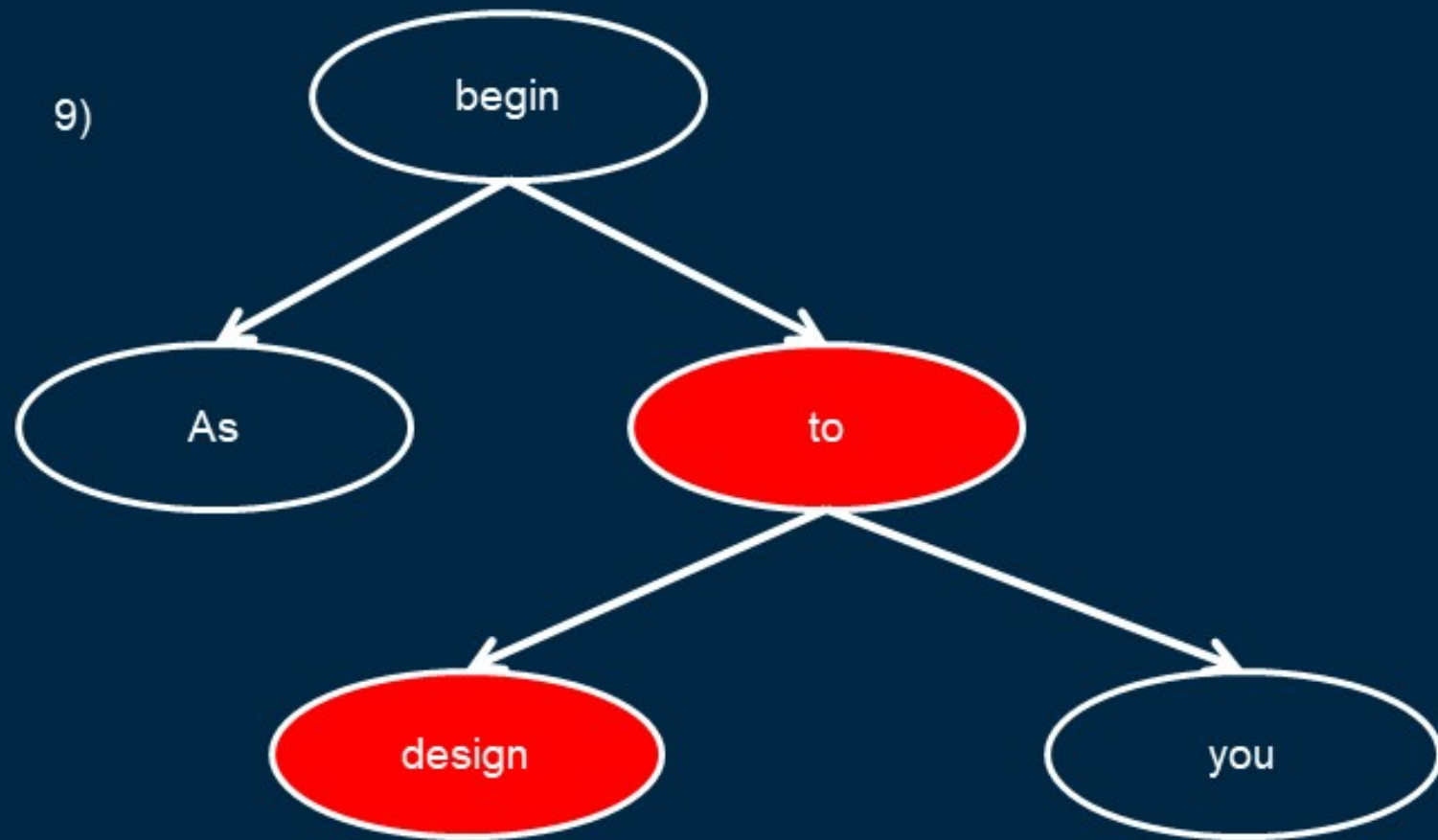
Schimbare culori

8)



