Translații între modelele bazate pe sisteme cu tranziții

Conf.dr. Lucian Luca
Universitatea "Tibiscus" din Timişoara

ABSTRACT. The actual study presents, starting from the results obtained in [SNW94], the formal relations between the different concurrent models, studied by the theory of categories. There are studied the models based on transition systems, expanding the study on the asynchronous transition systems, on the concurrent transition systems, on the trace automata, on the transition systems with independence and on the Petri nets. There are defined the morphism which structure the models, forming categories.

1 Introducere

Dacă teoriile tradiționale în teoria sistemelor concurente s-au concentrat asupra *modelării alegerilor* şi a *concurenței*, cercetări recente s-au focalizat pe elaborarea de teorii despre alte aspecte ale comportării sistemelor, cum ar *timpul real*, *probabilitatea* şi *securitatea*.

În [SNW94] relațiile formale dintre aceste diferite modele concurente sunt studiate cu ajutorul teoriei categoriilor, folosindu-se ca instrument *adjuncțiile*, mai precis, translațiile dintre modele sunt considerate a fi *reflecții* şi *coreflecții*, adică, o categorie este complet şi cu încredere scufundată (eng.: *fully and faithfully*) în cealaltă. Pentru a fi transformat în categorie, fiecare model este echipat cu o noțiune de morfism. Morfismele vor păstra comportamentul, respectând, în acelaşi timp, alegerea granularității modificărilor atomice din descrierea proceselor, adică morfismele sunt forme de *simulări*. Un rol al morfismelor este de a lega comportamentul unei construcții pe procese de cel al componentelor sale.

Reflecțiile şi coreflecțiile sunt un mijloc pentru a exprima faptul că un model este scufundat în (este mai abstract decât) altul, chiar dacă cele două modele sunt exprimate în termeni matematici foarte diferiți. Un adjunct ne va spune cum să scufundăm modelul mai abstract în celălalt, iar celălalt adjunct va abstractiza în continuare câteva aspecte ale reprezentării. Proprietățile de conservare ale adjuncților se pot folosi pentru a arăta cum se traduce semantica dintr-un model în semantica din celălalt model.

Principalul rezultat din [SNW94] este un cub (figura 1), în care vârfurile reprezintă opt modele de concurență, pe care le explicităm jos, iar muchiile sunt tocmai reflecții (săgețile duble) şi coreflecții (săgețile simple).

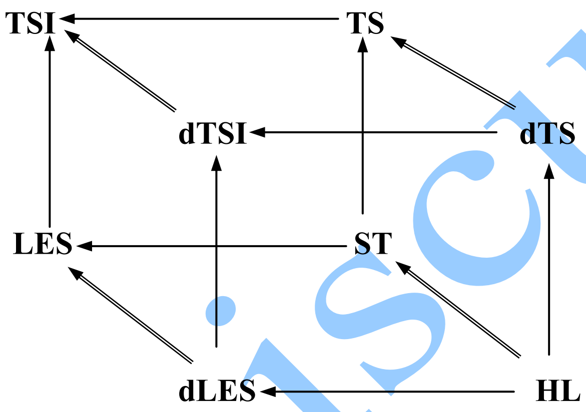

Figura 1: Relații între modele ale concurenței

HL	Limbaje Hoare
ST	Arbori de sincronizare
dLES	Structuri de evenimente etichetate deterministe
LES	Structuri de evenimente etichetate
dTS	Sisteme cu tranziții deterministe
TS	Sisteme cu tranziții
dTSI	Sisteme cu tranziții deterministe cu independență
TSI	Sisteme cu tranziții cu independență

Reamintim aici definiția noțiunii de adjunct [McL71] care este cea mai utilă (dintre multele care se găsesc în literatură) pentru scopurile noastre:

