

Compararea grafetului cu rețeaua Petri

Prof. dr. ing. Sorin Larionescu – UTCB

Grafetul este un *model* folosit foarte mult pentru *sistemele discrete logice*, a fost standardizat internațional¹ și a devenit unul din instrumentele de bază pentru întocmirea caietelor de sarcini și programarea automatelor programabile logice. Denumirea de *GRAFFCET*² provine³ din inițialele GRAF Funcțional de Comandă Etape - Tranziții și indică pe scurt că acest graf servește la modelarea sistemelor de conducere discrete logice și este format din etape, tranziții și legături orientate între acestea. Definierea grafetului, simplă și fără interpretări posibile este dată de standardele internaționale și naționale. Ea s-a schimbat însă în timp și probabil se va modifica și în continuare. Inițial grafetul a fost o rețea Petri particularizată⁴ pentru definirea riguroasă a caietelor de sarcini. Modul de reprezentare grafică a grafetului era aproape identic cu cel de la rețeaua Petri. Dar grafetul a cunoscut o nouă tinerete în momentul în care s-a dovedit foarte util pentru programarea automatelor programabile logice. Reprezentarea grafică s-a schimbat radical. La o primă vedere nu mai seamănă deloc cu o rețea Petri. Totuși multe din proprietățile modelului grafet actual se regăsesc și la modelul rețea Petri. Prezentarea pe scurt a asemănarilor și diferențelor dintre cele două tipuri de modele reprezintă scopul acestei lucrări.

Rețelele Petri *interpretate, sigure, viabile și fără conflicte efective* sunt numite *conforme* și coincid cu o clasă de grafeturi frecvent utilizate în practică. *Interpretarea* înseamnă că pozițiilor rețelei Petri li s-au atașat acțiuni care se execută atunci când pozițiile devin active, iar tranzițiilor li s-au atașat evenimente care, atunci când apar, declanșează tranzițiile pe care le interpretează, dacă acestea sunt validate. Rețeaua Petri este denumită *sigură* dacă toate pozițiile au un număr de mărci egal sau mai mic decât unu pentru toate situațiile de funcționare. Proprietatea de *viabilitate* asigură că rețeaua Petri nu se blochează în timpul funcționării într-o poziție sau grup de poziții. *Conflictele efective* se rezolvă diferit, după cum vom vedea, la grafet și la rețeaua Petri. Absența lor face cele două modele echivalente.

Etapa din grafet corespunde *poziției* (locației) din rețeaua Petri și modelează o situație în care comportamentul sistemului modelat, sau numai a unei părți ale sale, rămâne neschimbat în raport cu intrările și ieșirile sale. Etapele sunt reprezentate prin pătrate iar pozițiile prin cercuri, după cum se vede în exemplul din Fig. 1. Semnificația lor este aceeași la ambele modele.

Etapa inițială, activă la momentul de timp inițial, este specificată la grafet printr-un pătrat dublu. Pentru grafetul din Fig. 1 b) etapa inițială are numărul 10 iar etapele active la momentul de timp curent sunt specificate prin plasarea unei mărci în pătratele corespunzătoare cu numărul 20 și 40. Aceasta înseamnă că grafetul din figură prezintă situația sistemului discret logic în care funcționează instalațiile I1, I2 și robinetul R se deschide⁵.

La rețeaua Petri nu se specifică starea inițială iar pentru cazul din Fig. 1 a) sistemul modelat se găsește în starea de la momentul de timp curent specificată prin plasarea unei mărci în poziția corespunzătoare cu numărul 10.

Acțiunile care interpretează etapele grafetului sunt trecute în dreptunghiuri atașate de acestea. Pentru etapele inițiale se consideră că acțiunile implicite sunt cele care opresc toate echipamentele și se

¹ IEC 848 (Function Chart for Control System – FCCS), 1987.

² GRAFCET – Graphe Fonctionnel de Commande Etape – Transition (fr.).
SFC – Sequential Function Chart (eng.).