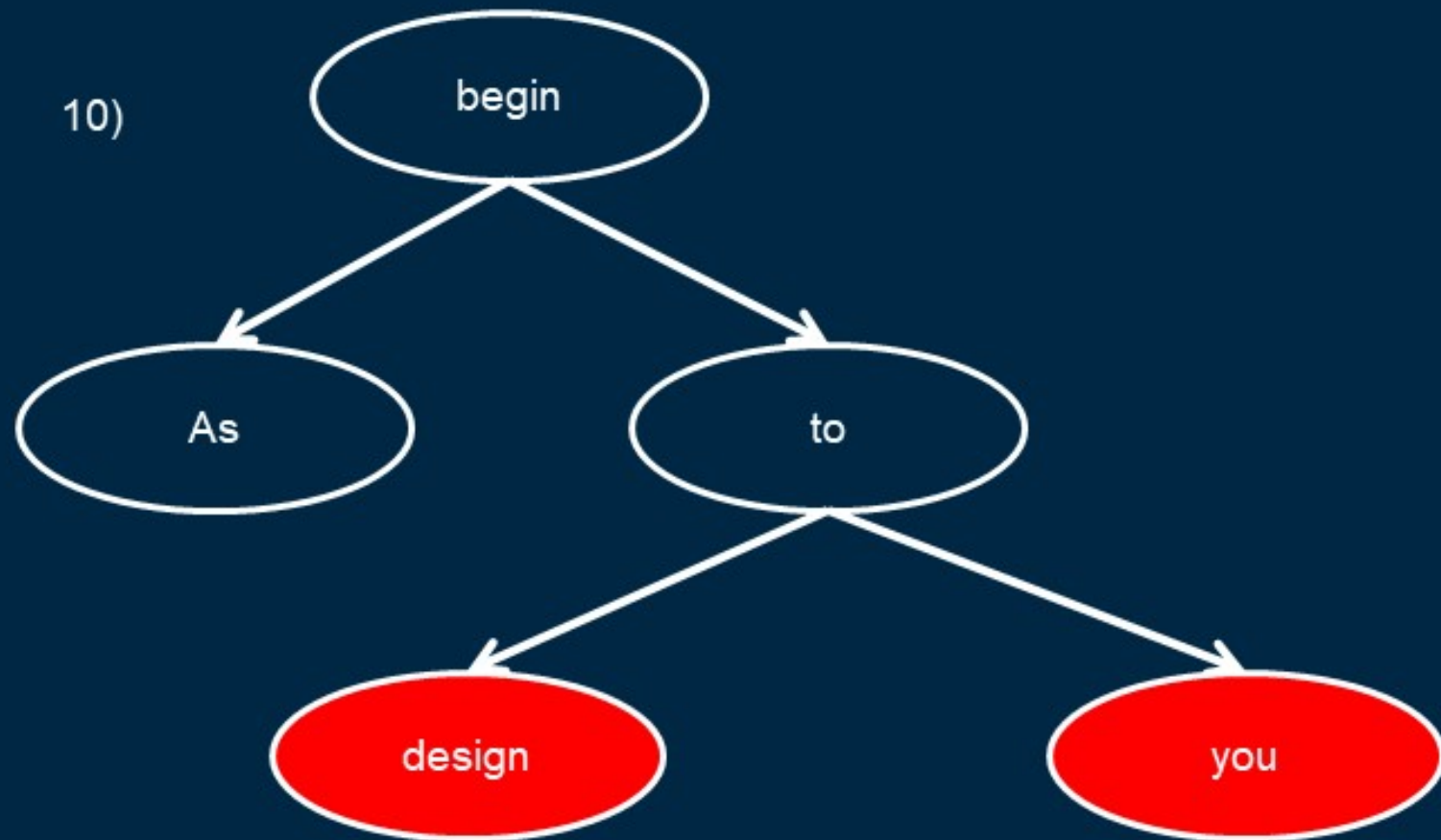
Rotatie

9)