Definiția 1 *Un functor $F{:}A \to B$ se spune a fi* ***adjunctul la stânga*** *al unui functor $G{:}B \to A$ şi invers, G este* ***adjunctul la dreapta*** *al lui F,*

$$F \dashv G \qquad \text{sau} \qquad \langle F,G \rangle : A \to B$$

dacă există o familie de săgeţi în A,

$$\eta=\{\ \eta_{\alpha}\!: a \to GF\in(a) \mid a\in A\},$$

numită unitatea adjuncţiei şi care se bucură de următoarea proprietate universală: pentru orice obiect $b\in B$ *şi orice săgeată* $f\!: a\to G(b)$ *în A, există o unică săgeată* $k\!: F(a)\to b$ *astfel încât diagrama din figura 2 comută.*

$$\begin{tikzcd} a \arrow[r, "\eta_a"] \arrow[dr, "f"'] & GF(a) \arrow[d, "G(k)"] \\ & G(b) \end{tikzcd}$$

Figura 2: Functor adjunct

Echivalent, F este adjunctul la stânga al lui G dacă există o familie de săgeţi în B,

$$\varepsilon = \{\varepsilon_b : FG(b) \to b \mid b \in B\},$$

numită *counitatea adjuncţiei*, astfel încât pentru orice săgeată $f : F(a) \to b$, $a \in A$, există o unică săgeată $k : a \to G(b)$ astfel încât $\varepsilon_b \circ F(k) = f$ adică diagrama din figura 3 comută.

$$\begin{tikzcd} FG(b) \arrow[r, "\varepsilon_b"] & \mathrm{B} \\ F(\mathrm{a}) \arrow[u, "F(k)"] \arrow[ur, "f"'] & \end{tikzcd}$$

Figura 3: Functori adjuncţi

Definiţia 2 *O adjuncţie se numeşte **reflecţie** (generalizată) a lui A în B, sau B se zice reflectivă în A, dacă elementele counităţii sunt izomorfisme. Dual, ea este o **coreflecţie** (generalizată) a lui B în A, sau A este coreflectivă în B, dacă componentele unităţii sunt izomorfisme.*

2 Sisteme ordinare cu tranziţii

Sistemele cu tranziţii sunt cele mai utilizate modele pentru calcule şi concurenţă. Ele nu sunt altceva decât grafuri cu tranziţii. De exemplu, să presupunem că valorile variabilelor unui program la diferite momente ale execuţiei lui sunt reprezentate de cinci stări, numite a, b, c, d şi e, iar tranziţiile dintre aceste stări sunt α, β, γ, δ (Figura 4), adică ele sunt cele patru instrucţiuni ale programului care modifică starea maşinii, din sursa săgeţii în ţinta ei.

$$\begin{tikzcd} a \arrow[r, "\alpha"] \arrow[d, "\beta"'] & d \arrow[r, "\gamma"] \arrow[d, "\beta"'] & e \\ b \arrow[r, "\alpha"] & c \arrow[ur, "\delta"] & \end{tikzcd}$$

Figura 4: Un sistem de tranziţii

În general, aceasta se formalizează folosind o relaţie de tranziţie.

Fie L o **mulţime de etichete**. Atunci:

L^* este mulţimea tuturor secvenţelor finite de elemente din L

L^ω este mulţimea tuturor secvenţelor infinite de elemente din L şi

$L^\infty = L^\omega \cup L^*$

Mai mult,

ε este *secvenţa vidă*, $\varepsilon \notin L$ şi fie $L_\varepsilon = L \cup \{\varepsilon\}$

$|\sigma|$ este lungimea secvenţei σ

τ este *acţiunea tăcută*, care merge de la o stare s la aceeaşi stare s, adică nu schimbă starea maşinii.

Atunci $L_\tau = L \cup \{\tau\}$ şi

$$(\bullet)^\circ : L_\tau \to L_\varepsilon$$

$$a \xrightarrow{l} b$$

$$l^\circ = \begin{cases} l, & dacă\ l \in L \\ \varepsilon, & dacă\ l = \tau \end{cases}$$

Definiția 3 *Un sistem cu tranziții este o structură* (Σ, i, L, A) *unde*

- Σ *este o mulțime de stări cu starea inițială* i
- L *este o mulțime de etichete*
- $A = \{ \xrightarrow{l} \subseteq \Sigma \times \Sigma / l \in L\}$ *este o mulțime de tranziții, adică relații binare pe* Σ *indexate după* L.

