



Verificarea protocoalelor



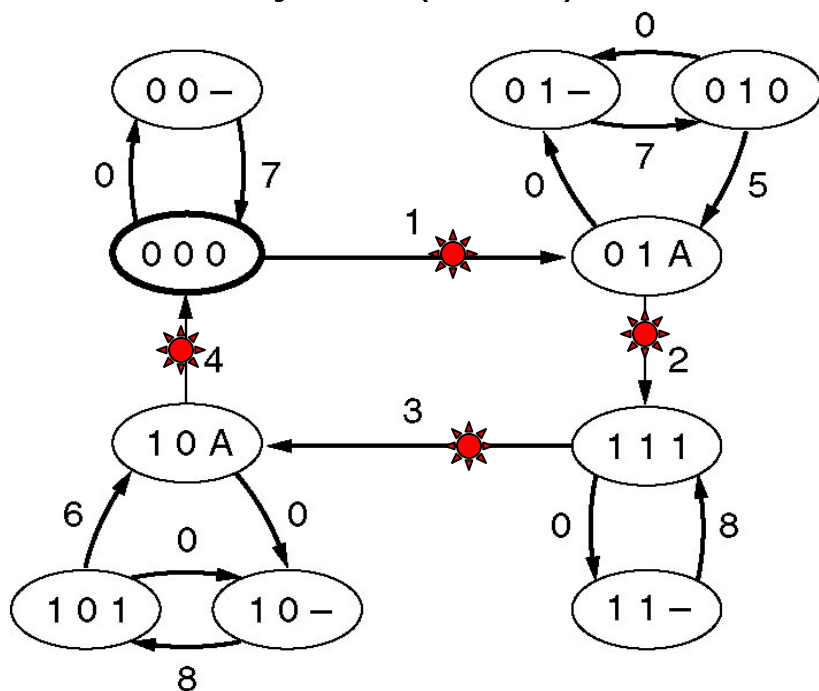
Verificarea Protocoalelor

Specificarea

- ne-formala (limbaj natural)
- formala - modele
 - tranziționale
 - automate - FSMs (Finite State Machines)
 - rețele Petri
 - algoritmice
 - hibride – FDTs (Formal Description Techniques)
 - Estelle (ISO 9074)
 - LOTOS (**L**anguage **O**f **T**emporal **O**rdering **S**pecification) – ISO 8807
 - SDL (**S**pecification and **D**escription **L**anguage) – ITU-T recomandarea Z.100
- TTCN-3 - Testing and Test Control Notation Version 3

Model de automat finit

- FSM: Finite State Machine (exemplu pentru protocolul 3).
- Reprezintă stările ca un triplet (T, R, C):
 - Transmitator: (0, 1): corespunde cadrului trimis;
 - Receptor: (0, 1): corespunde cadrului așteptat;
 - Canal: (0, 1, A, -): corespunde conținutului canalului.
- Stare inițială: (0,0,0)



(a)

Transition	Who runs?	Frame accepted	Frame emitted	To network layer
0	-	(frame lost)		-
1	R	0	A	Yes
2	S	A	1	-
3	R	1	A	Yes
4	S	A	0	-
5	R	0	A	No
6	R	1	A	No
7	S	(timeout)	0	-
8	S	(timeout)	1	-

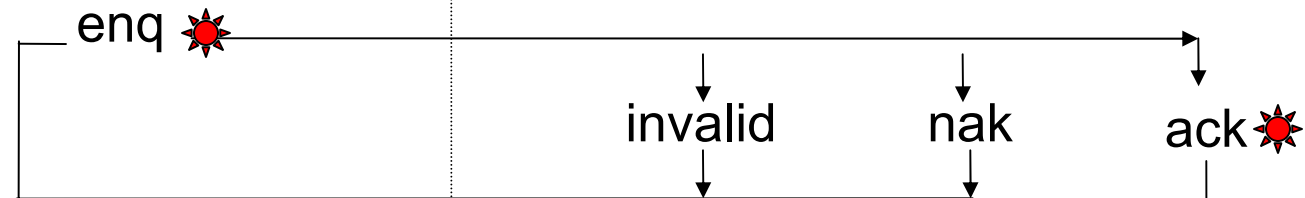
(b)

Modelul diagramelor de tranziții

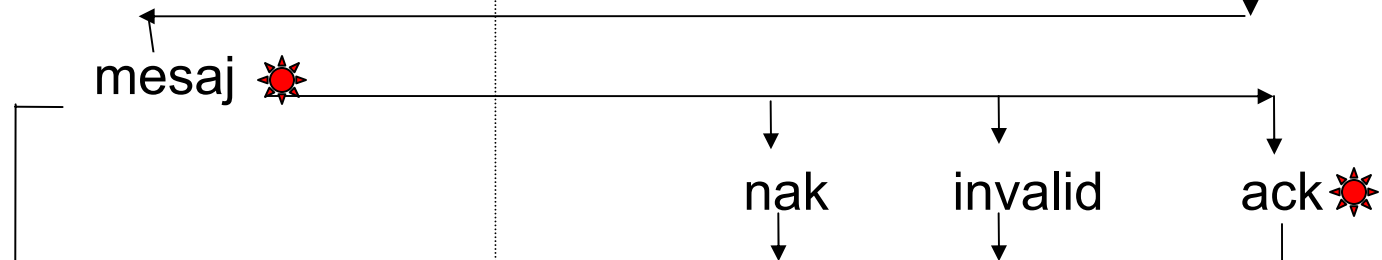
stția 1 -> stția 2

stția 2 -> stția 1

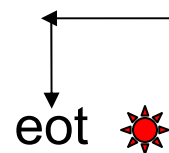
stabilire



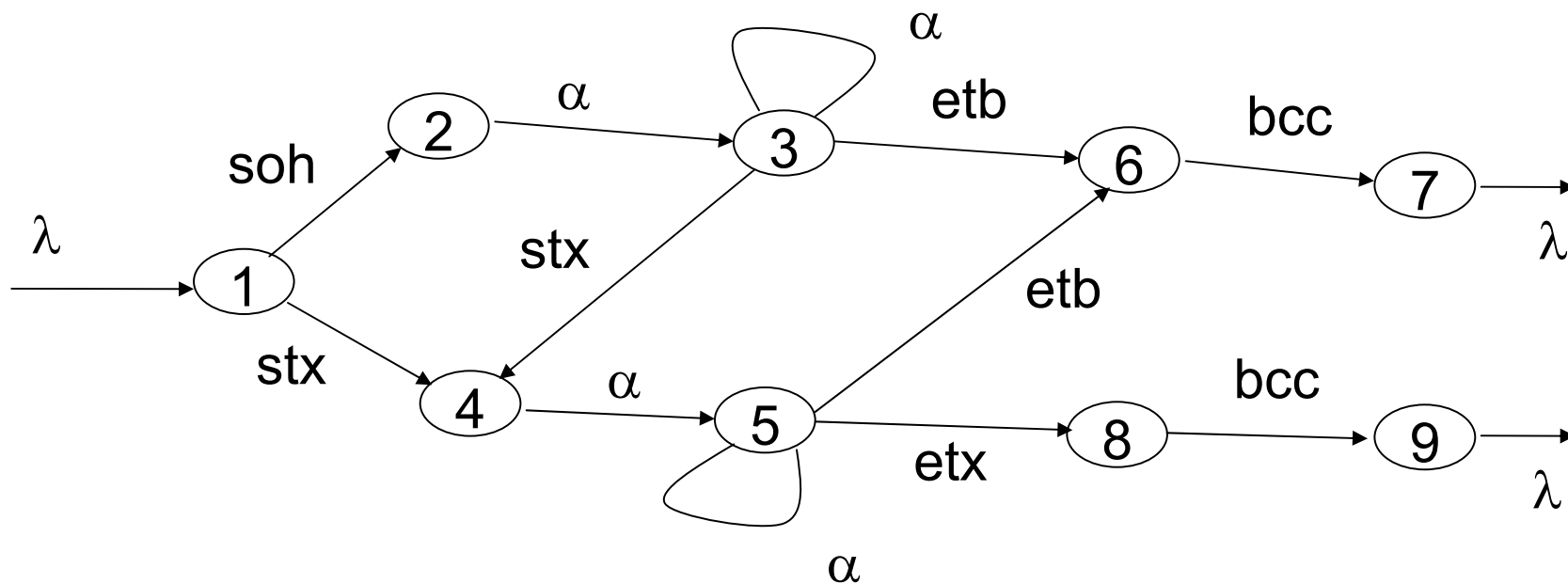
transfer



terminare



Modelul transmițătorului / receptorului





Acțiuni:

Tabel acțiuni

a - inițializare tampon mesaj;

b - inițializare tampon text și variabile text;

c - start text transparent;

d - primul caracter; pune contor pe valoarea 1;

e - alte caractere de text; increment contor;

f - sfârșit antet, memorează și interpretează început text;

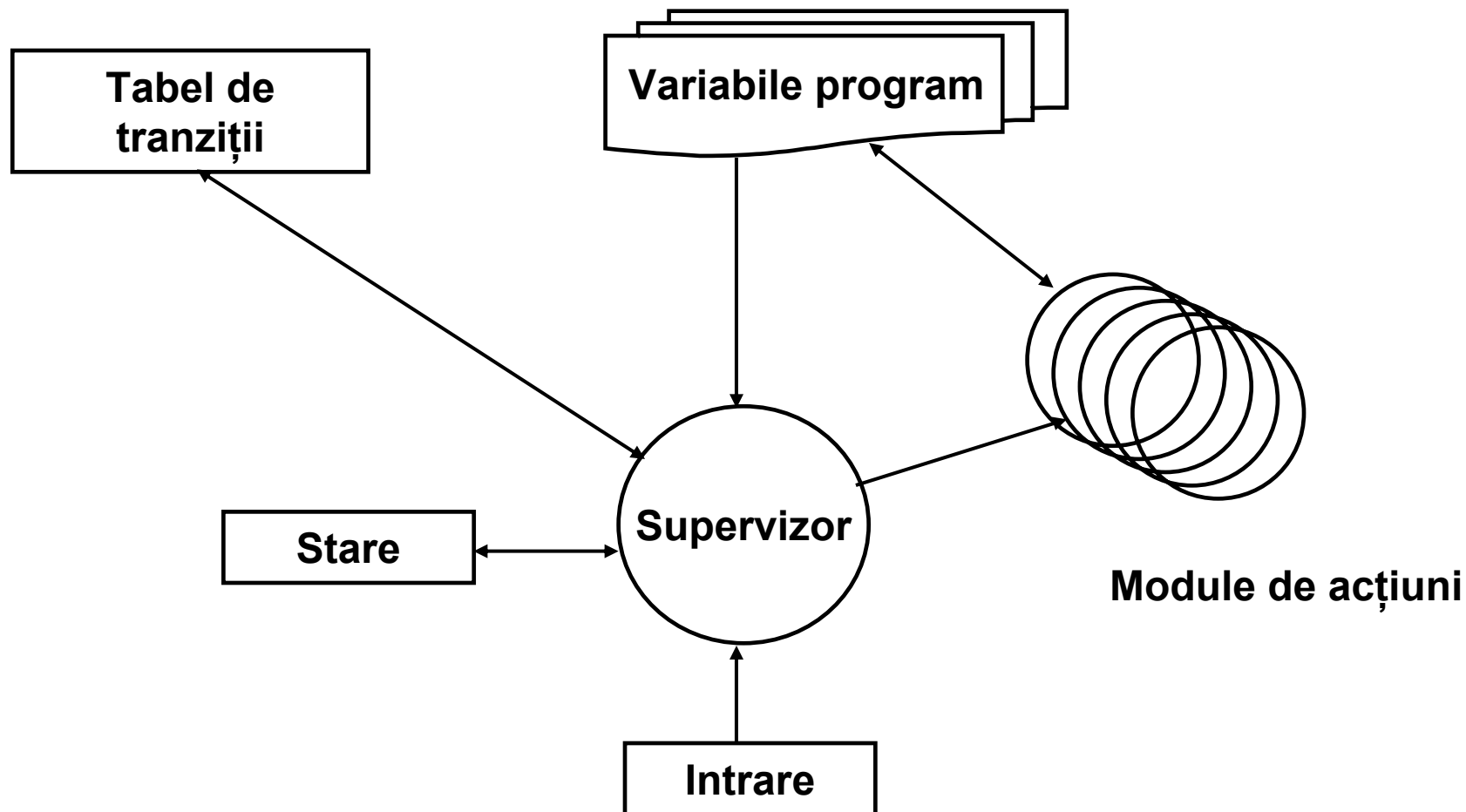
g - verificare erori;

k - sfârșit bloc/text;

stare\intr	soh	stx	α	dle	syn	etb	etx	β	itb	bcc
1	2 a	4 b		10 c						
2			3 d		2					
3		4 f	3 e		3	6 k				
4			5 d		4					
5			5 e		5	6 k	8 k			
6										7 g
7										
8										9 g
9										