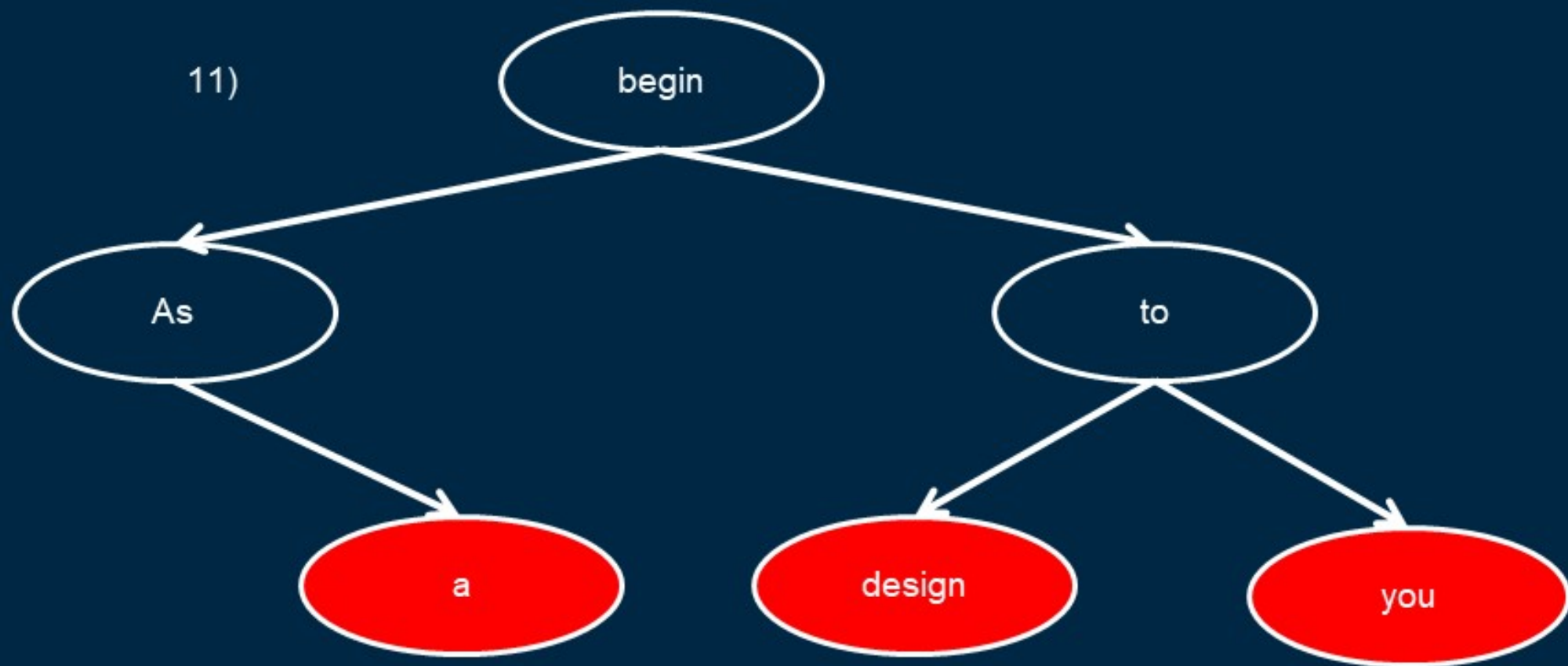


Schimbare culori