$(a,b) \in \xrightarrow{l}$ se notează uneori ca $a \xrightarrow{l} b$ și se citește: **sistemul poate evolua din starea *a* în starea *b* efectuând acțiunea** *l*.

Sistemele cu tranziții sunt puse într-o categorie, definind morfismele ca fiind un tip de simulări:

un sistem cu tranziții TS^1 *simulează* un sistem cu tranziții TS^0 dacă ori de câte ori TS^0 poate executa o anume acțiune l într-un anumit context, atunci TS^1 poate executa pe l într-un context înrudit. Echivalent [SNW94]:

1. stările inițiale trebuie mapate pe stări inițiale;
2. pentru fiecare acțiune pe care o poate executa primul sistem dintr-o stare dată, trebuie să fie posibil pentru al doilea sistem să execute acțiunea corespunzătoare - dacă ea există - din starea corespunzătoare.

Definiția 4 *Fie* $TS^0 = (\Sigma^0, i^0, L^0, A^0)$ *și* $TS^1 = (\Sigma^1, i^1, L^1, A^1)$ *două sisteme cu tranziții. Un* ***morfism*** $f: TS^0 \rightarrow TS^1$ *este o pereche* $f = (\sigma, \lambda)$, *unde:*

$$\sigma : \Sigma^0 \rightarrow \Sigma^1 \quad \text{și} \quad \lambda : L^0_\tau \rightarrow L^1_\tau$$

sunt astfel încât $\sigma(i^0) = i^1$

$$\lambda(\tau) = \tau$$

$$a \xrightarrow{l} b \text{ în } TS^0 \Rightarrow \begin{cases} \sigma(a) \xrightarrow{\lambda(l)} \sigma(b) & \text{în } TS^1, \text{ dacă } \lambda(l) \neq \tau \\ \sigma(a) \xrightarrow{\lambda(l)=\tau} \sigma(b)=\sigma(a), & \text{dacă } \lambda(l) = \tau \end{cases}$$

(adică, $\sigma(b)=\sigma(a)$, *dacă* $\lambda(l) = \tau$*)*

În literatură [SNW94] se folosesc două tipuri de λ-morfisme: parțiale și totale, $\lambda : L^0 \rightarrow L^1$.

Se arată ușor că sistemele cu tranziții etichetate și morfismele pentru ele, împreună cu compunerea pe componente a morfismelor, dau o categorie, notată în [SNW94] cu **TS**, având o subcategorie completă **ST**, constând din arbori de sincronizare.

Vom prezenta, în continuare, câteva modele abstracte de concurenţă, bazate pe sisteme cu tranziţii, care nu se regăsesc în cubul din [SNW94], pe care le vom înzestra cu morfisme, formând categorii care se vor o completare şi, pe alocuri, o generalizare a rezultatelor din [SNW94].

3 Sisteme cu tranziţii asincrone

Sistemele cu tranziţii asincrone (ATS) au fost introduse separat în [Bed88] şi [Shi85].

Ele pot fi gândite ca o generalizare a limbajelor pentru urme Mazurkiewicz. Ele deosebesc întrepătrunderea a două acţiuni de execuţia lor adevărat concurentă prin folosirea unei relaţii binare de independenţă.

Definiţia 5 *Un sistem cu tranziţii asincrone este ATS=(Σ, i, L, I, A), unde:*

- *(Σ,i, L, A) este un sistem cu tranziţii şi*
- *$I \subseteq L \times L$ este o relaţie simetrică ireflexivă - relaţia de independenţă - astfel încât:*
  1. $l \in L \Rightarrow \exists a,b \in \Sigma,\ a \xrightarrow{l} b$
  2. $a \xrightarrow{l} b^1$ şi $a \xrightarrow{l} b^2 \Rightarrow b^1 = b^2$
  3. $l_1\, I\, l_2$ şi $a \xrightarrow{l_1} b^1$ şi $a \xrightarrow{l_2} b^2 \Rightarrow \exists c,\ b^1 \xrightarrow{l_2} c$ şi $b^2 \xrightarrow{l_1} c$
  4. $l_1\, I\, l_2$ şi $a \xrightarrow{l_1} b^1$ şi $b^1 \xrightarrow{l_2} c \Rightarrow \exists b^2,\ a \xrightarrow{l_2} b^2$ şi $b^2 \xrightarrow{l_1} c$

Putem relaxa condiţia 1. spunând că toate evenimentele trebuie să fie folosite.