stare următoare / acțiune

Implementare





Rețele Petri

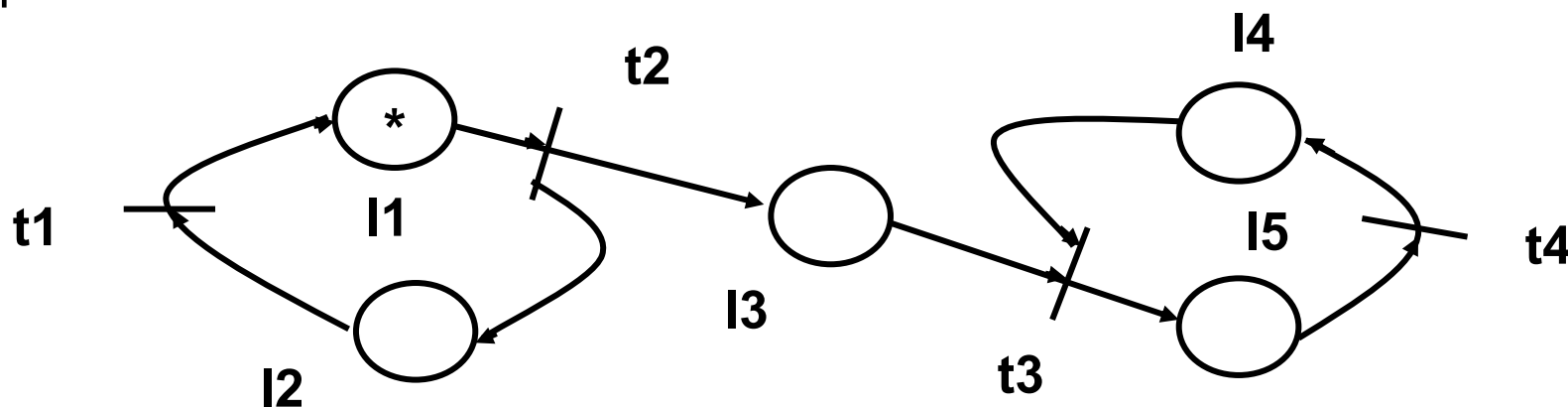


Rețele Petri

- Propuse în forma inițială de Carl Adam Petri în 1962;
- Au evoluat spre mai multe variante:
 - Rețele Petri elementare;
 - Rețele Petri generalizate;
 - Rețele Petri cu arce inhibitoare;
 - Rețele Petri colorate;
 - Rețele Petri continue;
 - Rețele Petri cu predicate;
 - Rețele Petri cu capacități;
 - Rețele Petri cu priorități.
- Unele variante sunt echivalente între ele, altele nu.
- Se folosesc pentru analiza (automată) a proprietăților sistemelor concurente, bazate pe evenimente:
 - Caz particular: verificarea protocoalelor.

Definiții (1)

- Rețea Petri este **RP** = (**L**, **T**, **I**, **O**)
 - **L**, mulțime finită de locuri;
 - **T**, mulțime finită de tranziții, $\mathbf{L} \cap \mathbf{T} = \Phi$;
 - **I**, funcție de incidență înainte $\mathbf{I} : \mathbf{L} \times \mathbf{T} \rightarrow \{0, 1\}$;
 - **O**, funcție de incidență înapoi $\mathbf{O} : \mathbf{T} \times \mathbf{L} \rightarrow \{0, 1\}$.
- Un Marcaj **M** : **L** → **N** asociază fiecărui loc un număr natural.
- Pentru fiecare **t** din **T** se definesc:
 - Mulțimea locurilor de intrare: **Pre**(**t**) = {**l** ∈ **L** | **I**(**l**, **t**) > 0};
 - Mulțimea locurilor de ieșire: **Post**(**t**) = {**l** ∈ **L** | **O**(**t**, **l**) > 0}.
- Reprezentare:



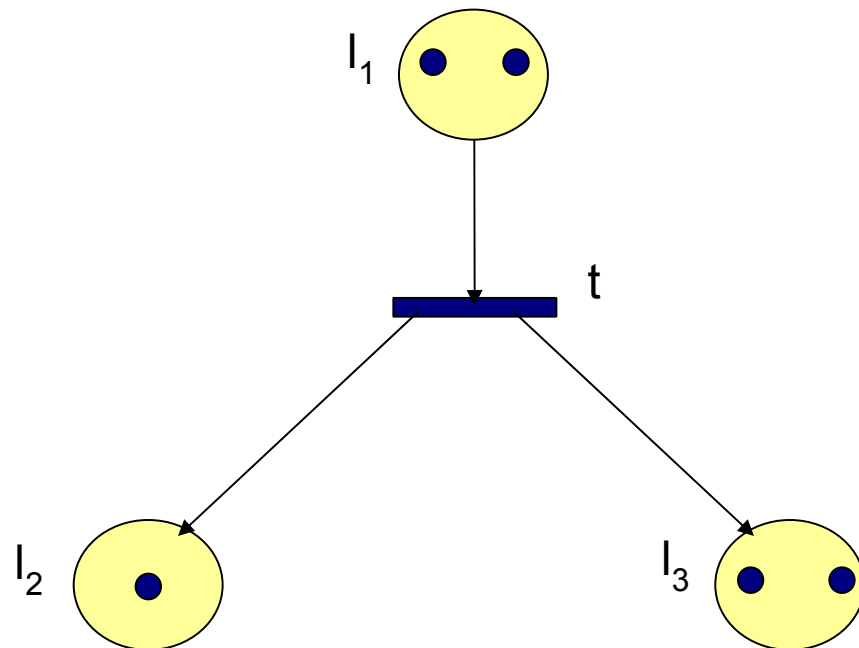


Definiții (2)

- Tranziția t este **executabilă** pentru marcajul M dacă:
 - $M(I) \geq I(I,t)$, $\forall I \in L$ (fiecare loc de intrare are cel puțin un punct).
- **Execuția** tranziției t în M schimbă marcajul în M' :
 - $M'(I) = M(I) - I(I,t) + O(t,I)$, $\forall I \in L$.
- Pentru execuția unei tranziții se folosește notația:
 - $M \xrightarrow{t} M'$
- O **secvență posibilă de execuții** din M în M' , notată $M \xRightarrow{\$} M'$, este $\$ = t_1, t_2, t_3, \dots, t_n$ dacă $\exists$ marcasele $M_1, M_2, M_3, \dots, M_n$ a.î.
 $M \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \xrightarrow{t_3} M_3 \dots \xrightarrow{t_n} M_n = M'$
- Fie RP și M_0 un marcaj inițial. Clasa marcajelor accesibile din M_0 este notată $A(M_0)$.

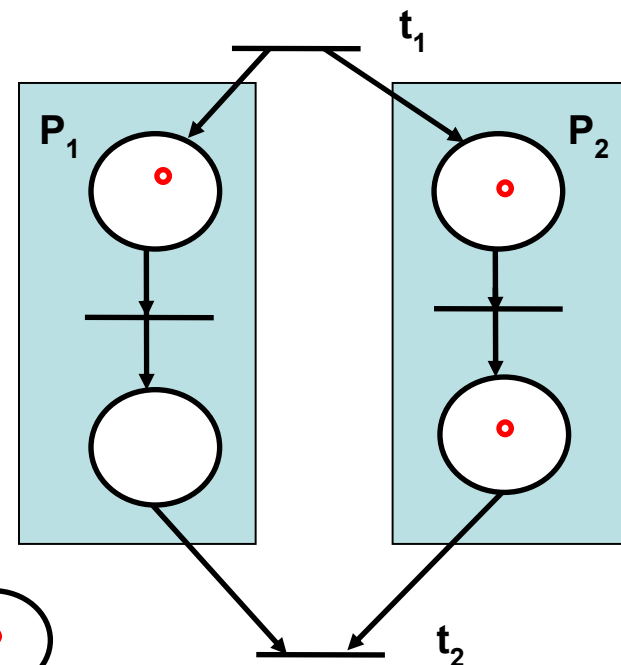
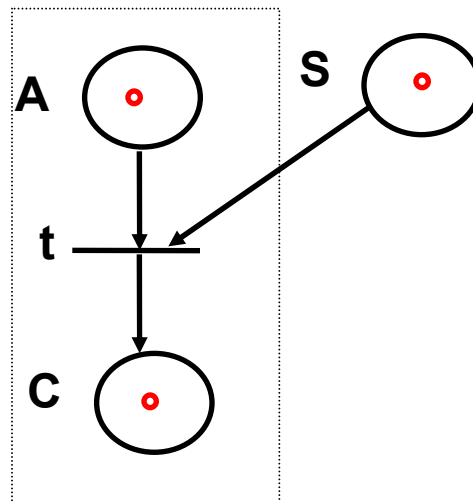
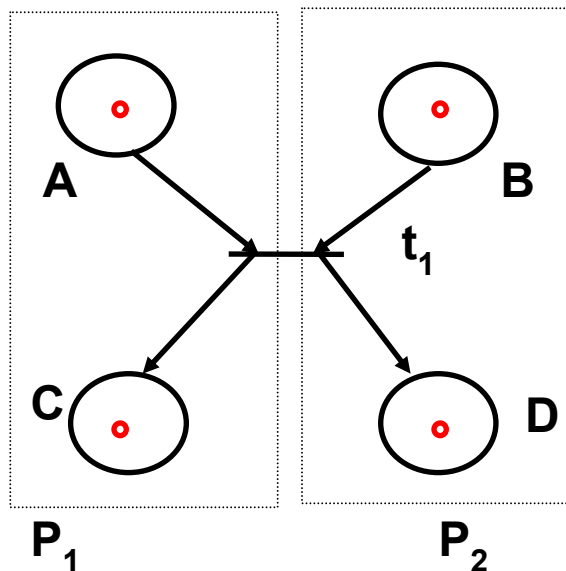
Execuția unei tranziții

- Fie:
 - Locurile l_1, l_2, l_3 ;
 - Marcajul inițial $M = [2, 0, 1]$;
 - Tranziția t , executabilă.
- Execuția este **instantanee**.
- Marcajul după execuția tranziției $M' = [1, 1, 2]$.
 - Fiecare loc de intrare pierde un punct;
 - Fiecare loc de ieșire primește un punct.



Ce modelăm?

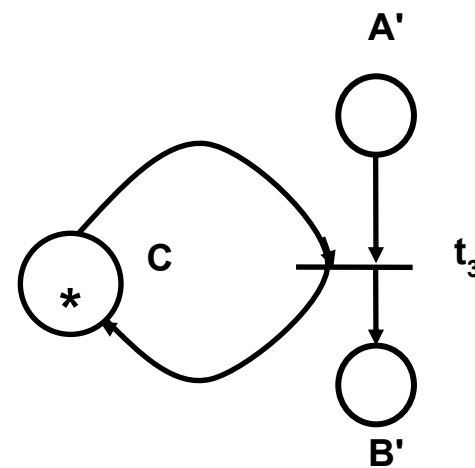
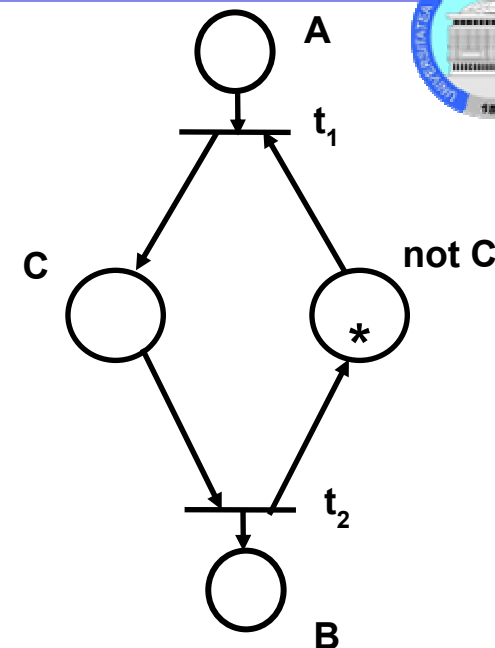
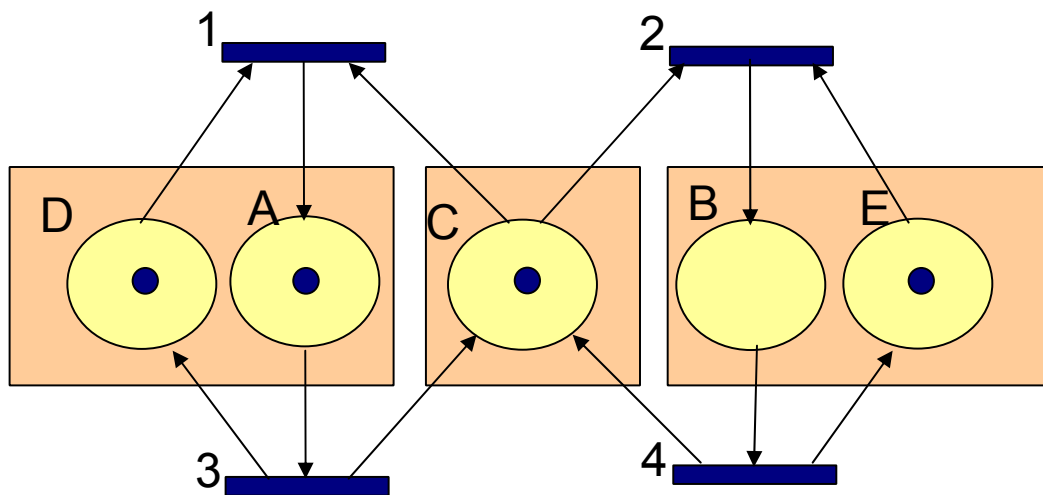
- Paralelismul:
 - P_1 și P_2 se execută în paralel.
- Sincronizarea:
 - Rendez-vous.
 - Semafor.