³ Prin proveniența sa *grafet* este un substantiv românesc, deși coincide cu cuvântul francez. Din această cauză, la fel ca în limba franceză, el se *pronunță* și se *folosește* după regulile limbii române.

⁴ Rapport de la Commission de normalisation de la representation du cahier des charges d'un automatisme logique, (Group de travail AFCET Sytemes logiques), Automatisme, Mars – Avril, 1978, p. 66 - 83.

⁵ Instalația I2 a fost pornită în etapa 30. Deoarece la activarea etapelor următoare nu s-a executat nici o comandă de oprire a instalației I2, aceasta rămâne pornită.

specifică, dacă este cazul, numai acțiunile diferite de acestea. Pozițiile rețelei Petri sunt interpretate de acțiuni care sunt specificate alături de cercul care reprezintă poziția.

Tranzițiile și evenimentele care le interpretează sunt reprezentate la fel în cazul rețelei Petri și a graficului. Denumirea tranzițiilor este formată de un număr plasat între paranteze rotunde. De exemplu, tranziția (5) este interpretată de evenimentul *o1*, o variabilă logică care ia valoarea logică 1 (adevărat) dacă se acționează butonul de oprire cu numărul 1. La grafic tranzițiile sunt formate din linii *orizontale* îngroșate, pe când rețelele Petri reprezintă tranzițiile prin dreptunghiuri¹ înnegrite care pot fi situate oricum.

Arcele orientate care indică legăturile dintre etape (poziții) și tranziții sunt formate la grafic numai din *linii drepte și verticale* cu sensul de sus în jos. Din această cauză săgețile nu mai sunt prezente. Excepție fac doar arcele orientate de întoarcere la starea inițială. La rețeaua Petri, Fig. 1 a), arcele orientate pot avea orice formă, pot fi orientate oricum și sensul este indicat întotdeauna prin săgeți. Din această cauză există o mare libertate la desenarea modelului rețea Petri.

La grafic fiecărei tranziții îi este asociată o *receptivitate* R_i , în care i este numărul tranziției. Receptivitatea este o funcție logică de evenimente, stări ale etapelor și temporizări. Tranzițiile rețelei Petri pot fi interpretate numai de evenimente și de temporizări, nu și de stări ale pozițiilor.

Evenimentele sunt, ca și la rețeaua Petri, *variabile logice* atașate semnalelor de intrare. În Fig. 2 se observă că pentru un semnal logic $s(t)$ de intrare în sistemul modelat se pot defini variabilele logice tip *nivel* E și tip *impuls* $\uparrow E$, $\downarrow E$ care apar pe frontul crescător, respectiv descrescător al semnalului E .

Fiecărei etape i se asociază, la grafic, o *variabilă de stare* logică notată X_i , în care i este numărul etapei. Această variabilă ia valoarea logică 1 cât timp etapa este activă sau 0 dacă etapa este inactivă.

Variabilele de temporizare se definesc față de momentul în care o etapă devine activă. În Fig. 3 se prezintă variabila de temporizare $t/X_{20}/2s$ care are valoarea logică 1 (adevărat) după o întârziere de 2 secunde de la activarea etapei cu numărul 20. Se observă că temporizarea apare numai dacă etapa este activă mai mult de 2 secunde. În momentul în care etapa se reactivează temporizarea revine la valoarea logică 0 (fals).

Fiecărei etape i se poate atașa o *variabilă logică*, numită *acțiune*, care este funcție de starea etapei la care este atașată, de semnale de intrare și de o temporizare.