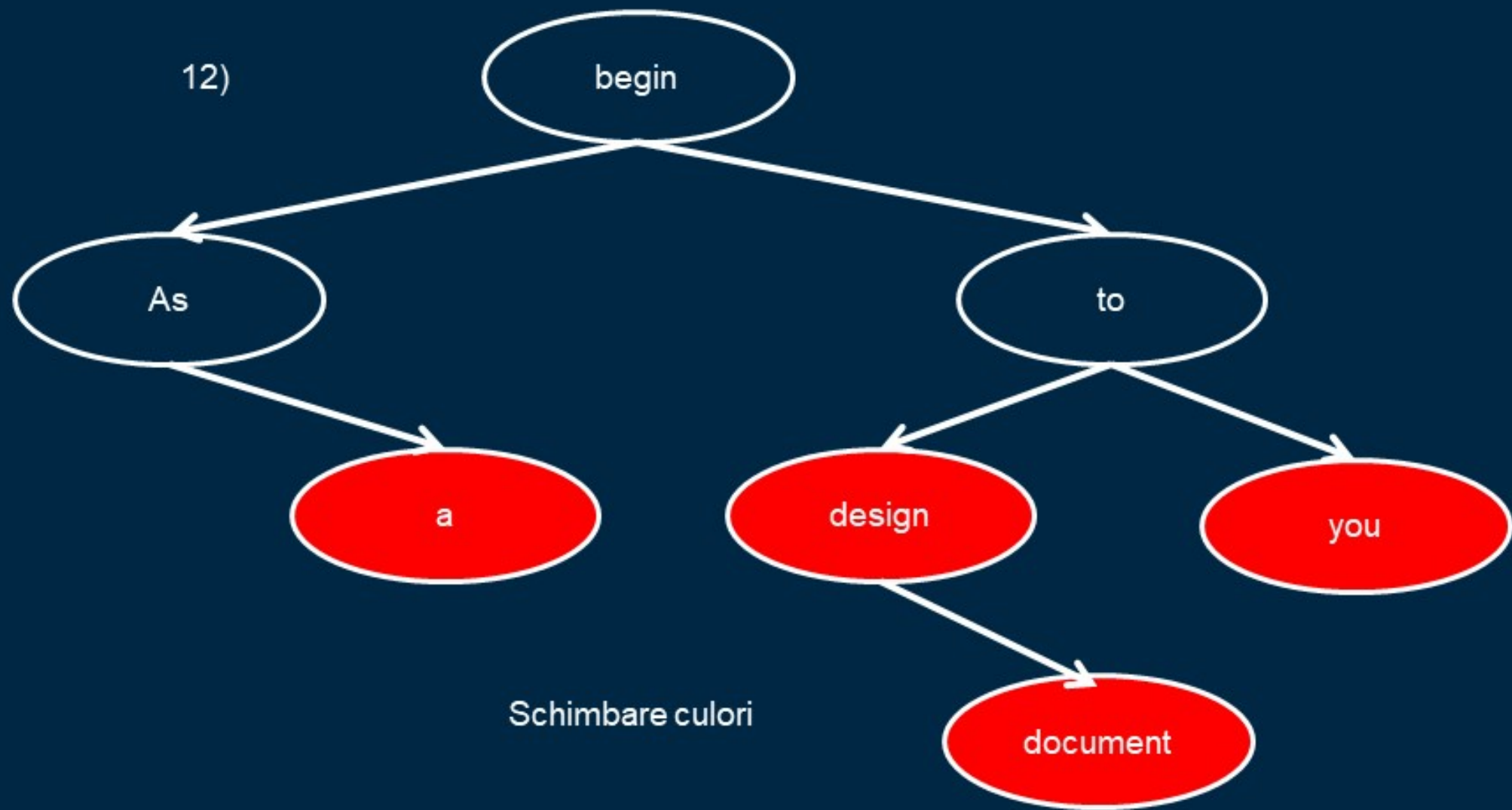
10)