Ce mai modelăm?

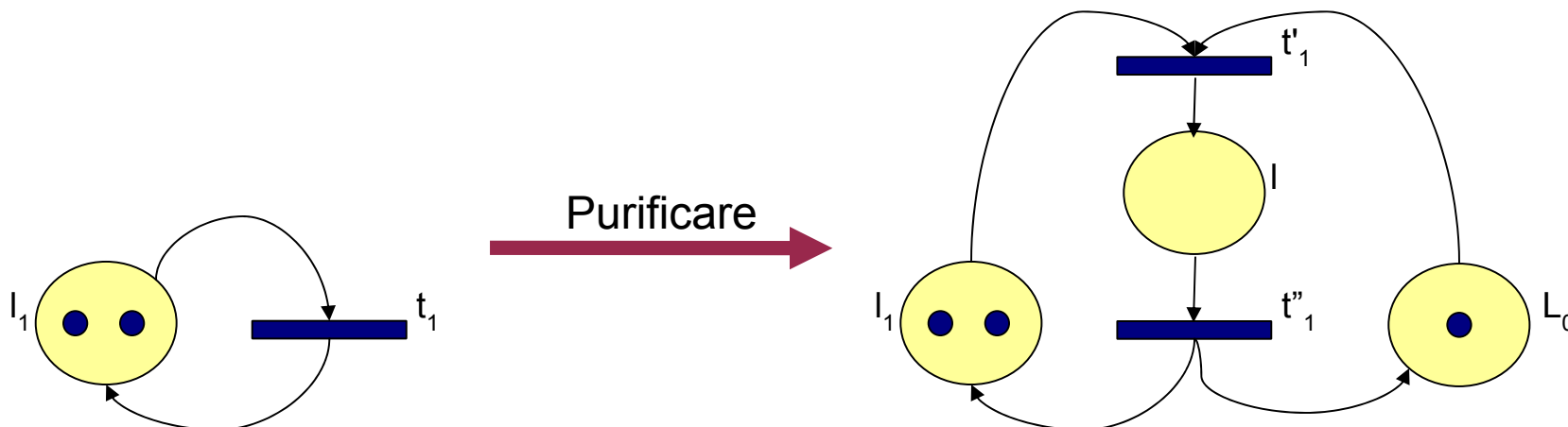
- Memorarea unei condiții și a opusului său.
- Citirea unei condiții, fără modificarea ei.
- Excluderea mutuală.

Resursa e eliberată,
poate fi refolosită



Rețele Petri pure

- O tranziție **impură** are locuri de intrare care sunt și de ieșire:
 - $\text{Pre}(t) \cap \text{Post}(t) \neq \emptyset$.
- O rețea Petri e **impură** dacă are cel puțin o tranziție impură.
- Rețelele impure mai dificil de analizat → **purificare**.
- Algoritm de purificare (crește **dimensiunea** rețelei!):
 - Se adaugă un **loc special** L_0 , cu marcaj 1;
 - Se transformă **fiecare tranziție** t într-un ansamblu de 2 tranziții t' , t'' și un loc intermediar l , cu marcaj 0;
 - Se adaugă un arc de la L_0 la fiecare tranziție t' și un arc de la fiecare tranziție t'' către L_0 .
- Exemplu:



Modelarea protocolului start-stop

- Model algoritmic:

- Transmițător:

```

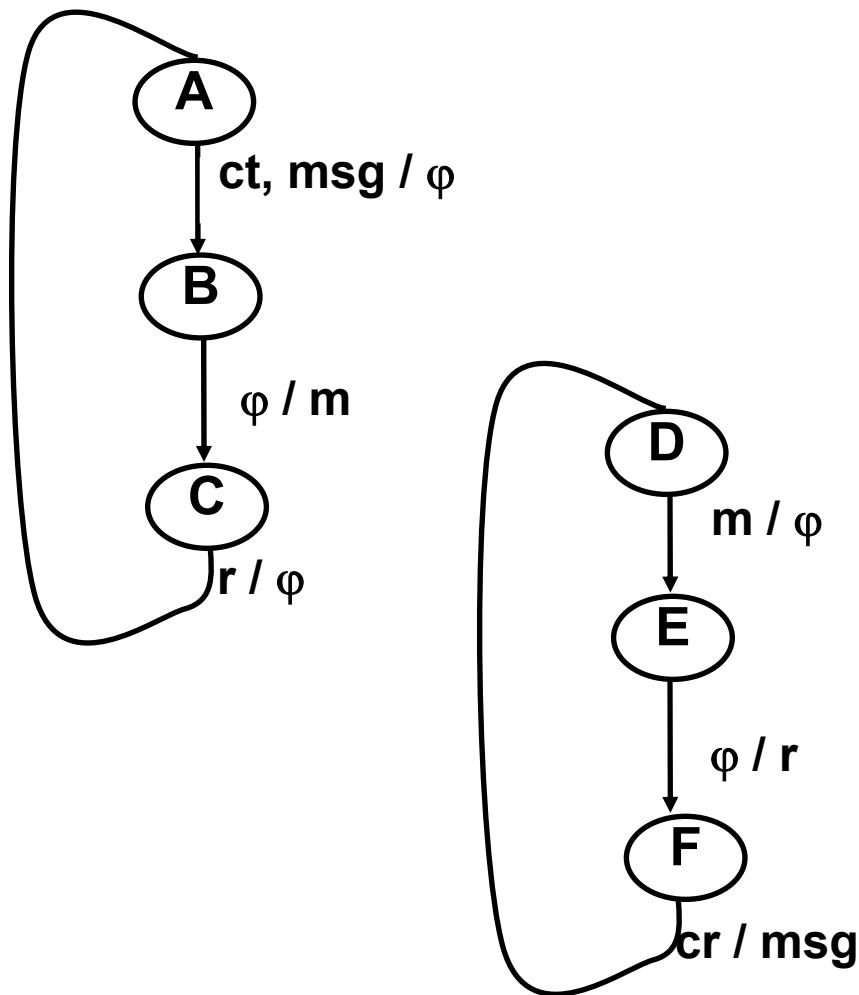
do
{
  asteapta cerere emisie (ct,msg) A
  pregateste mesaj (msg,m)
  transmite mesaj (m) B
  asteapta confirmare (r) C
} forever;
  
```

- Receptor:

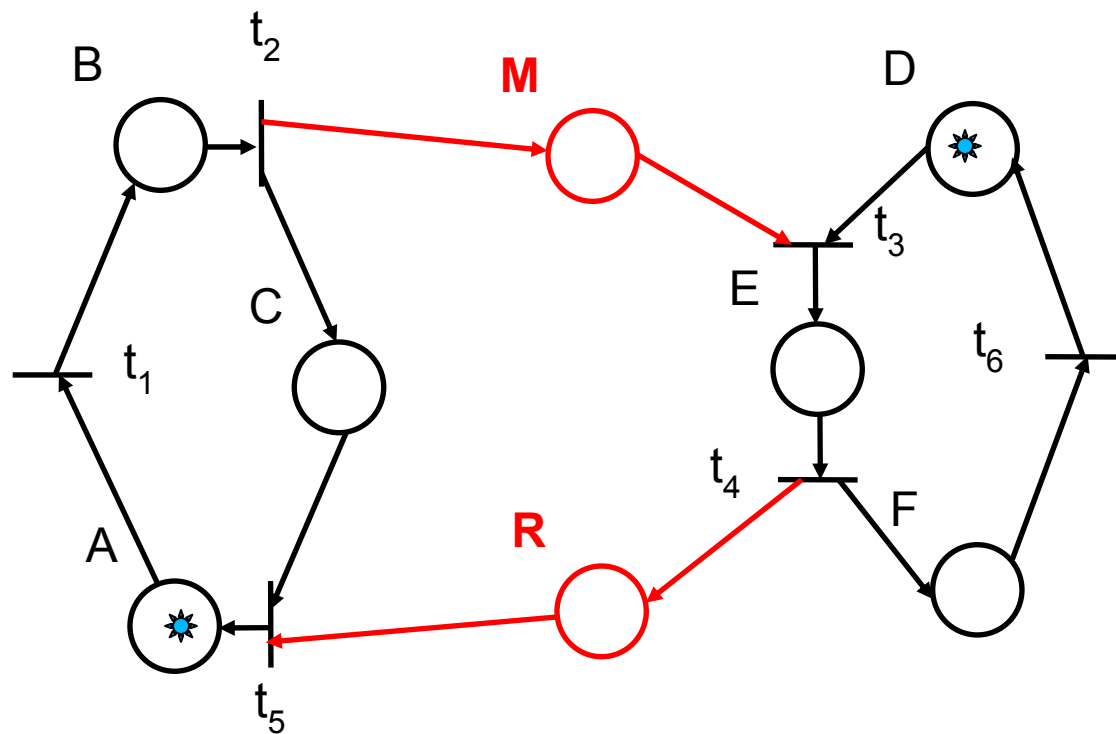
```

do
{
  asteapta mesaj (m) D
  pregateste raspuns (m,r, msg)
  transmite confirmarea (r) E
  asteapta cerere receptie (cr)
  transfera mesaj (msg) F
} forever
  
```

- Model de automat finit:



Rețeaua Petri asociată



Tranziții:

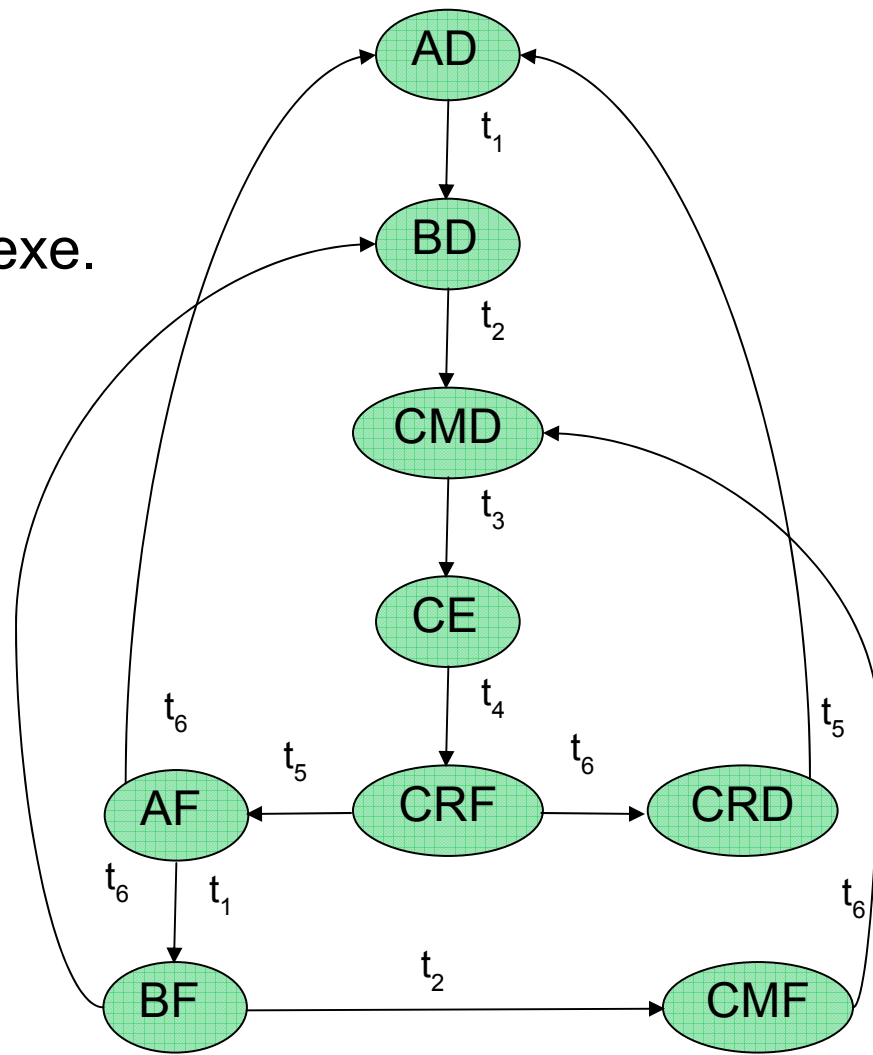
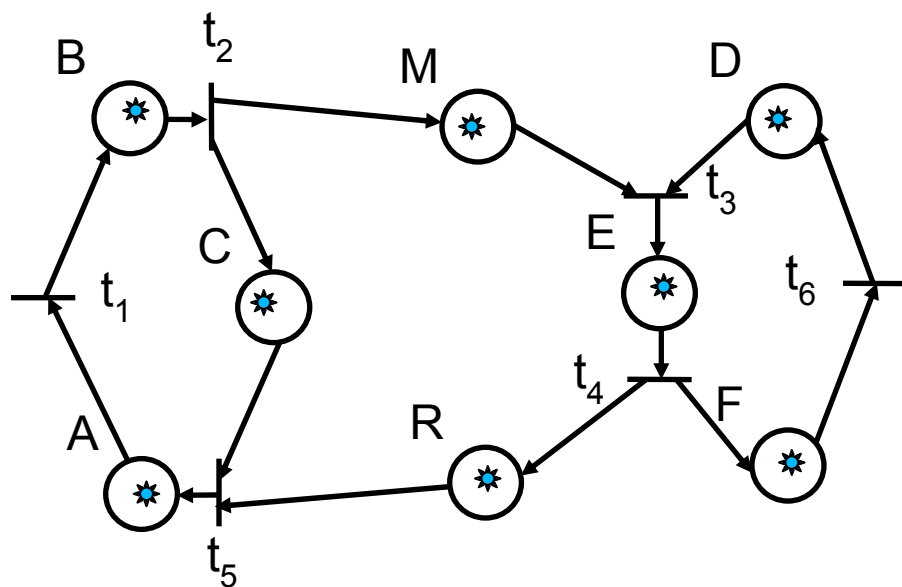
- t1** preluare mesaj produs de utilizatorul transmițător;
- t2** transmitere mesaj mediului de comunicare;
- t3** recepție mesaj de la mediu;
- t4** transmitere confirmare;
- t5** recepție confirmare;
- t6** consumare mesaj de utilizator receptor.