De exemplu, acțiunea $I1$ atașată etapei X_{20} din Fig. 1 b) și definită de relația (1) apare, adică ia valoarea logică 1 (adevărat), dacă etapa cu numărul 20 este activă, adică $X_{20}=1$, semnalul logic de intrare $a=1$ și temporizarea $t/X_{20}/5s$ a fost realizată.

$$I1 = X_{20} \cdot a \cdot (t / X_{20} / 5s) \quad (1)$$

Schimbările în timp din grafic, evoluția sa, respectă următoarele reguli:

1. **Condiții inițiale.** La momentul inițial numai etapele inițiale sunt active.
2. **Condiții de validare.** Pentru ca o tranziție să fie validată este necesar ca toate etapele precedente să fie active.
3. **Condiții de declanșare.** O tranziție este declanșată dacă este validată și receptivitatea care o interpretează are valoarea logică unu. Pentru a declanșa o tranziție trebuie, obligatoriu în această ordine, ca etapele precedente să fie dezactivate iar etapele posterioare să fie activate.
4. **Declanșării simultane.** Toate tranzițiile declanșabile la un moment dat sunt declanșate simultan.
5. **Conflictul de activare.** Dacă o etapă trebuie simultan activată și deaktivată de declanșările simultane ale unei tranziții precedente și ale unei tranziții posterioare, atunci ea rămâne activă².

La rețelele Petri regulile 4 și 5 nu se aplică deoarece tranzițiile nu se pot declanșa simultan. Dacă rețeaua Petri este conformă nu apare necesitatea aplicării acestor reguli și funcționarea este identică cu cea a graficului.

¹ La rețelele Petri generale tranzițiile consumă un număr resurse (mărci) din pozițiile precedente și produc un alt număr de mărci în pozițiile posterioare. Deci ele au consistență și apar ca niște dreptunghiuri. Pentru rețelele Petri sigure, cu maximum o marcă în fiecare poziție, folosite la modelarea sistemelor discrete logice, se adoptă, în general, o reprezentare apropiată de cea folosită de grafic: dreptunghiuri înnegrite subțiri

² Prin analogie cu pornirea unui motor electric se poate spune că există o prioritate la pornire.

Structurile tip folosite în grafcet sunt aceleași ca la rețeaua Petri: *secvența* Fig. 4 a), *saltul* Fig. 4 b), *repetarea* Fig. 4 c), *alegerea* Fig. 5 a), *convergența* Fig. 6 a), *paralelismul* Fig. 7 a), și *sincronizarea* Fig. 8 a). După cum se vede din aceste figuri și Fig. 1 reprezentarea grafică este diferită.

O problemă gravă apare la structura tip alegere Fig. 5 a. La rețeaua Petri această structură realizează o alegere tip SAU EXCLUSIV iar la grafcet realizează o alegere tip SAU. Deosebirea apare atunci când evenimentele R1 și R2 apar simultan. La grafcet, dacă etapa 10 este activă, se declanșează tranzițiile pe care aceste evenimente le interpretează și etapele 20 și 30 devin active simultan, în *paralel*. Deoarece la rețeaua Petri nu se pot declanșa simultan mai multe tranziții nu este posibil ca o alegere să se transforme într-un paralelism! Diferența dintre modul de funcționare al celor două modele apare din modul diferit de definire a mărcii.

La rețeaua Petri marca este o resursă și este indivizibilă. Nu se poate deplasa în același timp din poziția 10 în 20 și 30. Din cauza aceasta apare conflictul efectiv care trebuie rezolvat. Dacă nu este rezolvat sistemul discret logic modelat de rețeaua Petri se va comporta *nedeterminist*, la apariția simultană a evenimentelor R1 și R2 se va activa etapa 20 sau (exclusiv) etapa 30 în funcție de factori care nu sunt cunoscuți.

La grafcet marca este un indicator, semnalizator, al stării de activitate sau inactivitate a etapei. Deoarece mai multe tranziții se pot declanșa simultan și marca se poate deplasa simultan din etapa 10 în etapele 20 și 30.

- *Avantajele* grafcetului față de rețeaua Petri sunt următoarele:

Grafcetul permite interpretarea tranzițiilor și în funcție de unele evenimente interne (endogene) cum ar fi, de exemplu, apariția stării *active* sau *inactive*¹ a pozițiilor. Drept urmare desenul, care are o reprezentare grafică standardizată internațional, este mult mai simplu și mai inteligibil.

Existența macro-etapelor în grafcet conduce de asemenea la o simplificare substanțială a caietelor de sarcini și programelor pentru automatele programabile logice.