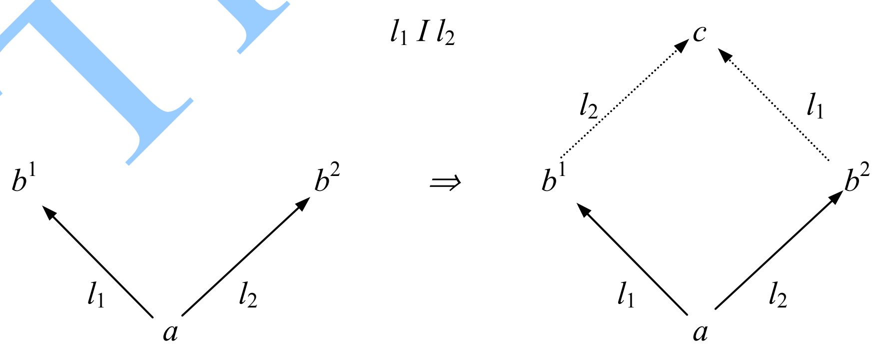

Figura 5: Condiţia 3. pentru sisteme cu tranziţii asincrone

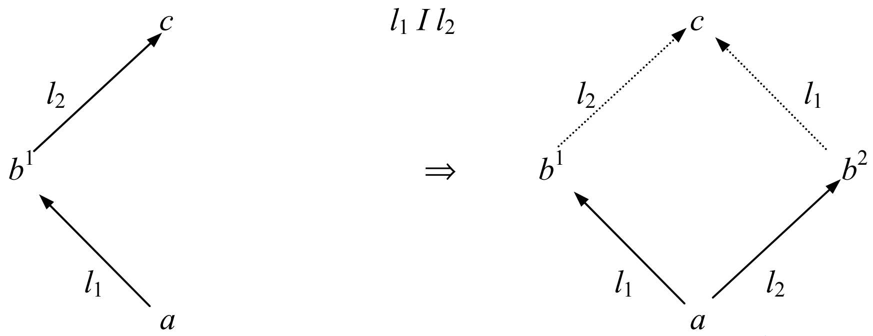

Figura 6: Condiția 4. pentru sisteme cu tranziții asincrone

Condiția 2. spune că sistemul cu tranziții din fundal trebuie să fie determinist. (l nu poate fi un generator aleator). Condițiile 3. şi 4. descriu confluența tranzițiilor venind din acțiuni independente (figurile 5 şi 6).

Morfismele f vor fi morfisme de sisteme cu tranziții care păstrează relația de independență I, adică:

$$m\ I^1\ n \Rightarrow f(m)\ I^2\ f(n)$$

şi astfel sistemele cu tranziții asincrone devin o categorie, notată **ATS**.

4 Automate pentru urme

Automatele pentru urme sunt asemănătoare cu sistemele cu tranziții, însă sunt mult mai generale. Ele au fost larg folosite ca şi modele operaționale pentru rețelele non-deterministe pentru flux de date ([Kah74], [KM77]).

Definiția 6 *Un **automat pentru urme** este un tuplu $TA = (\Sigma, L, A)$ unde:*

- *Σ este o mulțime de stări.*
- *L este un alfabet concurent, adică o mulțime de evenimente echipată cu o relație binară simetrică şi ireflexivă $\|_L$, numită relația de concurență.*
- *$A = \{ \xrightarrow{l} \subseteq \Sigma \times \Sigma / l \in L_\varepsilon \}$ este mulțimea de tranziții satisfăcând următoarele condiții:*
  1. *$a \xrightarrow{\varepsilon} b^1 \Leftrightarrow a = b$*
  2. *dacă $a \xrightarrow{l} b^1$ şi $a \xrightarrow{l} b^2$ atunci $b^1 = b^2$ (similar cu condiția 2 de la ATS)*

