

Curs 6

ANALIZA FLUXURILOR DE DATE SI ACTIVITATI - I

1. Tehnici de analiza

- **Aplicate definitiei procesului pentru ca:**
 1. Procesul sa fie lipsit de erori.
 2. Timpul de executie al procesului sa fie minim.
 3. Numarul resurselor folosite sa fie minim , iar utilizarea acestora sa fie caracterizata de eficienta.
- **Distingem intre:**
 1. Analiza calitativa – corectitudinea din punct de vedere logic a procesului definit (absenta deadlock-urilor si a livelock-urilor)
 2. Analiza cantitativa – performantele procesului (indicatori: timpul de executie, gradul de utilizare a resurselor)



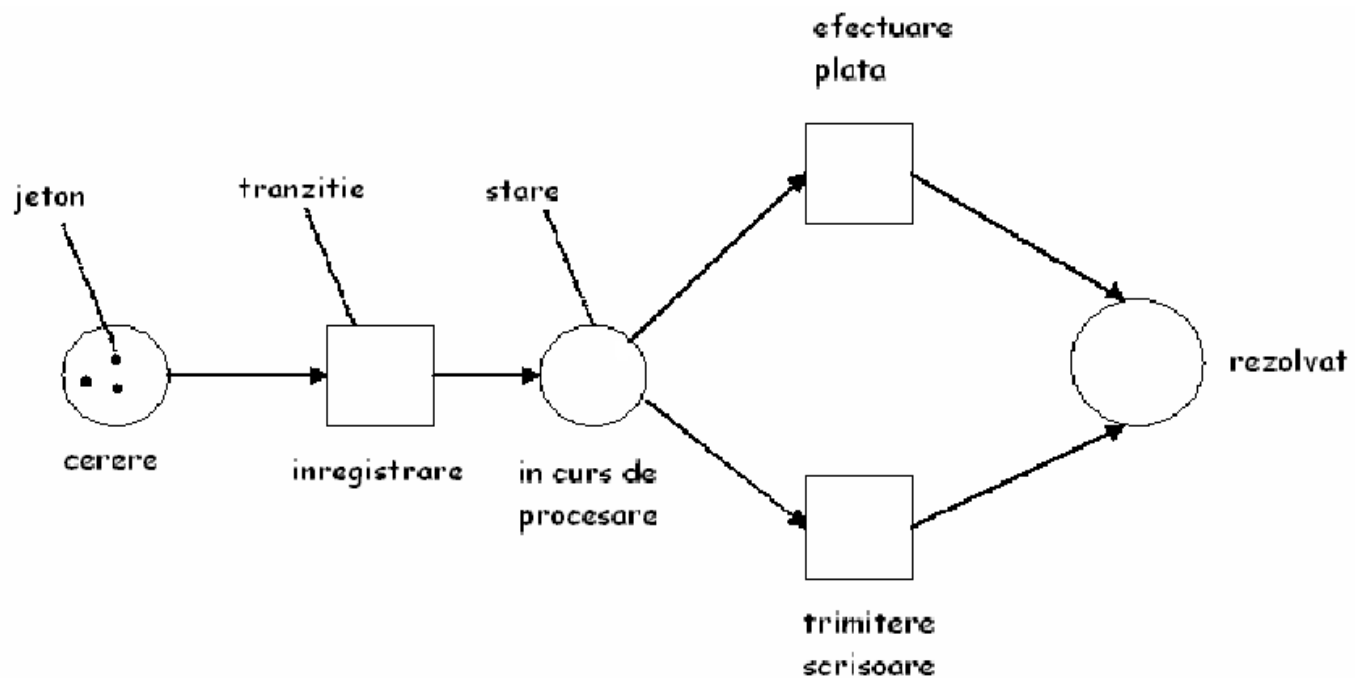
2. Analiza accesibilitatii

- Vom folosi o retea Petri pentru a descrie posibila evolutie a procesului modelat.
- O Retea Petri impreuna cu starea sa initiala determina:
 1. Starile care vor fi accesibile.
 2. Ordinea in care vor fi atinse acestea.
- Construirea grafului de marcaje

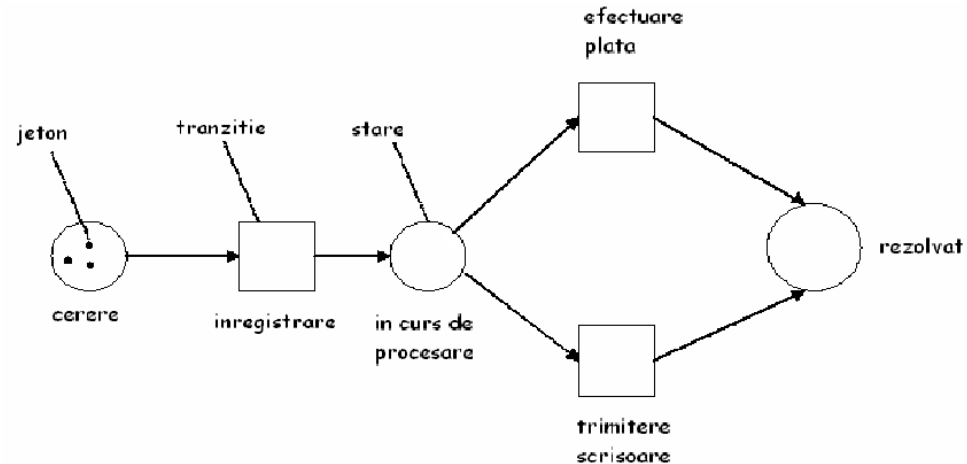


Exemplu (1)

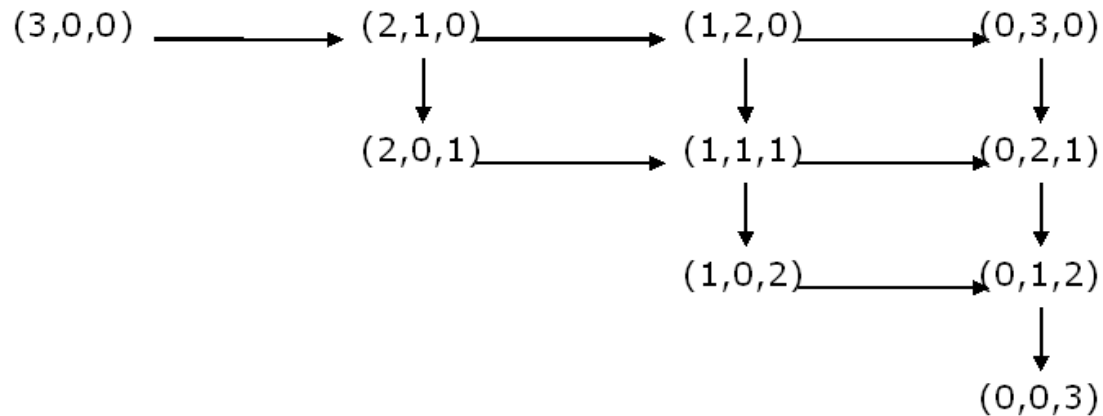
Fie următoarea RP:



Exemplu (2)



Graful de marcaje



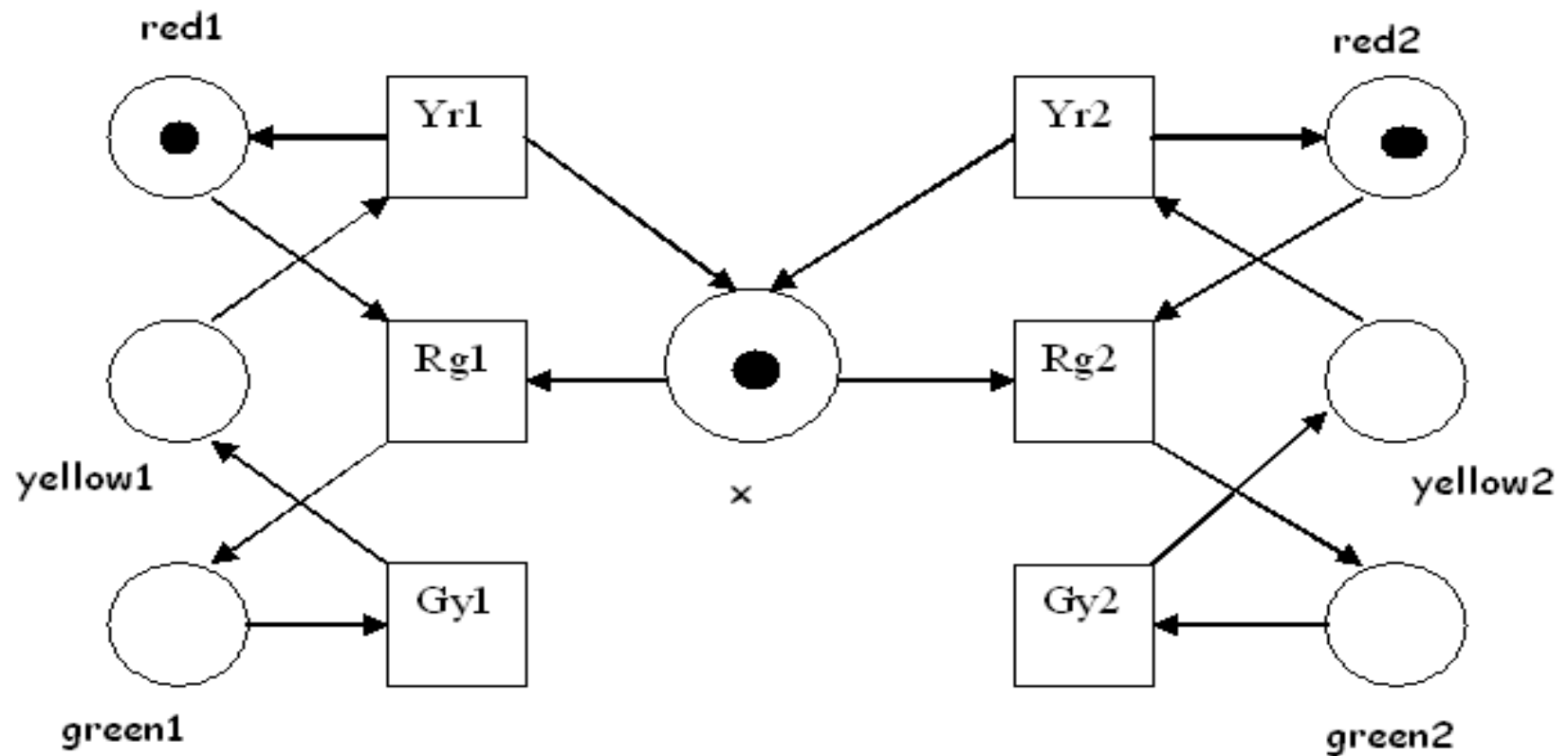
OBSERVATII:

1. Fiecare nod reprezinta o stare accesibila.
2. Fiecare tranzitie reprezinta o posibila schimbare de stare.
3. Chiar daca o stare este accesibila ea nu este neaparat atinsa.
4. Daca dintr-un nod pornesc mai multe sageti, indicand multiple posibilitati de evolutie, atunci, in mod evident, starea urmatoare nu este predeterminata = **alegere nedeterminista**

Exemplu:

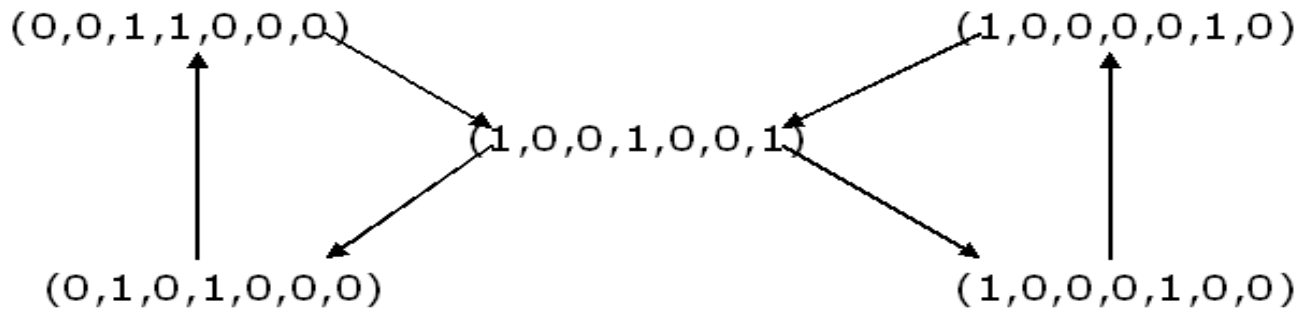
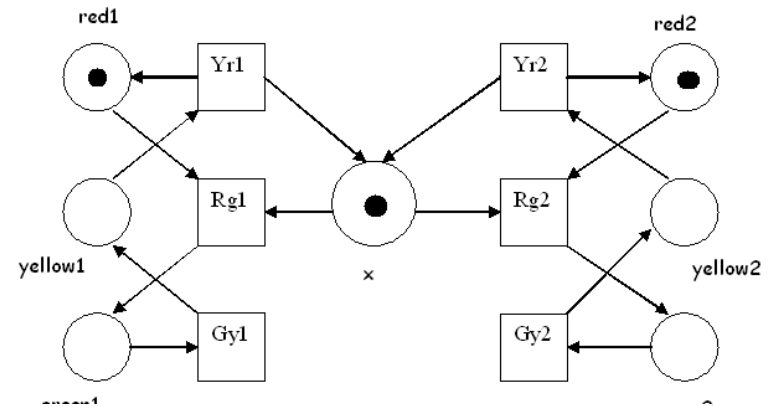
- Modelarea functionarii semafoarelor aflate la intersectia a doua strazi cu sens unic
- Observatii:
 1. Semafoarele functioneaza de o asemenea maniera incat unul dintre ele are mereu culoarea **rosu**.
 2. Cand ambele semafoare au culoarea **rosu** exista un jeton in starea x .
 3. Cand un semafor are culoarea **verde**, jetonul dispare din x , avand ca rezultat blocarea celuilalt semafor pe culoarea **rosu**.

