4.4. Metode de proiectare si verificare a protocoalelor

Metodele de proiectare a protocoalelor, la care ne referim in continuare permit descrierea protocoalelor in conformitate cu modelul de referinta pentru interconectarea sistemelor deschise. Ele sint aplicabile protocoalelor oricarui nivel al modelului arhitectural.

Fiecare nivel al modelului arhitectural realizeaza un anumit set de functii, folosind serviciile nivelului adiacent inferior. La rindul sau el furnizeaza un set bine definit de servicii accesibile nivelului adiacent superior. Din punctul acestuia de vedere, el poate fi vazut ca o cutie neagra, fiind interesante serviciile asigurate si nu modul in care aceste servicii sint realizate. Descrierea comportarii unui nivel fata de nivelul superior este cuprinsa in specificatia serviciilor. Functia unui nivel este distribuita entitatilor acestuia, care coopereaza intre ele conform protocolului nivelului. Structura stratului este reflectata de specificatia protocolului, care descrie functionarea fiecarei entitati ca raspuns la comenzile primite de la nivelul superior, la mesajele transmise de celelalte entitati similare si la actiunile initiate intern (de exemplu, trecerea anumitor intervale de timp). Specificatia protocolului reprezinta o rafinare a specificatiei serviciilor, ea precizind si modul in care serviciile sint realizate. Aceasta rafinare constituie ceea ce uzual se denumeste proiectarea unui protocol.

Specificarea unui protocol poate fi data in limbaj natural, dar si prin descrieri formale care inlatura ambiguitatile sau incompletitudinea frecventei in prima varianta.

Putem imparti modelele de specificare a protocoalelor in: tranzitionale, algoritmice si hibride. Modelele tranzitionale se bazeaza pe observatia ca protocoalele executa operatii simple, ca urmare a producerii unor evenimente (receptie de mesaje sau executii de primitive de serviciu). Comportarea entitatilor poate fi descrisa prin stari si tranzitii care modeleaza evenimentele si actiunile asociate. In aceasta categorie de modele pot fi incluse: diagramele de tranzitii, retelele Petri, grafurile UCLA, colocviile. Dezavantajul lor este explozia starilor in cazul unor protocoale complexe.

Comportarea entitatilor poate fi descrisa algoritmic. Modelele algoritmice folosesc limbaje de nivel inalt, realizind o descriere concisa si clara a functionarii entitatilor. Dezavantajul este apropierea de implementare a modelelor realizate, unele caracteristici importante ale protocoalelor fiind ascunse de amanuntele de implementare.

Modelele hibride imbina avantajos celelalte doua categorii. Un astfel de model este grefat pe un model tranzitional, tranzitiilor fiindu-le asociate module program, ale caror actiuni depind de si / sau modifica o serie de variabile numite variabile program sau variabile context.

Preocupari recente au condus la definirea unor limbaje de specificare, cum ar fi ESTELLE, precum si la adoptarea de catre ISO a limbajului LOTOS de specificare a protocoalelor.

Modelul formal al unui protocol obtinut in urma proiectarii poate fi validat, prin studiul diferitelor proprietati. Metodele de validare se impart in doua categorii: prima are la baza generarea si analiza spatiului starilor accesibile ale protocolului, iar cea de a doua foloseste demonstrarea corectitudinii programelor.

Pentru a combina avantajele celor doua abordari, au fost propuse si tehnici de validare hibride. O alta cale urmata este cea constructiva, in care se urmareste obtinerea unor protocoale corecte prin respectarea unor reguli de proiectare.

Modelul formal al unui protocol poate sta la baza diferitelor sale implementari. In functie de tipul modelelor utilizate, implementarea poate fi mai usoara sau mai dificila. In anumite cazuri, procesul de implementare poate fi automatizat. Desigur si in cazul implementarii se poate realiza validarea produselor obtinute, analizindu-se concordanta lor cu specificatia protocolului. Uzual aceasta se face prin testare sau simulare, care considera comportarea rezultatului implementarii pentru anumite cazuri reprezentative. Obtinerea datelor pentru testele de conformanta porneste chiar de la specificarea protocoalelor, existind preocupari pentru obtinerea lor, automat, prin reguli aplicate constructiilor limbajului de specificare.

4.4.1. Modele tranzitionale

Deoarece protocoalele guverneaza cooperarea intre entitati comunicante care executa operatii paralele asincrone, modelele cele mai potrivite pentru descrierea lor formala sint cele care expliciteaza mecanismele de conectare intre entitati. Dintre acestea, de o larga raspindire se bucura retelele Petri (RP), cu diferitele lor variante si extinderi. Concepute si folosite ca instrument general de specificare si analiza a sistemelor concurente, RP si-au gasit aplicarea fireasca in domeniul protocoalelor.

O retea Petri RP este un cvadruplu RP = (L, T, I, O), unde:

L este o multime finita de locuri (locatii),

T este o multime finita de tranzitii, L^T=0,

I este o functie I: LxT->{ 0, 1 }, numita functie de incidenta inainte,

O este o functie O: TxL->{ 0, 1 }, numita functie de incidenta inapoi.

Un marcaj M al une RP este o functie M:L->N, din multimea locurilor in multimea numerelor naturale N. Notam prin M(l), marcajul locului l.

O RP poate fi reprezentata ca un graf dirijat bipartit, locurile fiind reprezentate prin patrate (sau cercuri), iar tranzitiile prin bare. Nodurile sint conectate prin arce, fiecare arc fiind dirijat de la un loc la o tranzitie sau de la o tranzitie la un loc. Pentru o tranzitie t si un loc l, exista I (l, t) arce dirijate de la l la t si O (t, l) arce dirijate de la t la l. Marcajul se reprezinta dispunind in cercul fiecarei locatii un numar de puncte egal cu M(l). Un exemplu de reprezentare grafica a unei retele Petri este cel din figura 4.7.

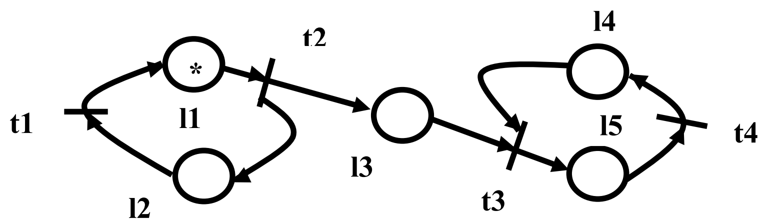


Fig.4.7. Reprezentarea grafica a unei retele Petri cu marcaj

Data fiind o retea Petri si o tranzitie t din T, multimea locurilor sale de intrare se defineste ca Pre(t) = { l in L | I(l, t)<>0 }. Similar, multimea locurilor de iesire este Post(t) = { l in L | O (t, l)<>0 }.

O tranzitie t a unei RP este executabila intr-un marcaj M daca si numai daca pentru orice locatie din Pre(t) este indeplinita relatia M(l) >= I(l, t). Deoarece I(l, t) = 0 pentru locuri neapartinind lui Pre(t), relatia precedenta se scrie M(l) >= I(l,t) pentru orice l din L.

Executia unei tranzitii t transforma marcajul M intr-un alt marcaj M' care indeplineste conditia M'(l) =M(l) - I(l, t) + O(t, l), pentru orice l din L. Notam aceasta executie prin M -t> M'. Executia micsoreaza marcajul fiecarei locatii de intrare cu o unitate si mareste marcajul fiecarei locatii de iesire cu o unitate.

Fie $ o secventa finita de tranzitii t_1 t_2 t_3... $t_{k.}$ Spunem ca $ este o secventa posibila de executii din marcajul M, daca si numai daca exista marcajele M_1, $M_{2,}$..., M_k astfel incit M[t_1>M_1 [t_2>M_2[... [t_k>$M_{k.}$ Se mai spune ca marcajul M_k este accesibil din M prin executia lui $ si se noteaza prin M [$> $M_{k.}$ Data fiind o RP si si un marcaj M_0, aspectele dinamice ale retelei sint puse in evidenta prin multimea secventelor posibile de executii din marcajul M_0 si de multimea marcajelor accesibile din $M_{0,}$ notate A($M_{0)}$. Clasa marcajelor accesibile poate fi reprezentata ca un graf ale carui noduri sint marcajele accesibile din M_0 si ale carui arce corespund executiei tranzitiilor. O astfel de reprezentare pune in evidenta relatia de precedenta asupra executiei actiunilor.

