Lecţia 1-3:

Retele de fluxul datelor

G Stefănescu — Universitatea Bucureşti

Dezvoltarea Aplicatiilor Interactive, Sem.2
Februarie-Iunie 2010

Retele de fluxul datelor

Cuprins:

- *Generalitati*
- Retele sincrone
- Functii de procesare a suvoaielor
- Retele asincrone deterministe
- Retele asincrone nedeterministe

Generalitati

Generalitati:

- O *reţea de fluxul datelor* (DFN) este o reţea de *procese concurente* care comunică prin *canale FIFO*.
- DFN-urile formează un model de *concurenţă adevărată* unde:
  - Starea globală este împărţită în părţi, numite *canale*.
  - Canalele sunt considerate cozi FIFO nemărginite, modelate prin *şuvoaie / stream-uri* (un şuvoi este o secvenţă finită ori infinită de date care reţine istoria datelor comunicate pe un canal specific);
  - Celulele din reţea consumă date din canalele lor de intrare şi produc date pe canalele de ieşire.

Generalitati

Generalitati:

- DFN-urile pot fi clasificate astfel:
  - după *timp*: *sincrone* ori *asincrone*;
  - după *comportament*: *deterministe* ori *nedeterministe*;

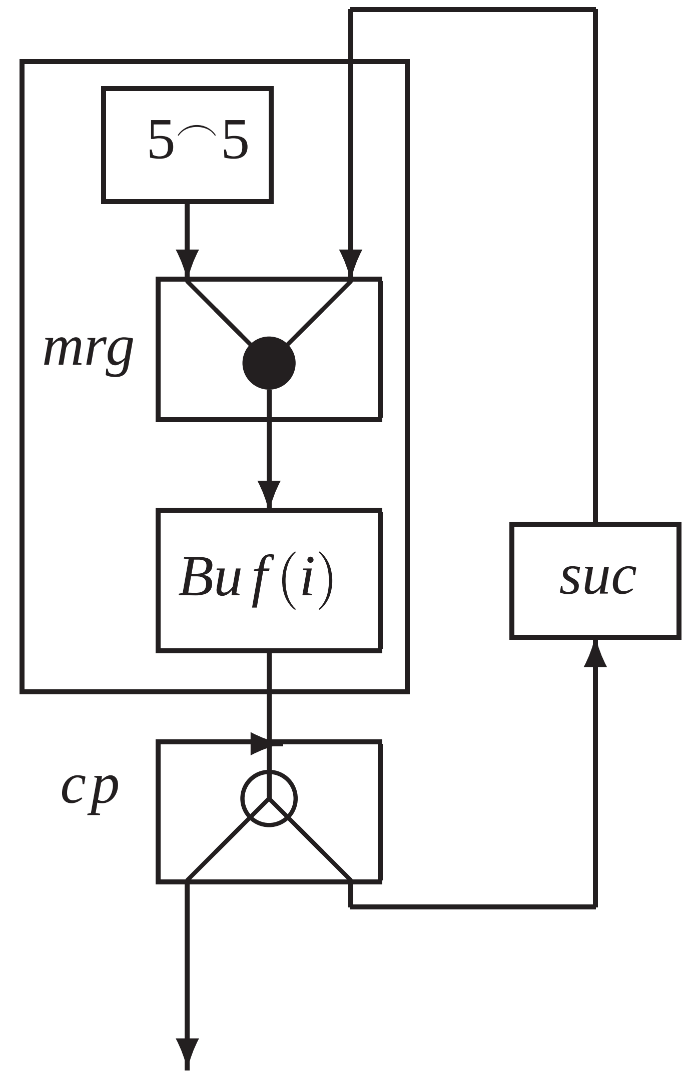
- DFN-urile *sincrone* pot fi folosite pentru procesoare ori circuite cu ceas global, având o teorie mai simplă.

- DFN-urile *asincrone* sunt utile pentru arhitecturi paralele unde un ceas global este greu de menţinut. Aici:
  - DFN-urile *deterministe* au o teorie mai *simplă* în care feedback-ul este definit ca punct fix minimal.
  - cele *nedeterministe* prezintă o *anomalie de timp*, i.e., semantică naturală cu şuvoaie *nu este compoziţională* şi trebuie rafinată.

..Generalitati

Exemplu: (*reţea asincronă, nedeterministă*)

- $5^\frown 5$ generează un 5, apoi altul, apoi stop;
- *mrg* este un merge special care combină (nondeterminist) $5^\frown 5$ cu prima data din canalul din dreapta (dacă există), apoi stop;
- *cp* copiază intrarea pe ambele ieşiri şi repetă acest comportament;
- *suc* consumă o dată, produce succesorul şi repetă acest comportament;
- $Buf(1)$ lasă să treacă o dată, apoi alta, apoi stop;
- $Buf(2)$ consumă o dată, apoi alta, apoi produce datele consumate (păstrând ordinea), apoi stop.



 G Stefanescu

..Generalitati

Exemplu (cont.)

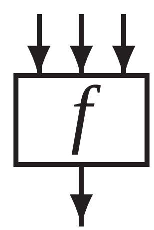
- Este un exemplu tipic de reţea DFN *asincronă* şi *nondeterministă*;
- Asincronicitatea se referă la lipsa unui ceas global care să forţeze operarea sincronă a tuturor celulelor;
- Aici celulele *operează* independent *când au date* la intrare;
- Nondeterminismul este prezent în defnirea comportamentului merge-ului *mrg*.

Exemplu (cont.)

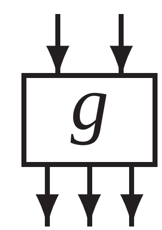
- Comportament în DFN-ul cu $Buf(1)$:
  - $5^\frown 5$ generează un 5 care: este pasat de *mrg* lui $Buf(1)$, copiat la ieşirea lui $Buf(1)$ şi pasat la ieşire şi lui *suc*;
  - *suc* consumă 5-ul şi produce un 6 care este pasat de *mrg* lui $Buf(1)$ şi pasat la ieşire şi lui *suc*;
  - pe scurt, DFN-ul cu $Buf(1)$ poate produce la ieşire secvenţa $5^\frown 6$.
- In DFN-ul cu $Buf(2)$ comportamentul de sus nu este posibil:
  - reţeaua cu $Buf(2)$ poate genera la ieşire doar secvenţa $5^\frown 5$.

Operatii cu retele:

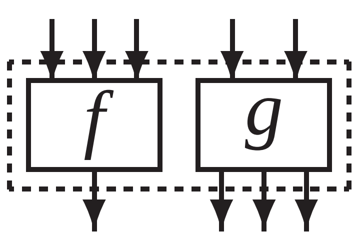
- Operatiile de bază sunt *juxtapunerea* "$\star$", *compunerea* "$\cdot$" şi *feedback-ul* "$\uparrow$".
- Ele au următoarea interpretare:



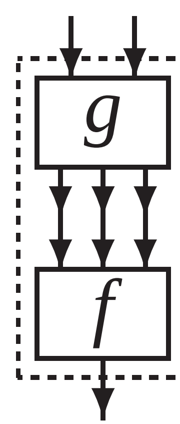
$f : 3 \longrightarrow 1$



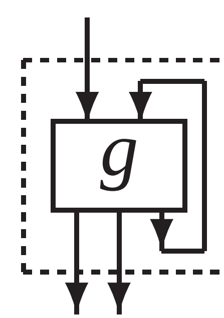
$g : 2 \longrightarrow 3$



$f \star g : 5 \longrightarrow 4$



$g \cdot f : 2 \longrightarrow 1$



$g \uparrow^1 : 1 \longrightarrow 2$

..Reprezentare textuala a retelelor

Reprezentarea relatiilor finite:

- Se poate demonstra că orice relaţie finită se obţine cu juxtapunere şi compunere din nişte relaţii elementare: *identităţi* $\mathsf{I}_a : a \longrightarrow a$, *transpoziţii* ${}^a\mathsf{X}^b : a+b \longrightarrow b+a$, *ramificări* $\wedge^a_k : a \longrightarrow ka$, şi *idenitificări* $\vee^k_a : ka \longrightarrow a$.
- In desen sunt ilustrate I_4, ${}^2\mathsf{X}^1$, $\wedge^3 (=_{\text{def}} \wedge^3_2)$, $\perp^3 (=_{\text{def}} \wedge^3_0)$, $\vee_2 (=_{\text{def}} \vee^2_2)$, $\top_2 (=_{\text{def}} \vee^0_2)$

$\mathsf{I}_4 : 4 \longrightarrow 4$ ${}^2\mathsf{X}^1 : 3 \longrightarrow 3$

$\wedge^3 : 3 \longrightarrow 6$ $\perp^3 : 3 \longrightarrow 0$ $\vee_2 : 4 \longrightarrow 2$ $\top_2 : 0 \longrightarrow 2$

..Reprezentare textuala a retelelor

Expresii: *Expresiile de reţele* se obţin astfel:

- ca elemente de plecare se folosesc *variabile* $x : a \longrightarrow b$ şi *relaţii finite* $f : a \longrightarrow b$;
- se aplică operaţiile de *juxtapunere*, *compunere*, şi *feedback*.