Semafoare Varianta 1



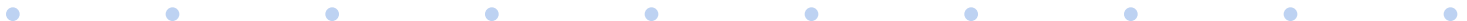
Semafoare1 Graf de marcaje

Fiecare stare din graf are
urmatoarea structura:
(rosu1, verde1, galben1,
rosu2, verde2, galben2, x)

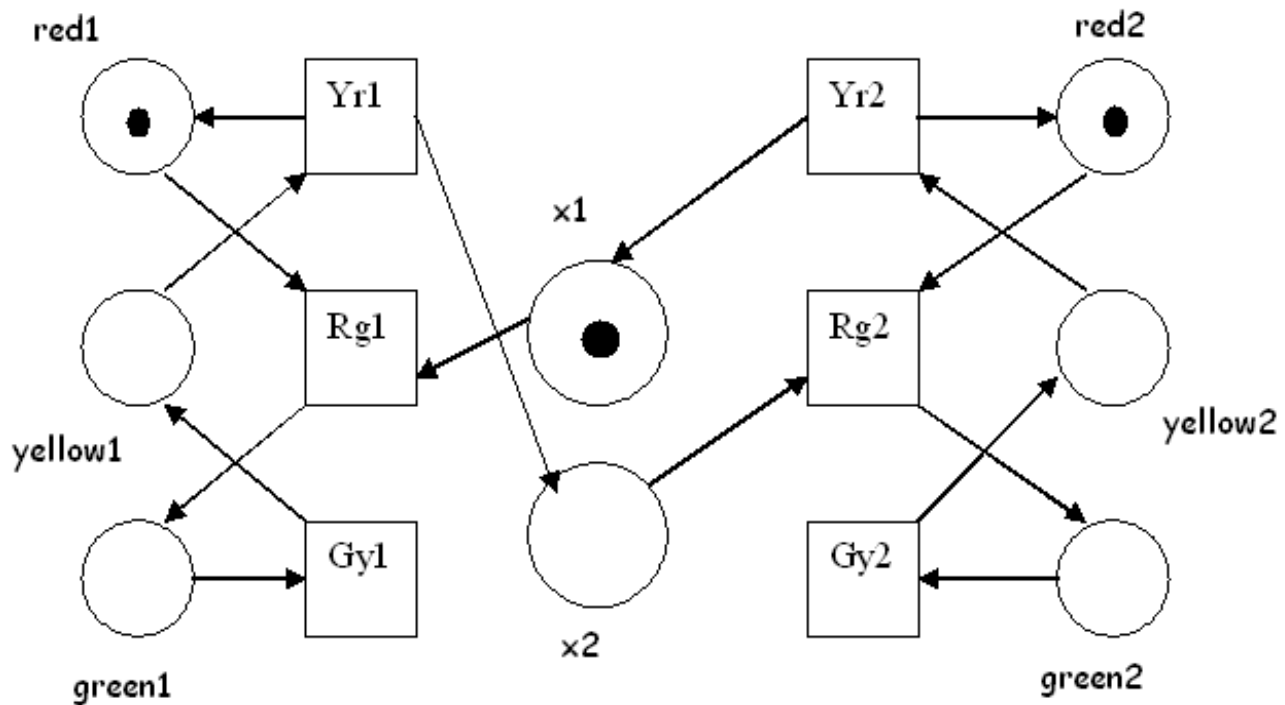


ATENȚIE!

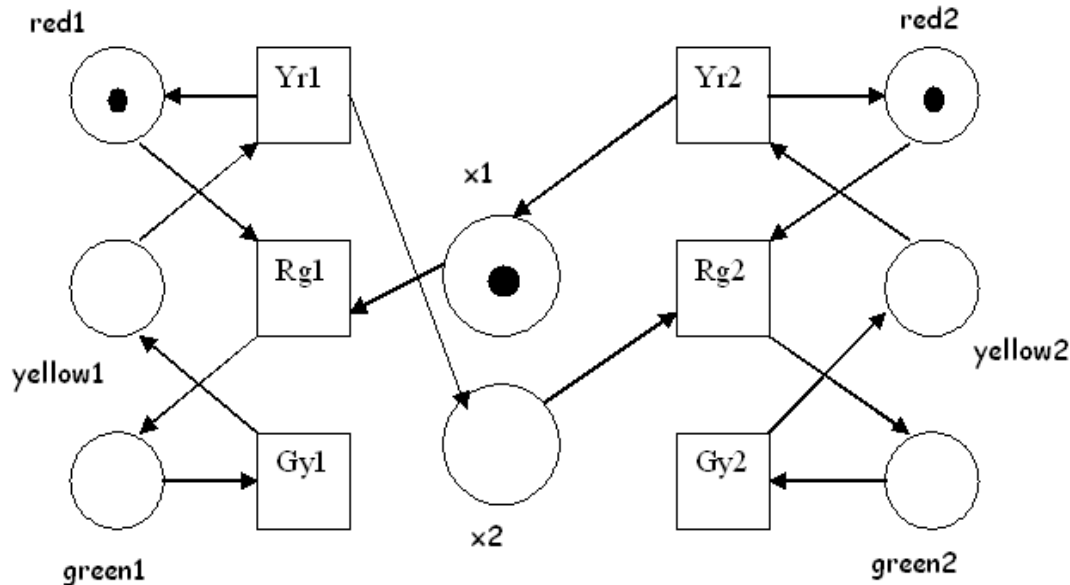
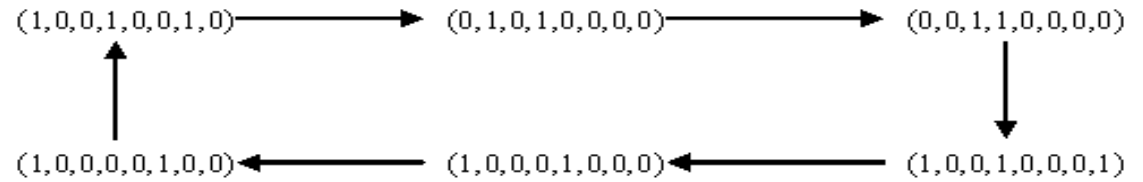
- Exista posibilitatea ca un semafor sa se schimbe intotdeauna pe culoarea verde, in timp ce, cel de-al doilea semafor sa ramana blocat pe culoarea rosu.
- Solutie: Introducerea celei de-a doua stari neutre x2.