3. *pentru toate stările* a *şi evenimentele* m, n, *dacă* $m \parallel_L n$, $a \xrightarrow{m} b$ *şi* $a \xrightarrow{n} c$, *atunci pentru o anume stare* d *există tranziţiile* $c \xrightarrow{m} d$ *şi* $b \xrightarrow{n} d$ *(similar cu condiţia 3. de la ATS)*

Se poate defini o **echivalenţă a permutărilor** pe urmele acestor TA, care să echivaleze urmele care sunt "esenţial aceleaşi":
dacă două acţiuni m şi n sunt independente, adică $m \parallel_L n$, atunci *mn* şi *nm* sunt doar două vederi secvenţiale ale aceleiaşi execuţii paralele şi deci ele pot fi egalate (echivalate).

Această echivalenţă este cea mai mică congruenţă relativ la concatenarea urmelor finite, astfel încât:
$a \xrightarrow{m} b \xrightarrow{n} c$ *şi* $a \xrightarrow{n} d \xrightarrow{m} c$ *sunt* $\sim -$ *legate dacă* $m \parallel_L n$
Relaţiile echivalenţă între urme sunt legate de problemele de planificare din sistemele concurente şi de serializabilitate.

5 Sisteme cu tranziţii concurente

Au fost introduse ca şi model operaţional pentru adevărata concurenţă în [Sta89]:

Definiţia 7 *Un graf cu identităţi este G = (O, A, dom, cod, id), unde:*
- *O este mulţimea stărilor proprii*
- *A este mulţimea tranziţiilor proprii*
- $dom : A \to O$ *mapează tranziţiile pe stările lor iniţiale*
- $cod : A \to O$ *mapează tranziţiile pe stările lor finale*
- $id : O \to A$ *mapează fiecare* $s \in O$ *într-o tranziţie specială (tranziţia nulă din secţiunea precedentă)* id_s *astfel încât* $dom(id_s) = s$ *şi* $cod(id_s) = s$, $s \xrightarrow{id_s} s$

Definiţia 8 *Un graf extins* $G^\# = (O^\#, A^\#, dom, cod, id)$ *este un graf cu identităţi, unde:*
- $O^\# = O \cup \{\Omega\}$, Ω *nu aparţine lui* O.
- $A^\# = A \cup \{\ \omega_q \mid q \in O^\#\}$ *şi* $dom(\omega_q) = q$, $cod(\omega_q) = \Omega$

Definiția 9 *Fiind dat un graf G, mulțimea tranzițiilor* ***coinițiale****, Coin(G), este mulțimea perechilor (t, u) de tranziții t și u ale lui G care au aceleași stări de start, dom(t) = dom(u).*

Definiția 10 *Un* ***sistem cu tranziții concurente*** *CTS = (G, $\uparrow$) este:*

- *G = (O, A, dom, cod, id) este un graf cu identități*
- *$\uparrow : Coin(G^{\#}) \to A^{\#}$ este operația reziduală*

care verifică următoarele condiții:

1. *pentru toți $t \in A^{\#}$ și $u \in A^{\#}$:*
   a. *dom $(t \uparrow u)$ = cod (u)*
   b. *cod $(t \uparrow u)$ = cod $(u \uparrow t)$*
2. *pentru toți $t : q \to r \in A^{\#}$:*
   a. *$id_q \uparrow t = id_r$*
   b. *$t \uparrow id_q = t$*
   c. *$t \uparrow t = id_r$*
3. *pentru toți coinițialii $t, u, v \in A^{\#}$:*
   $(v \uparrow t) \uparrow (u \uparrow t) = (v \uparrow u) \uparrow (t \uparrow u)$ (axioma "cubului", figura 7)
4. *pentru toți coinițialii $t, u \in A^{\#}$:*
   dacă $t \uparrow u$ și $u \uparrow t$ sunt ambele identități, atunci $t = u$.