Grafcetul permite declanșarea simultană a tranzițiilor fără să existe noțiunea de conflict. Aceasta ușurează întocmirea și folosirea caietelor de sarcini de către nespecialiști. O poziție activată și desactivată simultan rămâne în grafcet activă². În felul acesta se evită ignorarea acțiunilor corespunzătoare poziției și modificarea algoritmului care a stat la baza întocmirii grafcetului. Ca un preț pentru facilitățile oferite, grafcetul nu posedă un model matematic atât de elaborat ca rețeaua Petri.

Desenul standardizat al grafcetului este mai adecvat pentru automatele programabile logice.

- *Dezavantajele* grafcetului față de rețeaua Petri sunt următoarele:

Grafcetul permite și paralelismul prin interpretarea tranzițiilor situate într-o structură tip alegere. În aceste situații pot apare frecvent greșeli la ieșirea din paralelism.

La modelarea sistemelor discrete logice cu ajutorul grafcetului pot apare erori la reactivarea etapelor. La modelul rețea Petri aceste erori sunt repede evidențiate.

Structura modelului grafcet fără interpretare oferă puține informații și nu permite verificările care se pot face pe un model rețea Petri.

În concluzie se poate aprecia că pentru elaborarea caietelor de sarcini sau programarea automatelor programabile logice grafcetul este preferabil. Pentru analiza, sinteza și validarea modelului este mai bună rețeaua Petri. La prima vedere ar părea avantajoasă elaborarea caietului de sarcini sub formă de grafcet și apoi să se treacă la rețeaua Petri pentru analiză, validare și sinteză. Din păcate trecerea grafcet – rețea Petri și invers nu se poate efectua direct decât în cazul modelelor mai simple de tip rețea Petri interpretată, sigură, viabilă și fără conflicte structurale. În acest caz se renunță însă la toate facilitățile pe care le oferă grafcetul. De fapt alegerea ar trebui să se facă în funcție de complexitatea sistemului modelat. Pentru sisteme relativ simple nu are importanță modelul folosit. La modelarea sistemelor complexe grafcetul este bun drept caiet de sarcini, dar pentru analiza, sinteza și validarea modelului este prudent să se folosească rețeaua Petri. În felul acesta greșeli greu detectabile pot fi evitate.

¹ Rețeaua Petri nu poate testa dacă o poziție este inactivă.

² Prioritate la pornire.