Semafoare Varianta 2



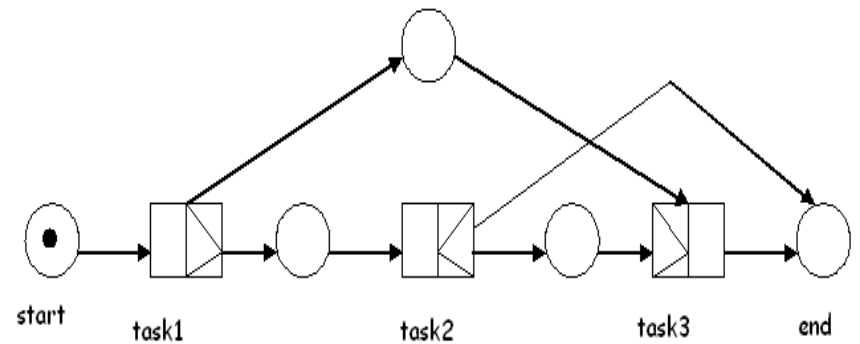
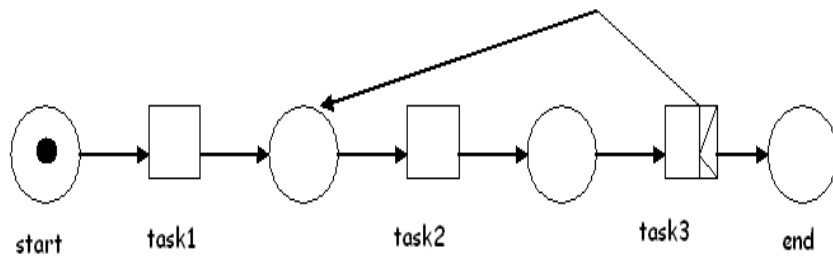
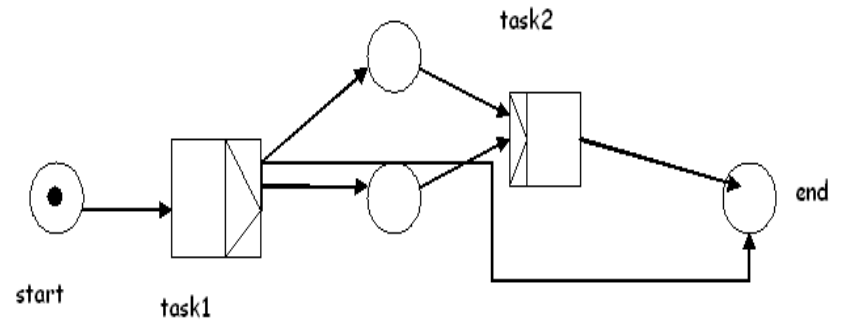
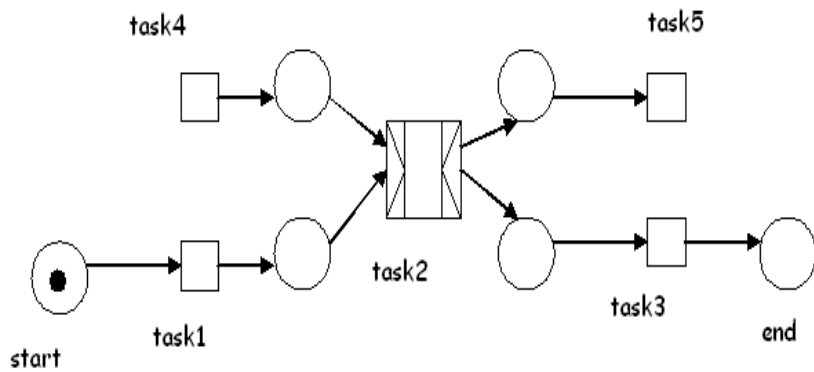
Semafoare Varianta 2



3. Analiza structurala

- Greseli frecvente in definirea unui proces
 1. Tranzitii fara conditii de intrare (este neclara situatia in care se pot executa) sau fara conditii de iesire (nu contribuie la rezolvarea unei situatii si se poate renunta la ele).
 2. Tranzitii “moarte” (nu se activeaza niciodata).
 3. Blocaje (deadlocks): pot compromite un caz inainte de a se ajunge la conditia da sfarsit.
 4. Livelock: inghetarea procesului intr-o bucla infinita.
 5. Activitati ce se executa dupa ce conditia de sfarsit a fost atinsa.
 6. Jetoane ramase in retea dupa ce executia a luat sfarsit.

Identificati greselile:



Viabilitatea rețelei

- Fiecare proces trebuie să îndeplinească următoarele *cerințe minimale* pentru a putea fi numit *viabil*: nu trebuie să conțină tranziții redundante și trebuie să se execute în totalitate (fără a mai exista jetoane în rețea).
- Un flux de activități definit în termeni de RP are o *singură* stare de *intrare* și o *singură* stare de *iesire*.
- !!! Orice tranziție poate fi executată plecând din starea de intrare (start) și urmărind un anumit număr de pași intermediari.
- !!! Starea de ieseire (end) este accesibilă din fiecare tranziție existentă, urmărind un anumit număr de pași intermediari.

Viabilitatea rețelei

- O rețea Petri care satisface condițiile anterior menționate poartă denumirea de:

RETEA DE FLUXURI DE ACTIVITATI
(WORKFLOW NET)



Viabilitatea rețelei

O rețea de fluxuri de activități este viabilă dacă:

1. Pentru fiecare jeton din *start* există un jeton în *end*.
 2. Când jetonul apare în *end*, în celelalte stări nu mai există niciun alt jeton.
 3. Pentru orice tranziție, este posibil să avansăm din starea inițială într-o altă stare din care tranziția să fie validă.
- Notiunea de viabilitate o include pe aceea de “obiectivitate” = dacă o tranziție poate fi executată, atunci acest lucru nu poate fi amânat la infinit.