Retelele Petri reprezinta instrumente adecvate modelarii sistemelor paralele asincrone: tranzitile modeleaza evenimente care se pot produce in sistem, in timp ce locatiile modeleaza conditiile necesare producerii evenimentelor. Fiecare marcaj reprezinta o stare a sistemului modelat. Faptul ca intr-o stare anumite conditii sint indeplinite poate provoca producerea unor evenimente. Acestea pot schimba starea sistemului, modificindu-se astfel conditiile anterioare.

Pentru a ilustra procesul de modelare folosind RP, consideram un protocol simplu, care guverneaza transferul unor mesaje, cu confirmarea receptiei, intre o entitate emitatoare si o entitate receptoare. In vederea alcatuirii modelului protocolului, prezentam elementele semnificative referitoare la interactiunile entitatilor cu nivelele adiacente si la comportarea entitatilor (figura 4.8).

Functionarea entitatilor se bazeaza pe transmiterea unor mesaje m de la Emitator la Receptor si pe transmiterea unor confirmari de la Receptor la Transmitator, asigurate ca servicii de nivelul adiacent inferior. Nivelul superior poate solicita urmatoarele servicii: transmiterea unui mesaj (ct), insotita de mesajul ce trebuie transmis (msg) ; receptia unui mesaj (cr), avind ca raspuns mesajul receptionat (msg).

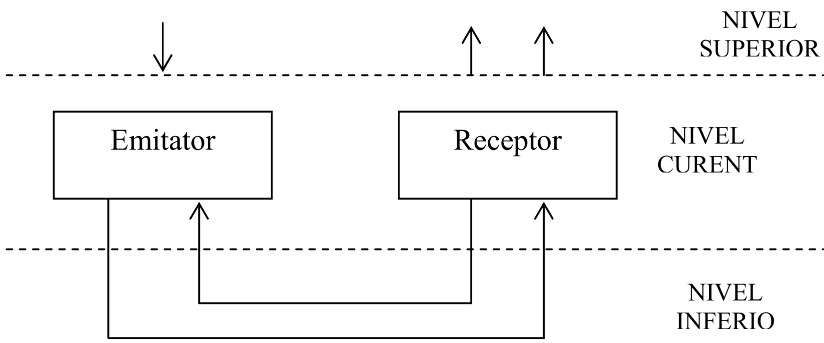


Figura 4.8. Arhitectura protocolului simplu

Operatiile executate de entitati sint urmatoarele:

```
Emitator()
{
do {
      A: asteapta_cerere_transmisie(ct, msg);
      B: pregateste_mesaj(msg, m);
      transmite_mesaj(m);
      C: asteapta_confirmare(r);
   } forever;
}

Receptor()
{
  do {
        D: asteapta_mesaj(m);
        E: pregateste_raspuns(m, r, msg);
        transmite_confirmare(r);
        F: asteapta_cerere_receptie(cr);
        transfera_mesaj(msg);
        } forever ;
}
```

Inainte de a trece la modelarea protocolului printr-o RP, sa observam ca din algoritmii prezentati se pot deduce cu usurinta doua automate, unul pentru sursa, celalalt pentru destinatie. Acestea sint prezentate in figura 4.9 in forma diagramelor de tranzitii, starile fiind corespunzatoare sectiunilor de algoritm cu acelasi nume (A, B etc).Pentru fiecare tranzitie sa dau intrarile si iesirile corespunzatoare, sub forma intrare / iesire.

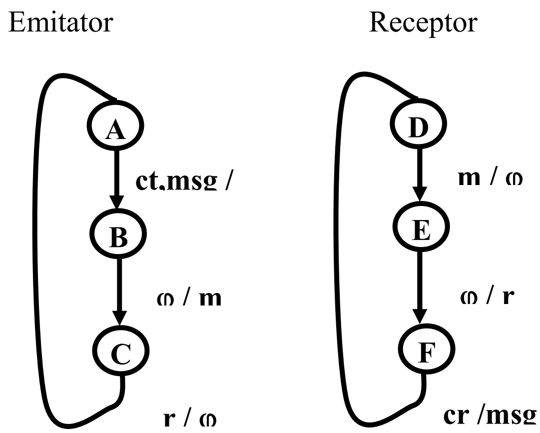


Figura 4.9. Modelarea entitatilor prin automate

In multimea intrarilor consideram si evenimente interne (@), care modeleaza actiuni ale entitatii cu efect doar asupra starii proprii. In multimea iesirilor, @ semnifica iesire nula. Desi aceasta reprezentare mentioneaza interactiunile dintre entitati prin legaturile care exista intre intrarile si iesirile lor, ea are dezavantajul ca nu prezinta in mod explicit aceste interactiuni, asa cum apar ele in modelele de retele Petri.

Pentru a alcatui RP corespunzatoare protocolului, completam modelele entitatilor cu modelul mediului de comunicare. Despre nivelul adiacent superior nu avem informatii ; considerind o comportare ideala, corecta a acestuia, putem neglija includerea lui in model. Ca urmare, RP va fi obtinuta prin conectarea modelelor entitatilor de protocol si a mediului de comunicare. Pentru fiecare conexiune a mediului distingem doua stari: libera (M') si cu mesaj in tranzit (M, figura 4.10).

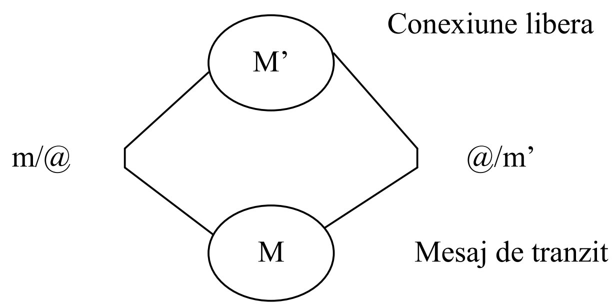


Fig.4.10. Modelarea mediului de comunicare

Trecerea de la M' la M modeleaza trecerea mesajului de la transmitator la mediu, iar tranzitia de la M la M' trecerea mesajului de la mediu la receptor. Tranzitia M->M' este determinata de terminarea propagarii mesajului de la un capat la celalalt al conexiunii.

In reteaua Petri corespunzatoare protocolului, starile componentelor devin noduri loc, iar tranzitiile devin noduri tranzitie. Tranzitiile a doua sau a mai multor componente care se produc simultan se contopesc, devenind un singur nod tranzitie.

De exemplu, pentru transmiterea mesajului m de la Transmitator la mediu, obtinem configuratia din figura 4.11.

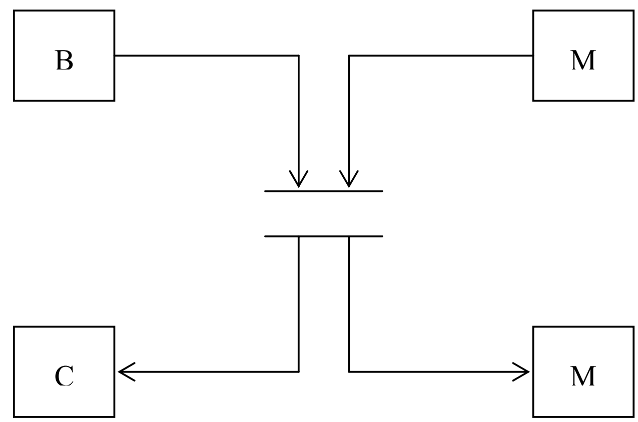


Figura 4.11. RP corespunzatoare transmiterii mesajului

Tinind cont ca locul M este suficient pentru reprezentarea celor doua stari ale conexiunii (prin marcaj 0 se poate codifica mediu liber, iar prin marcaj 1, mesaj in tranzit), putem renunta la locul M' in RP. Pentru intregul protocol, modelul obtinut este reprezentat in figura 4.12.

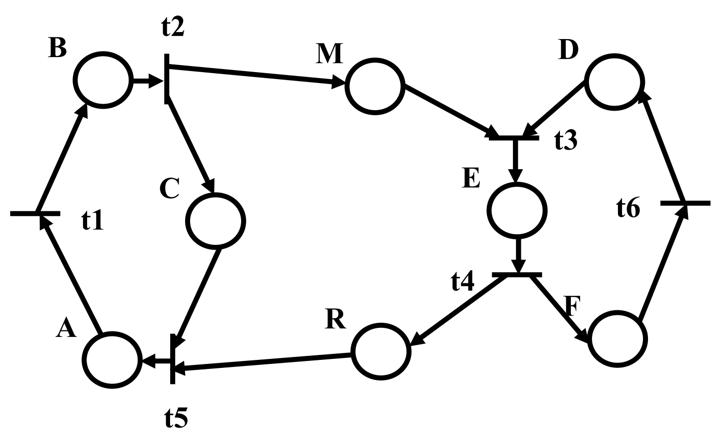


Figura 4.12.