Marcajul corespunzător unei stări inițiale M_0 :

- Entitatea emițătoare așteaptă producerea unui mesaj (A);
- Entitatea receptoare este pregătită pentru recepție (D);
- Mediul de transmisie este gol.

Mașina de puncte

- Diagramă de tranziții.
- Explorează marcajele posibile.
- Dificil de folosit la sisteme complexe.



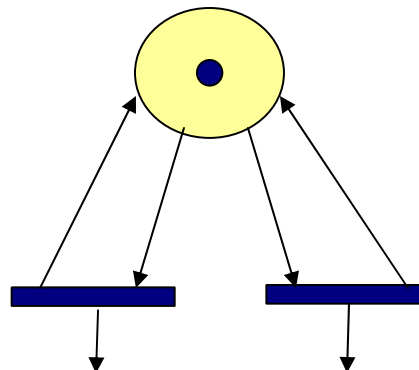
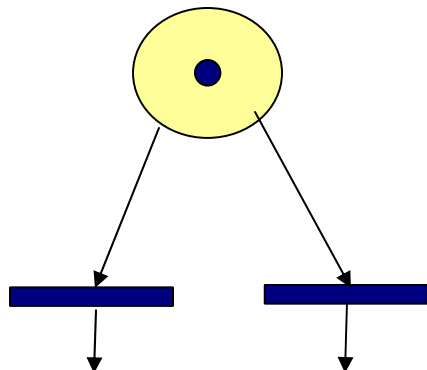


Validarea protocoalelor - proprietăți generale

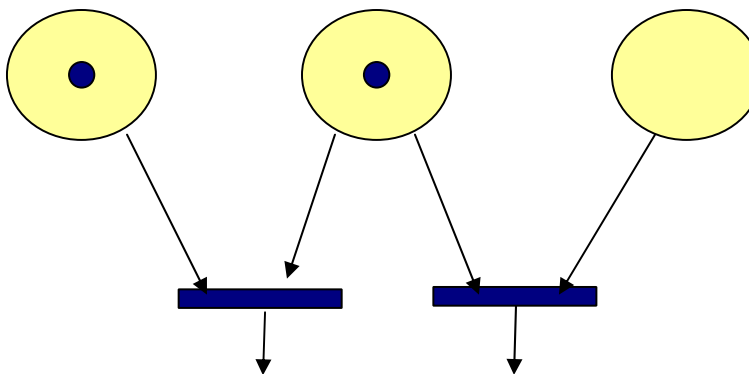
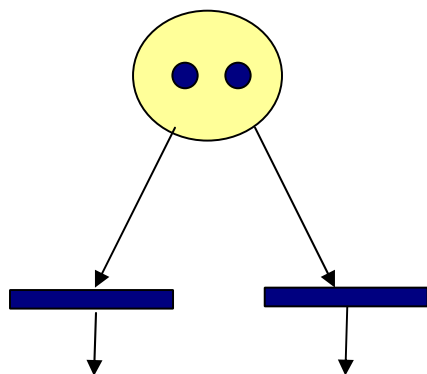
- Mărginire:
 - Orice $M \in A(M_0)$ și orice l din $L \rightarrow M(l) \leq n$.
- Siguranță:
 - Mărginire pentru $n=1$.
- Viabilitate:
 - Din oricare M accesibil din M_0 există o secvență de execuții care conține t .
- Cvasi-viabilitate:
 - există o secvență de execuții din M_0 care conține t .
- Home state:
 - Din oricare M accesibil din M_0 există o secvență de execuții care conduce în H .

Persistența

- Conflicte efective:



- Conflicte structurale, dar ne-efective:



Persistență: În orice M accesibil din M_0 , în care t_j și t_k sunt executabile, t_j , t_k și, prin simetrie t_k , t_j sunt secvențe posibile de execuții din M .

Validarea protocoalelor - proprietăți specifice

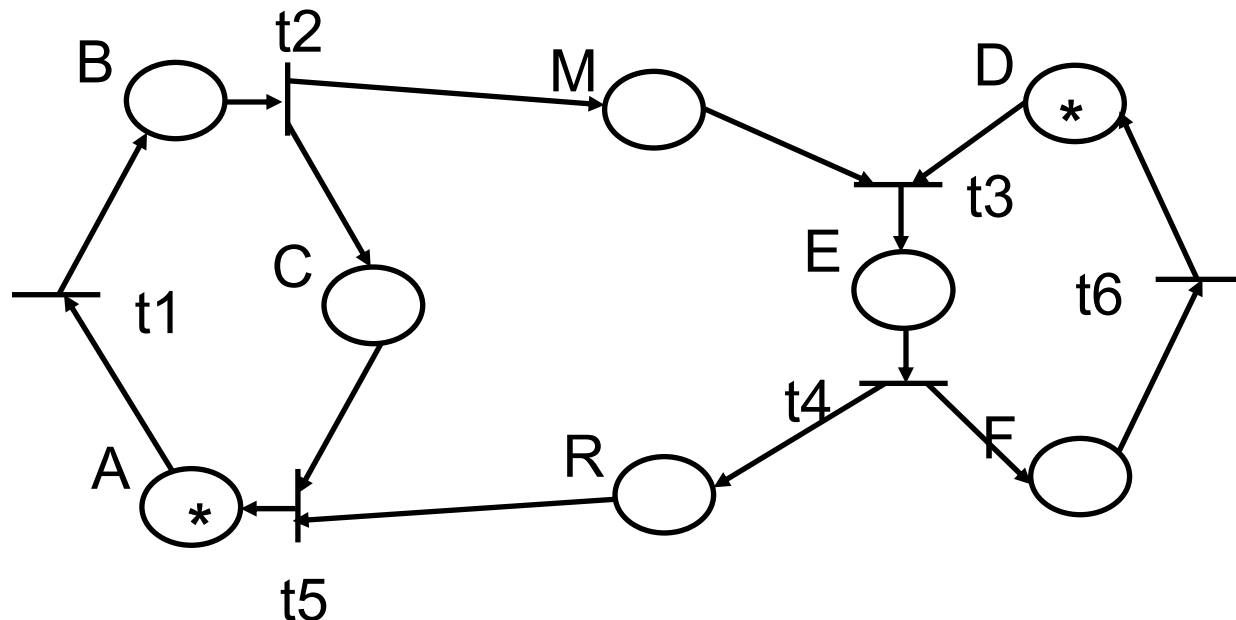
- Invarianți:

- Pe locuri (L-invarianți):

- $M(A) + M(B) + M(C) = 1$, pentru orice $M \in A(M_0)$.
 - Componentă / rețea conservativă.

- Pe tranziții (T-invarianți):

- Avans sincron $0 \leq N(t3) - N(t4) \leq 1$.
 - Secvențe repetitive.



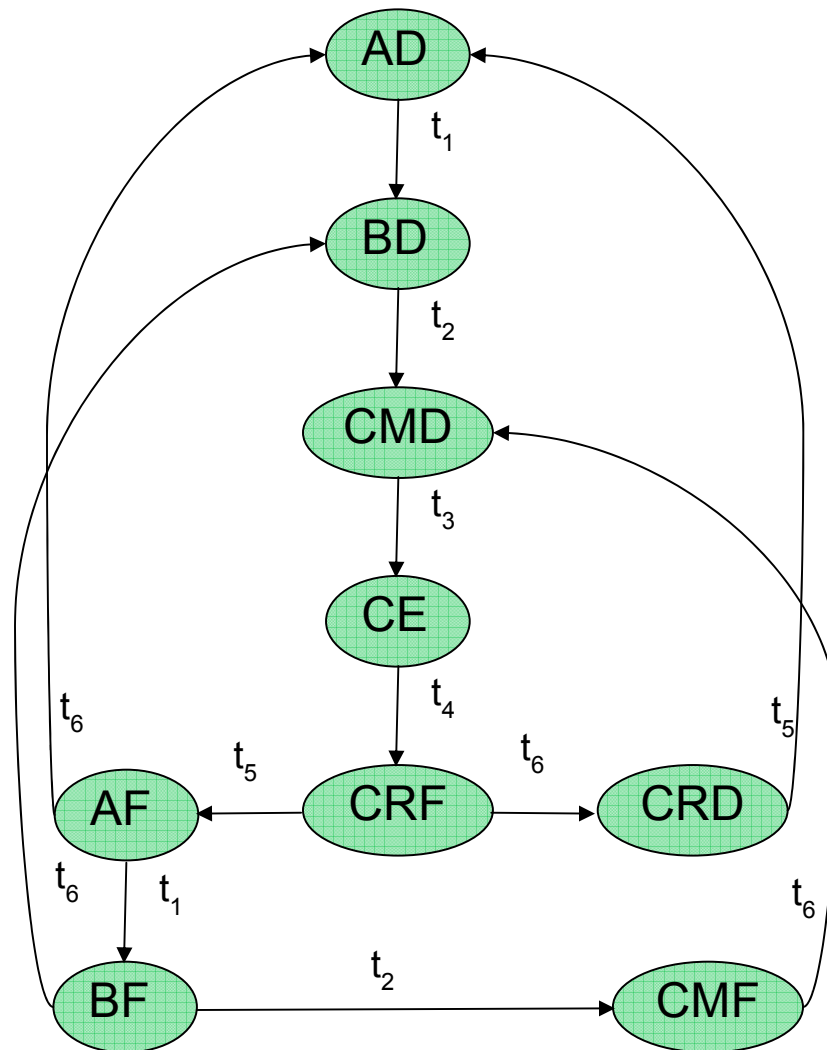


Validarea protocoalelor - metode

- Mașina de puncte.
- Arbori și grafuri de acoperire.
- Calculul invariantilor (model algebric).
- Reducerea modelelor.