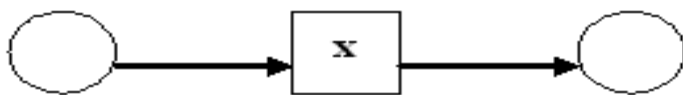
Cum verificam daca un proces corespunde unei retele de fluxuri de activitati (WFN)?

- V1: Construim graful de marcaje.
 - Nu poate fi construit pentru procese de scara larga fara ajutorul unui computer.
 - Nu ofera solutii pentru reabilitarea procesului corespunzator unei retele neviabile.
- V2:Metoda cu suport computerizat
 - Adaugam o stare aditionala retelei: t^* , avand ca intrare *end* si ca iesire *start* => *Retea Scurt-Circuitata (Short-Circuited Net SCN)*
 - viabilitate WFN =(absenta blocajelor(liveness) + marginire) SCN
 - Starea “live” = cand este posibil sa execut fiecare tranzitie din retea

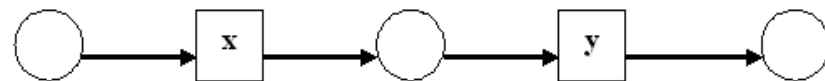
Cum verificam daca un proces corespunde unei retele de fluxuri de activitati (WFN)? (2)

- Marginire = exista o limita superioara finita a numarului de jetoane dintr-o stare.
- Cand un proces nu este viabil se pot genera diagnostice pentru a vedea motivele pentru care acest lucru se intampla.
- V3: Metoda fara suport computerizat
 - Adaugam la absenta blocajelor si marginire proprietatea de "retea sigura" = in fiecare stare nu poate fi mai mult de un jeton (marginita superior de 1).
 - O retea viabila se comporta ca o tranzitie!
 - Pot descompune retea in blocuri structurale despre care se stie ca sunt viabile!

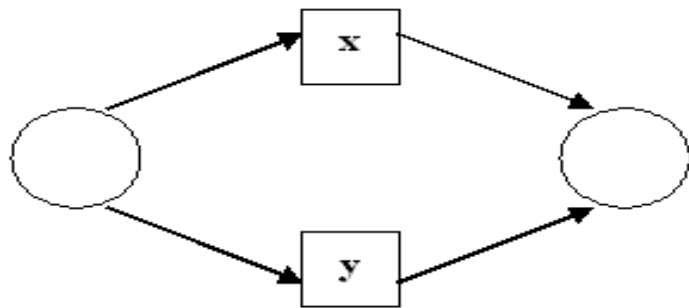
Blocuri structurale (1)



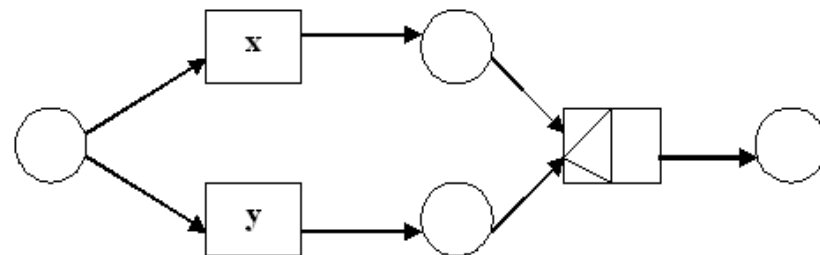
bloc de baza



constructie secventiala

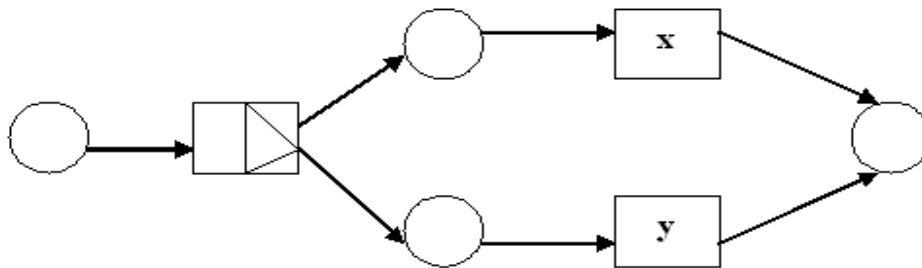


Ramificatie SAU implicit

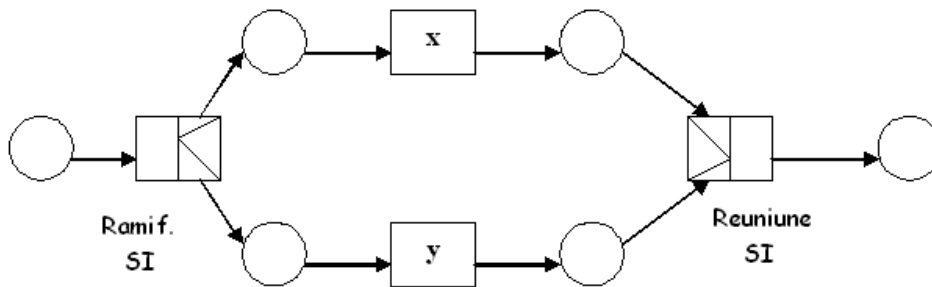


Reuniune de tip SAU

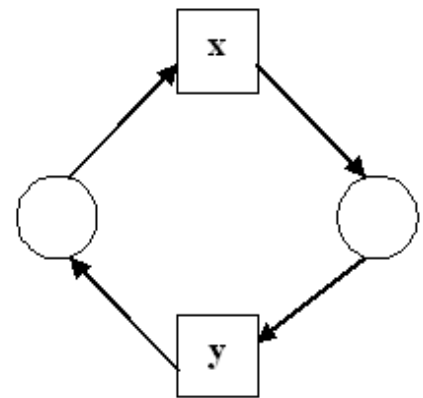
Blocuri structurale (2)