Semnificatiile locurior sint aceleasi cu ale starilor automatelor din care provin. Similar, tranzitiile retelei capata semnificatiile tranzitiilor automatelor din care sint derivate, si anume:

t1 = preluare mesaj produs de utilizatorul transmitator,
t2 = transmitere de mesaj mediului de comunicare,
t3 = receptie mesaj de la mediu,
t4 = transmitere confirmare,
t5 = receptie confirmare,

t6 = consumare mesaj de utilizator receptor.

Modelul contine marcajul corespunzator unei stari initiale posibile a protocolului, in care entitatea emitatoare asteapta producerea unui mesaj (A), entitatea receptoare este pregatita pentru receptie (D), iar mediul de transmisie este gol. Convenim sa notam un marcaj oarecare al modelului prin colectia numelor locurilor care contin puncte, un nume aparind de atitea ori cite puncte contine locul respectiv (marcajul initial este AD).Multimea marcajelor si a tranzitiilor intre ele definesc masina de puncte a retelei, care se reprezinta grafic sub forma unei diagrame de tranzitii. Pentru reteaua Petri a protocolului simplu, masina de puncte este prezentata in figura 4.13.

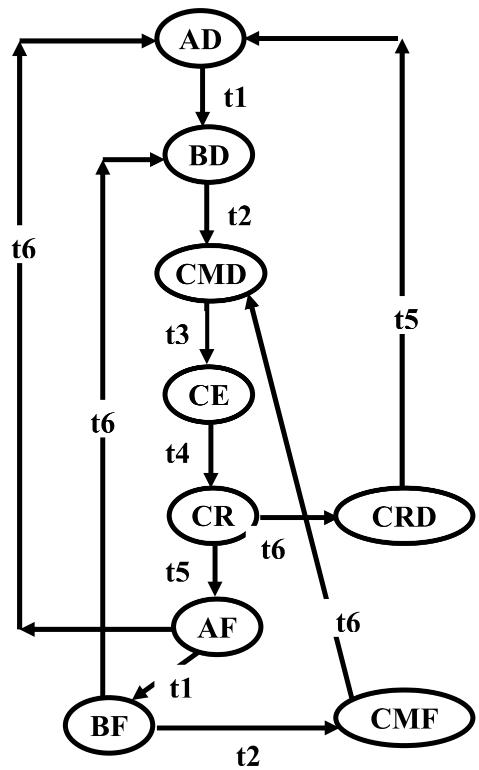


Figura 4.13.

Exemplul anterior reliefeaza citeva caracteristici importante ale retelelor Petri. Una este absenta oricaror masuri cantitative privind durata de mentinere a conditiilor sau timpul scurs intre doua evenimente oarecare. Singurul aspect important este ordinea partiala a evenimentelor. A doua caracteristica este nedeterminismul executiei tranzitiilor: daca la un moment dat sint mai multe tranzitii executabile, alegerea uneia din ele se face la intimplare. O asemenea comportare este acceptabila, deoarece executia unei tranzitii este considerata instantanee. De aici rezulta insa obligatia de a modela prin tranzitii doar actiuni primitive, indivizibile.

4.4.2. Caracteristici structurale ale RP

In cazul in care structura RP se supune anumitor reguli, se obtin retele cu proprietati interesante. Prezentam in continuare citeva RP particulare.

Graf de stari: orice tranzitie are exact un loc de intrare si un loc de iesire.

Graf de evenimente: orice loc are exact o tranzitie de intrare si una de iesire.

RP fara conflict: orice loc are cel mult o tranzitie de iesire.

RP cu alegere libera (free choice): exista doua sau mai multe tranzitii avind exact aceleasi locuri de intrare.

RP simpla: fiecare tranzitie este implicata in cel mult un conflict.

RP pura: nici un loc nu este intrare si iesire a aceleiasi tranzitii. Se demonstreaza ca orice RP impura poate fi transformata intr-o RP pura.

RP fara bucle: daca o tranzitie are un loc ce este simultan intrare si iesire, atunci ea mai are cel putin un alt loc de intrare.

Urmatoarele variante reprezinta generalizari ale RP.

RP generalizata: arcelor le sint asociate ponderi (numere naturale), deci functiile de incidenta inainte si inapoi sint definite prin I: LxT -> N si O: TxL -> N. Regulile de executie a tranzitiilor ramin nemodificate. Ca urmare, o tranzitie t este executabila daca marcajul fiecarui loc de intrare l este cel putin egal cu ponderea arcului (l, t). Executia tranzitiei este, ca si mai inainte, caracterizata de relatia M'(l) = M(l) - I(l, t) + O(t, l). Se arata ca orice RP generalizata poate fi transformata intr-o RP obisnuita.

RP cu capacitati: fiecarui loc ii este asociata o capacitate (numar natural), reprezentind valoarea maxim admisibila a marcajului locului. Executia unei tranzitii este permisa doar daca nu determina depasirea capacitatii vreunui loc. Se arata ca orice RP cu capacitati poate fi transformata intr-o retea obisnuita.

RP colorate: au marcaje carora le sint asociate culori. Orice RP colorata cu un numar finit de culori poate fi transformata intr-o RP obisnuita.

RP cu arce inhibitoare: in afara arcelor mentionate la RP obisnuite, are arce orientate de la o locatie la o tranzitie, avind efect daca marcajul locatiei este zero. In cazul general, o RP cu arce inhibitoare nu poate fi transformata intr-o retea obisnuita. Aceasta transformare este posibila doar in cazul RP marginite (marcajul oricarui loc nu depaseste niciodata o valoare finita, cunoscuta).

RP cu prioritati: asupra tranzitiilor este impusa o relatie de ordine partiala, luata in consideratie atunci cind mai multe tranzitii sint executabile, pentru selectia uneia dintre ele. Ea nu poate fi transformata intr-o RP obisnuita.

RP continua: marcajul unui loc este numar real, traversarea (executia) tranzitiilor facindu-se ca un flux continuu.

4.4.3. Validarea protocoalelor

Modelul unui protocol este util in validarea sa (verificarea corectitudinii), inainte de implementare. Prin validare se urmareste a se demonstra ca entitatile care coopereaza realizeaza serviciile specificate.

Retelele Petri permit atit analiza unor proprietati generale, pe care trebuie sa le aiba orice protocol, cit si a unor proprietati specifice, dependente de caracteristicile sistemului modelat. Prezentam mai intii citeva din proprietatile generale, dintre care marginirea si viabilitatea joaca un rol deosebit.

O RP este **marginita** de n pentru un marcaj initial M_0 daca si numai daca, pentru orice marcaj M accesibil din M_0 si orice locatie l din L, este satisfacuta relatia $M(l)<=n$. In cazul in care n = 1 spunem ca reteaua este **sigura.** Marginirea este o proprietate practica importanta, putind fi utilizata in depistarea depasirii capacitatii canalelor de comunicatie. In cazul protocolului simplu, reteaua Petri este sigura.

O tranzitie t din T a unei RP este **viabila** pentru un marcaj initial M_0 daca si numai daca, oricare ar fi marcajul M accesibil din M_0 exista o secventa de executii care contine pe t. O RP este viabila daca si numai daca fiecare tranzitie a retelei este viabila.

Viabilitatea poate fi utilizata in caracterizarea blocarii definitive a sistemelor modelate. Spunem ca un sistem prezinta o blocare definitiva daca doua sau mai multe componente ale sale isi conditioneaza reciproc actiunile, nemaiputind executa nici o operatie. Uzual, pentru a demonstra evitarea blocarii definitive, trebuie sa se arate ca anumite tranzitii "cheie" sint viabile. Pentru reteaua din figura 4.12, existenta a doua entitati ciclice sugereaza alegerea a doua tranzitii cheie, una pentru fiecare entitate (de exemplu t_1 si t_3). Daca una sau mai multe tranzitii nu sint viabile, functionarea sistemului modelat se poate degrada in timp. Atunci cind intregul sistem este blocat definitiv, nici una din tranzitii nu este executabila.

O tranzitie t din T a unei RP este **cvasiviabila** pentru un marcaj initial M_0, daca exista o secventa de executii din M_0 care contine t. O RP este cvasiviabila daca si numai daca toate tranzitiile sale sint cvasiviabile.

O RP are o stare de **revenire** (home) H, pentru un marcaj initial M_0, daca oricare ar fi marcajul M accesibil din M_0 exista o secventa de executii care conduce in H. O RP este **proprie** sau reinitializabila daca M_0 este stare de revenire.