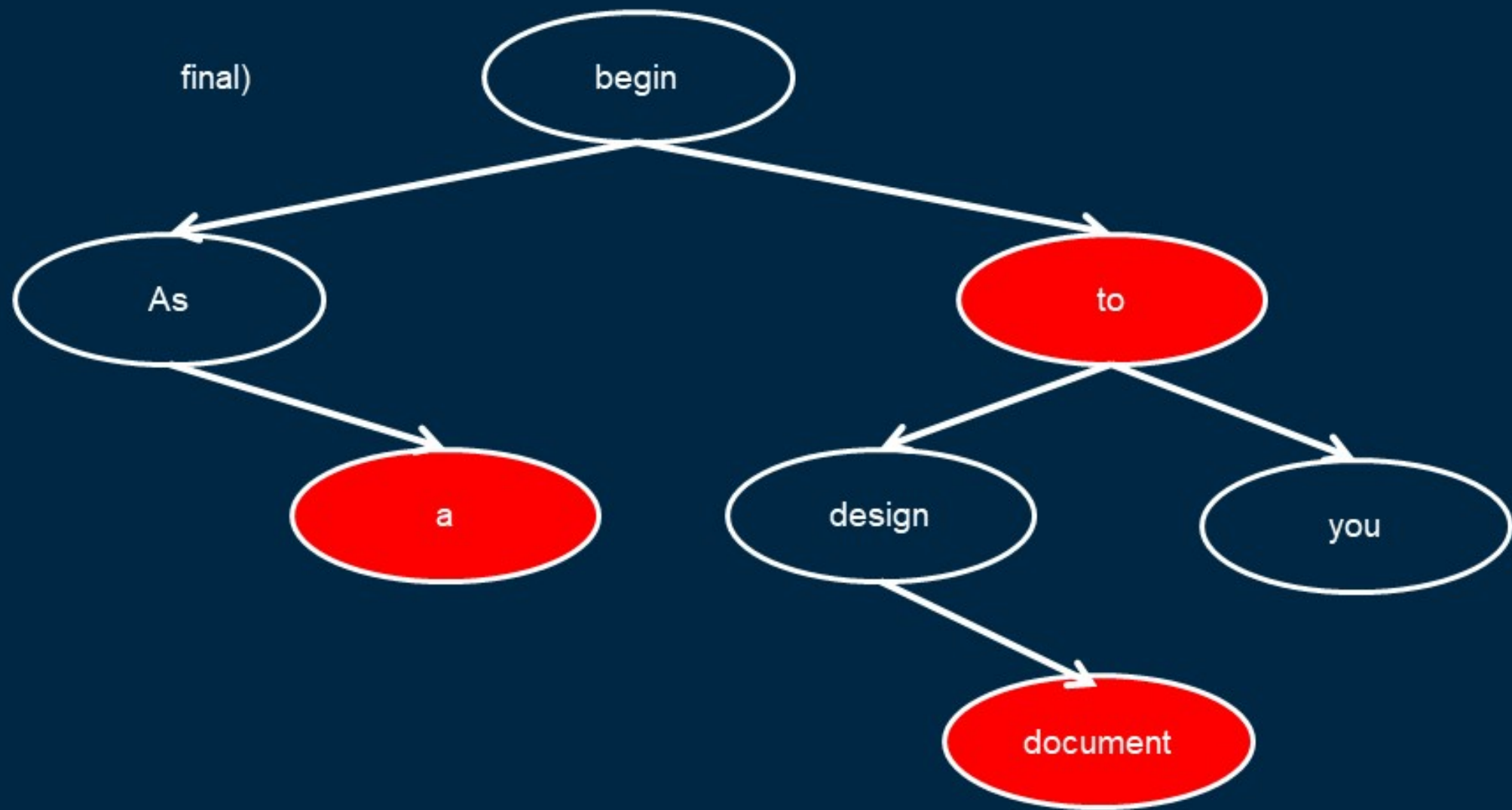
11)



12)



Schimbare culori



Intrebari?

dorin.lordache@365.univ-ovidius.ro

Multumesc

CREDITS: This presentation template was created by [Slidesgo](#),
including icons by [Flaticon](#), and infographics & images by [Freepik](#)