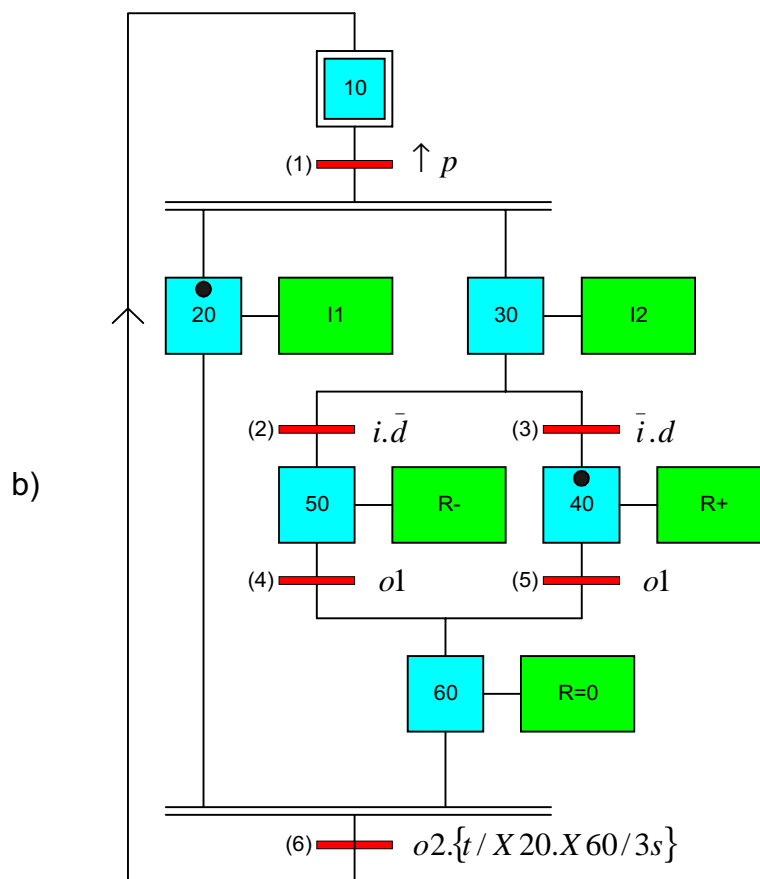
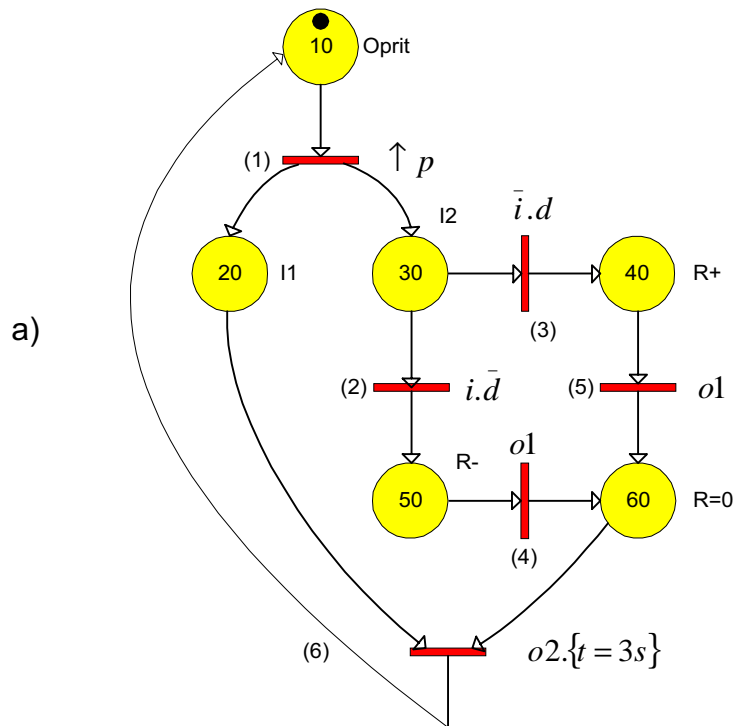


Fig. 1 Modelul unui sistem discret logic prezentat sub forma unei rețele Petri interpretate, sigure, viabile și fără conflicte a) și sub forma unui grafet b).

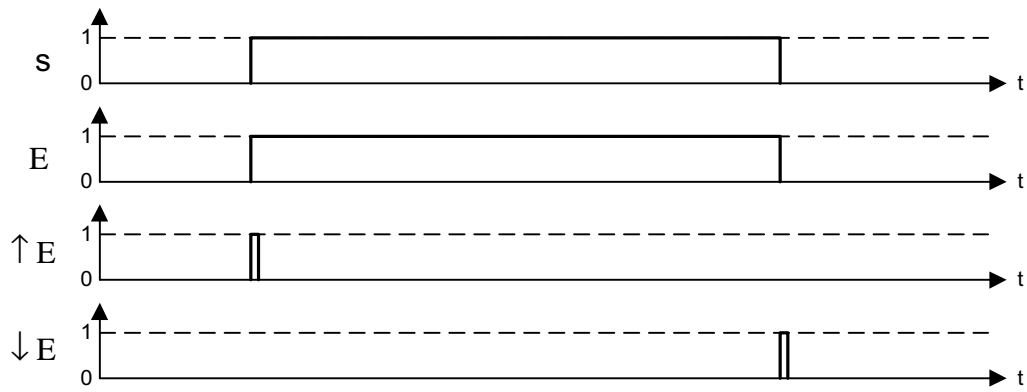


Fig. 2 Semnalul logic s de intrare în sistem și evenimentele tip nivel E și tip impuls $\uparrow E$, $\downarrow E$ atașate.