In unele cazuri, un marcaj accesibil in care nici una dintre tranzitiile retelei nu este executabila poate fi considerat corect. El poate corespunde unei stari **terminale,** a carei atingere poate semnifica indeplinirea corecta a unei anumite functii a protocolului. Modelul protocolului simplu nu are stari terminale.

O proprietate importanta a RP este cea de **persistenta.** Deoarece aceasta este legata de notiunea de conflict, precizam ca un **conflict structural** corespunde unor tranzitii { $t_1, t_2, ..., t_k$ } cu un loc de intrare comun l. Conflictul structural devine **conflict efectiv** intr-un marcaj M daca M(l) este inferior numarului de tranzitii din { $t_1, t_2, ..., t_k$ }, care sint validate de M. De exemplu, variantele din figura 4.14 (a) si (b) sint conflicte efective, in timp ce variantele (c) si (d) nu sint.

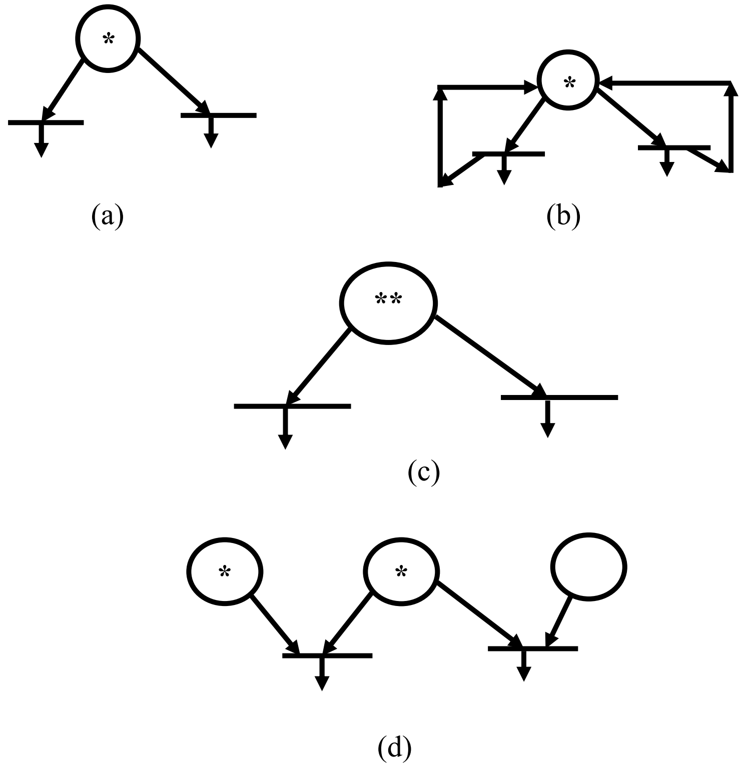


Figura 4.14.

Configuratiile (a) si (b) difera intre ele: in primul caz, executia lui t_1 face ca t_2 sa devina neexecutabila, in timp ce in al doilea caz tranzitiile pot fi executate in orice ordine, t_1, t_2 sau t_2, t_1. Spunem ca reteaua din (b) este persistenta. Mai precis, o RP este **persistenta** pentru un marcaj initial M_0, daca pentru orice marcaj M accesibil din M_0, in care t_j si t_k sint executabile atunci t_j, t_k si, prin simetrie t_k, t_j sint secvente posibile de executii din M. Aceasta proprietate arata ca, intr-o retea persistenta, alegerea uneia din tranzitiile aflate eventual in conflict se poate face oricum.

In privinta analizei proprietatilor specifice, tehnica cea mai cunoscuta este cea a **invariantilor.** Ideea de baza este definirea unor predicate sau asertiuni care ramin invariante pentru orice stare a sistemului modelat. In cazul retelelor Petri, se utilizeaza atit invarianti asupra locurilor cit si invarianti asupra tranzitiilor. Pentru reteaua Petri a protocolului simplu, un exemplu din prima categorie este: M(A) + M(B) + M (C) = 1, care subliniaza ca, la un moment dat, entitatea transmitatoare se afla intr-una din starile A, B sau C.

In general, data fiind o RP cu L = { $l_1, l_2, ..., lr$ } si un marcaj initial M_0 exista un invariant linear de locuri, sau **l-invariant,** daca se gaseste un ansamblu de numere naturale (ponderi) $p_1, p_2, ..., p_r$, nu toate nule, astfel ca:

$p_1.M(l_{1)} + p_2.M(l_{2)} +...+ p_r. M(l_{r)} = const,$

pentru orice M accesibil din M_0. Multimea L' inclusa in L, a locurilor l_i pentru care $p_i>0$ este o **componenta conservativa.** Reteaua Petri se numeste conservativa daca L este o componenta conservativa. Reteaua corespunzatoare protocolului simplu este conservativa si avem $2M(A) + 2M(B) + M(C) + M(D) + 2M(E) + M(F) +M(M) + M(R) = 3$ pentru orice M accesibil din marcajul M_0 dat in figura 4.12.

De regula, o componenta conservativa are o semnificatie fizica: ea corespunde unei entitati aflate intr-o singura stare la un moment dat, sau unui ansamblu de entitati al caror numar se conserva.

Notind cu N(t) numarul de executii, la un moment dat ale tranzitiei t, un exemplu de invariant din a doua categorie este $N(t_3) - N(t_{4)}>=0$, care arata ca niciodata numarul de confirmari nu poate depasi numarul de mesaje transmise. Totodata $N(t_{3)} - N(t_{4)}<=1$ ceea ce inseamna ca avansul sincron al lui t_3 fata de t_4 nu poate depasi 1. Avansurile sincrone maximale constituie invarianti relativi la executia tranzitiilor.

Secventa de executii t_1 t_2 t_3 t_4 t_5 t_6 din M_0 readuce reteaua in marcajul initial. Ea poate fi deci repetata, motiv pentru care se numeste **secventa repetitiva.** Mai mult, ea este o secventa repetitiva completa, deoarece contine toate tranzitiile.

Inainte de a caracteriza ansamblul secventelor posibile de executii, sa amintim elementele utilizate in constructia expresiilor regulate pe alfabetul $T = \{ t_{1,}..., t_m \}$ al simbolurilor asociate tranzitiilor unei RP:

- t_i este o expresie legala, semnificind executia tranzitiei t_i ;

- daca $1 si $2 sint expresii legale, atunci $1 $2 este de asemenea legala si semnifica executia secventei $1 urmata de executia secventei $2 (concatenarea) ;

- daca $1 si $2 sint expresii legale atunci $1+$2 este de asemenea legala si semnifica executia secventei $1 sau (exclusiv) a secventei $2 (alternativa) ;

- daca $ este o secventa legala atunci [$] este de asemenea legala si arata ca $ este optionala (se executa 0 data sau de zero ori) ;

- daca $ este o expresie legala atunci { $ } este de asemenea legala si semnifica repetarea indefinita a secventei $ (de 0, 1, 2,... ori).

Concatenarea este prioritara fata de alternativa. Parantezele rotunde pot fi utilizate pentru gruparea secventelor si au prioritate maxima, ca [] si {}.

Secventa $1 este un prefix al lui $2 daca exista $3 astfel incit $2 = $1 $3. Expresia $\{ t_1 + t_2 +... + t_m \}$ contine toate secventele care pot fi construite pornind de la alfabetul $T = \{ t_{1,} t_2,..., t_m \}$.

Fie $ o secventa repetitiva care contine tranzitiile din multimea T' inclusa in T. Multimea tranzitiilor T' este o componenta repetitiva. Reteaua Petri este repetitiva daca T este o componenta repetitiva. RP corespunzatoare protocolului simplu este repetitiva. Secventele repetitive caracterizeaza comportarea ciclica a unui model.

4.4.4. Arbori si grafuri de acoperire

Asa cum s-a aratat anterior, generarea masinii de puncte permite gasirea proprietatilor unei RP. Masina de puncte nu poate fi alcatuita in cazul unei retele nemarginite. In aceasta situatie, unele informatii despre comportarea modelelor pot fi obtinute din arborele de acoperire si din graful de acoperire asociate RP, care au intotdeauna un numar finit de noduri. Constructia este ilustrata de exemplul din figura 4.15.