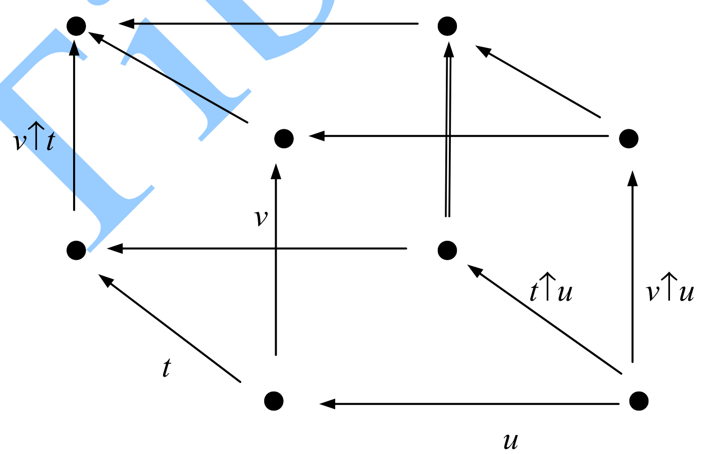

Figura 7: Relații între modele ale concurenței

În figura 7 $\Uparrow$ înseamnă $(v \uparrow t) \uparrow (u \uparrow t) = (v \uparrow u) \uparrow (t \uparrow u)$. Tranzițiile coinițiale t, u ale lui CTS se numesc **consistente** dacă $t \uparrow u$ este o tranziție proprie (adică este în A, nu în $A^{\#}$).

Definiția 11 *Un morfism între sistemele cu tranziții concurente* $CTS_1 = (G_1, \uparrow_1)$ *și* $CTS_2 = (G_2, \uparrow_2)$ *este o pereche de aplicații*

$$\rho = (\rho_O, \rho_A), \quad \rho_O: O_1 \to O_2, \quad \rho_A: A_1 \to A_2$$

astfel încât

1. $dom_2 \circ \rho_A = \rho_O \circ dom_1$
2. $cod_2 \circ \rho_A = \rho_O \circ cod_1$
3. $\rho_A \circ id_1 = id_2 \circ \rho_O$ *(simulare a grafului de tranziție a stărilor din fundal)*
4. *dacă* t *și* u *sunt tranziții consistente proprii ale lui* G_1 *atunci* $\rho(t \uparrow_1 u) = \rho(t) \uparrow_2 \rho(u)$

Extindem morfismele la tranziții neproprii luând $\rho(\omega_q) = \omega_{\rho(q)}$. Aceasta face din sistemele cu tranziții concurente o categorie, pe care o notăm **<u>CTS</u>**.

6 Sisteme cu tranziții cu independență

Sunt sisteme cu tranziții îmbogățite cu o noțiune de concurență pentru a modela adevărata concurență. Relația de independență dintre acțiuni este acum o funcție și de stări, ceea ce rafinează modelul. Au fost introduse în [SNW94] și pot fi considerate ca o variație a automatelor concurente [Sta89] și a sistemelor cu tranziții asincrone [Bed88].

Definiția 12 *Un sistem cu tranziții cu independență este o structură* $TSI = (\Sigma, i, L, A, I)$, *unde* (Σ, i, L, A) *este un sistem cu tranziții și* $I \subseteq A \times A$ *este o relație simetrică și ireflexivă astfel încât:*

1. $(a \xrightarrow{l} b^1) \sim (a \xrightarrow{l} b^2) \Rightarrow b^1 = b^2$ *(condiția 2. de la ATS).*
2. $(a \xrightarrow{l} b^1)\ I\ (a \xrightarrow{l} b^2) \Rightarrow$
   $\exists\ c \mid (a \xrightarrow{l} b^1)\ I\ (b^1 \xrightarrow{m} c) \wedge (a \xrightarrow{l} b^2)\ I\ (b^2 \xrightarrow{l} c)$
   (similară condiției 3. de la ATS).
3. $(a \xrightarrow{l} b^1)\ I\ (b^1 \xrightarrow{m} c) \Rightarrow$
   $\exists\ b^2 \mid (a \xrightarrow{l} b^1)\ I\ (a \xrightarrow{m} b^2) \wedge (a \xrightarrow{m} b^2)\ I\ (b^2 \xrightarrow{l} c)$
   (similară condiției 4. de la ATS).