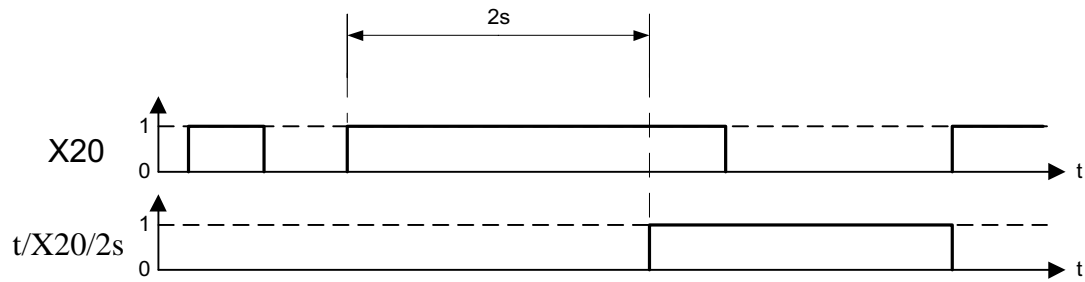


Fig. 3 Variabila $t/X20/2s$ de temporizare față de activarea etapei 20

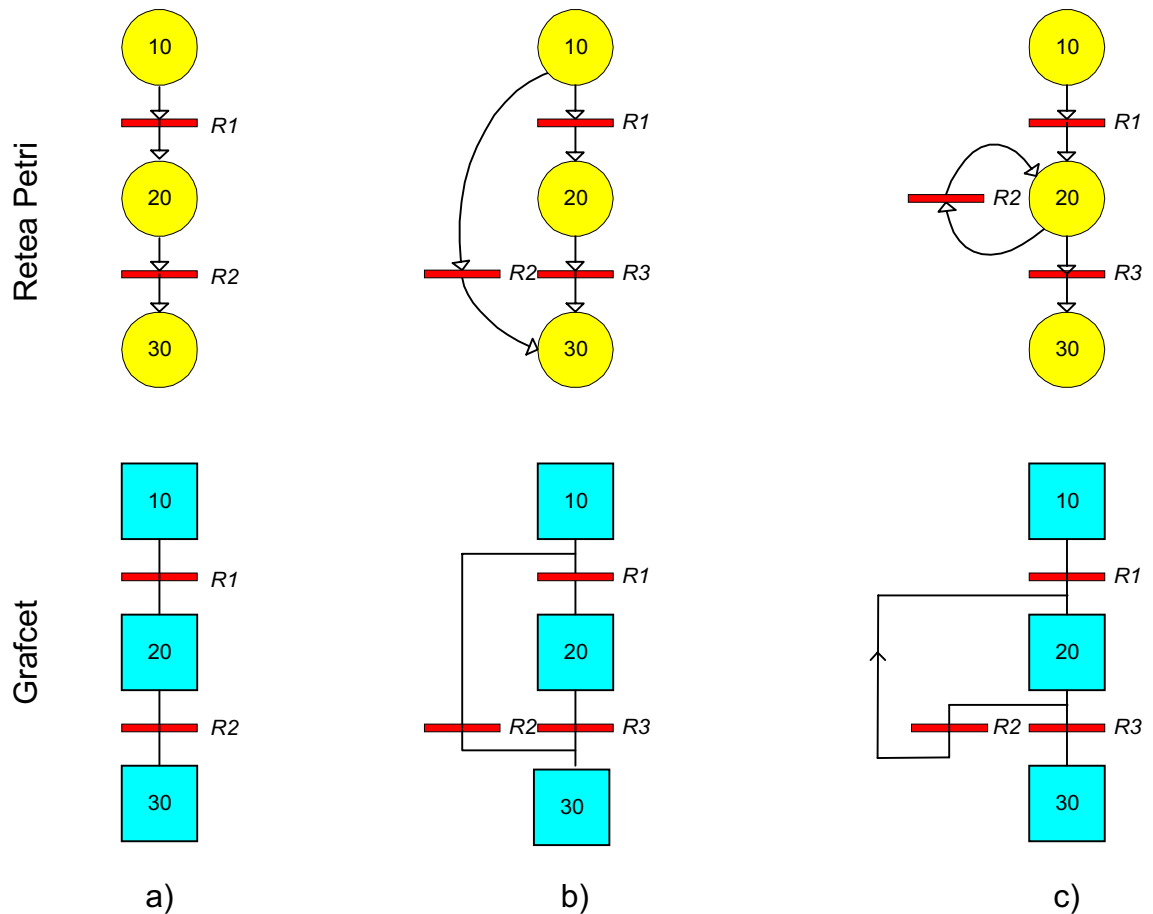
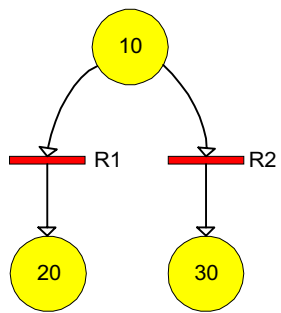
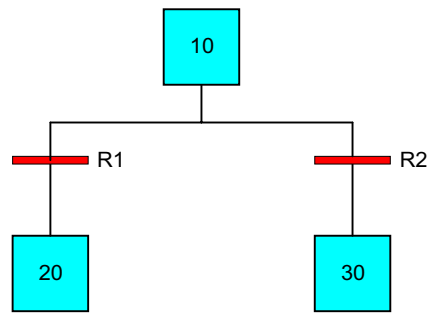