Ramificatie SAU explicit

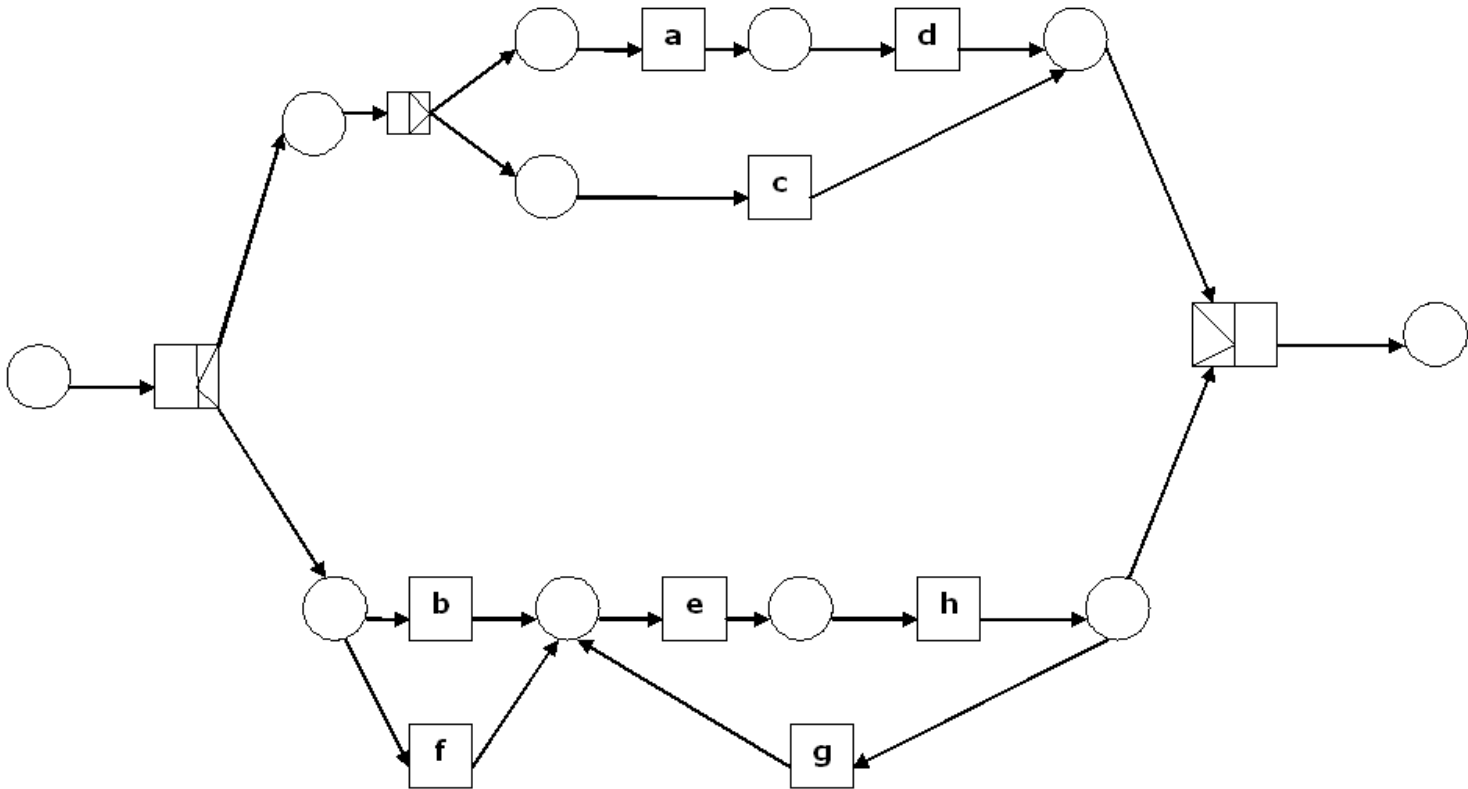


Constructie de tip SI



Constructie iterativa

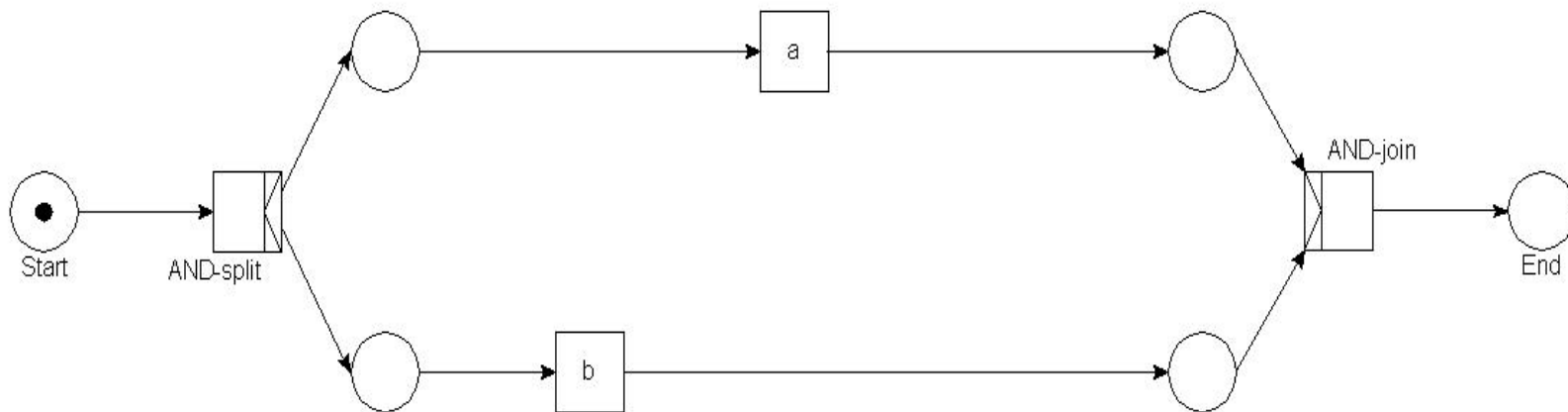
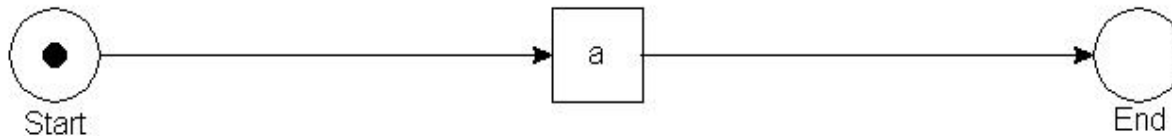
Demonstrati ca urmatoarea retea este viabila prin descompunerea sa in blocuri de baza:

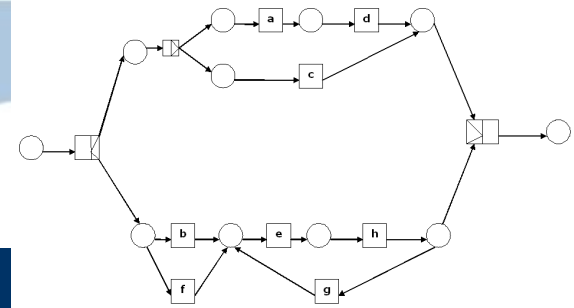


SOLUTIE :