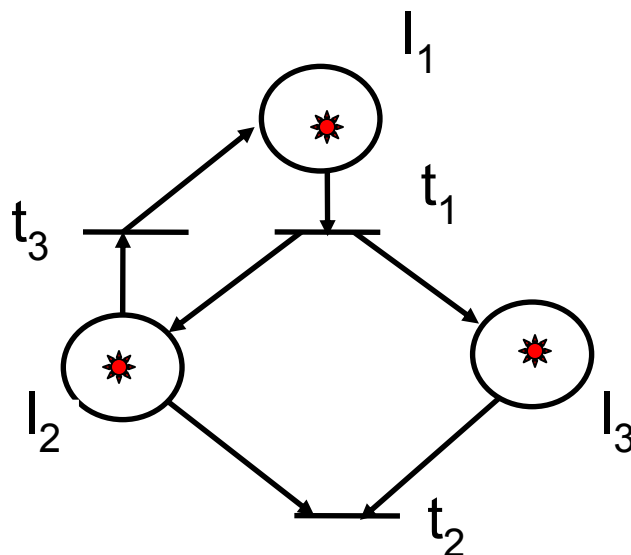
Mașina de puncte

- Proprietăți:
 - sigură;
 - viabilă;
 - home state;
 - **invarianti?**



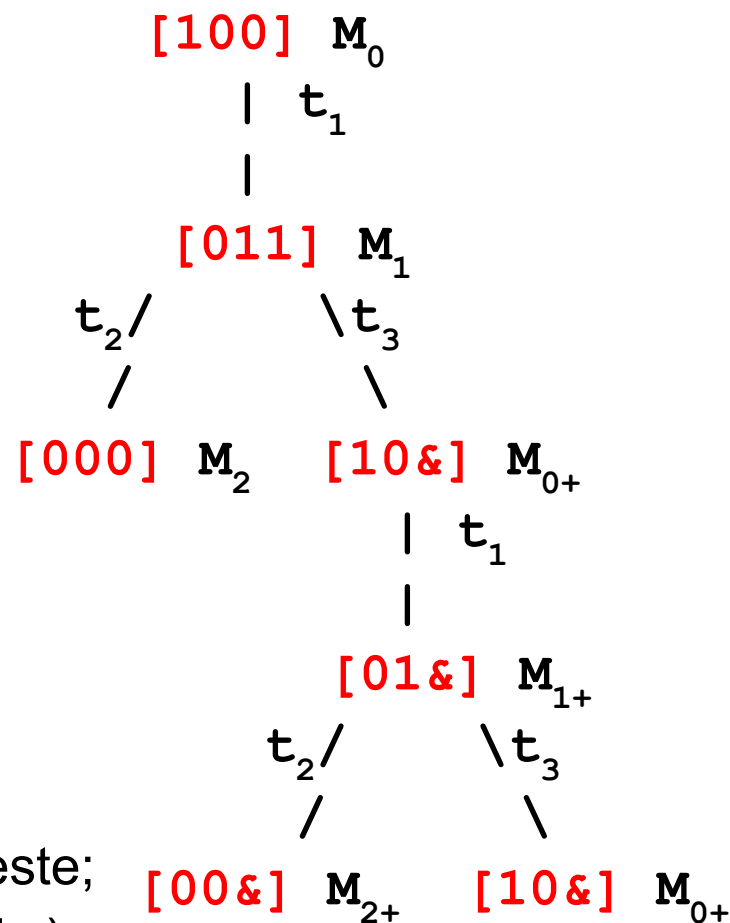
Arbori și grafuri de acoperire

- Analiza rețelelor nemarginite.



- Proprietăți:

- Locurile I_1 și I_2 sunt mărginite; I_3 nu este;
- Există o infinitate de blocări (M_2 și M_{2+});
- RP este cvasi-viabilă.





Construire arbore de acoperire

```
construire_arbore_acoperire()
{
    calculeaza succesoarele lui M0;
    for (fiecare succesor M)
        if (M>M0)
            marcheaza cu & fiecare componenta a lui M superioara componentei
            corespunzatoare din M0;
    while (exista un marcaj nou Mi, neconsiderat)
        if (nu exista pe calea de la M0 la Mi un marcaj Mj=Mi)
        {
            calculeaza succesoarele lui Mi;
            for (fiecare succesor Mk al lui Mi)
            {
                o componenta & a lui Mi ramine & in Mk;
                if (exista un marcaj Mj pe calea de la M0 la Mk cu Mk>Mj)
                    marcheaza cu & fiecare componenta din Mk superioara
                    componentei coresp. din Mj;
            }
        }
}
```



Analiza RP prin calculul invariantilor

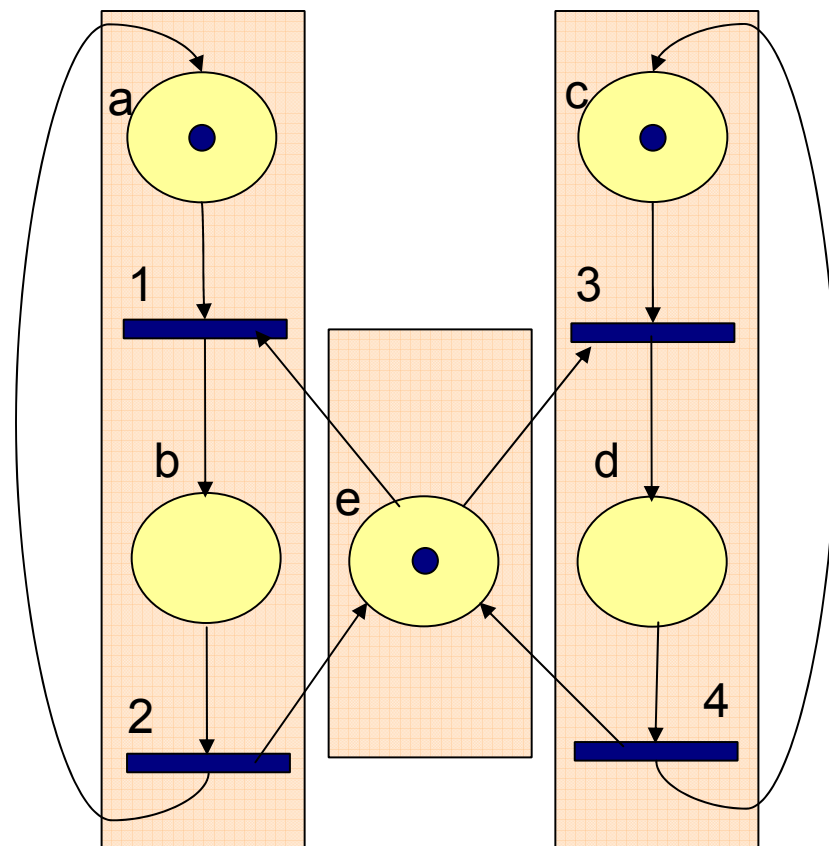
- Fie RP o rețea Petri **pură** (fără bucle), în care L și T sunt ordonate (arbitrar):
 - L: $l_1 < l_2 < \dots < l_m$,
 - T: $t_1 < t_2 < \dots < t_n$.
- Matricea $A : L \times T \rightarrow \mathbb{Z}$ cu $A[l_i, t_j] = O(t_j, l_i) - I(l_i, t_j)$ este **matricea de incidențe** a lui RP.
- Notăm:
 - $A[l_i, -]$ = linia l_i ;
 - $A[-, t_j]$ = coloana t_j .
- **L-vector** = o matrice coloană indexată după L.
- **T-vector** = o matrice coloană indexată după T.

Modelul excluderii mutuale

Matricea de incidențe:

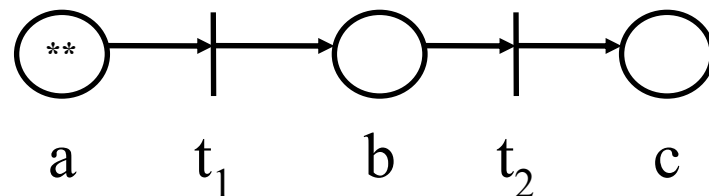
A		1	2	3	4
a		-1	1		
b		1	-1		
c				-1	1
d				1	-1
e		-1	1	-1	1

$$A[a,1] = O[a,1] - I[a,1] = 0 - 1 = -1$$



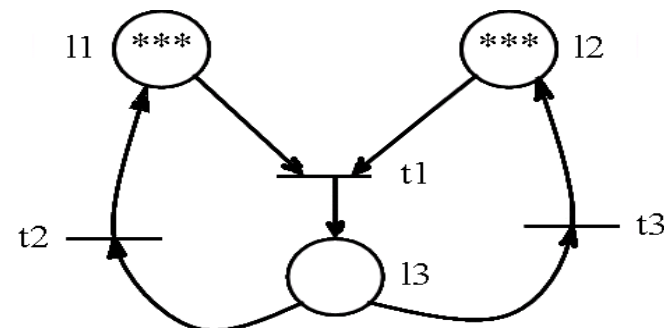
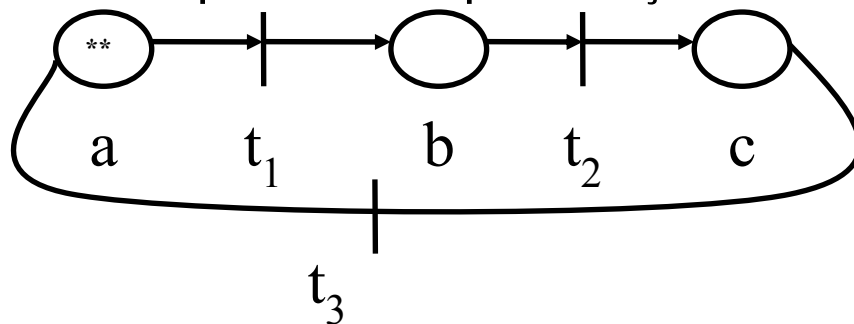
Aspecte de corectitudine

- Aspecte de corectitudine:
 - Garantare că nu se pierde puncte;
 - Posibilitate reproducere marcaje.



- Exemple:

- RP fără pierderi de puncte dar cu marcaj nereproductibil.
- RP fără pierderi de puncte și cu marcaj reproductibil;



- RP cu pierderi de puncte și cu marcaj nereproductibil.



L-invarianți

- Dacă M și M' au $M \xrightarrow{-t} M'$; rezulta
 - $M' = M + A[-, t]$;
- Pentru modelul excluderii mutuale avem **invariantul** :

$$M[a] + 2M[b] + M[c] + 2M[d] + M[e] = 3 \text{ (orice } M\text{)}.$$
- Pentru $g^T = [1, 2, 1, 2, 1]$ și M, M' reprezentați ca **M-vectori**
 - $g^T.M = g^T.M' = g^T.M + g^T.A[-, t]$.
- Rezultă:
 - $g^T.A[-, t] = 0$;
 - $g^T.A = 0$ (relația anterioară valabilă pentru orice t).
- **g este L-invariant.**
- Un l-vector l este un L-invariant $\Leftrightarrow l^T.A = 0$.
- Un L-invariant ne-negativ l se numește **minimal** $\Leftrightarrow$ nu există un l' a.i. $0 < l' < l$.

Exemplul excluderii mutuale (1)

U A						1	2	3	4
a		1	0	0	0	0	-1	1	
b		0	1	0	0	0	1	-1	
c		0	0	1	0	0		-1	1
d		0	0	0	1	0		1	-1
e		0	0	0	0	1	-1	1	-1
a+b		1	1	0	0	0	0	0	0
b+e		0	1	0	0	1	0	0	-1

Pentru $j=1$ se **adaugă**
liniile **a+b** și **b+e**

Exemplul excluderii mutuale (2)

U A						1	2	3	4

c		0	0	1	0	0		-1	1
d		0	0	0	1	0		1	-1
a+b		1	1	0	0	0			
b+e		0	1	0	0	1		-1	1

c+d		0	0	1	1	0			
d+b+e		0	1	0	1	1			

Pentru $j=3$ se **adaugă** liniile $c+d$ și $d+b+e$

Pentru $j=3$ se **adaugă** liniile **c+d** și **d+b+e**



Exemplul excluderii mutuale (3)

U A						1	2	3	4

d+b+e	0	1	0	1	1	0	0	0	0
c+d	0	0	1	1	0	0	0	0	0
a+b	1	1	0	0	0	0	0	0	0

$$I1 = \begin{array}{|c|} \hline 0 \\ \hline 1 \\ \hline 0 \\ \hline 1 \\ \hline _ 1 _ \\ \hline \end{array}$$

$$I2 = \begin{array}{|c|} \hline 0 \\ \hline 0 \\ \hline 1 \\ \hline 1 \\ \hline _ 0 _ \\ \hline \end{array}$$

$$I3 = \begin{array}{|c|} \hline 1 \\ \hline 1 \\ \hline 0 \\ \hline 0 \\ \hline _ 0 _ \\ \hline \end{array}$$



Calcul invarianți

```
calcul_invarianti()  
{  
    construiește matricea (U|A);  
    for (fiecare indice j al tranziției tj)  
    {  
        adauga la matricea (U|A) atâtea linii i câte  
        combinații lineare de câte două linii cu  
        coeficienți întregi pozitivi în care se anulează  
        elementul [i,j] există;  
        elimină din matricea (U|A) liniile i în care  
        elementul [i,j] este nenul.  
    }  
}
```



Folosire invarianți

- Regulă:
 - Dacă M este un marcaj și I un L-invariant atunci pentru orice M' accesibil din M :
 - $I^T.M' = I^T.M$.
- Utilizare:
 - Verificarea evitării anumitor marcaje:
 - Dacă există un invariant I a.î. $I^T.M' \neq I^T.M$ atunci M' nu poate fi accesibil din M .
 - Găsirea condițiilor necesare completării unui marcaj M' accesibil din M și cunoscut parțial;
 - Deducerea unor proprietăți generale ale marcajelor accesibile.

Exemplu pentru excluderea mutuală

- Invarianții găsiți sunt:

- $I_1 = [0 \ 1 \ 0 \ 1 \ 1];$
- $I_2 = [0 \ 0 \ 1 \ 1 \ 0];$
- $I_3 = [1 \ 1 \ 0 \ 0 \ 0];$
- $I = I_1 + I_2 + I_3 = [1 \ 2 \ 1 \ 2 \ 1]$ (invariantul **global**).