Exemplu:

- Desenul anterior se reprezintă cu expresia

$$[((5^\frown 5) \star \mathsf{I}_1) \cdot mrg \cdot Buf(i) \cdot cp \cdot (\mathsf{I}_1 \star suc)] \uparrow^1$$

Algebra de retele

- Algebra *BNA* (Basic Network Algebra) este dată de ecuaţiile:

I. Axiome fara feedback

B1 $f\star(g\star h) = (f\star g)\star h$

B2 $\mathsf{I}_0\star f = f = f\star\mathsf{I}_0$

B3 $f\cdot(g\cdot h) = (f\cdot g)\cdot h$

B4 $\mathsf{I}_a\cdot f = f = f\cdot\mathsf{I}_b$

B5 $(f\star f')\cdot(g\star g') = (f\cdot g)\star(f'\cdot g')$

B6 $\mathsf{I}_a\star\mathsf{I}_b = \mathsf{I}_{a+b}$

B7 ${}^a\mathsf{X}^b\cdot{}^b\mathsf{X}^a = \mathsf{I}_{a+b}$

B8 ${}^a\mathsf{X}^{b+c} = ({}^a\mathsf{X}^b\star\mathsf{I}_c)\cdot(\mathsf{I}_b\star{}^a\mathsf{X}^c)$

B9 $(f\star g)\cdot{}^c\mathsf{X}^d = {}^a\mathsf{X}^b\cdot(g\star f)$
pentru $f : a\to c,\quad g : b\to d$

II. Axiome pentru feedback

R1 $f\cdot(g\uparrow^c)\cdot h = ((f\star\mathsf{I}_c)\cdot g\cdot(h\star\mathsf{I}_c))\uparrow^c$

R2 $f\star g\uparrow^c = (f\star g)\uparrow^c$

R3 $(f\cdot(\mathsf{I}_b\star g))\uparrow^c = ((\mathsf{I}_a\star g)\cdot f)\uparrow^d$
pentru $f : a+c\to b+d,\quad g : d\to c$

R4 $f\uparrow^0 = f$

R5 $(f\uparrow^b)\uparrow^a = f\uparrow^{a+b}$

R6 $\mathsf{I}_a\uparrow^a = \mathsf{I}_0$

R7 ${}^a\mathsf{X}^a\uparrow^a = \mathsf{I}_a$

- Algebra de reţele *NA* (Network Algebra) se obţine din BNA cu axiome suplimentare pentru $\wedge_k, \vee^k$.

Retele de fluxul datelor

Cuprins:

- Generalitati
- *Retele sincrone*
- Functii de procesare a suvoaielor
- Retele asincrone deterministe
- Retele asincrone nedeterministe

Retele sincrone

Retele sincrone:

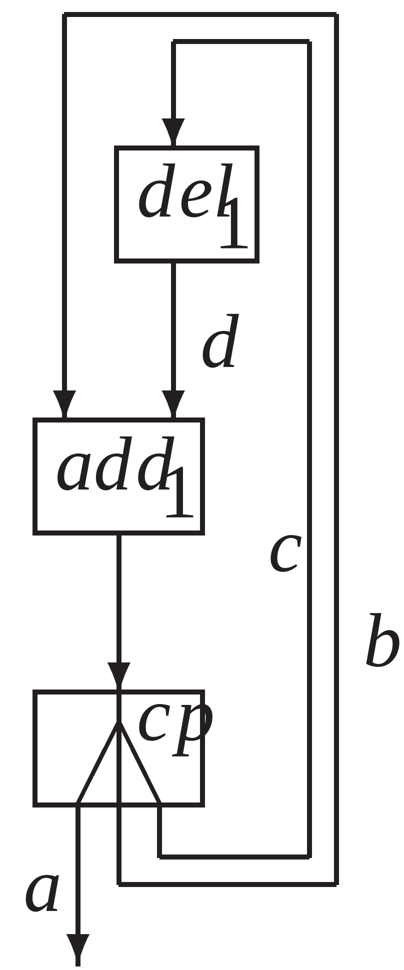
- Retele sincrone folosesc un *ceas global*;
- Intre două ticuri de ceas, o celulă *consumă* o dată din fiecare canal de intrare (produsă în ciclul anterior), *calculează*, şi *produce* o dată pe canalele de ieşire;
- Fiecare celulă produce o *întâziere de un ciclu* de ceas între intrări şi ieşiri, dar firele de conexiune nu produc întâzieri;
- Fiecare celulă conţine un tuplu de *date iniţiale* pentru canalele de ieşire.
- Comportament tipic: După o perioadă de *încălzire*, când ieşirea este irelevantă, reţeaua produce rezultatele procesării intrărilor.

..Retele sincrone

Exemplu: (*secvenţa Fibonacci*)

- Considerăm reţeaua ($cp = \overset{\circ}{\wedge}{}^1_3$)

$$E = [(\mathsf{I}_1 \star del) \cdot add \cdot \overset{\circ}{\wedge}{}^1_3] \uparrow^2 : 0 \to 1$$



- Comportamenul celulelor $\mathsf{I}_1 : 1 \to 1, \quad \overset{\circ}{\wedge}{}^1_3 : 1 \to 3,$ $del : 1 \to 1, \quad add : 2 \to 1$ este:

  $-\mathsf{I}_1(x) = x;$
  $-\overset{\circ}{\wedge}{}^1_3(x) = (x,x,x);$
  $-del(x(0)^\frown x(1)^\frown \ldots) = 1^\frown x(0)^\frown x(1)^\frown \ldots;$
  $-add(x(0)^\frown x(1)^\frown \ldots, y(0)^\frown y(1)^\frown \ldots)$
  $\quad = 1^\frown(x(0)+y(0))^\frown(x(1)+y(1))^\frown \ldots.$

- Rezultatul, care nu depinde de intrare, este secvenţa Fibonacci $a(n+2) = a(n+1) + a(n)$, cu $a(0) = 1$ şi $a(1) = 2$, anume:

$$||E||() = 1^\frown 2^\frown 3^\frown 5 \ldots$$

Semantica

Semantica I. Procedură *monolitică* pentru semantică:

- folosim o *variabilă* pentru *fiecare canal* (arc);
- canalele care nu-s de intrare se iniţializează cu *valorile iniţiale* ale celulelor din care pleacă;
- se calculează *recursiv* noile valori de la un ciclu la altul;

Pe exemplu, luăm variabilele a, b, c, d ca în desen; atunci:

$timp$	0	1	2	3	4	5	6	...
a	1	2	3	5	8	13	21	
b	1	2	3	5	8	13	21	
c	1	2	3	5	8	13	21	
d	1	1	2	3	5	8	13	

Relaţia de recursivitate este:

$$a(n+1) = b(n+1) = c(n+1) = a(n) + d(n);\ d(n+1) = c(n)$$

..Semantica

Semantica II. Procedură *modulară* pentru semantică:

- *descompunem reţeaua* în componente folosind operaţiile de la algebra de reţele $\star, \cdot, \uparrow$;
- calculăm inductiv *transformarea pe şuvoaie* a fiecărei *componente*:
  - *juxtapunerea* $\star$ şi *compunerea* $\cdot$ au definiţii naturale, simple pe (tuple) de şuvoaie;
  - *feedback-ul* $\uparrow$ este este un *feedback multiplicativ* calculat folosind valoarea iniţială pe arcul de feedback;
  - Notă: feedback-ul *nu este definit* când între intrare şi ieşire nu există măcar o celulă ce produce întârzieri (pe constante ca $\curlywedge_3^1$ feedback-ul nu este definit);

 G Stefanescu

Feedback multiplicativ

Feedback multiplicativ: Un feedback într-un mediu multiplicativ

$$f : D^{m+p} \to D^{n+p} \quad \mapsto \quad f\uparrow^p : D^n \to D^n$$

se poate defini prin una din următoarele metode:

Neconstructiv (pentru specificaţii):

$$(x,y) \in f\uparrow^p \quad dnd \quad \exists z.((x,z),(y,z)) \in f$$

Constructiv (iterativ), folosind o relaţie de ordine pe D, ce poate fi:

- *temporală* - folosim şuvoaie ($D = S^\omega$) şi ordinea "prefix" indusă (pentru reţele de fluzul datelor);
- *spaţială* - folosim submulţimi ($D = 2^S$) şi ordinea indusă (pentru verificarea programelor, cu mecanismul de "cea mai slabă precondiţie").

..Semantica

Pe exemplu:

- $(\mathsf{I}_1 \star del)(x(0)^\frown x(1)^\frown \ldots, y(0)^\frown y(1)^\frown \ldots)$
  $= (x(0)^\frown x(1)^\frown \ldots, 1^\frown y(0)^\frown y(1)^\frown \ldots)$

- $[(\mathsf{I}_1 \star del) \cdot add](x(0)^\frown x(1)^\frown \ldots, y(0)^\frown y(1)^\frown \ldots)$
  $= 1^\frown x(0) + 1^\frown x(1) + y(0)^\frown \ldots$

- $[(\mathsf{I}_1 \star del) \cdot add \cdot \overset{\circ}{\wedge}{}^1_3](x(0)^\frown x(1)^\frown \ldots, y(0)^\frown y(1)^\frown \ldots)$
  $= (1^\frown x(0) + 1^\frown x(1) + y(0)^\frown \ldots, 1^\frown x(0) + 1^\frown x(1) + y(0)^\frown \ldots,$
  $1^\frown x(0) + 1^\frown x(1) + y(0)^\frown \ldots)$

- $\{[(\mathsf{I}_1 \star del) \cdot add \cdot \overset{\circ}{\wedge}{}^1_3] \uparrow^1\}(x(0)^\frown x(1)^\frown \ldots)$
  $= (1^\frown x(0) + 1^\frown x(1) + 1^\frown \ldots, 1^\frown x(0) + 1^\frown x(1) + 1^\frown \ldots)$

- $\{\{[(\mathsf{I}_1 \star del) \cdot add \cdot \overset{\circ}{\wedge}{}^1_3] \uparrow^1\} \uparrow^1\}$
  $= 1^\frown 2^\frown 3^\frown \ldots$

Pentru feedback identificam suvoaiele canalelor respective; e.g., pentru primul feedback, $y(0)^\frown y(1)^\frown y(2)^\frown \ldots = 1^\frown x(0) + 1^\frown x(1) + y(0)^\frown \ldots$*.*

Retele de fluxul datelor

Cuprins:

- Generalitati
- Retele sincrone
- *Functii de procesare a suvoaielor*
- Retele asincrone deterministe
- Retele asincrone nedeterministe

Functii de procesare a suvoaielor

Modelul $\mathsf{SPF}(M)$, constând din *funcţii de procesare a şuvoaielor*, se construieşte astfel:

Date: D - o mulţime de date atomice;

Suvoaie infinite: - $D^\infty = \{f : \mathbb{N} \to D\}$; un element $x \in D^\infty$ se poate reprezenta ca şuvoi $x(0)^\frown x(1)^\frown x(2)^\frown \ldots$;

Suvoaie: - secvenţe finite ori infinite de date din D, anume $M = D^\omega$, unde $D^\omega =_{def} D^* \cup D^\infty$;

Concatenare: $x^\frown y$ reprezintă concatenarea lui x cu y; rezultatul este x dacă x este infinit;

Ordinea prefix: $x \sqsubseteq y \quad =_{def} \quad \exists z \in M.\ x^\frown z = y$;

..Functii de procesare a suvoaielor

Funcţie de procesare a şuvoailor: o funcţie $f : M^m \longrightarrow M^n$; spunem ca are tipul $m \mapsto n$;

Funcţii monotone: o funcţie $f : M^m \longrightarrow M^n$ este *monotonă* dacă $\forall x,y \in M^m.\, x \sqsubseteq y \quad \Rightarrow \quad f(x) \sqsubseteq f(y)$;

Supremum: pentru $S \subseteq M$, $\bigsqcup S$ notează *supremum*-ul lui S în M (cel mai mic majorant), dacă el există; altfel spus: (1) $\forall s \in S.\, s \sqsubseteq \bigsqcup S$ şi (2) dacă $\forall s \in S.\, s \sqsubseteq u$, atunci $\bigsqcup S \sqsubseteq u$;

Submulţimi direcţionate: $S \subseteq M$ este *direcţionată* dacă $\forall s', s'' \in S.\, \exists \bigsqcup \{s', s''\}$; în M, orice mulţime direcţionată are supremum;

Funcţii continue: o funcţie $f : M^m \longrightarrow M^n$ este *continuă* dacă este monotonă şi pentru orice mulţime direcţionată $S \subseteq M$, $f(\bigsqcup S) = \bigsqcup \{f(x) : x \in S\}$

..Functii de procesare a suvoaielor

Modelul SPF*:*

$$\mathsf{SPF}(M)(a,b) = \{f : M^a \to M^b : \; f \text{ (prefix) continuă } \}$$

o funcţie $f : M^m \to M^n$; spunem ca are tipul $m \mapsto n$;

Structura de algebră de reţea:

- *Juxtapunere (compunere paralelă):* $(f \star g)(x,y) = (f(x), g(y))$;
- *Compunere secvenţială:* $(f \cdot g)(x) = g(f(x))$;
- *Identitate:* $\mathsf{I}_a(x) = x$, pentru $x \in M^a$;
- *Transpoziţie:* ${}^a\mathsf{X}^b(x,y) = (y,x)$, pentru $x \in M^a, y \in M^b$;
- *Copie:* $\curlywedge^a(x) = (x,x)$, pentru $x \in M^a$;
- *Sink (scurgere):* $\downarrow^a(x) = (\,)$, pentru $x \in M^a$; $(\,)$ este tuplul vid;
- *Sursă goală:* $\uparrow_a(\,) = (\langle\,\rangle_a)$ (a-tuplu de şuvoaie vide);
- *Constanta* $\vee$ rămând *neinterpretată* (în cazul determinist).

 G Stefanescu

..Functii de procesare a suvoaielor

Structura de algebră de reţea (cont.)

- *Feedback:* Pentru $f \in \mathsf{SPF}(M)(a \star c, b \star c)$, feedback-ul $f \uparrow^c \in \mathsf{SPF}(M)(a,b)$ este definit astfel:

  Dacă $x \in M^a$, definim

$$f \uparrow^c (x) = \bigsqcup_{k \geq 1} y_k$$

  cu $y_k \in M^b$ şi $z_k \in M^c$ inductiv definite astfel:

$$(y_1, z_1) = f(x, \langle\ \rangle), \quad (\langle\ \rangle \text{ este şuvoiul vid}) \text{ şi}$$
$$(y_{k+1}, z_{k+1}) = f(x, z_k), \quad \text{pentru } k \geq 1.$$

Notă: Definiţia are sens (f este continuă) şi produce *punctul fix minimal*; e.g., $\overset{\circ}{\wedge}{}^a \uparrow^a = \overset{\bullet}{|}_a$.

..Functii de procesare a suvoaielor

Teoremă: $(\mathsf{SPF}(M), \star, \cdot, \uparrow, \mathsf{I}_a, {}^a\mathsf{X}^b)$ *este BNA.*

Dem.

- Axiomele fără feedback sunt uşor de verificat;
- Din cele pentru feedback, doar axioma R5 (feedback-ul multiplu simultan este echivalent cu cel unar repetat) este un pic mai dificilă;
- Axioms R5 este: pentru $f \in \mathsf{SPF}(M)(a+c+d,\ b+c+d)$:

$$(f \uparrow^{c+d}) = (f \uparrow^{d}) \uparrow^{c}$$

..Functii de procesare a suvoaielor

- Fie $f \in \mathsf{SPF}(M)(a+c+d,\ b+c+d)$.
  Atunci $f \uparrow^{c+d} \in \mathsf{SPF}(M)(a,b)$ este definită prin

  $$(f \uparrow^{c+d})(x) = y$$
  unde $y = \bigsqcup_{k\geq 1} y_k$

  iar y_k, z_k, w_k sunt definite inductiv de

  $$(y_k, z_k, w_k) = f(x, z_{k-1}, w_{k-1}) \quad \text{pentru } k \geq 1$$
  cu $z_0 = \langle\ \rangle, w_0 = \langle\ \rangle$.

  Notăm: $z := \bigsqcup_{k\geq 1} z_k$ şi $w := \bigsqcup_{k\geq 1} w_k$.

..Functii de procesare a suvoaielor

- Similar, $(f\uparrow^d)\uparrow^c \in \mathsf{SPF}(M)(a,b)$ este definit de:

$$((f\uparrow^d)\uparrow^c)(x) = \bar{y}, \quad \text{unde } \bar{y} = \bigsqcup_{i\geq 1}\bar{y}_i$$

iar $\bar{y}_i, \bar{z}_i$ sunt definite inductiv de

$$(\bar{y}_i, \bar{z}_i) = (f\uparrow^d)(x, \bar{z}_{i-1}) \text{ pentru } i \geq 1, \text{ cu } \bar{z}_0 = \langle\ \rangle$$

Din definiţia lui $f\uparrow^d$ există elemente $\tilde{y}_{i,j}, \tilde{z}_{i,j}, \tilde{w}_{i,j}$ pentru $i, j \geq 1$ astfel ca pentru toţi $i \geq 1$:

$$\bar{y}_i = \bigsqcup_{j\geq 1}\tilde{y}_{i,j}, \quad \bar{z}_i = \bigsqcup_{j\geq 1}\tilde{z}_{i,j} \quad \text{şi}$$
$$(\tilde{y}_{i,j}, \tilde{z}_{i,j}, \tilde{w}_{i,j}) = f(x, \bar{z}_{i-1}, \tilde{w}_{i,j-1}) \text{ pentru } j \geq 1 \text{ cu } \tilde{w}_{i,0} = \langle\ \rangle$$

Secvenţele $(\tilde{y}_{i,j}), (\tilde{z}_{i,j})$, şi $(\tilde{w}_{i,j})$ sunt crescătoare în ambii indicii i, j, deci au sens notaţiile

$$\bar{z} := \bigsqcup_{i\geq 1}\bar{z}_i \quad \text{şi}$$
$$\bar{w} := \bigsqcup_{i\geq 1}\bar{w}_i, \quad \text{unde pentru } i \geq 1: \ \bar{w}_i := \bigsqcup_{j\geq 1}\tilde{w}_{i,j}.$$

..Functii de procesare a suvoaielor

- (1) $f\uparrow^{c+d} \sqsubseteq (f\uparrow^{d})\uparrow^{c}$. Demonstrăm prin inducţie că

$$(y_k, z_k, w_k) \sqsubseteq (\tilde{y}_{k,k}, \tilde{z}_{k,k}, \tilde{w}_{k,k}), \quad \forall k \geq 1$$

$$
\begin{array}{lrl}
k = 1: & (y_1, z_1, w_1) & = f(x, z_0, w_0) \\
 & & = f(x, \langle\,\rangle, \langle\,\rangle) \\
 & & = f(x, \bar{z}_0, \tilde{w}_{1,0}) \\
 & & = (\tilde{y}_{1,1}, \tilde{z}_{1,1}, \tilde{w}_{1,1}).
\end{array}
$$

$$
\begin{array}{lrl}
k \Rightarrow k+1: & (y_{k+1}, z_{k+1}, w_{k+1}) & = f(x, z_k, w_k) \\
 & & \sqsubseteq f(x, \tilde{z}_{k,k}, \tilde{w}_{k,k}) \\
 & & \sqsubseteq f(x, \bar{z}_k, \tilde{w}_{k,k}) \\
 & & \sqsubseteq f(x, \bar{z}_k, \tilde{w}_{k+1,k}) \\
 & & = (\tilde{y}_{k+1,k+1}, \tilde{z}_{k+1,k+1}, \tilde{w}_{k+1,k+1}).
\end{array}
$$

..Functii de procesare a suvoaielor

Deci:

$$\begin{aligned}
(f\uparrow^{c+d})(x) &= y\\
&= \textstyle\bigsqcup_{k\geq 1} y_k\\
&\sqsubseteq \textstyle\bigsqcup_{k\geq 1} \tilde{y}_{k,k}\\
&= \textstyle\bigsqcup_{i\geq 1}\bigsqcup_{j\geq 1} \tilde{y}_{i,j}\\
&= \textstyle\bigsqcup_{i\geq 1} \overline{y}_i\\
&= \overline{y}\\
&= ((f\uparrow^{d})\uparrow^{c})(x).
\end{aligned}$$