P1. Bloc de baza cu tranzitia a

P2. Constructie de tip Si prin adaugarea tranzitiei b

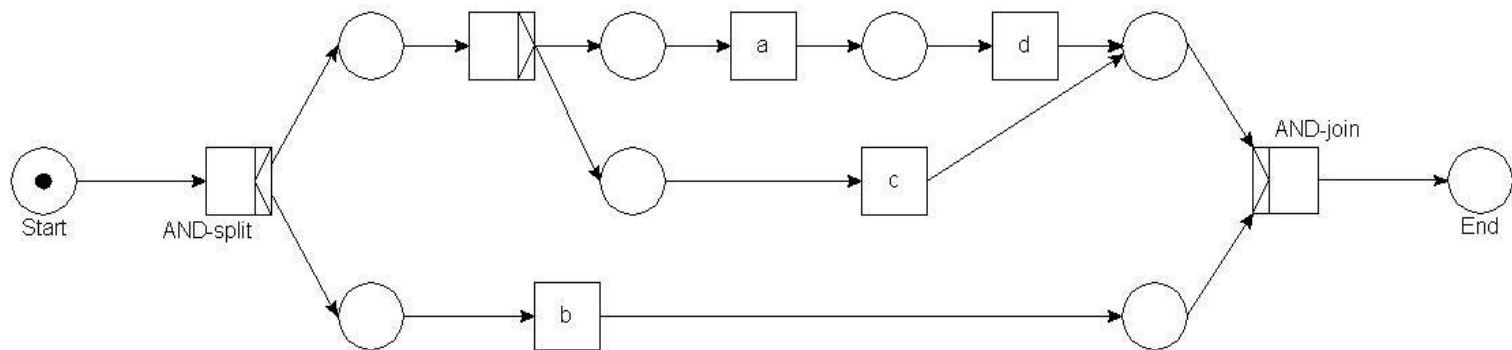
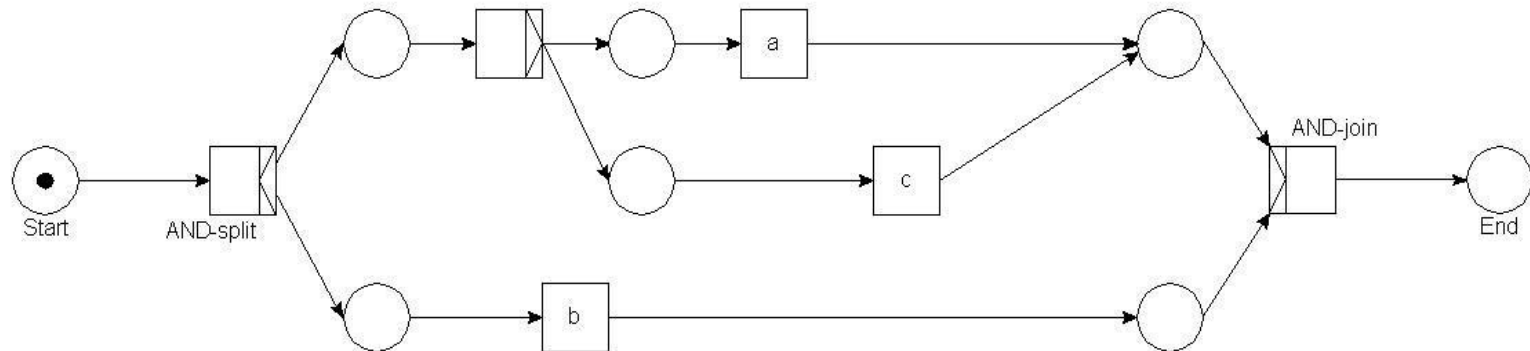


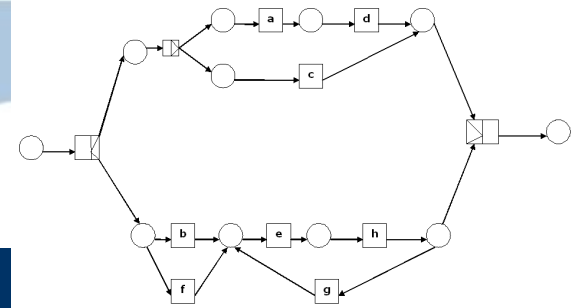


SOLUTIE (continuare) :

P3) Constructie SAU explicit cu tranzitia c

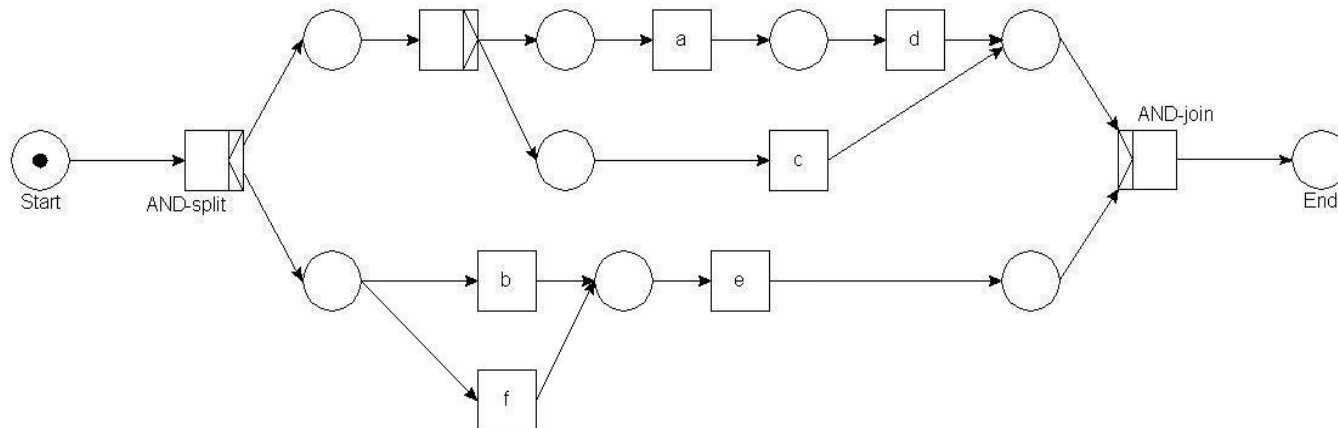
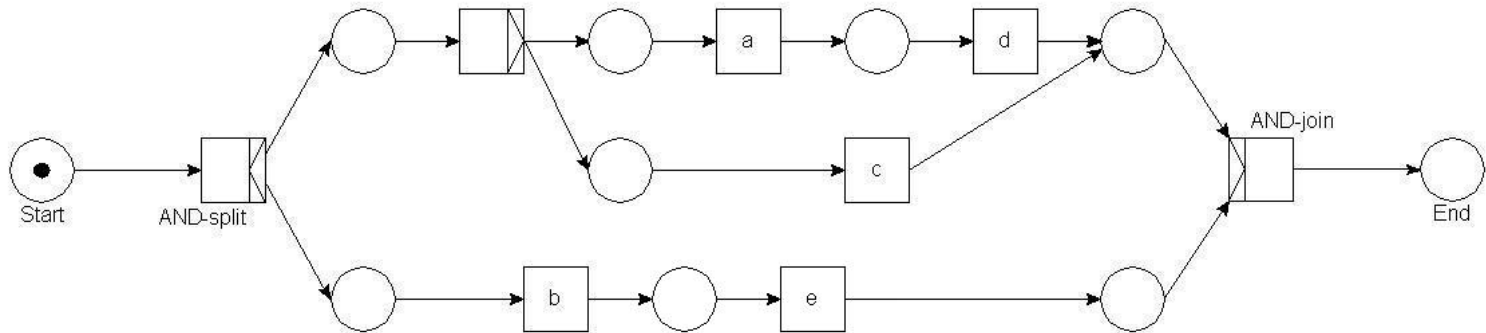
P4) Constructie secventiala cu tranzitia d