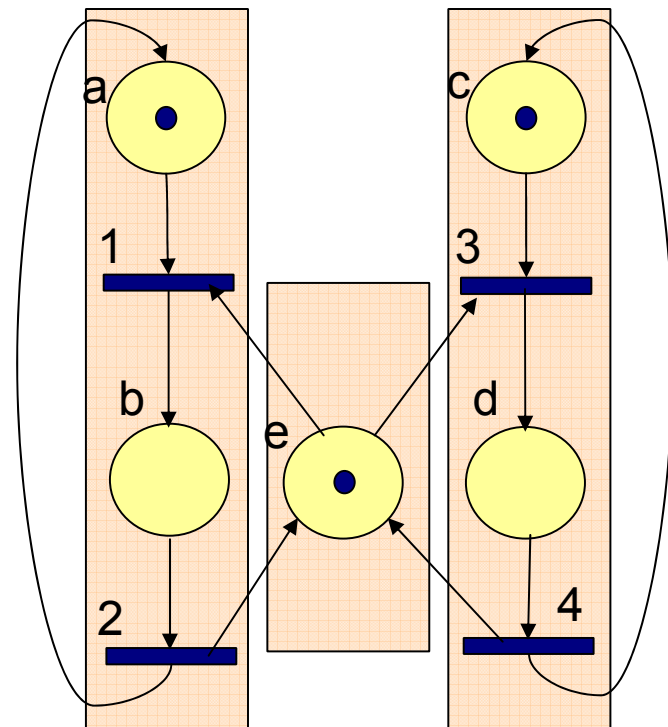
- Din $I^T \cdot M = I^T \cdot M_0$ obținem ptr invarianți:

- $M[b] + M[d] + M[e] = 1;$
- $M[c] + M[d] = 1;$
- $M[a] + M[b] = 1.$

- Relațiile exprimă:

- Condiția de excludere mutuală (prima relație);
- Siguranța: $M[l_i] \leq 1$ pentru orice l_i ;
- Rețea conservativă: din $g = I_1 + I_2 + I_3$ se obține:

$$M[a] + 2M[b] + M[c] + 2M[d] + M[e] = 3.$$



Reproducerea marcajelor

- Efectul tranziției 1, scris $M_0 + A[-,1] = M_1$ este echivalent cu:

$$M0 + A. \begin{bmatrix} \overline{1} & 1 & \overline{1} \\ | & 0 & | \\ | & 0 & | \\ | & 0 & | \end{bmatrix} = M1$$

- Efectul cumulat al tranzițiilor 1 și 2 poate fi scris:

$$M0 + A. \begin{bmatrix} \overline{1} & 1 & \overline{1} \\ | & 1 & | \\ | & 0 & | \\ | & 0 & | \end{bmatrix} = M0$$

T-invarianți

- T-vectorul J care reprezintă **numărul de execuții** ale tranzițiilor și este o soluție a ecuației $A.y = 0$ este **T-invariant**:

$$J = \begin{bmatrix} \bar{} & 1 & \bar{} \\ & 1 & \\ & 0 & \\ _ & 0 & _ \end{bmatrix}$$

- J este un T-invariant $\Leftrightarrow A.J = 0$.
- Un T-invariant ne-negativ J se numește minimal $\Leftrightarrow$ nu există J' a.î. $0 < J' < J$.
- Dacă J este un T-invariant atunci există un marcaj reproductibil prin execuția tranzițiilor în conformitate cu J .
- Pentru modelul excluderii mutuale, RP revine în marcajul inițial prin execuția tranzițiilor 1 și 2 (J_1) sau 3 și 4 (J_2).



Calculul T-invariantilor

- Din $x^T \cdot A = 0$ și $A \cdot y = 0$ (sau $y^T \cdot A^T = 0$) rezultă că:
 - **T-invariantii asociați lui A sunt L-invariantii lui A^T .**
- A^T este matricea de incidențe corespunzătoare **RP duale**.
- RP duală se obține astfel:
 - Fiecărui loc în RP îi corespunde o tranziție în RP duală;
 - Fiecărei tranziții în RP îi corespunde un loc în RP duală;
 - Fiecărui arc în RP îi corespunde un arc orientat în sens contrar în RP duală.



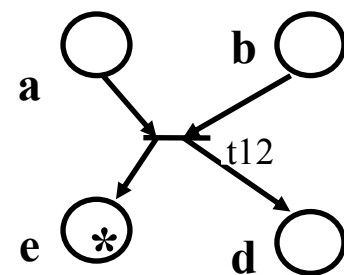
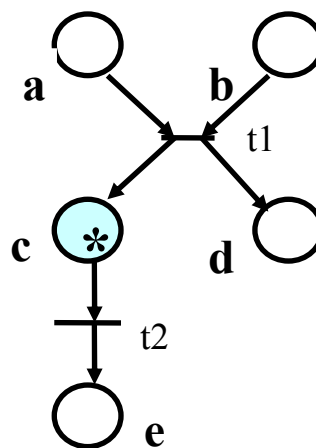
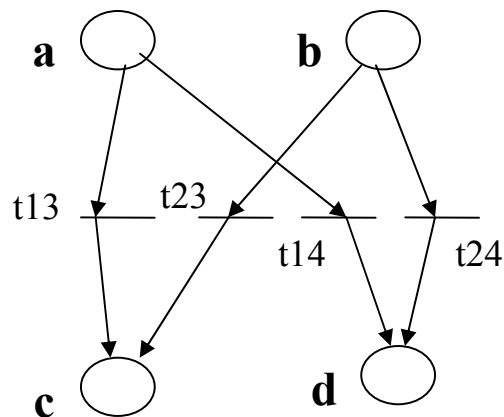
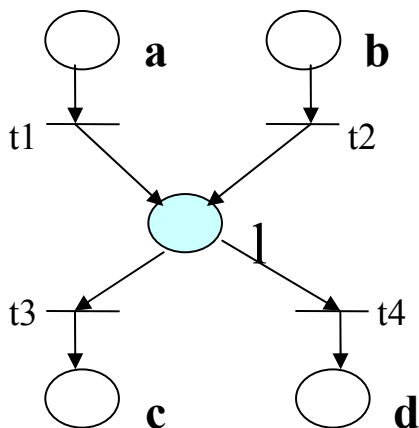
Reducerea RP

- Reducerea are ca scop scăderea dimensiunilor rețelei.
- Reducerea trebuie să păstreze unele din proprietățile rețelei.
- Reducere fără păstrarea invariantilor:
 - R1 (Reducerea unui **loc**);
 - R2 (Reducerea unui **loc implicit**);
 - R3 (Reducerea unei **tranziții neutre**);
 - R4 (Reducerea **tranzițiilor identice**).
- Reducere cu păstrarea invariantilor:
 - Ra (Reducerea unei **tranziții impure**);
 - Rb (Reducerea unei **tranziții pure**).

R1: Reducerea unui loc

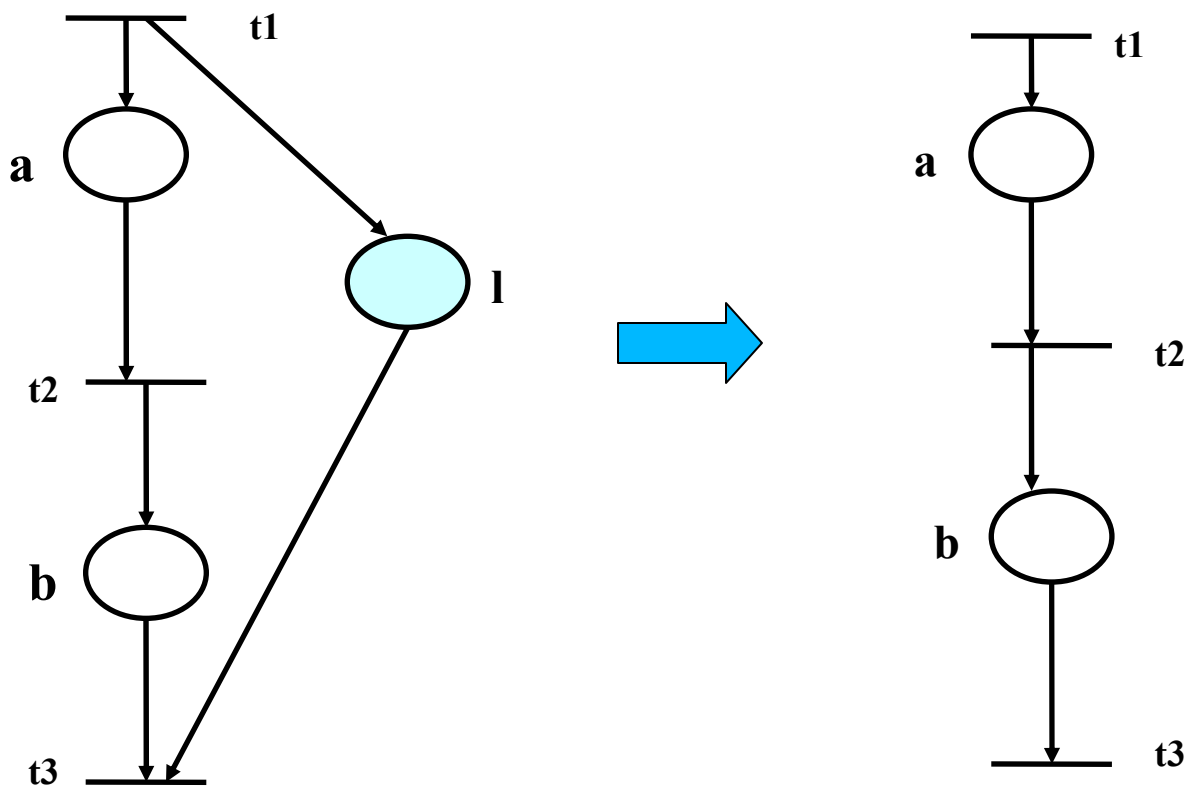
Eliminarea unui loc l:

- Dacă l are j intrări și k ieșiri, ele sunt înlocuite prin $j \cdot k$ tranziții, obținute prin contopirea unei tranziții de intrare cu una de ieșire;
- Ieșirile unei tranziții de intrare (ex. d - ieșirea lui t1) devin ieșiri ale tranziției obținută prin contopire (t12).



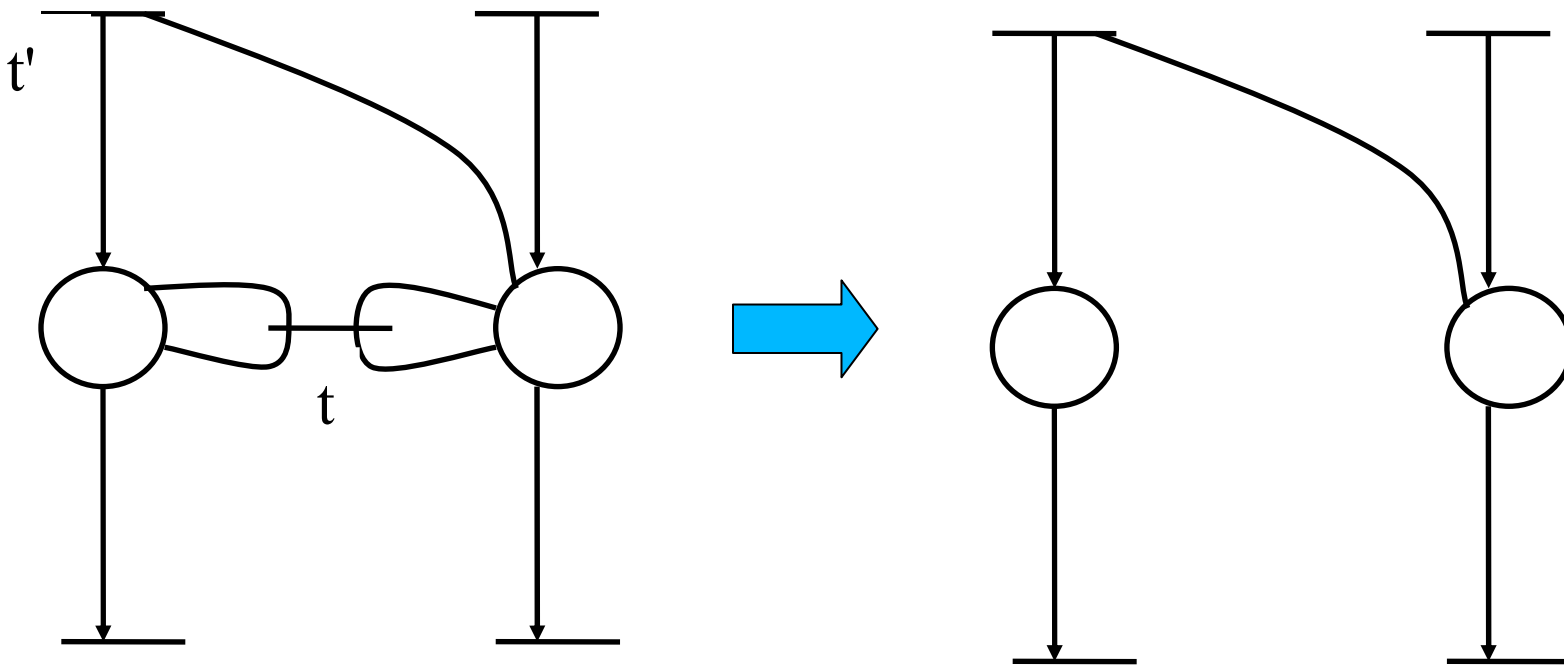
R2: Reducerea unui loc implicit

- Loc implicit:
 - Marcajul său permite întotdeauna execuția oricărei tranziții de ieșire, care ar fi executabilă dacă se ignoră l;
 - Marcajul său poate fi determinat din marcajul celorlalte locuri.