..Functii de procesare a suvoaielor

- (2) $(f\uparrow^d)\uparrow^c \sqsubseteq f\uparrow^{c+d}$. Demonstrăm prin inducţie dublă că

$$(\tilde{y}_{i,j},\tilde{z}_{i,j},\tilde{w}_{i,j}) \sqsubseteq (y,z,w), \quad \forall i,j \geq 1$$

Notăm că $(y,z,w) = f(x,z,w)$. Intr-adevăr,

$$\begin{aligned} f(x,z,w) &= f(x,\textstyle\bigsqcup_{k\geq 1} z_k,\bigsqcup_{k'\geq 1} w_{k'})\\ &= \textstyle\bigsqcup_{k\geq 1} f(x,z_k,w_k)\\ &= \textstyle\bigsqcup_{k\geq 1}(y_{k+1},z_{k+1},w_{k+1})\\ &= (y,z,w). \end{aligned}$$

Verificăm condiţiile inductive:

$i=1,\ j=1$:
$$\begin{aligned} (\tilde{y}_{1,1},\tilde{z}_{1,1},\tilde{w}_{1,1}) &= f(x,\bar{z}_0,\tilde{w}_{1,0})\\ &= f(x,\langle\,\rangle,\langle\,\rangle)\\ &= (y_1,z_1,w_1)\\ &\sqsubseteq (y,z,w). \end{aligned}$$

 G Stefanescu

..Functii de procesare a suvoaielor

$j \Rightarrow j+1$ (pentru $i=1$): $$\begin{aligned}(\tilde{y}_{1,j+1}, \tilde{z}_{1,j+1}, \tilde{w}_{1,j+1}) &= f(x, \bar{z}_0, \tilde{w}_{1,j}) \\ &= f(x, \langle\ \rangle, \tilde{w}_{1,j}) \\ &\sqsubseteq f(x,z,w) \\ &= (y,z,w)\end{aligned}$$

$i \Rightarrow i+1,\ j=1$: $$\begin{aligned}(\tilde{y}_{i+1,1}, \tilde{z}_{i+1,1}, \tilde{w}_{i+1,1}) &= f(x, \bar{z}_i, \tilde{w}_{i+1,0}) \\ &= f(x, \bigsqcup_{j' \geq 1} \tilde{z}_{i,j'}, \langle\ \rangle) \\ &\sqsubseteq f(x,z,w) \\ &= (y,z,w)\end{aligned}$$

$j \Rightarrow j+1$ (pentru $i+1$): ca în cazul $i=1$.

Retele de fluxul datelor

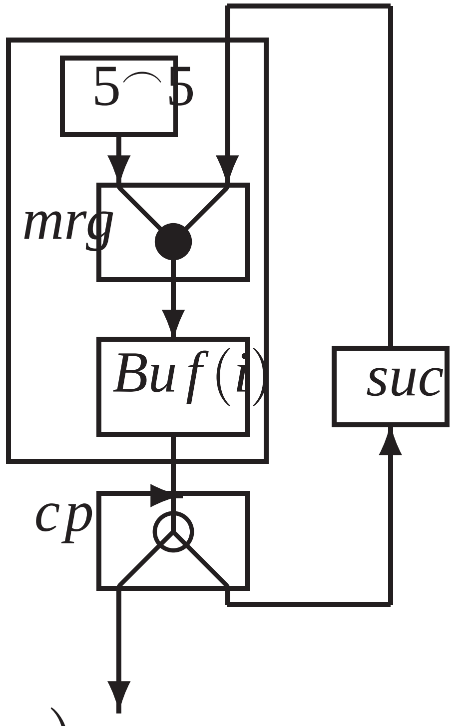
Cuprins:

- Generalitati
- Retele sincrone
- Functii de procesare a suvoaielor
- *Retele asincrone deterministe*
- *Retele asincrone nedeterministe*

Retele asincrone deterministe:

- Studiem reţelele asincrone *deterministe* şi interpretarea lor prin *funcţii de procesare a şuvoialelor*;

- Exemplu: In DFN-ul nedeterminist anterior înlocuim *mrg* cu un *merge determinist*, e.g. $mrg1 = 1^\frown 2^\frown 1$ (transmite, în ordine, a dată de pe canalele 1, apoi 2, apoi 1, apoi stop); obtinem un DFN asincron şi determinist.



- Notăm $F(i) = N(i) \uparrow^1$, unde $N(i) = ((5^\frown 5) \star \mathsf{I}_1) \cdot mrg1 \cdot Buf(i) \cdot cp \cdot (\mathsf{I}_1 \star suc)$

- Interpretarea prin funcţii SPF:
  - calculăm valoarea expresiei de mai sus *în modelul* SPF, plecând cu funcţiile SPF particulare asociate atomilor.

Retele asincrone deterministe

Retele asincrone deterministe (cont.)

- Semantica lui $F(1)$: Calculăm iteraţiile

$$(y_1, z_1) = N(1)(\langle\,\rangle) = (5,6)$$
$$(y_2, z_2) = N(1)(z_1) = N(1)(6) = (5^\frown 6, 6^\frown 7)$$
$$(y_3, z_3) = N(1)(z_2) = N(1)(6^\frown 7) = (5^\frown 6, 6^\frown 7)$$
$$\ldots$$

deci $F(1) = \bigsqcup_{k\geq 1} y_k = 5^\frown 6$.

- Semantica lui $F(2)$: Calculăm iteraţiile

$$(y_1, z_1) = N(2)(\langle\,\rangle) = (\langle\,\rangle, \langle\,\rangle)$$
$$(y_2, z_2) = N(2)(z_1) = N(2)(\langle\,\rangle) = (\langle\,\rangle, \langle\,\rangle)$$
$$\ldots$$

deci $F(2) = \bigsqcup_{k\geq 1} y_k = \langle\,\rangle$.

Algebra de retele

- Algebra *NA* (Network Algebra) este obţinută adăugând la BNA următoarele *axiome slabe* pentru constantele de ramificare:

I. Axiome fara feedback

A0 $\vee_a^1 = \wedge_1^a = \mathsf{I}_a$

A1 *asociativitate*

$\wedge_m^a \cdot (\sum_{j\in[m]} \wedge_{m_j}^a) = \wedge_{\sum_{j\in[m]} m_j}^a$

$(\sum_{i\in[n]} \vee_a^{n_i}) \cdot \vee_a^n = \vee_a^{\sum_{i\in[n]} n_i}$

A2 *comutativitate*: pentru $f : m \longrightarrow m$ bijectie

$\wedge_m^a \cdot f = \wedge_m^a$

$f \cdot \vee_a^m = \vee_a^m$

A3 *comutare identificari si ramificari*

$\vee_a^n \cdot \wedge_m^a = (\sum_{i\in[n]} \wedge_m^a) \cdot \varphi_{n,m} \cdot (\sum_{i\in[m]} \vee_a^n)$

A4 *idempotenţă*

$\wedge_n^a \cdot \vee_a^n = \mathsf{I}_a$

II. Axiome pentru feedback

R8 $\wedge^a \uparrow^a = \top_a$

R9 $\vee_a \uparrow^a = \perp^a$

R10 $[(\mathsf{I}_a \star \wedge^a) \cdot ({}^a\mathsf{X}^a \star \mathsf{I}_a) \cdot (\mathsf{I}_a \star \vee_a)] \uparrow^a = \mathsf{I}_a$

Am notat cu $\sum$ juxtapunerea repetată, iar cu $\varphi_{n,m}$ bijectia naturală care rearanjază n grupuri de m elemente în m grupuri de n elemente.

..Algebra de retele

Dincolo de izomorfism de grafuri: Ne depărtăm de izomorfismul de grafuri adăugând două tipuri de axiome:

Axiomele de comutare tare: Pentru $f : m \to n$,

$$C1 \quad f \cdot \wedge_k^n = \wedge_k^m \cdot (f \star \ldots \star f)$$

$$C2 \quad \vee_m^k \cdot f = (f \star \ldots \star f) \cdot \vee_n^k$$

Regula enzimatică: Pentru $f : m+p \to n+p, \quad g : m+q \to n+q$ şi o relaţie finită $y : p \to q$,

$$Enz \quad f(\mathsf{I}_n \star y) = (\mathsf{I}_m \star y)g \quad \Rightarrow \quad f \uparrow^p = g \uparrow^q$$

 G Stefanescu

Algebra SPF

Algebra SPF**:** In modelul $\mathsf{SPF}(M)$ sunt definite toate constantele de ramificare (copiere) $\wedge_k^m, k \geq 0$, dar numai constantele de identificare $\vee_m^k, k \in \{0,1\}$;

Teorema: $(\mathsf{SPF}(M), \star, \cdot, \uparrow, \mathsf{I}_a, {}^a\mathsf{X}^b, \overset{\circ}{\wedge}, \underset{\circ}{|}, \overset{\bullet}{|})$ *este* teorie Elgot duala, *anume BNA şi satisface:*