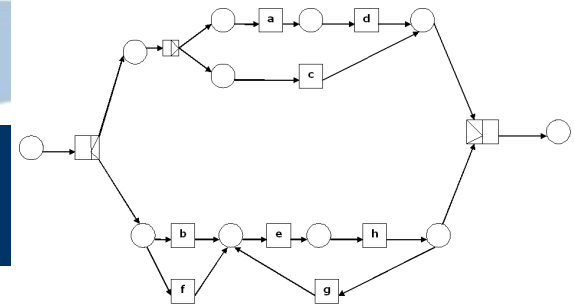


SOLUTIE (continuare) :

- P5) Constructie secventiala cu tranzitia e
- P6) Constructie SAU implicit cu tranzitia f

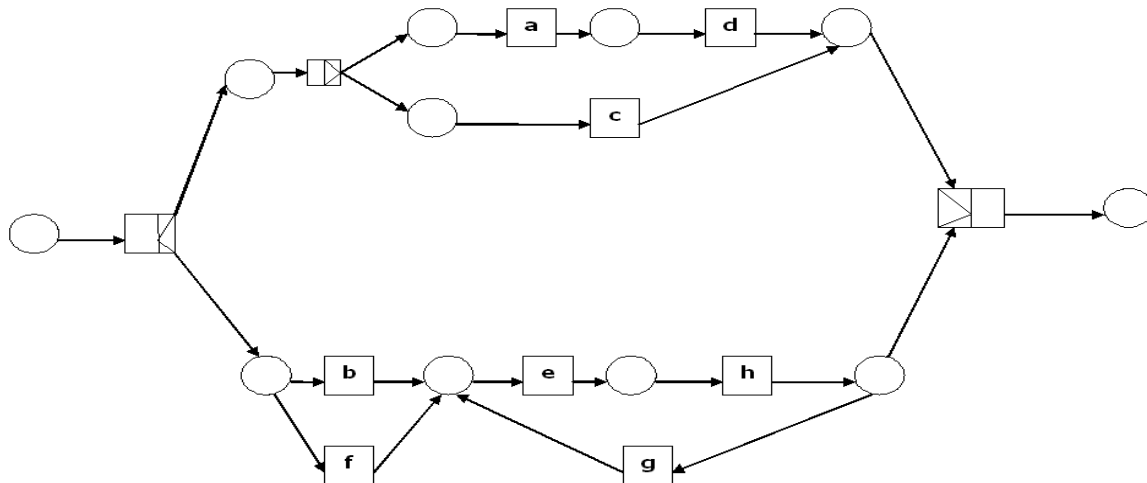
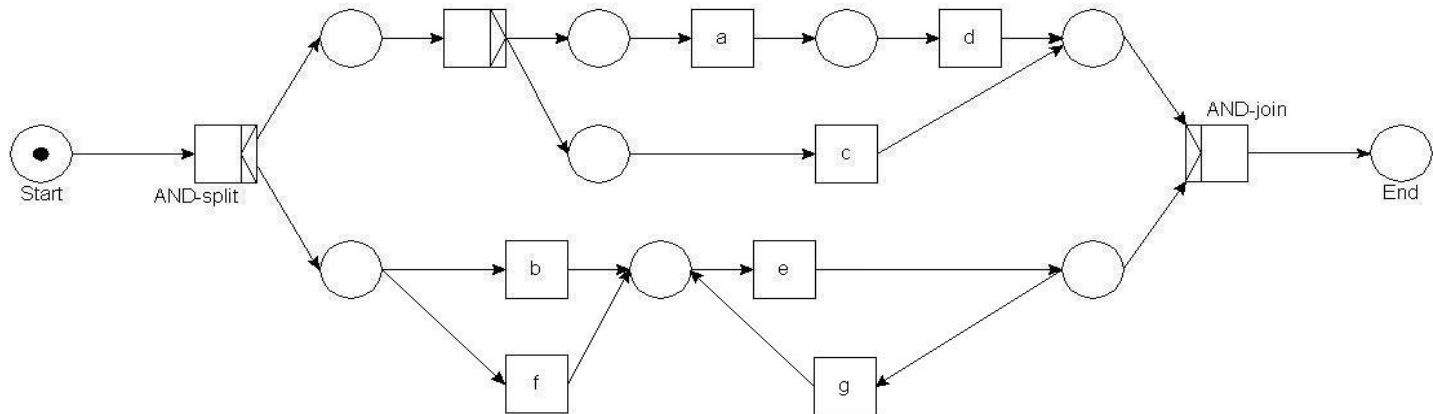


SOLUTIE (continuare) :



Pasul 7) Constructie iterativa cu tranzitia g

Pasul 8) Constructie secventiala cu tranzitia h



OBSERVATII:

1. Pot exista mai multe posibilitati de a descompune o retea.
2. Nu toate retelele viabile pot fi descompuse in subblocuri de baza.
3. In cazul in care, pentru o retea, nu putem gasi o descompunere, nu este cazul sa concluzionam ca retea initiala nu este viabila.