R3: Reducerea unei tranziții neutre

- Tranziția t este neutră $\Leftrightarrow \text{Pre}(t) = \text{Post}(t)$.
- Eliminare $\Leftrightarrow$ există $t' \neq t$ cu $O(t', l) \geq l(l, t)$ pentru orice l din $\text{Pre}(t)$.





R4: Reducerea tranziții identice

- Tranziții identice: au aceleași locuri de intrare și de ieșire.
- Dacă există n tranziții identice, se elimină $n-1$ din ele.

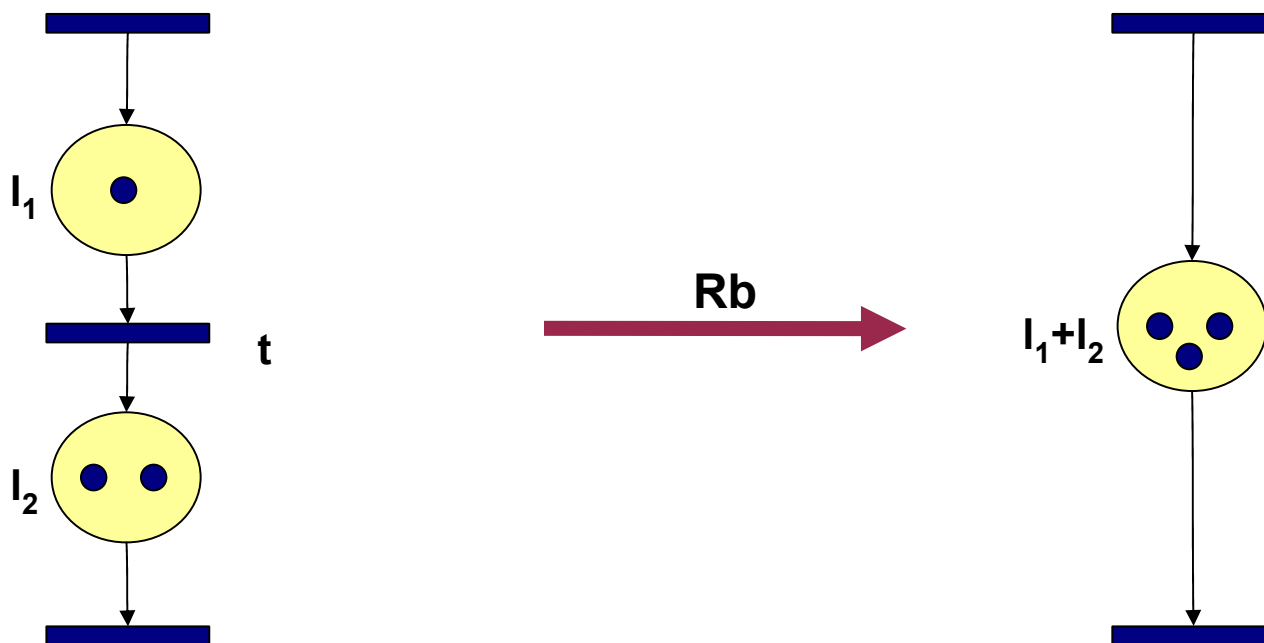
Ra: Reducerea unei tranziții impure

- O tranziție **impură** are locuri de intrare care sunt și de ieșire:
 - $\text{Pre}(t) \cap \text{Post}(t) \neq \emptyset$.



Rb: Reducerea unei tranziții pure

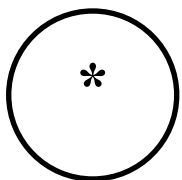
- Reducere tranziție pură:
 - Se elimină tranziția pură t ;
 - Fiecărui cuplu de locuri l_i din $\text{Pre}(t)$ și l_j din $\text{Post}(t)$ i se asociază un loc $l_i + l_j$ al cărui marcaj este $M(l_i) + M(l_j)$.



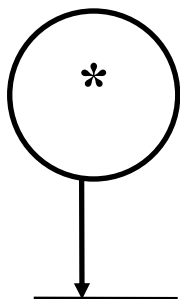
Cazuri ireductibile

- Rețea conservativă: $M(I1)+M(I2)=1$.

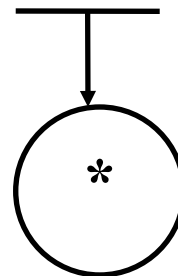
11 + 12



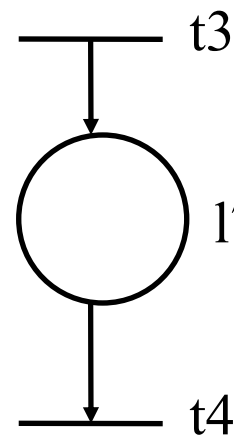
t1



t2



15 + 16

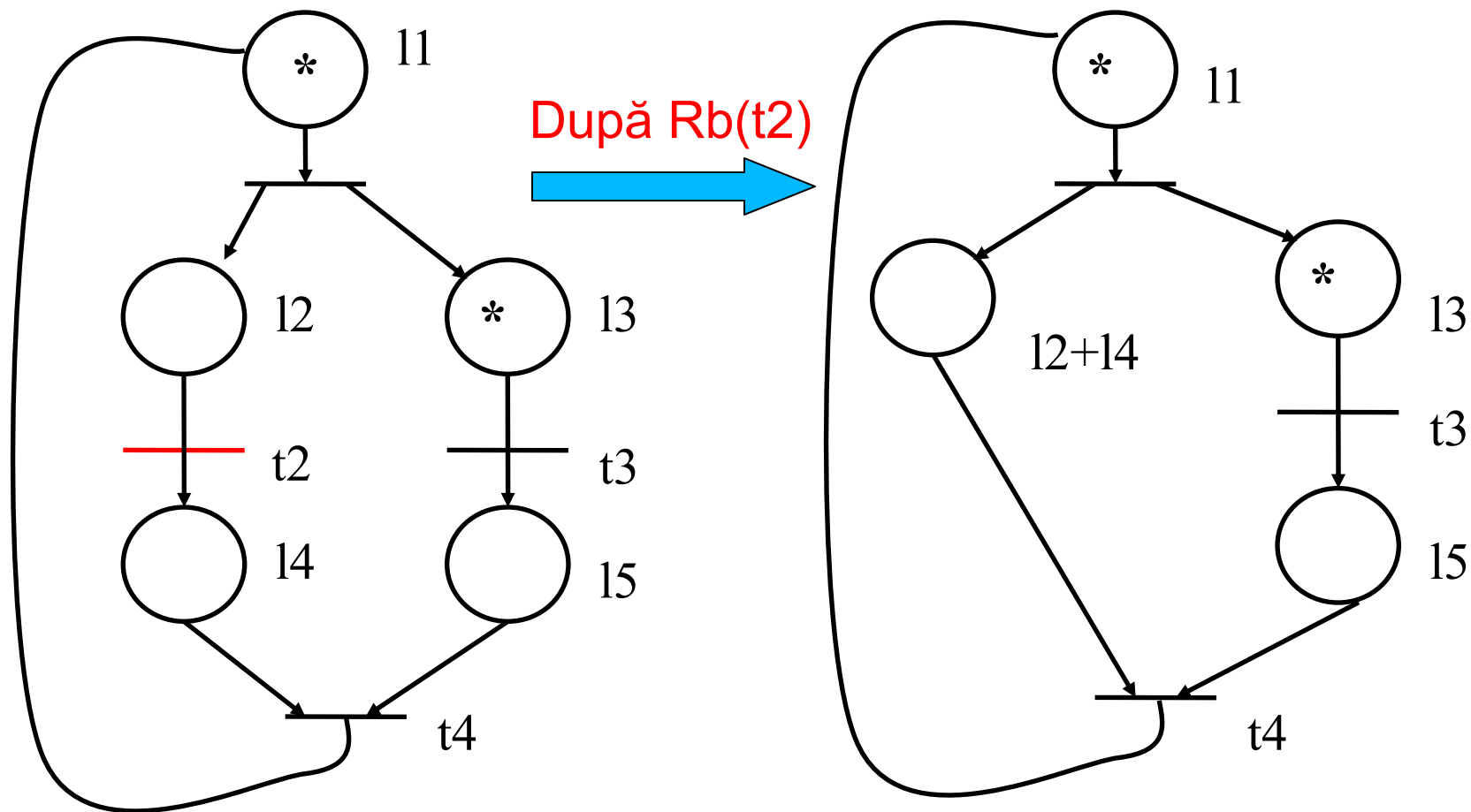


17 + 18

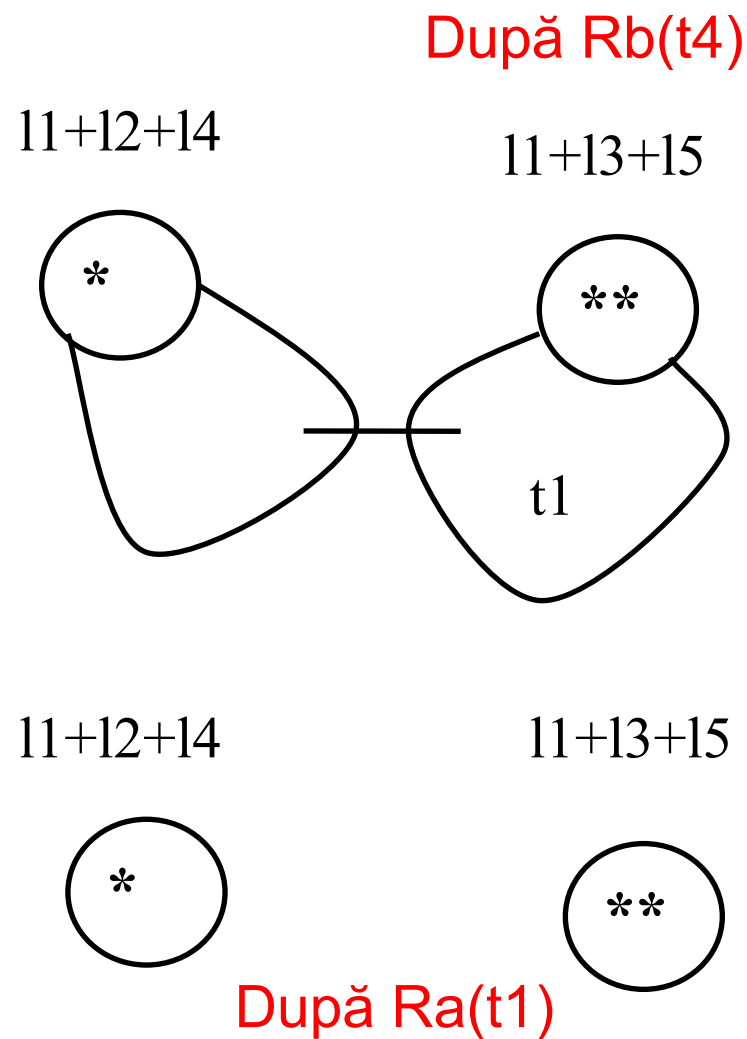
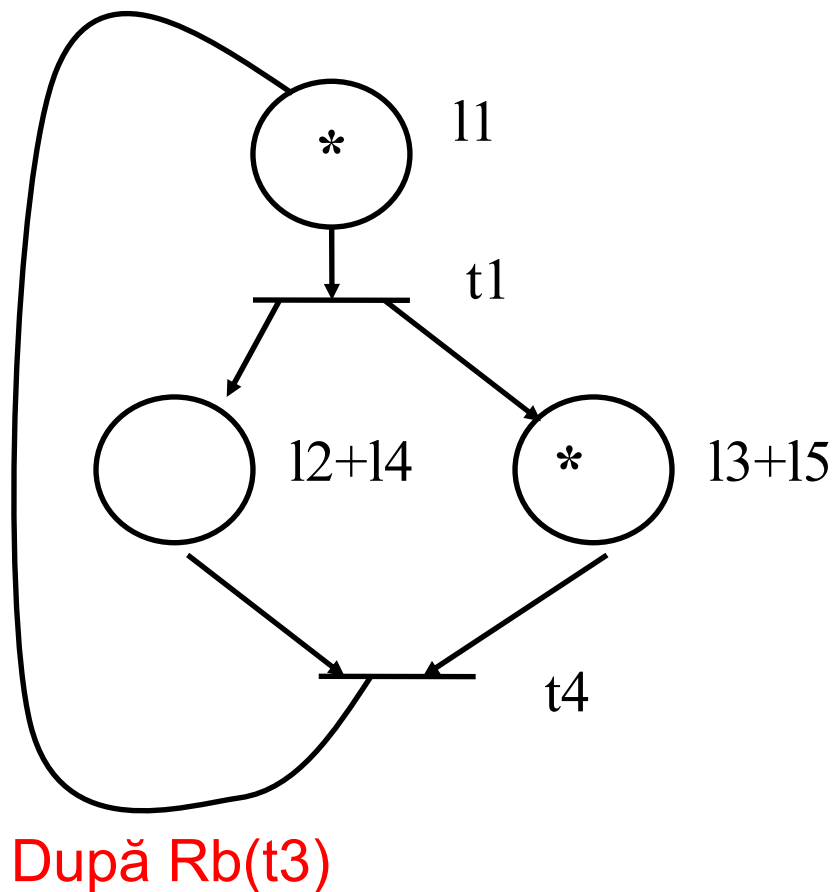
Proprietăți păstrate prin reducere