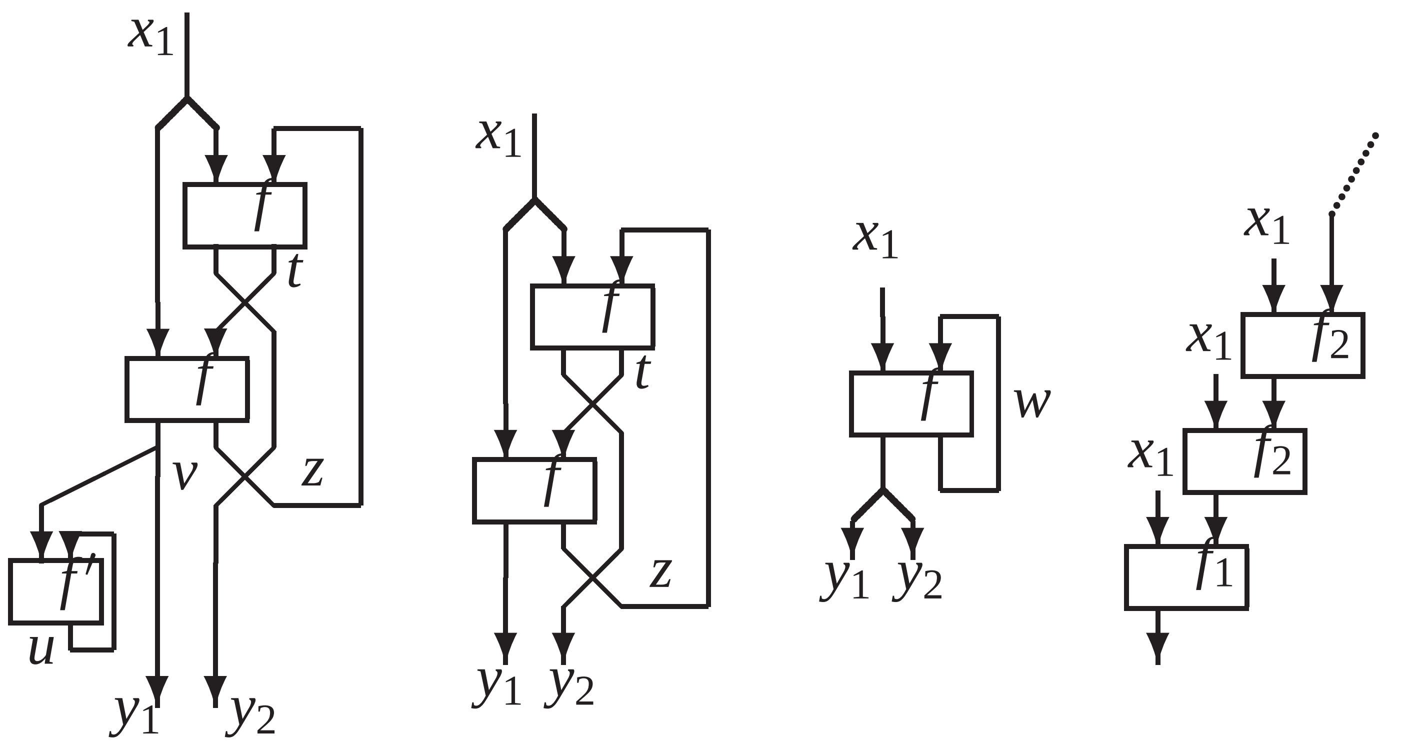
1. *axiomele slabe pentru constantele* $\wedge_k^m, k \geq 0$ *şi* $\vee_m^k, k \in \{0,1\}$, *i.e., A0-A4 (cu restricţiile de mai sus) şi R8 (primele generate de* $\overset{\circ}{\wedge}, \underset{\circ}{|}$, *celelalte de* $\overset{\bullet}{|}$*);*
2. *axiomele tari C*1 *(pentru* $\wedge_k^m$*);*
3. *axioma enzimatică, când relaţia y este inversă de funcţie;*

Consecinţă: *Transformările asociate axiomelor de mai sus sunt corecte dacă se aplică reţelelor* DFN *asincrone, deterministe.*

Transformari algebrice

Transformari algebrice:

- Vom folosi reţelele asincrone, deterministe de mai jos:



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

- Vom vedea că reţelele din (a)-(c) au acceaşi semantica SPF; rezultatul poate fi reprezentat şi prin *arborele infinit* din (d);

Transformari algebrice:

- Reţeaua din (a) poate fi reprezentată prin expresia ($\wedge = cp$)

$$F0 =_{\mathrm{def}} \wedge^1 \cdot [(\mathsf{I}_1 \star f \cdot {}^1\mathsf{X}^1) \cdot (f \star \mathsf{I}_1) \cdot (\wedge^1 \cdot (f' \uparrow^1 \star \mathsf{I}_1) \star {}^1\mathsf{X}^1)] \uparrow^1$$

- Notând cu f_1, f_2 cele două componente ale lui f, putem reprezenta echivalent reţelele prin sisteme de ecuaţii:

$$\begin{array}{cccc}
\begin{array}{rcl} y_1 &=& v \\ y_2 &=& f_1(x_1,z) \\ v &=& f_1(x_1,t) \\ z &=& f_2(x_1,t) \\ t &=& f_2(x_1,z) \\ u &=& f'(v,u) \end{array} &
\begin{array}{rcl} y_1 &=& v \\ y_2 &=& f_1(x_1,z) \\ v &=& f_1(x_1,t) \\ z &=& f_2(x_1,t) \\ t &=& f_2(x_1,z) \end{array} &
\begin{array}{rcl} y_1 &=& f_1(x_1,t) \\ y_2 &=& f_1(x_1,z) \\ z &=& f_2(x_1,t) \\ t &=& f_2(x_1,z) \end{array} &
\begin{array}{rcl} y_1 &=& f_1(x_1,w) \\ y_2 &=& f_1(x_1,w) \\ w &=& f_2(x_1,w) \end{array} \\
(\mathrm{N1}) & (\mathrm{N2}) & (\mathrm{N3}) & (\mathrm{N4})
\end{array}$$

..Transformari algebrice

Pas 1. Forma normală: Cu regulile BNA, o expresie ca $F0$ poate fi adusă la o formă normală ce corespunde sistemelor de ecuaţii de mai sus:

$$F1 =_{\text{def}} \{([2,4,6,8][1,10][3,9][5,7][11]) \cdot (\mathsf{I}_1 \star f_1 \star f_1 \star f_2 \star f_2 \star f')\} \uparrow^4$$

Explicaţii:

- variabilele se înlocuiesc cu numere; la intrare: $(x_1, v, z, t, u) = (1,2,3,4,5)$; la ieşire $(y_1, y_2, v, z, t, u) = (1,2,3,4,5,6)$;
- expresia $\mathsf{I}_1 \star f_1 \star f_1 \star f_2 \star f_2 \star f'$ strânge funcţiile din membrul drept al ecuaţiilor;
- $([2,4,6,8][1,10][3,9][5,7][11])$ reprezintă relaţia care duce: pe 1 în $2,4,6,8$; pe 2 în $1,10$; etc.

..Transformari algebrice

Pas 2. Regula enzimatica, inverse de injectii: Regula enzimatică cu inverse de injecţii permite eliminarea de părţi nefolosite pentru ieşiri (necoaccesibile);

E.g., f' nu este folosit pentru ieşiri, deci poate fi eliminat:

$$
\begin{aligned}
&([2,4,6,8][1,10][3,9][5,7][11]) \cdot (\mathsf{I}_1 \star f_1 \star f_1 \star f_2 \star f_2 \star f') \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad \cdot (\mathsf{I}_2 \star \underline{\mathsf{I}_3 \star \perp^1}) \\
&= ([2,4,6,8][1][3,9][5,7][]) \cdot (\mathsf{I}_1 \star f_1 \star f_1 \star f_2 \star g) \\
&= (\mathsf{I}_1 \star \underline{\mathsf{I}_3 \star \perp^1}) \cdot ([2,4,6,8][1][3,9][5,7]) \cdot (\mathsf{I}_1 \star f_1 \star f_1 \star f_2 \star f_2)
\end{aligned}
$$

deci cu regula enzimatică pentru inverse de injecţii $F1 = F2$, unde

$$F2 =_{\mathrm{def}} \{([2,4,6,8][1][3,9][5,7]) \cdot (\mathsf{I}_1 \star f_1 \star f_1 \star f_2 \star f_2)\} \uparrow^3$$

..Transformari algebrice

Pas 3. Izomorfism de grafuri: Cu un izomorfism de grafuri, deci reguli BNA, putem elimina redenumirea $y_1 = v$, deci $F2 = F3$, unde

$$F3 =_{\text{def}} \{([1,3,5,7][4,8][2,6]) \cdot (f_1 \star f_1 \star f_2 \star f_2)\} \uparrow^2$$

De notat ca primul f_1 din aceasta expresie corespunde celui de-al doilea f_1 din expresia anterioara (din definitia lui $F2$).

..Transformari algebrice

Pas 4. Regula enzimatica, inverse de surjectii: Regula enzimatică cu inverse de surjecţii permite identificarea de variabile, când membrul drept devine egal, după identificare:

E.g., putem identifica z şi t (redenumiţi w) căci, deşi membrul drept este diferit, devine egal după identificare (anume $f_2(x_1,w)$):

$$(\mathsf{I}_1\star\underline{\wedge}^1)\cdot([1,3,5,7][4,8][2,6])\cdot(f_1\star f_1\star f_2\star f_2)$$

$$=([1,3,5,7][2,4,6,8])\cdot(f_1\star f_1\star f_2\star f_2)$$

$$=([1,3,5][2,4,6])\cdot(\mathsf{I}_4\star\wedge^2)\cdot(f_1\star f_1\star f_2\star f_2)$$

$$=([1,3,5][2,4,6])\cdot(f_1\star f_1\star f_2)\cdot(\mathsf{I}_2\star\underline{\wedge}^1)$$

deci cu regula enzimatică pentru inverse de surjecţii $F3=F4$, unde

$$F4=_{\text{def}}\{([1,3,5][2,4,6])\cdot(f_1\star f_1\star f_2)\}\uparrow^1$$

..Transformari algebrice

Pas 5. Simplificare finală: Cu o simplă transformare aciclică, putem transforma $F4$ într-o formă mai simplă (dar *nu normală*!), anume $F4 = F5$, unde

$$F5 =_{\text{def}} \{\wedge^2 \cdot (\mathsf{I}_2 \star f_2)\} \uparrow^1 \cdot f_1 \cdot \wedge^1$$

ce corespunde poate celei mai simple reprezentări a acestei transformări:

$$y_1 = y_2 = f_1(x_1, w), \text{ unde } w = f_2(x_1, w).$$

Axiomatizarea dfn-urilor: intrare-iesire

Schita:

- Cu transformări ca cele de mai sus, *orice reţea poate fi adusă la o formă normală minimală*;
- *Formele normale minimale sunt în bijecţie cu tuplele de arbori de desfăşurare* a reţelelor (în particular, ele sunt unice până la un izomorfism);
- Rămâne *de văzut că*:
  - *două reţele definesc aceeaşi funcţie* SPF *dnd produc acelaşi tuplu de arbori de desfăşurare*.