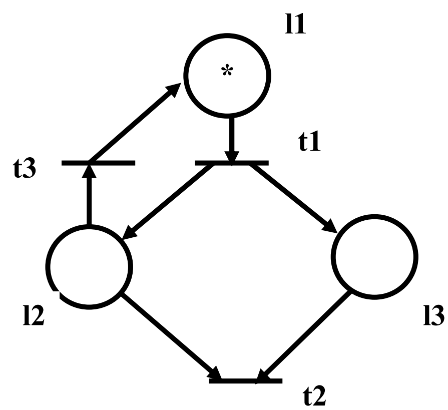


Figura 4.15 (a) Retea Petri

```
        [100] M0
          | t1
          |
        [011] M1
     t2/      \t3
      /        \
[000] M2   [10&] M0+
                | t1
                |
            [01&] M1+
          t2/     \t3
           /       \
      [00&] M2+  [10&] M0+
```

Figura 4.15 (b) arbore de acoperire

```
      [100] M0
           | t1
           |
        [011] M1
     t2/     \t3
      /        \
[000] M2   [10&] M0+
             t1 | ^
                v | t3
             [01&] M1+
           t2/
            /
      [00&] M2+
```

Figura 4.15 (c) graf de acoperire

Fiecare nod al arborelui de acoperire este un marcaj. Considerind locatiile din L in ordinea $l_1, l_2, \ldots$ putem reprezenta un marcaj oarecare M printr-un vector $[M(l_1)\ M(l_2) \ldots]$. Pentru alcatuirea arborelui, se calculeaza succesoarele fiecarui marcaj, care se adauga arborelui. Operatia se executa o data pentru fiecare marcaj distinct.

Pornind din marcajul initial M_0 = [100], prin executia lui t_1 se obtine M_1 = [011], de unde prin t_2 se obtine M_2 = [000], iar prin t_3 se obtine M3 = [101]. Deoarece $M_3 > M_0$, secventa t_1 t_3 se poate repeta indefinit, $M(l_3)$ crescind nelimitat. Asociind acestui marcaj simbolul & echivalent cu o valoare oricit de mare, se obtine o notatie concisa, in care o infinitate de marcaje posibile sint inlocuite cu un singur "pseudomarcaj", M_0+=[10&]. Pentru marcajele urmatoare, notatia & se va pastra, deoarece modificarea marcajului (cresterea sau scaderea lui cu un numar finit) lasa in locul respectiv tot o valoare oricit de mare. Pentru obtinerea grafului de acoperire, se contopesc nodurile arborelui de acoperire care corespund aceluiasi marcaj. Algoritmul de construire a arborelui de acoperire este urmatorul:

construire_arbore_acoperire() {
calculeaza succesoarele lui M_0;
for (fiecare succesor M)
    if ($M>M_0$) marcheaza cu & fiecare componenta a lui M
    superioara componentei corespunzatoare din M_0;
    while (exista un marcaj nou M_i, neconsiderat)
        if (nu exista pe calea de la M_0 la M_i un marcaj $M_j=M_i$)
        { calculeaza succesoarele lui M_i;
        for (fiecare succesor M_k al lui M_i)
            { o componenta & a lui M_i ramine & in M_k;
            if (exista un marcaj M_j pe calea de la M_0 la
            Mk cu $M_k>M_j$)
                marcheaza cu & fiecare componenta din M_k
                superioara componentei coresp. din M_j;
            }
        }
}

Pornind de la arborele de acoperire sau de la graful de acoperire, se pot obtine informatii despre model: locurile l_1 si l_2 sint marginite, in timp ce l_3 nu este ; exista o infinitate de blocari, corespunzatoare marcajelor M_2 si M_2+ ; RP este cvasiviabila ; ansamblul de secvente de executii corespunde limbajului
$L = t_1\{ t_3 t_1\}t_2.$

4.4.5. Analiza retelelor Petri prin calculul invariantilor.

Fie RP = (L, T, I, O) o retea Petri. RP se numeste pura daca pentru orice (l, t) din LxT cu I(l, t)<>0 avem O(t, l) = 0.

Fie RP o retea Petri pura, in care se considera ca multimile L si T sint ordonate (arbitrar):

L: $l_1 < l_2 < ... < l_m$, m fiind cardinalul lui L,

T: $t_1 < t_2 <... < t_n$, n fiind cardinalul lui T.

Matricea A: LxT > Z indexata dupa L si T cu

$A [l_i, t_j] = O (t_j, l_i) - I (l_i, t_j)$

se numeste matricea de incidente a lui RP. Linia l_i si coloana t_j ale lui A se noteaza prin A [li, -], respectiv A [-, tj].

Exemplu: modelul excluderii mutuale (vezi figura 4.16).

Matricea de incidente este urmatoarea:

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

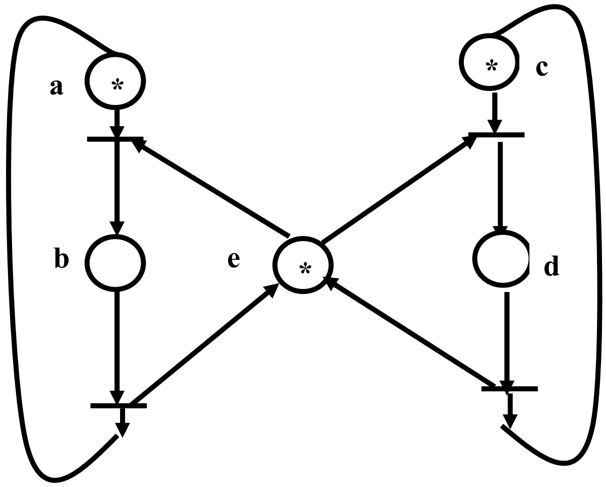


Figura 4.16.

A[a, 1] = -1 si A[b, 1] = 1 arata ca la executia tranzitiei 1 se ia un punct din a si se adauga un punct la b. Astfel, efectul executiei unei tranzitii este inregistrat in intregime in coloana corespunzatoare a matricei A.

O matrice coloana indexata dupa L se numeste L-vector. Similar, o matrice coloana indexata dupa T se numeste T-vector. Marcajul unei retele Petri poate fi reprezentat ca un L-vector, iar efectul tranzitiei t din M in M' se poate exprima prin relatia:

M' = M + A[-, t].

Comportarea dinamica a RP depinde de structura retelei si de marcajul initial. Influenta structurii retelei asupra comportarii prezinta un interes deosebit ea fiind aceeasi pentru orice marcaj initial si, in plus, putind fi investigata independent de orice proces dinamic.

Pentru modelarea sistemelor dinamice prin retele Petri prezinta interes urmatoarele doua aspecte:

- daca este garantat ca nu se pierd puncte (cel putin nu intr-o maniera necontrolabila) ;

- daca este posibila reproducerea marcajelor.

Exemple.

a) RP fara pierdere de puncte dar cu marcaj nereproductibil (figura 4.17).

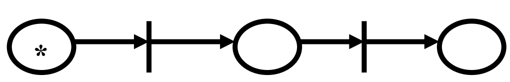


Figura 4.17.

b) RP fara pierdere de puncte cu marcaje reproductibile (figura 4.18).

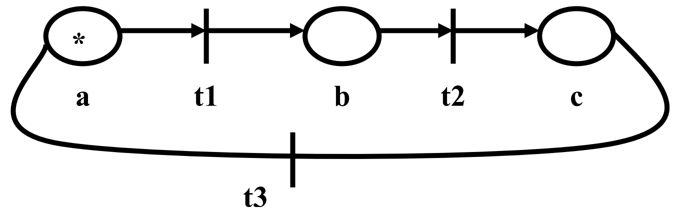


Figura 4.18.

c) RP cu pierdere de puncte si marcaje nereproductibile (figura 4.19).

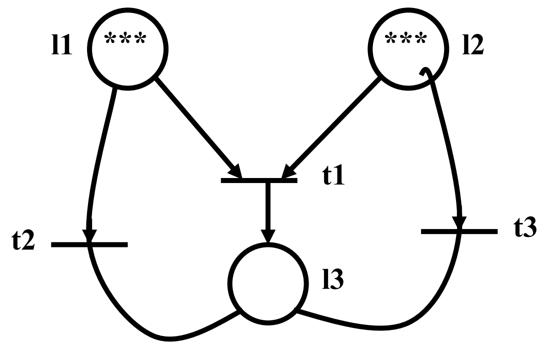


Figura 4.19.

In primul exemplu, tranzitiile 1 si 3 pierd un punct, iar tranzitiile 2 si 4 adauga un punct, ceea ce face ca pierderea sa fie temporara. Daca luam marcajele din b si d cu pondere dubla fata de celelalte, observam ca marcajul ponderat este invariant:

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

indiferent de M. Daca notam cu g vectorul ponderilor ordonat dupa L, obtinem relatia $g^T.M = 3$ pentru orice marcaj M (unde g^T este transpusa lui g).