Reduceri	R1	R2	R3	R4	Ra	Rb
Proprietăți						
Mărginirea	X	X	X	X		
Siguranța	X		X	X		
Viabilitatea	X	X	X			
Cvasi-viabilitatea	X	X	X	X		
Evitarea blocării	X	X	X	X		
Starea de revenire	X	X	X	X		
Conservabilitatea	X	X	X	X		
Invarianti					X	X

Exemplu reduceri (1)

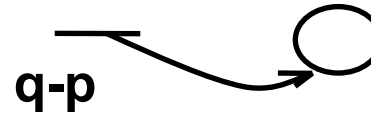
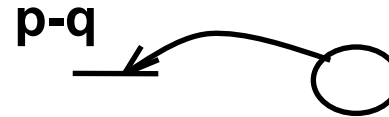
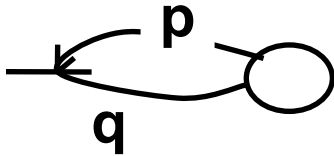


Exemplu reduceri (2)

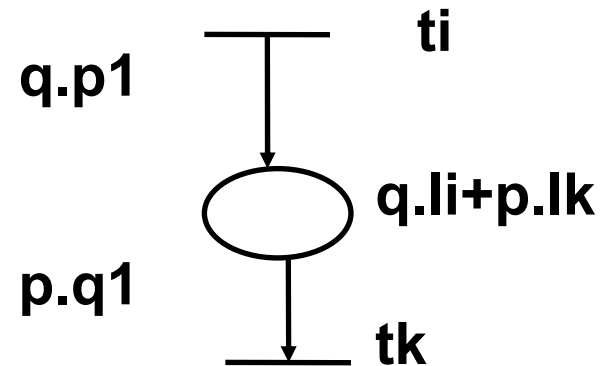
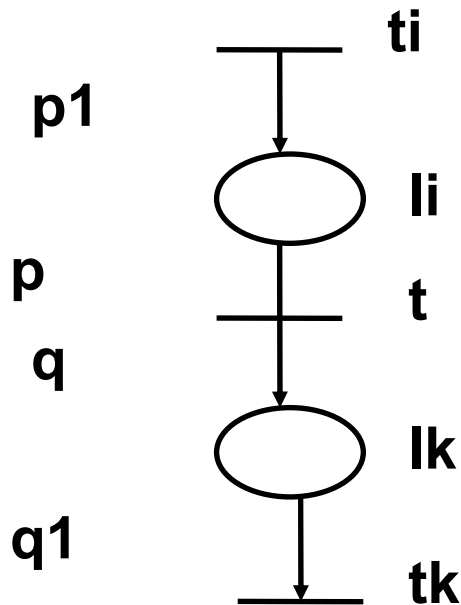


Reduceri pentru RP generalizate

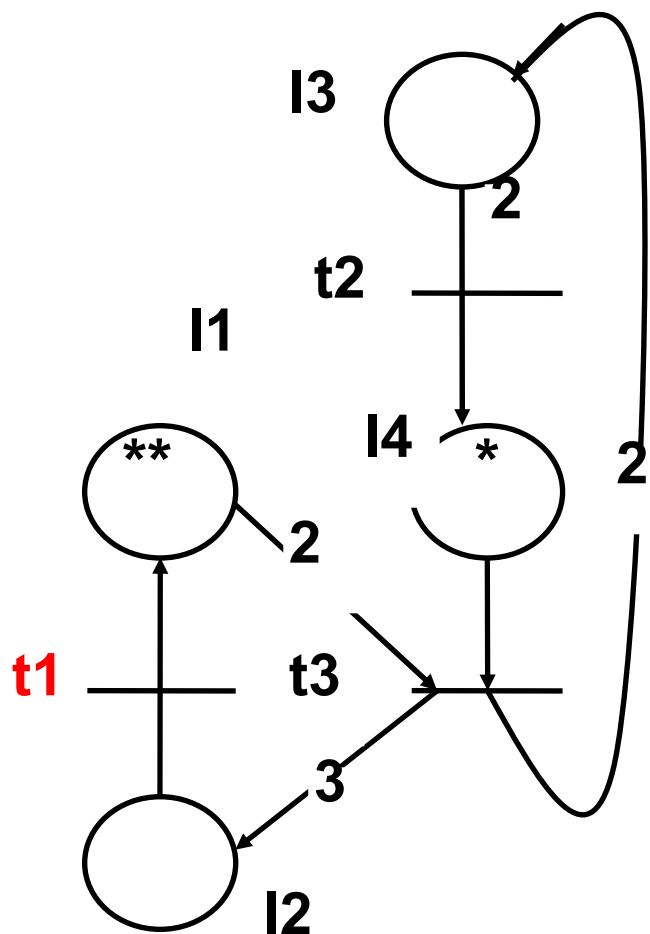
- Reducerea R'a (tranziție impură):



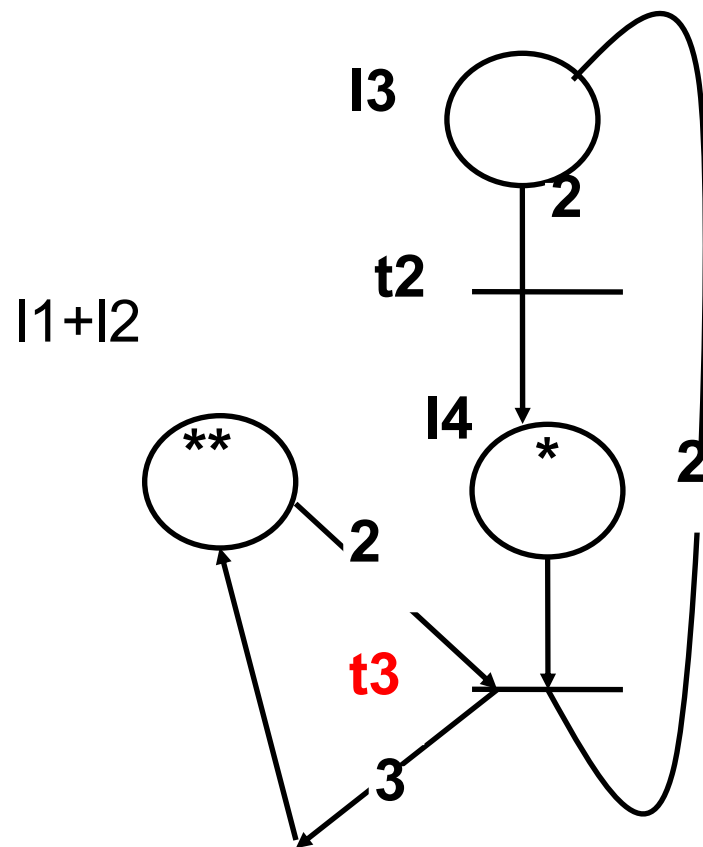
- Reducerea R'b (tranziție pură):



Exemplu reduceri RP generalizate (1)

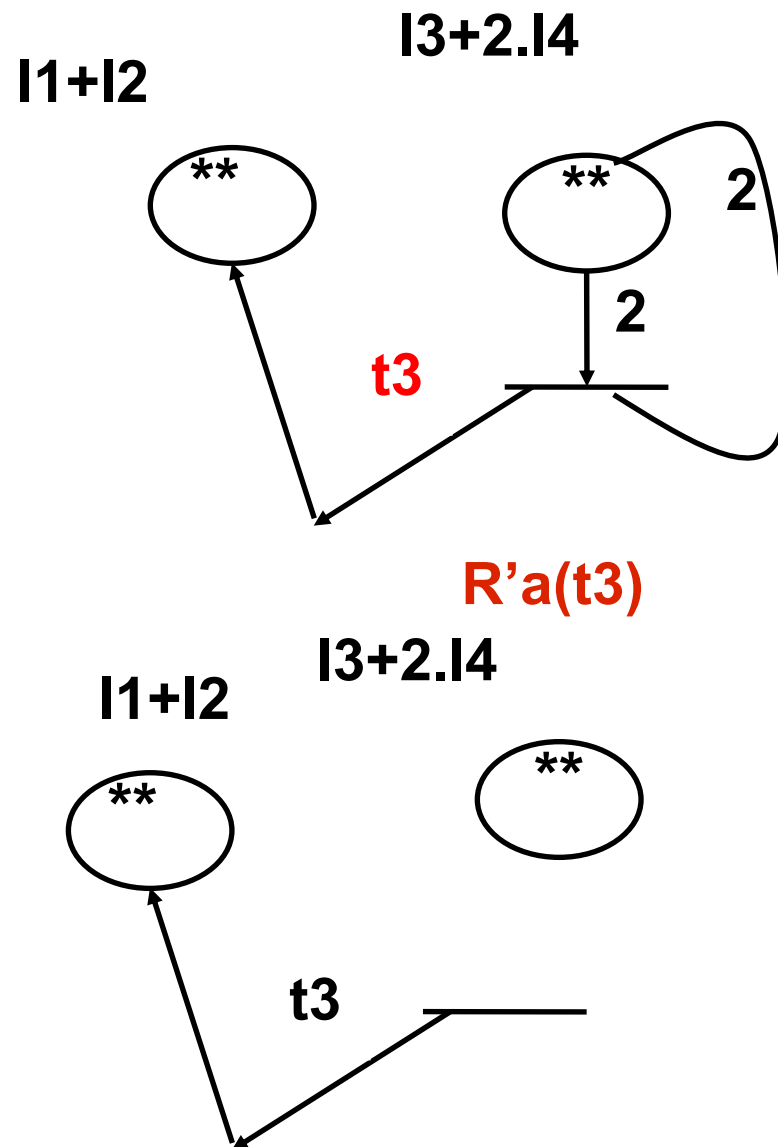
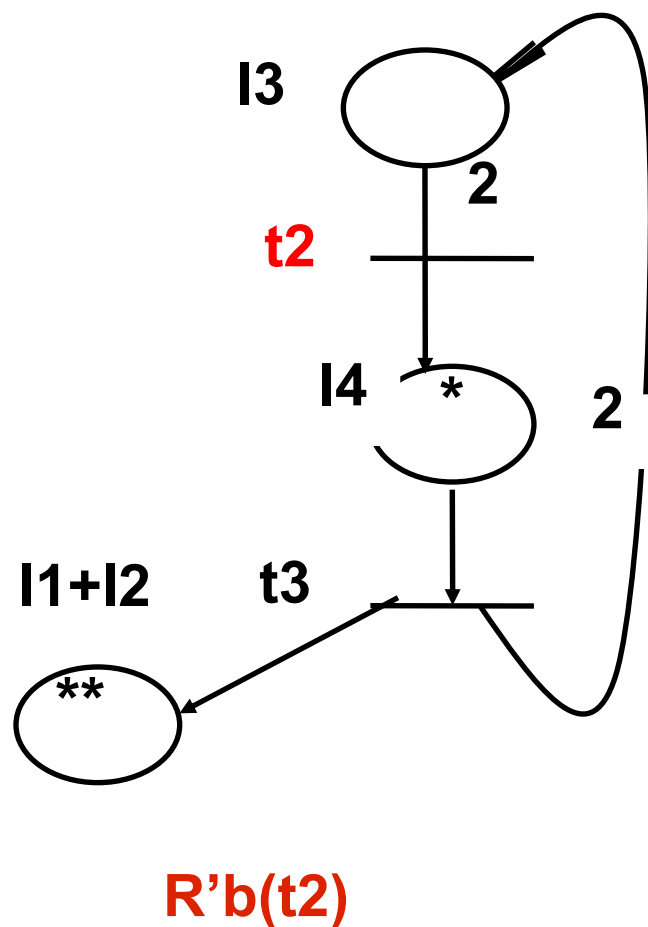


Se aplică $R'b(t1)$: $M(I1+I2) = 1*0+1*2$



$R'a(t3)$

Exemplu reduceri RP generalizate (2)





Sumar

- Verificarea protocoalelor se face înaintea implementării.
- Modele de specificare a protocoalelor:
 - Tranziționale;
 - Algoritmice;
 - Hibride.
- Rețele Petri: model tranzițional care poate fi analizat prin:
 - Mașina de puncte.
 - Arbori și grafuri de acoperire.
 - Calculul invariantilor (model algebric).
 - Reducerea modelelor.
- În cazul sistemelor complexe are loc explozia stărilor:
 - Analiza vizuală imposibilă;
 - Analiza automată prin programe (utilitare).