Axiomatizarea dfn-urilor: intrare-iesire

Desfăşurărea:

- Fiecare atom cu mai multe ieşiri se înlocuieşte cu un tuplu de *atomi cu o unică ieşire* folosind transformarea
$$f : m \to n \quad \longleftrightarrow \quad \wedge_n^m \cdot (\textstyle\sum_{i=1}^n f(\wedge_0^{i-1} \star \mathsf{I}_1 \star \wedge_0^{n-i}))$$
- Punem pe *primul nivel* al arborelui de desfăşurare *copii ale celulelor* care au ca ieşiri *ieşirile din reţea*;
- Pe *nivelul 2*, punem *copii ale celulelor* care au ca ieşiri *intrări ale celulelor de pe nivelul 1*;
- Repetăm procedura de mai sus, obţinând un arbore (potenţial infinit);
- *Exemplu:* Arborele desenat anterior în figura (d) conţine desfăşurarea reţelelor (a)-(c).

..Axiomatizarea dfn-urilor: intrare-iesire

Corectitudinea desfăşurării:

- *Notaţii:* $F \equiv_{unfold} G$ dacă F şi G se desfăşoară în acelaşi tuplu de arbori; $F \equiv_{IO} G$ dacă F şi G definesc aceeaşi funcţie SPF;

- *Ecuaţia de punct fix*

$$(f \cdot \overset{\circ}{\wedge}^{b}) \uparrow^{b} = \overset{\circ}{\wedge}^{a} \cdot (\mathsf{I}_a \star (f \cdot \overset{\circ}{\wedge}^{b}) \uparrow^{b}) \cdot f, \quad \text{cu} \quad f : a + b \to b$$

  este validă în SPF, deci desfăşurarea este corectă, conservând funcţia SPF asociată reţelei.

- In cazul DFN-urilor nedeterminste, ecuaţia de mai sus (la fel ca şi procedul de desfăşurare) poate să *nu fie validă* dacă se iau comportamente *diferite* pentru cele *două apariţii* ale lui f în membrul drept.

..Axiomatizarea dfn-urilor: intrare-iesire

Prop: $F \equiv_{unfold} G$ *dnd* $F \equiv_{IO} G$.

Dem: (a) $(F \equiv_{unfold} G) \Rightarrow (F \equiv_{IO} G)$: rezultă din *validitatea ecuaţiei de punct fix*;

(b) $(F \equiv_{IO} G) \Rightarrow (F \equiv_{unfold} G)$: Demonstrăm că dacă F' şi F'' sunt arbori diferiţi, există o interpretare a.î. F' şi F'' specifică funcţii diferite.

- Fie D un domeniu de date ce conţine *Σ-arbori parţiali* peste X ("parţial" = unele argumente într-un termen $\sigma(x_1, \ldots, x_n)$ pot fi nedefinite, notate cu "?") unde:
  - X este mulţime infinită de *variabile*;
  - Σ este signatură cu *un simbol* σ_f pentru *orice atom* f care apare în F' ori F'', cu aritatea corespunzătoare;

..Axiomatizarea dfn-urilor: intrare-iesire

- Celulele au următoarea acţiune:
  - Cazul $m \geq 1$: O celulă $f : m \to 1$ execută:

$$f(v_1, \ldots, v_m) = \sigma_f(?, \ldots, ?)^\frown g_f(v_1, \ldots, v_m)$$
$$g_f(t_1{}^\frown w_1, \ldots, t_m{}^\frown w_m) = \sigma_f(t_1, \ldots, t_m)^\frown g_f(w_1, \ldots, w_m)$$
$$\text{unde } t_1, \ldots, t_m \in D \text{ şi } v_1, \ldots, v_m, w_1, \ldots, w_m \in D^\omega$$

    (Folosim definiţia şi pentru şuvoaie finite - ieşire este definită până la lungimea maximă a intrărilor, apoi vidă.)
  - Cazul $m = 0$: O celulă $f : 0 \to 1$ produce ieşirea $(\sigma_f)^\infty$;
- Luăm o variabilă x_i pentru fiecare intrare i şi considerăm ca intrare tuplul $((x_1)^\infty, \ldots, (x_m)^\infty)$;

..Axiomatizarea dfn-urilor: intrare-iesire

- Rezultatul $|F|((x_1)^\infty,\ldots,(x_m)^\infty)$ unui arbore $F : m \longrightarrow 1$ este un şuvoi de termeni $t_1{}^\frown t_2{}^\frown \ldots$, unde t_i este *aproximarea parţială a lui F până la nivelul i*;

- Cum F' şi F'' sunt diferiţi, există un nivel i astfel că aproximările până la nivelul i sunt diferite, deci $|F'|((x_1)^\infty,\ldots,(x_m)^\infty) \neq |F''|((x_1)^\infty,\ldots,(x_m)^\infty)$.

- *Exemplu:* Pentru arborele din figura (d) transformarea este

$$\begin{array}{l}
\sigma_{f_1}(?,?)^\frown \\
\sigma_{f_1}(x_1,\sigma_{f_2}(?,?))^\frown \\
\sigma_{f_1}(x_1,\sigma_{f_2}(x_1,\sigma_{f_2}(?,?)))^\frown \\
\sigma_{f_1}(x_1,\sigma_{f_2}(x_1,\sigma_{f_2}(x_1,\sigma_{f_2}(?,?))))^\frown \ldots
\end{array}$$

Axiomatizarea dfn-urilor: intrare-iesire

Din cele de mai sus dedcem următoarea:

Teoremă fundamentală:

Două DFN*-urile asincrone, deterministe definesc aceeaşi funcţie de procesare a şuvoialelor dacă şi numai dacă se pot transforma una în alte folosind axiomele de teorie Elgot duală.*

Retele de fluxul datelor

Cuprins:

- Generalitati
- Retele sincrone
- Functii de procesare a suvoaielor
- Retele asincrone deterministe
- *Retele asincrone nedeterministe*

Retele asincrone nedeterministe

Generalitati:

- Modelele pentru DFN-uri asincrone *nedeterministe* se obţin *greu* din cele *deterministe*: ecuaţiile cu apariţii multiple ale unei variabile ca

$$\begin{array}{ll} C1 & f \cdot \wedge_k^n = \wedge_k^m \cdot (f \star \ldots \star f); \\ Fix & (f \cdot \overset{\circ}{\wedge}{}^b) \uparrow^b = \overset{\circ}{\wedge}{}^a \cdot (\mathsf{I}_a \star (f \cdot \overset{\circ}{\wedge}{}^b) \uparrow^b) \cdot f; \end{array}$$

  valide determinist, pot fi invalide nedeterminist (se aleg comportări diferite pentru apariţii diferite ale lui f).

- Avntajul algebrei *BNA* este că axiomele sunt *"liniare"*, variabilele apărând numai o dată în stânga/dreapta în orice axiomă.

- In plus, există o *anomalie de timp* (anomalia Brock–Ackermann), care arată că modelul SPF *nu se potriveşte cu semantica operaţională* a DFN-urilor asincrone nedeterministe.

Anomalia de timp

Anomalia de timp:

- Sursa de anomalie este *feedback*-ul:
  - la DFN-urile nedeterministe, *datele* care vin pe *feedback* pot *interfera inpredictibil* cu cele de pe alte canale;
  - la cele deterministe, ele aşteaptă pe canalul de feedback până la corecta lor utilizare;
- Anomalia Brock–Ackermann speculează astfel de situaţii spre a arăta că *semantica relaţională nu este compoziţională*, anume
  - reţele ce specifică *aceeaşi relaţie IO* puse în *acelaşi context* pot produce reţele cu *relaţii IO diferite*.

..Anomalia de timp

Anomalia de timp: Posibile soluţii sunt: (1) *slabirea modelului semantic*; ori (2) *întărirea modelului operaţional*;

In prima linie, s-au propus diverse modele:

- **Modelul cu urme (Trace model):** Se foloseşte un *timp global*, liniarizându-se (într-un trace) şuvoialele de pe toate canalele; [Exemplu: pentru $N(i) =_{\text{def}} ((5^\frown 5) \star \mathsf{I}_1) \cdot mrg \cdot Buf(i) \cdot cp \cdot (\mathsf{I}_1 \star suc)$, am $out_1(5)^\frown in_1(6)^\frown out_1(6)$ urmă în $N2$ dar nu în $N1$.]
- **Model bazat pe intervale de timp:** Se divid şuvoailele în secvenţe finite, care corespund *ciclurilor de ceas*; modelul se reduce la cel sincron, unde nu avem o astfel de anomalie;
- **Model bazat pe oracole:** Comportamentul nedeterminist se reduce la un *set de comportamente deterministe*, folosind oracole.

..Anomalia de timp

Anomalia de timp: Există o propunere şi în a doua linie:

Cozi cu ghicit-si-imprumut (guess-and-borrow queues): Se extinde capacitatea canalelor FIFO permiţând *conţinuturi negative*:

- coada se *împrumută* urmând a primi datele ulterior;
- dacă primeşte ce a folosit e OK; dacă nu, calculul rezultat se ignoră;

Astfel semantica operaţională devine la fel de puternică ca cea relaţională şi anomalia dispare.