Considerind doua marcaje M si M' cu M [t> M', adica M' = M + A[-, t] obtinem $g^T.M = g^T.M' = g^T.M + g^T.A[-, t]$. Rezulta $g^T.A[-, t] = 0$ pentru orice t, de unde $g^T.A = 0$. Deci g este o solutie a ecuatiei $x^T.A = 0$ si se numeste un L-invariant.

In general, pentru o RP cu matricea de incidente A, un L-vector I este un L-invariant daca si numai daca $I^T.A = 0$. Un L-invariant ne-negativ I se numeste minimal daca si numai daca nu exista un alt L-invariant I' a.i. 0< I' < I.

Un algoritm simplu si eficient de calcul al invariantilor este dat in continuare. Fie U matricea unitate de ordinul m (m fiind numarul locatiilor) si A matricea de incidente.

calcul_invarianti()
{ construieste matricea (U|A);
for (fiecare indice j al tranzitiei t_j)

{
adauga la matricea (U|A) atitea linii i cite combinatii lineare de cite doua linii, cu coeficienti intregi
pozitivi, in care se anuleaza elementul [i,j] exista;
elimina din matricea (U|A) liniile i in care elementul [i,j] este nenul.
}
}
L-invariantii corespund liniilor nenule din U.

Pentru modelul excluderii mutuale obtinem:

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	1

Pentru j=1 se adauga liniile a+b si b+e obtinindu-se:

a+b	1	1	0	0	0	0	0	0	0
b+e	0	1	0	0	1	0	0	-1	1

dupa care se suprima liniile a, b si e (avind elementele din coloana 1 diferite de 0), ceea ce conduce la forma modificata a matricei U|A urmatoare:

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

Urmatoarea coloana considerata este j=3, pentru care se adauga liniile c+d si d+b+e obtinindu-se:

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

Se elimina liniile c, d precum si b+e, obtinindu-se invariantii:

$$I_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \qquad I_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \qquad I_3 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Daca M este un marcaj al unei retele Petri si I un L-invariant atunci pentru orice marcaj M' accesibil din M avem $I^T.M' = I^T.M$.

Acest rezultat se poate folosi astfel:

- verificarea evitarii anumitor marcaje ; astfel, daca exista un invariant I a.i. $I^T.M' <> I^T.M$ atunci M' nu poate fi accesibil din M ;
- gasirea conditiilor necesare completarii unui marcaj M' accesibil din M si cunoscut partial ;
- deducerea unor proprietati generale ale marcajelor accesibile.

De exemplu, folosind relatia $I^T.M = I^T.M_0$ (unde M_0 este un marcaj initial, iar M un marcaj oarecare accesibil din $M_{0)}$ obtinem pe rind, pentru cei trei invarianti:

M[b] + M[d] + M[e] =1,
M[c] + M[d] = 1,
M[a] + M[b] = 1.

Prima relatie corespunde conditiei de excludere mutuala si exprima faptul ca resursa este fie libera (M[e] = 1), fie utilizata de unul din procese (M[b] = 1), fie de al doilea proces (M[d] = 1). Luate in ansamblu, relatiile conduc la M[l_i] <= 1 pentru orice locatie li, deci reteaua este sigura. Ponderile asociate locatiilor in vectorul g rezulta din $g = I_1 + I_2 + I_3$. Relatia

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

probeaza ca reteaua este conservativa.

In privinta reproducerii marcajelor, observam ca pentru exemplul excluderii mutuale, marcajul initial se reproduce prin executia tranzitiilor 1 si 2. Efectul tranzitiei 1 asupra marcajului este dat de relatia M_0 + A[-,] = M_1 care este echivalenta cu:

$$M_0 + A* \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = M_1$$

Ca urmare, efectul cumulat al tranzitiilor 1 si 2 poate fi reprezentat prin:

$$M_0 + A* \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = M_0$$

unde T-vectorul:

$$e = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

reprezinta numarul de executii ale tranzitiilor si este o solutie a ecuatiei A.y = 0. El se numeste T-invariant.

In general, pentru o RP cu matricea de incidente A, un T-vector J este un T-invariant daca si numai daca A.J = 0. Un T-invariant ne-negativ J se numeste minimal daca si numai daca nu exista un alt T-invariant ne-negativ J' a.i. 0 < J' < J.

Daca J este un T-invariant al unei RP atunci exista un marcaj reproductibil prin executia tranzitiilor in conformitate cu J.

Din relatiile $x^T.A = 0$ si A.y = 0 corespunzatoare l-invariantilor x, respectiv t-invariantilor y, se deduce ca t-invariantii asociati lui A se pot afla calculind l-invariantii asociati matricei de incidente transpusa $A^{T.}$ Cu alte cuvinte, componentele repetitive ale unei RP sint componente conservative in RP duala definita astfel:

- fiecarui loc in RP ii corespunde o tranzitie in RP duala ;

- fiecarei tranzitii in RP ii corespunde un loc in RP duala ;

- fiecarui arc in RP ii corespunde un arc orientat in sens contrar in RP duala.

Pentru exemplul excluderii mutuale obtinem invariantii:

$$J1 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad J2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

RP revine in marcajul initial prin executia tranzitiilor 1 si 2 sau 3 si 4.

Aplicatie: transmiterea unui mesaj cu confirmare (vezi fig.4.12).

Matricea de incidente este:

A	1	2	3	4	5	6
a	-1	0	1	0	0	0
b	1	-1	0	0	0	0
c	0	1	-1	0	0	0
d	0	0	0	-1	0	1
e	0	0	0	1	-1	0
f	0	0	0	0	1	-1
m	1	0	0	-1	0	0
r	0	-1	0	0	1	0

Se obtin invariantii urmatori:

$$I_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad I_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad I_3 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Luind marcajul initial din figura, rezulta relatiile:

M[a] + M[b] + M[c] = 1,

M[d] + M[e] + M[f] = 1,

M[a] + M[b] + M[e] + M[m] + M [r] =1.

Din ele rezulta $M[l_i] <= 1$, deci reteaua este sigura, iar prin insumarea relatiilor rezulta:

2M[a] + 2M[b] + M[c] + M[d] + 2M[e] +M[f] + M[m] + M[r] = 3

deci reteaua este conservativa.

4.4.6. Reducerea RP

Construirea masinii de puncte si utilizarea matricelor de incidente devin complicate pentru RP mari. In asemenea situatii, reducerea RP la unele mai simple, dar care pasteaza proprietati cum sint marginirea, viabilitatea sau invarianta este foarte utila. Reducerea nu contribuie la obtinerea unor modele echivalente, deci nu trebuie sa cautam interpretari practice ale acesteia, nici sa luam ca model pentru implementare o retea simplificata.

4.4.6.1. Eliminarea unui loc (reducerea R_I)

Un loc l_i poate fi eliminat daca indeplineste conditiile:

1) tranzitiile sale de iesire nu au alte locuri de intrare ;

2) nici o tranzitie nu este simultan intrare si iesire ;

3) cel putin o tranzitie de iesire a lui l_i are, la rindul sau, cel putin un loc de iesire.

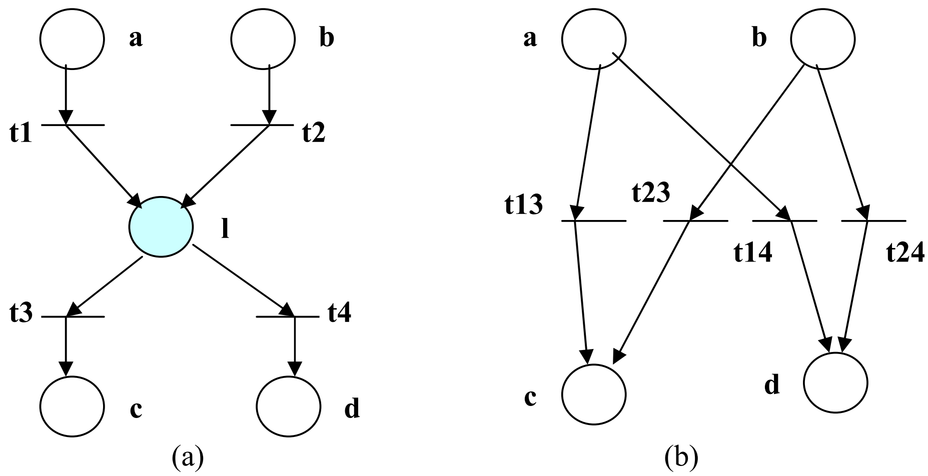


Figura 4.20.
(a) reteaua initiala ; (b) reteaua dupa reducere

Eliminarea lui l_i este insotita de transformarea tranzitiilor sale din aval si amonte, precum si a marcajului, dupa cum urmeaza:

- daca l_i are j intrari si k iesiri, ele sint inlocuite prin j.k tranzitii, fiecare reprezentind contopirea unei tranzitii de intrare cu una de iesire (vezi figura 4.20) ;

- iesirile unei tranzitii de intrare devin iesiri ale tranzitiei obtinuta prin contopire (vezi figura 4.21) ;

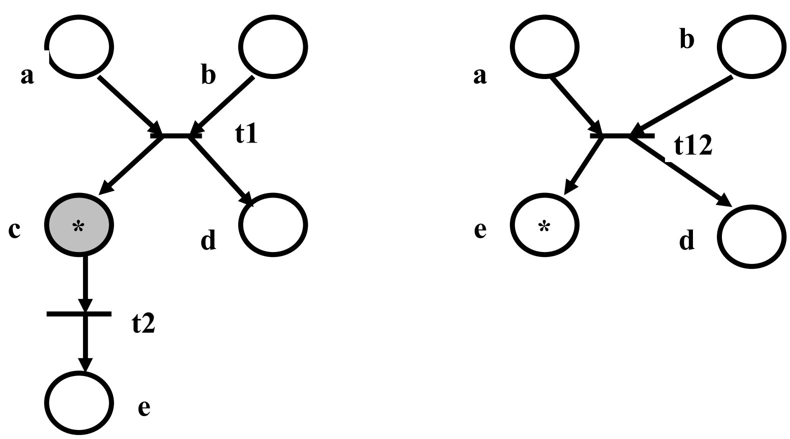


Figura 4.21.

- daca li are k iesiri si este marcat atunci prin eliminare se obtin k retele distincte, marcajul fiind plasat in fiecare caz in locurile corespunzatoare unei alte tranzitii de iesire.

Proprietatile conservate prin reducerea R1 sint marginirea, siguranta, viabilitatea, cvasiviabilitatea, evitarea blocarii, starea de revenire, conservabilitatea.

4.4.6.2. Reducerea unui loc implicit (R_2)

Un loc l_i este implicit daca indeplineste conditiile:

1) marcajul sau permite intotdeauna executia oricarei tranzitii de iesire, care ar fi executabila daca se ignora l_i ;

2) marcajul sau poate fi determinat din marcajul celorlalte locuri.

Un loc implicit poate fi eliminat, impreuna cu arcele sale de legatura. Un exemplu de transformare R_2 este prezentat in figura 4.22.

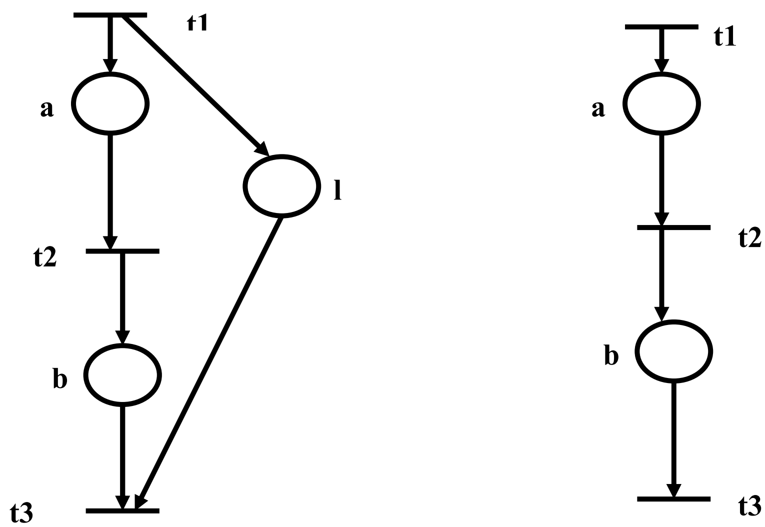


Figura 4.22.

(a) Reteaua initiala; (b) reteaua redusa

De remarcat ca modelul prezentat ramine reductibil daca M(l2)>0, dar devine nereductibil daca M(l1)=M(l2)=0 si M(l3)=1.

Proprietatile de marginire, viabilitate, cvasiviabilitate, evitarea blocarii, starea de revenire si conservabilitatea sint verificate pentru RP redusa daca si numai daca sint verificate pentru RP initiala. RP poate sa nu fie sigura chiar daca RP redusa este.

4.4.6.3. Tranzitie neutra (R_3)

O tranzitie t este neutra daca si numai daca multimea locurilor sale de intrare este identica cu multimea locurilor sale de iesire. Ea poate fi eliminata daca si numai daca exista o tranzitie t'<>t avind O(t', l)>=I(l, t) pentru orice l din Pre(t).

In exemplul din figura 4.23, tranzitia t_5 este neutra, putind fi eliminata impreuna cu arcele care o leaga de locurile de intrare si de iesire l1 si l2.

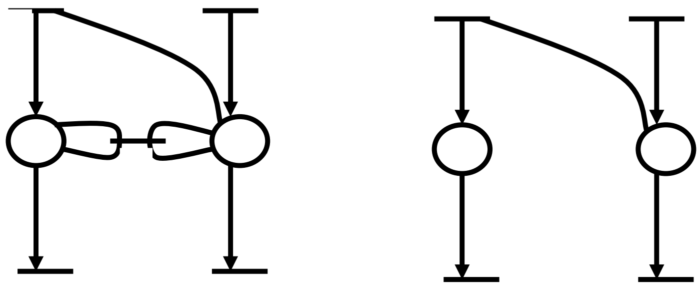


Figura 4.23.
(a) reteaua initiala; (b) reteaua redusa

Fiecare dintre proprietatile de marginire, siguranta, viabilitate, cvasiviabilitate, absenta blocarii, stare de revenire si conservabilitate este verificata pentru RP initiala daca si numai daca este verificata pentru RP redusa.

4.4.6.4. Tranzitii identice

Doua tranzitii sint identice daca au aceleasi locuri de intrare si aceleasi locuri de iesire. Se poate elimina una din ele si arcele corespunzatoare. Proprietatile conservate sint marginirea, siguranta, cvasiviabilitatea, absenta blocarii, starea de revenire, conservabilitatea.

Unele din configuratiile prezentate se intilnesc mai rar in modelele bine construite. Ele pot apare insa dupa aplicarea anumitor reduceri, ceea ce face sa creasca importanta lor practica. Regulile prezentate nu sint singurele reduceri posibile.

4.4.7. Conservarea invariantilor

Urmatoarele transformari conserva invariantii retelelor. Primele doua se aplica RP obisnuite. Cea de a doua transformare poate conduce la obtinerea unei RP generalizate, pentru care sint aplicabile alte doua reguli, similare primelor.

Reducerea R_a se refera la tranzitii impure intr-o RP obisnuita. O tranzitie impura t are un loc de intrare l, care este simultan nod de iesire. Reducerea ei se face astfel:

- se suprima arcele (l, t) si (t, l) ;

- se suprima t daca ea devine izolata.

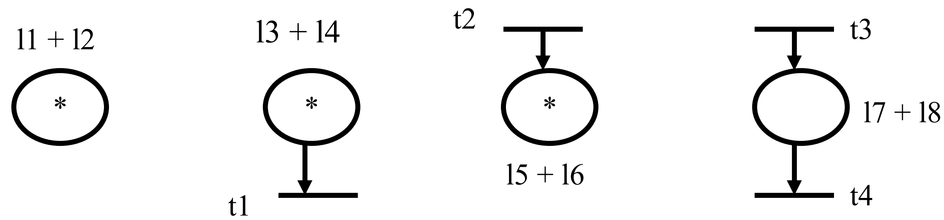
Reducerea R_b elimina o tranzitie pura intr-o RP obisnuita. Este necesar ca atit multimea locurilor de intrare cat si multimea locurilor de iesire sa fie nevide. Transformarea se face astfel:

- se elimina tranzitia pura t ;

- fiecarui cuplu de locuri l_i din Pre(t) si l_j din Post(t) i se asociaza un loc l_i+l_j al carui marcaj este $M(l_{i)}+M(l_j)$;

- tranzitiile de intrare ale lui l_i+l_j sint tranzitiile de intrare ale lui l_i reunite cu tranzitiile de intrare ale lui l_j, mai putin t ; tranzitiile de iesire ale lui l_i+l_j sint tranzitiile de iesire ale lui l_i reunite cu tranzitiile de iesire ale lui l_j, mai putin t.

Regulile se aplica repetat pina la obtinerea unor cazuri ireductibile sau a unei RP generalizate. Cazurile ireductibile sint prezentate in figura 4.24.