4. $(a \xrightarrow{l} b^1) \sim (b^2 \xrightarrow{l} c)\ I\ (d^1 \xrightarrow{m} d^2) \Rightarrow$

$$(a \xrightarrow{l} b^1)\ I\ (d^1 \xrightarrow{m} d^2)$$

unde ~ este cea mai mică echivalență pe tranziții care include relația R, definită de

$$(a \xrightarrow{l} b^1)\ R\ (b^2 \xrightarrow{l} c) \Leftrightarrow \begin{cases} (a \xrightarrow{l} b^1)\ I\ (a \xrightarrow{m} b^2) \text{ şi} \\ (a \xrightarrow{l} b^1)\ I\ (b^1 \xrightarrow{m} c) \text{ şi} \\ (a \xrightarrow{m} b^2)\ I\ (b^2 \xrightarrow{l} c) \end{cases}$$

Morfismele sunt morfisme totale ale sistemului cu tranziții de fundal care păstrează independența: o pereche de aplicații (σ,λ), cu σ o aplicație între stări şi λ o aplicație între etichete astfel încât

$$(a \xrightarrow{l} b^1) \in A^1 \Rightarrow (\sigma(a) \xrightarrow{\lambda(l)} \sigma(b^1)) \in A^2$$

$$(a \xrightarrow{l} b^1)\ I^1\ (c \xrightarrow{m} d^1) \Rightarrow (\sigma(a) \xrightarrow{\lambda(l)} \sigma(b^1))\ I^2\ (\sigma(c) \xrightarrow{\lambda(m)} \sigma(d^1))$$

Notăm cu **<u>TSI</u>** categoria sistemelor cu tranziții cu independență cu morfisme totale.

7 Rețele Petri

Sunt modele sistem şi ale concurenței adevărate. Pot fi gândite ca "sisteme cu tranziții distribuite": formalismul pe care se bazează este puțin diferit şi se axează pe "jocul jetonului". Se pot folosi pentru a da o semantică a adevăratei concurențe lui CCS ([DNM88]).

Definiția 13 *O rețea Petri N = (P, T, pre, post) constă din:*

- *P este o mulțime de locuri;*
- *T este o mulțime de tranziții;*
- *pre : T* $\rightarrow P^{\sim}$ *este aplicația pre-condiție, unde* $P^{\sim}$ denotă mulțimea multiset-urilor lui P.
- *post : T* $\rightarrow P^{\sim}$ *este aplicația post-condiție.*

Rețelele Petri există în mai multe versiuni. Cea din figura 8 este adesea numită P/T-rețele (eng.: Place/Transition nets). Locurile reprezintă resurse care pot fi folosite de unul sau mai multe procese. Aceasta se formalizează prin noțiunea de marcaj: un marcaj este o multimulțime de locuri.

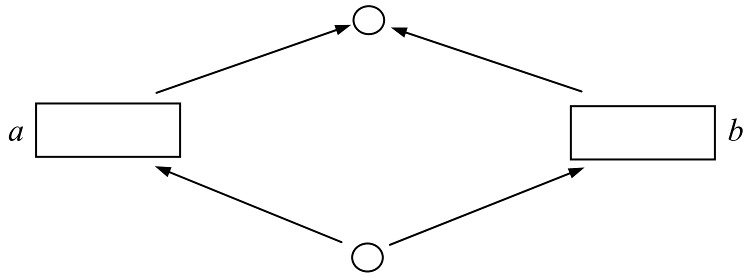

Figura 8: Rețea Petri - excludere mutuală între a şi b

Aplicația pre-condiție descrie cum "consumă" tranzițiile resursele. Aplicația post-condiție arată cum "crează" tranzițiile noi resurse. Aceasta defineşte o relație de tranziție între marcaje:

Dacă M_1 şi M_2 sunt două marcaje ale unei rețele N, şi $t \in T$ este o tranziție a lui N, scriem $M_1 t \rangle M_2$ pentru "t se declanşează din M_1 în M_2" dacă şi numai dacă $\exists M_3 \in P^{\sim}$ astfel încât

$$M_1 = M_3 + pre(t) \quad \text{şi} \quad post(t) + M_3 = M_2$$

Figura 9: Rețea Petri – o execuție adevărat concurentă între a şi b

În general, rețelele Petri se reprezintă grafic astfel: locurile sunt cercuri, iar tranzițiile sunt dreptunghiuri. Pentru a reprezenta aplicația pre-condiție, se folosesc arce de la locuri la tranziții, cu o multiplicitate potrivită. Aplicația post-condiție se reprezintă prin arce de la tranziții la locuri, cu o multiplicitate potrivită. Marcajele sunt jetoane puse în locuri potrivite.

Un caz particular de rețele Petri este reprezentat de rețelele C/E (rețele Condiție/Eveniment): locurile sunt acum condiții care pot fi doar adevărate sau false. Adevărul este identificat cu un marcaj ocupând un loc, iar falsul prin nici un marcaj ocupând un loc, aşadar putem avea cel mult un marcaj per condiție.

Un caz special al rețelelor C/E este o rețea de apariții (ocurențe), în care o tranziție poate fi declanşată exact o dată. Toate aceste tipuri de rețele Petri se pot considera categorii cu următoarea noțiune de morfism. Vom considera că ele sunt date cu un marcaj inițial M_0, aşa că putem scrie rețeaua ca şi $N = (P, T, M_0, pre, post)$.

Definiția 14 *Fie două rețele Petri* $N^1 = (P^1, T^1, M_0^1, pre^1, post^1)$ *şi* $N^2=(P^2, T^2, M_0^2, pre^2, post^2)$. *Un morfism* $f: N^1 \to N^2$ *este o pereche* (β, η), $\beta: P^1 \to P^2$ şi $\eta: T^1 \to T^2$ *astfel încât:*

1. $\beta M_0^1 = M_0^2$
2. $\beta \circ pre^1(e) = pre^2 \circ \eta(e)$
3. $\beta \circ post^1(e) = post^2 \circ \eta(e)$

Ca de obicei, ne limităm la morfisme care sunt funcții totale. Obținem o categorie izomorfă de rețele Petri dacă toate rețelele sunt ridicate (eng.: lifted) prin adăugarea unui eveniment nul *, cu $pre(*) = post(*) = \emptyset$.

Ca o consecință directă a definiției de mai sus, morfismele păstrează relația de tranziție: dacă $M_1^1\ e\rangle\ M_2^1$ în N^1, atunci $\beta M_1^1\ \eta(e)\rangle\ \beta M_2^1$ în N^2

Notăm cu **PT** categoria rețelelor Petri cu morfisme ca în definiția de mai sus. Rețelele Petri au inspirat majoritatea modelelor bazate pe evenimente.

Bibliografie

[Bed88] **M. A. Bednarczyk** - *Categories of asynchronous systems*, PhD thesis, University of Sussex, 1988.

[DNM88] **P. Degano, R. de Nicola, and U. Montanari** - *A distributed operational semantics for CCS based on condition/event systems*, Acta Informatica, 26(1/2):59:91, 1988

[Kah74] **G. Kahn** - *The semantics of a simple language for parallel programming*, in J.L. Rosenfeld, editor, Information Processing, 74:471-475, North-Holland, 1974.

[KM77] **G. Kahn and D. B. MacQueen** - *Coroutines and networks of parallel processes*, in B. Gilchrist, editor, Information Processing, 77:993-998, North-Holland, 1977.

[LD00] **L. Luca and I. Despi** – *Modele matematice pentru concurență*, Analele Universității "Tibiscus", vol. X: 101-106, Timişoara, 2000

[McL71] **S. Mac Lane** – *Categories for the working mathematician*, Springer-Verlag, 1971

[Shi85] **M.W. Shie**lds - *Concurrent machines*, Computer Journal 28, 1985.

[SNW94] **V.Sassone, M. Nielsen and G. Winskel** – *Relationships between models of concurrency*, in Proceedings of the REX 3 school and symposium, 1994.

[Sta89] **A. Stark** - *Concurrent Transition Systems*, Theoretical Computer Science, 64:221-269, 1989