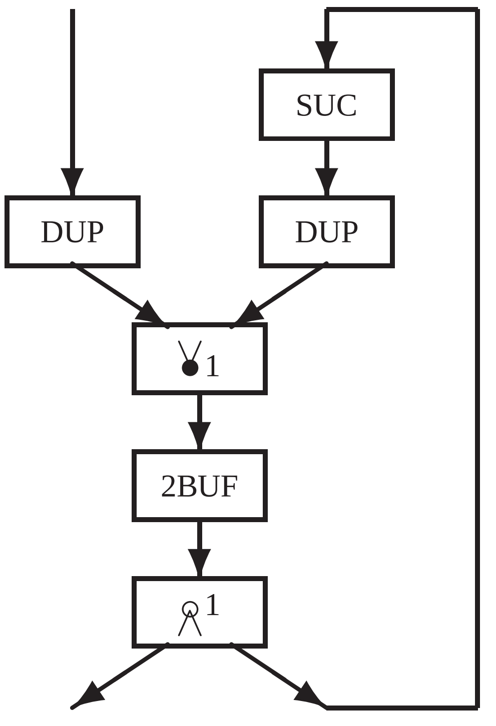
..Anomalia de timp

Modele complet-abstracte:

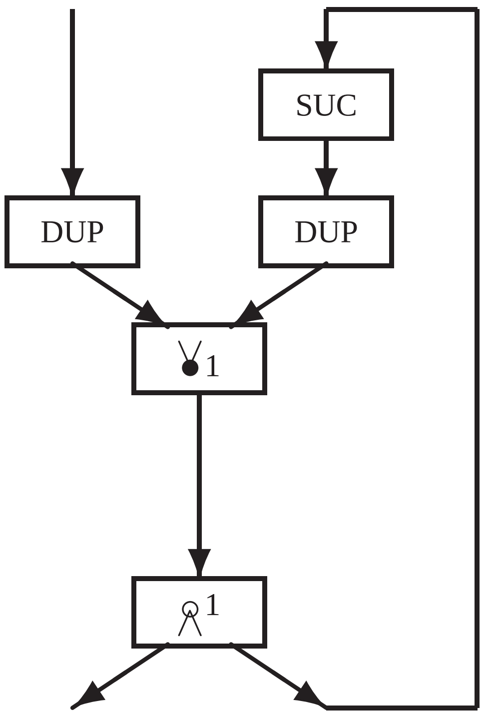
- Un model este *complet-abstract (fully-abstract)* dacă
  - este compoziţional şi
  - între el şi modelul relaţional intrare-ieşire nu mai sunt modelel compoziţionale;
- Din modele anerioare,
  - cele cu *urme* şi cu *intervale* temporale *sunt complet-abstracte*;
  - cel cu *oracole nu este complet-abstract*;

..Anomalia de timp

Exemplul Brock–Ackermann:



$f\uparrow^1$



$f'\uparrow^1$

..Anomalia de timp

Reţelele sunt $f\uparrow^1$, $f'\uparrow^1$, unde

$$f = (\mathtt{DUP}\star\mathtt{SUC}\cdot\mathtt{DUP})\cdot\curlyvee_1\cdot\mathtt{2BUF}\cdot\curlywedge^1$$
$$f' = (\mathtt{DUP}\star\mathtt{SUC}\cdot\mathtt{DUP})\cdot\curlyvee_1\cdot\mathsf{I}_1\cdot\curlywedge^1$$

Celulele au următoarea semnificaţie

- $\mathtt{DUP}$ - duplică datele: citeşte d, emite $d^\frown d$, apoi repetă;
- $\mathtt{SUC}$ - calculează succesorul: citeşte d, emite $d+1$, apoi repetă;
- $\mathtt{2BUF}$ - pasează câte 2 date: citeşte d_1, citeşte d_2, emite $d_1{}^\frown d_2$, apoi repetă;
- $\curlyvee_1$ - mixează nedeterminist datele de pe canalele de intrare;
- $\curlywedge^1$ - copiază datele de intrare pe ambele canale de ieşire;

..Anomalia de timp

Anomalia

- Reţelele f, f' specifică aceeaşi relaţie intrare ieşire (DUP produce un număr par de date, astfel că diferenţa dintre 2BUF şi I_1 dispare);
- Pe de altă parte, $1^\frown 2^\frown 2^\frown 3^\frown 3^\frown \ldots$ se poate obţine în $[f' \uparrow^1](1)$, dar nu în $[f \uparrow^1](1)$;

Merge:

- In exemplul Brock–Ackermann, *anomalia este introdusă de merge* (restul este determinist).

..Anomalia de timp

Exemplul Russell:

Exemplu Russell produce *anomalia fară merge*, doar cu celule nedeterministe:

- $G : 1 \to 1$: (citeşte d, emite $0^\frown 1$) ori (emite 0, citeşte d, emite 0), apoi repetă;

- $G' : 1 \to 1$: (citeşte d, emite $0^\frown 1$) ori (emite 0, citeşte d, emite 0) ori (emite 0, citeşte d, emite 1), apoi repetă;

Anomalia: G şi G' calculează acceaşi relaţie intrare-ieşire, dar nu şi $(G \cdot \mathring{\wedge}^1) \uparrow^1$ şi $(G' \cdot \mathring{\wedge}^1) \uparrow^1$:

- $0^\frown 1^\frown \ldots$ este o ieşire posibilă în $[(G' \cdot \mathring{\wedge}^1) \uparrow^1]()$, dar nu şi în $[(G \cdot \mathring{\wedge}^1) \uparrow^1]()$.

Semantica cu oracole

Mulţimi de functii SPF**:** Modelul PSPF(M) constă din *mulţimi de funcţii* SPF, care modelează comportamentul nedeterminist prin *oracole*, astfel:

- Un oracol alege un comportament predefinit pentru celulele nedeterministe din reţea;
- Nedeterminismul (rezolvat din exterior) rezidă din variaţia oracolelor;
- Dat un oracol, reţeaua devine deterministă, deci putem aplica rezultatele din cazul determinist;

..Semantica cu oracole

Modelul $P\mathsf{SPF}$: Formal

- $P\mathsf{SPF}(M)(a,b) := \{F \mid F \subseteq \mathsf{SPF}(M)(a,b)\}$;

- Operaţiile se extind pe componente:

$$\begin{array}{rcl} F \star G & = & \{f \star g\colon\ f \in F, g \in G\} \\ F \cdot G & = & \{f \cdot g\colon\ f \in F, g \in G\} \\ F \uparrow^a & = & \{f \uparrow^a\colon\ f \in F\} \end{array}$$

- O constantă $c \in \{\mathsf{I}, \mathsf{X}, \wedge, \perp, \top\}$ din SPF induce una $\{c\}$ în $P\mathsf{SPF}$;

- Constante adiţionale (nedeterministe):

..Semantica cu oracole

Split: $\overset{\bullet}{\wedge}{}^1 = \{\, \overset{\bullet}{\wedge}{}^1_{\phi} \mid \phi : \omega \longrightarrow \{1,2\}\} : 1 \longrightarrow 2$, unde pentru ϕ dat

$\overset{\bullet}{\wedge}{}^1_{\phi}\ (x) = (y,z)$, unde y şi z se obţin *despicând pe x după indicaţiile lui* ϕ: când $\phi(i) = 1$ intrarea i se emite pe canalui 1, iar când este 2, pe canalul 2.

Se extinde la blocuri de canale folosind oracole independente pe fiecare canal.

..Semantica cu oracole

Merge: $\curlyvee_1 = \{ \overset{\phi}{\curlyvee}_1 \mid \phi : \omega \to \{1,2\}\} : 2 \to 1$, unde pentru ϕ dat

$\overset{\phi}{\curlyvee}_a (x,y) = z$, unde z se obţine *din x şi y după indicaţiile lui* ϕ: când $\phi(i) = 1$ ieşirea i se ia de pe canalui 1, iar când este 2, de pe canalul 2. (Dacă nu există dată pe intrarea specificată, celula merge se blochează.)

Se extinde la blocuri de canale folosind oracole independente pe fiecare canal.

(Rich) Source: $\text{♀}_1 = \{g_x \mid x \in M\}$, unde pentru $x \in M$, $\quad g_x : 0 \to 1$ este funcţia dată de $g_x(\ ) = x$.

..Semantica cu oracole

BNA:

Teoremă: $(P\mathsf{SPF}, \star, \cdot, \uparrow, \mathsf{I}, \mathsf{X})$ *este BNA.*

Dem: Se reduce la rezultatul similar despre SPF.

Comentarii:

- Teorema arată că semantica cu oracole este *compoziţională*.
- Axiome “neliniare” ca $F \cdot \overset{\circ}{\wedge}{}^{b} = \overset{\circ}{\wedge}{}^{a}(F \star F)$, valide în SPF, nu mai sunt valide în $P\mathsf{SPF}$;
- Modelul $P\mathsf{SPF}$ dă un atu în plus axiomelor BNA (pe lângă evidenta “naturaleţe”).

Teoremă: $(P\mathsf{SPF}, \star, \cdot, \uparrow, \mathsf{I}, \mathsf{X}, \overset{\bullet}{\wedge}, \underset{\circ}{|}, \underset{\bullet}{\vee}, \overset{\bullet}{|})$ *este BNA şi satisface axiomele suplimentare din tabelul următor.*

..Semantica cu oracole

Axiome pentru split/merge:

I′. Axiome pentru constantele aditionale $\top$, $\bot$, $\vee$, $\wedge$, fara feedback

A1a	$\wedge^a \cdot (\wedge^a \star \mathsf{I}_a) = \wedge^a \cdot (\mathsf{I}_a \star \wedge^a)$
A1b*	$\wedge^a \cdot (\bot^a \star \mathsf{I}_a) \supset \mathsf{I}_a$
A1c	$(\vee_a \star \mathsf{I}_a) \cdot \vee_a = (\mathsf{I}_a \star \vee_a) \cdot \vee_a$
A1d*	$(\top_a \star \mathsf{I}_a) \cdot \vee_a \supset \mathsf{I}_a$
A2a	$\wedge^a \cdot {}^a\mathsf{X}^a = \wedge^a$
A2b	${}^a\mathsf{X}^a \cdot \vee_a = \vee_a$
A3a	$\top_a \cdot \bot^a = \mathsf{I}_\varepsilon$
A3b	$\top_a \cdot \wedge^a = \top_a \star \top_a$
A3c	$\vee_a \cdot \bot^a = \bot^a \star \bot^a$
A3d*	$\vee_a \cdot \wedge^a \subset (\wedge^a \star \wedge^a) \cdot (\mathsf{I}_a \star {}^a\mathsf{X}^a \star \mathsf{I}_a) \cdot (\vee_a \star \vee_a)$
A4*	$\wedge^a \cdot \vee_a \supset \mathsf{I}_a$

II′. Feedback pe constante split/merge

R8	$\vee_a \uparrow^a = \bot^a$
R9	$\wedge^a \uparrow^a = \top_a$
R10*	$[(\mathsf{I}_a \star \wedge^a) \cdot ({}^a\mathsf{X}^a \star \mathsf{I}_a) \cdot (\mathsf{I}_a \star \vee_a)] \uparrow^a \supset \mathsf{I}_a$

..Semantica cu oracole

Dem:

Constante: Ce constante de ramificare folosim?