Fig. 4 Structurile tip secvență a), salt b) și repetare c) la rețeaua Petri și grafcet

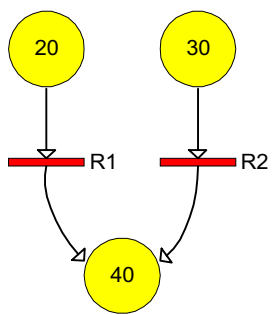


a)

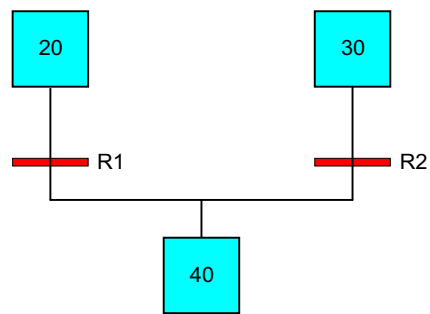


b)

Fig. 5 Structura tip alegere la rețeaua Petri a) și grafcet b).



a)



b)

Fig. 6 Structura tip convergență la rețeaua Petri a) și grafcet b).

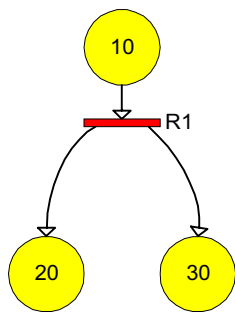
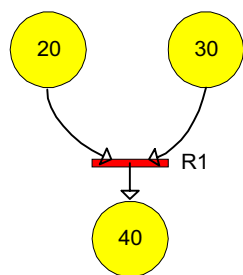
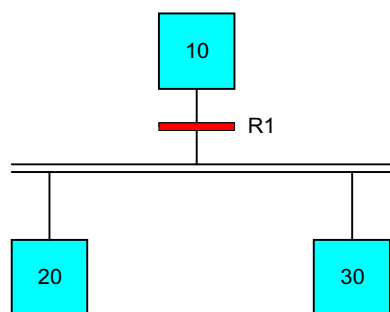
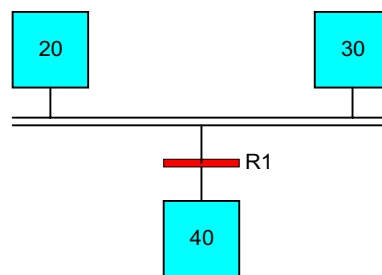


Fig. 7 Structura tip paralelism la rețeaua Petri a) și grafcet b).



a)



b)

Fig. 8 Structura tip sincronizare la rețeaua Petri a) și grafcet b).