(a) (b) (c) (d)

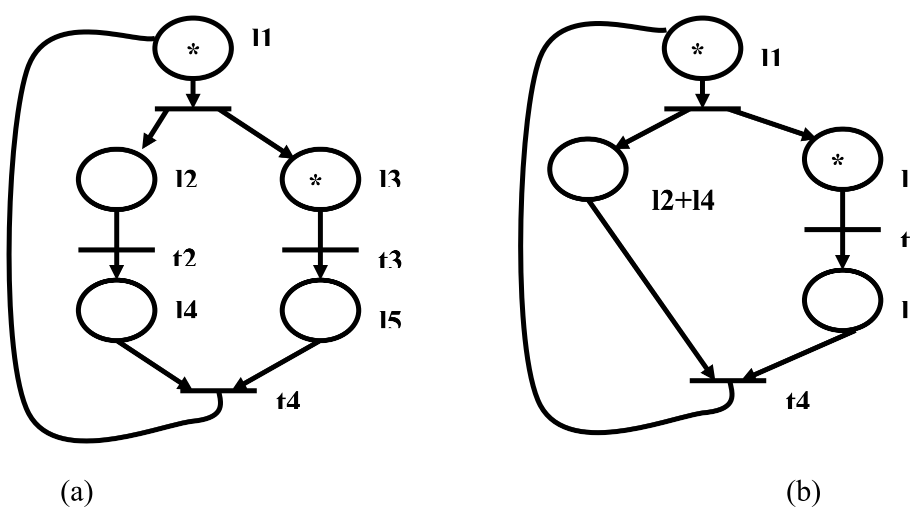
Figura 4.24

Dintre ele, doar cazul (a) corespunde unei componente conservative cu invariantul $M(l_1)+M(l_2)=1$.

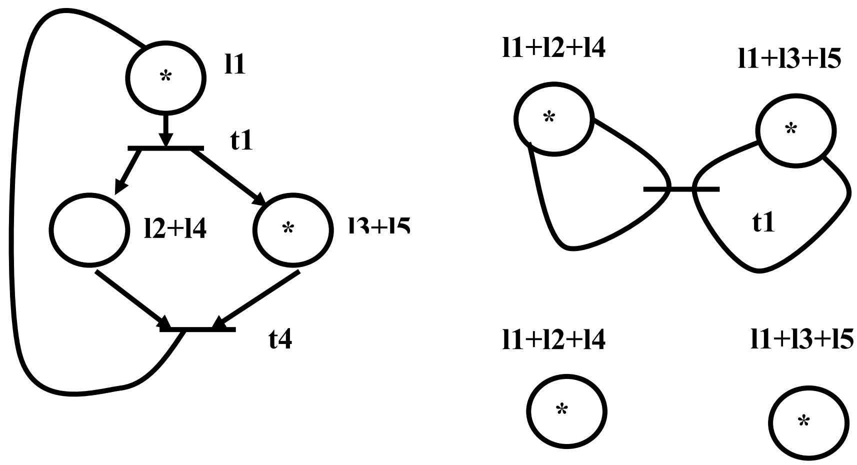
Aplicarea regulilor este ilustrata in figura 4.25. Sint aplicate succesiv reduceri R_b pentru tranzitiile t_2, t_3 si t_4, apoi reducerea R_a pentru tranzitia t_1. Rezultatul este obtinerea a doi l-invarianti si anume:

$M(l_1) + M(l_2) + M(l_4) = 1$ si $M(l_1) + M(l_3) + M(l_5)+2$.

Graful redus poate depinde de ordinea reducerilor. Deoarece evita duplicarea locurilor, este recomandabila aplicarea prioritara a reducerilor R_b pentru tranzitii cu o intrare si o iesire.



(a) (b)

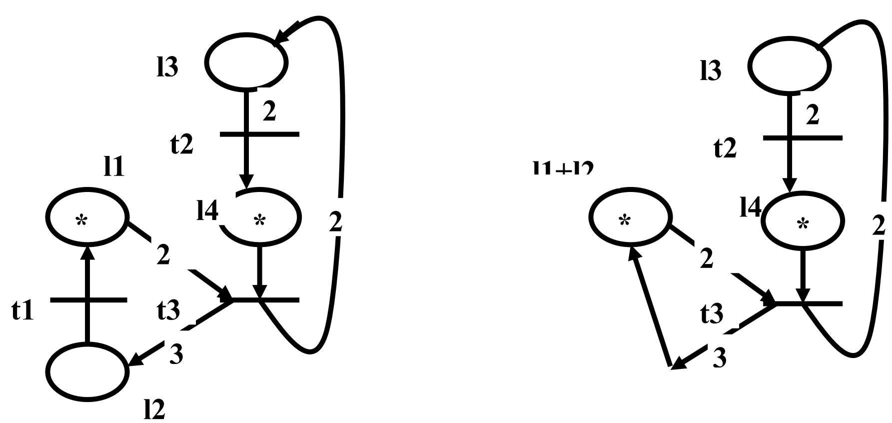


(c) (d) si (e)

Figura 4.25

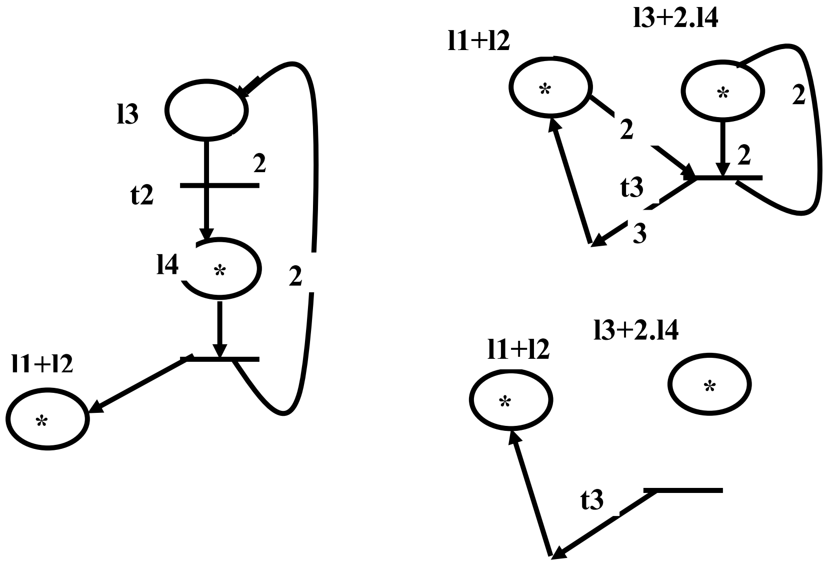
(a) reteaua initiala; (b) dupa Rb(t2); (c) dupa Rb(t3); (d) dupa Rb(t4); (e) dupa Ra(t1)

Reducerea $R_{a'}$ se refera la o tranzitie impura intr-o RP generalizata. Fie t o astfel de tranzitie si l un loc pentru care I(l, t)=p>0 si O(t, l)=q>0. Reducerea depinde de relatia dintre p si q si consta in eliminarea arcului (l, t) sau (t, l) cu pondere mai mica, celuilalt asociindu-i-se o pondere egala cu |p-q|. Daca p = q atunci se elimina ambele arce. In fine, se elimina si t, daca ea devine izolata.Reducerea $R_{b'}$ se refera la o tranzitie pura t intr-o RP generalizata. Ea trebuie sa aiba cel putin un loc de intrare si un loc de iesire. Reducerea se face astfel:



(a) (b)

Figura 4.26
(a) reteaua initiala ; (b) dupa $R_{b'}(t_1)$



(c) (d) si (e)

Figura 4.26
(c) dupa $R_{a'}(t_3)$; (d) dupa $R_{b'}(t_2)$; (e) dupa $R_{a'}(t_3)$

- se elimina tranzitia pura t;

- fiecarei perechi (l_i, l_k) formata dintr-o locatie de intrare l_i si una de iesire l_k ale lui t_i se asociaza un loc $q.l_i+p.l_k$, unde $q = O(t, l_k)$ iar $p = I(l_i, t)$, cu marcajul egal cu $q.M(l_i) + p.M(l_k)$

- fiecare tranzitie de intrare (iesire) a lui l_i (exceptind t) devine tranzitie de intrare (iesire) a lui $q.l_i + p.l_k$ cu ponderea multiplicata prin q ; fiecare tranzitie de intrare (iesire) a lui l_k (exceptand t) devine tranzitie de intrare (iesire) a lui $q.l_i+p.l_k$ cu ponderea multiplicata prin p.

Reducerile sint ilustrate pe un exemplu in figura 4.26.

Rezultatul obtinut evidentiaza existenta unui l-invariant $M(l_3) +2.M(l_4) = 2$.

Reteaua nu este marginita.