- Plecăm cu $\wedge$ şi $\vee$ interpretate ca split şi merge;
- Teoria trebuie să fie închisă la feedback şi:
  $\overset{\bullet}{\wedge}{}^s \uparrow^s = \overset{\bullet}{|}{}_s$, căci $\underset{\phi}{\overset{\bullet}{\wedge}}{}^s \uparrow^s = \underset{\phi}{\overset{\bullet}{|}}{}_s$ pentru orice oracol ϕ;
  $\underset{\bullet}{\vee}{}_s \uparrow^s = \underset{\circ}{|}{}^s$, căci, similar, $\underset{\bullet}{\vee}{}_s \uparrow^s = \underset{\circ}{|}{}^s$ pentru orice ϕ
- Deci folosim: $\{\overset{\bullet}{\wedge}, \underset{\bullet}{\vee}, \underset{\circ}{|}, \overset{\bullet}{|}\}$.

Oracole extinse:

- Folosim oracole extinse *în lăţime*: $\phi : \omega \to \{1,\ldots,k\}$cu $k \geq 1$
  şi constantele corespunzătoare $\underset{\phi}{\overset{\bullet}{\wedge}}{}^1_k : 1 \to k$ şi $\overset{\phi}{\underset{\bullet}{\vee}}{}^k_1 : k \to 1$.

Aximele valide: - uşor de demonstrat.

 G Stefanescu

..Semantica cu oracole

A3 invalidă:

- $[(\overset{\bullet}{|}_s \otimes \mathsf{I}_s)\ \underset{\bullet}{\overset{\phi}{\vee}}_s\ ](x)$ este prefixul lui x până la cel mai mare k pentru care $\phi(1) = \ldots = \phi(k) = 2$.

A7 invalidă:

- $[\ \underset{\phi}{\overset{\bullet}{\wedge}}^s\ (\underset{\circ}{|}^s \otimes \mathsf{I}_s)](x)$ este sub-stream-ul lui x dat de poziţiile k cu $\phi(k) = 2$.

A10 invalidă:

- Fie $E(\phi', \phi'', \psi', \psi'') = (\ \underset{\phi'}{\overset{\bullet}{\wedge}}^s \otimes \underset{\phi''}{\overset{\bullet}{\wedge}}^s\ )(\mathsf{I}_s \otimes {}^s\mathsf{X}^s \otimes \mathsf{I}_s)(\ \underset{\bullet}{\overset{\psi'}{\vee}}_s \otimes \underset{\bullet}{\overset{\psi''}{\vee}}_s\ )$ şi $F(\sigma, \tau) = \underset{\bullet}{\overset{\sigma}{\vee}}_s \cdot \underset{\tau}{\overset{\bullet}{\wedge}}^s$;

..Semantica cu oracole

- Mulţimea $\{E(\phi',\phi'',\psi',\psi'') : \phi',\phi'',\psi',\psi'\}$ este mai mare ca $\{F(\sigma,\tau) : \sigma,\tau\}$. Intr-adevăr,

  – $E(\phi',\phi'',\psi',\psi'')(1^\frown 2^\frown \ldots, a^\frown b^\frown \ldots) = (2^\frown \ldots, b^\frown \ldots)$ cu $\phi' = 1^\frown 2 \ldots$; $\phi'' = 1^\frown 2 \ldots$; $\psi' = 2 \ldots$; $\psi'' = 1^\frown 1^\frown \ldots$;
  In $F(\sigma,\tau)(1^\frown 2^\frown \ldots, a^\frown b^\frown \ldots)$ prima ieşire pe cel puţin un canal este din $\{1, a\}$.

  – Pe dos, $F(\sigma,\tau)$ poate fi simulată de $E(\phi',\phi'',\psi',\psi'')$, cu ϕ' şi ϕ'' restricţii adecvate din τ, iar ψ' şi ψ'' din σ.
  [Pentru un oracol α şi o submulţime $A \subseteq \omega$, notăm $\alpha|_A$ oracolul obţinut restricţionând pe α la A: dacă A are elementele $a_1 < a_2 < \ldots$, atunci $\alpha|_A(i) = \alpha(a_i)$ pentru $i = 1, 2, \ldots$. Acum, $\phi' = \tau|_{\sigma^{-1}(1)}$, $\phi'' = \tau|_{\sigma^{-1}(2)}$, $\psi' = \sigma|_{\tau^{-1}(1)}$, $\psi'' = \sigma|_{\tau^{-1}(2)}$ (Dacă unele din oracolele obţinute sunt finite, se extind arbitrar la infinit.)]

..Semantica cu oracole

A11 invalidă:

- $E(\phi,\psi) = \underset{\phi}{\curlywedge}^{s} \cdot \overset{\psi}{\curlyvee}_{s}$ generează o clasă de funcţii care include strict I_s, permiţând "permutări", e.g., $1^\frown 2^\frown 3^\frown 4^\frown \ldots \mapsto 2^\frown 1^\frown 4^\frown 3^\frown \ldots$.

R10 invalidă:

- Partea stângă a lui R10 specifică un "bag/sac" (elementele puse pot fi scoase în orice ordine), deci include strict I_s.

Modele complet-abstracte

Modelul cu urme:

Teoremă: *Modelul cu urme este complet abstract relativ la semantica relaţională intrare-ieşire.*

Dem (schita):

- In modelul cu urme liniarizăm evenimementele de pe toate canalele (de intrare/ieşire); notăm:

  $\langle in_i, x\rangle$ = citit x de pe canalul de intrare i şi
  $\langle out_j, y\rangle$ = emis y pe canalul de ieşire j.

- Notăm:

  $[\![\ ]\!]_{Tr}$ = semantica cu urme;
  $[\![\ ]\!]_{H}$ = semantica relaţională I/O;
  $[\![\ ]\!]_{O}$ = semantica operaţională.

..Modele complet-abstracte

- Fapte:

  1. $[\![N_1]\!]_{Tr} = [\![N_2]\!]_{Tr}$ implică $[\![N_1]\!]_O = [\![N_2]\!]_O$;
  2. $[\![N_1]\!]_{Tr} = [\![N_2]\!]_{Tr}$ implică $\forall C.[\![C(N_1)]\!]_{Tr} = [\![C(N_2)]\!]_{Tr}$;
  3. $[\![N_1]\!]_{Tr} \neq [\![N_2]\!]_{Tr}$ implică $\exists C.[\![C(N_1)]\!]_H \neq [\![C(N_2)]\!]_H$

- Demonstraţiile pentru 1 şi 2 sunt relativ simple;

- Pentru 3, trebuie găsit un context C care liftează diferenţele de la urme la relaţia I/O specificată. Folosim

$$MergeMark : 1 + q \to 1 \quad \text{şi} \quad CopyRelese : 1 \to 1 + p$$

  şi contextul

$$C(N) = (MergeMark \cdot CopyRelease \cdot (\mathsf{I}_1 \star N)) \uparrow^q$$

..Modele complet-abstracte

MergeMark: amestecă datele de pe primul canal cu cele din restul de q canale. In plus:

- fiecare dată d din canalul $1 + j$ este transformată în tupletul $\langle out_j, d \rangle$;
- data de tipul $\langle in_j, d \rangle$ din primul canal este pasată neschimbată;
- merge-ul este onest (fair), i.e., nici un canal nu este neglijat la infinit.

CopyRelease: acest bloc copiază data de intrare pe primul canal. In plus:

- dacă data curentă este de tipul $\langle in_j, d \rangle$ transmite d pe canalul $1 + j$.

..Modele complet-abstracte

Presupunem că D poate codifica perechile $\langle in_j, d\rangle$ ori $\langle out_j, d\rangle$. [E.g., este suficient ca D să aibă 2 elemente şi citim un calup de k date odată, pentru un k suficient de lung.]

Dacă $[\![N_1]\!]_{Tr} \neq [\![N_2]\!]_{Tr}$, atunci:

- există o urmă $t \in [\![N_1]\!]_{Tr} \setminus [\![N_2]\!]_{Tr}$ (ori pe dos);
- t este stream (finit ori infinit) din evenimente $\langle in_j, d\rangle$ ori $\langle out_j, d\rangle$;
- fie $in(t) = t|_{in}$ restricţia lui t la evenimentele de tip in;
- o analiză fină arată că:

$$(in(t), t) \in [\![N_i]\!]_H \quad \text{dnd} \quad t \in [\![N_i]\!]_{Tr}$$

- deci $[\![C(N_1)]\!]_H \neq [\![C(N_2)]\!]_H$.

..Modele complet-abstracte

Concluzii:

- Modelul cu urme (ca şi cel cu intervale) adaugă *cantitatea minimă de informaţie* la modelul relaţional spre a-l transforma într-unul compoziţional.

- Din cauza timpului global, modelul cu urme *nu admite o teorie algebrică simplă*, i.e., operaţiile BNA nu au o exprimare simplă în model.

- Modelul cu *oracole* este mai *in spiritul sitemelor interactive*: un oracol este un tip de date de interacţie (temporal) care controlează evoluţia reţelei nedeterministe.