Event structures for the reversible early internal π-calculus

Eva Graversen, Iain Phillips, and Nobuko Yoshida
Imperial College London

9th of June 2020

Background

- Process calculi describe concurrent processes.
- RCCS (Danos and Krivine, 04) was the first reversible process calculus. It stored memories in stacks separate from processes (dynamic reversibility).
- CCSK (Phillips and Ulidowski, 06) used keys to denote past actions in CCS while maintaining the structure of the process (static reversibility).
- Extrusion histories containing past outputs of bound names and inputs of free names have been used to define stable non-interleaving early operational semantics of π-calculus (Hildebrandt et al., 17).

Background

- Event structures have been used to define semantics for CCS, π-calculus, LOTOS, etc. in forwards-only setting.
- Event structures provide true concurrency semantics, which describe the causal relationships between actions.
- Existing true concurrency semantics of reversible process calculi (Cristescu et al., 15, Aubert and Cristescu, 15) needed to reverse all past actions before mapping to their model of true concurrency.

Motivation

- Currently existing reversible π-calculus semantics are either late (Cristescu et al., 13, Medic et al., 18) or reduction semantics (Lanese et al., 10, Tiezzi and Yoshida, 10).
- This is despite early semantics being more common than late in forward-only π-calculi.
- Reduction semantics are unlabelled, late and early are labelled.
- Early semantics lets the process receive existing free names from the environment, late do not.
- Using early semantics gives us some non-structural causation, e.g. if $a(x) \mid (\nu\ b)\overline{a}(b)$ receives b after sending b.

Motivation

- πI-calculus is a variant of π-calculus where output names are bound.
- πI-calculus avoids most of the non-structural causation of the π-calculus, including situations like $(\nu x)(\overline{a}\langle x\rangle \mid \overline{b}\langle x\rangle \mid x(y))$.

- We want true concurrency semantics of a dynamically reversible early πI-calculus, and we use denotational event structure semantics.
- We go through a statically reversible early πI-calculus.

πIH-calculus

πIH-calculus

Hildebrandt et al. defined stable non-interleaving early operational semantics for the π-calculus using extrusion histories.

We extend the histories to contain enough information to reverse.

$$
\begin{aligned}
& a(x).\,((\nu b)\,(\overline{x}\,\langle b\rangle)\,|x(y))\,|\,(\nu c)\,(\overline{a}\,\langle c\rangle) && \xrightarrow{\overline{a}\langle c\rangle} \\
& a(x).\,((\nu b)\,(\overline{x}\,\langle b\rangle)\,|x(y))\,|0 && \xrightarrow{a(c)} \\
& ((\nu b)\,(\overline{c}\,\langle b\rangle)\,|c(y))\,|0 && \xrightarrow{\tau} \\
& (\nu b)\,(0|0)\,|0 &&
\end{aligned}
$$

πIH-calculus

Hildebrandt et al. defined stable non-interleaving early operational semantics for the π-calculus using extrusion histories.

We extend the histories to contain enough information to reverse.

$$(\emptyset, \emptyset) \vdash a(x).\,((\nu b)\,(\overline{x}\langle b\rangle)\,|\,x(y))\,|\,(\nu c)\,(\overline{a}\langle c\rangle) \xrightarrow[1[\overline{a}\langle c\rangle][0]]{\overline{a}\langle c\rangle}$$

$$(\{(c, 1[\overline{a}\langle c\rangle][0])\}) \vdash a(x).\,((\nu b)\,(\overline{x}\langle b\rangle)\,|\,x(y))\,|\,0 \xrightarrow[0[a(x).((\nu b)(\overline{x}\langle b\rangle)|x(y))][((\nu b)(\overline{c}\langle b\rangle)|c(y))]]{a(c)}$$

$$\begin{pmatrix} \{(c, 1[\overline{a}\langle c\rangle][0])\}, \\ \{(c, 0[a(x).\,((\nu b)\,(\overline{x}\langle b\rangle)\,|\,x(y))][((\nu b)\,(\overline{c}\langle b\rangle)\,|\,c(y))])\} \end{pmatrix} \vdash ((\nu b)\,(\overline{c}\langle b\rangle)\,|\,c(y))\,|\,0 \xrightarrow[0\langle 0[(\overline{c}\langle b\rangle)][0], 1[c(y)][0]\rangle]{\tau}$$

$$\begin{pmatrix} \{(c, 1[\overline{a}\langle c\rangle][0])\}, \\ \{(c, 0[a(x).\,((\nu b)\,(\overline{x}\langle b\rangle)\,|\,x(y))][((\nu b)\,(\overline{c}\langle b\rangle)\,|\,c(y))])\} \end{pmatrix} \vdash (\nu b)\,(0|0)\,|\,0$$

πIH-calculus

Hildebrandt et al. defined stable non-interleaving early operational semantics for the π-calculus using extrusion histories.

We extend the histories to contain enough information to reverse.

$$(\emptyset, \emptyset, \emptyset) \vdash a(x).\,(\overline{x}(b)|x(y))\,|\overline{a}(c) \xrightarrow[1[\overline{a}(c)][0]]{\overline{a}(c)}$$

$$(\{(\overline{a}(c), 1[\overline{a}(c)][0])\}, \emptyset, \emptyset) \vdash a(x).\,(\overline{x}(b)|x(y))\,|0 \xrightarrow[0[a(x).(\overline{x}(b)|x(y))][(\overline{c}(b)|c(y))]]{a(c)}$$

$$\left(\begin{array}{l}\{(\overline{a}(c), 1[\overline{a}(c)][0])\},\\ \{(a(c), 0[a(x).\,(\overline{x}(b)|x(y))][(\overline{c}(b)|c(y))])\}, \emptyset\end{array}\right) \vdash (\overline{c}(b)|c(y))\,|0 \xrightarrow[0\langle 0[\overline{c}(b)][0], 1[c(y)][0]\rangle]{\tau}$$

$$\left(\begin{array}{l}\{(\overline{a}(c), 1[\overline{a}(c)][0])\},\\ \{(a(c), 0[a(x).\,(\overline{x}(b)|x(y))][(\overline{c}(b)|c(y))])\},\\ \{((\overline{c}(b), c(b))\,, 0\,\langle 0[\overline{c}(b)][0], 1[c(y)][0]\rangle)\}\end{array}\right) \vdash (\nu b)\,(0|0)\,|0$$

Existing Approach

Process with past actions

$\downarrow$ Reverse all actions

Standard process

$\downarrow$ Denotational semantics

Truely concurrect structure

$\downarrow$ Redo actions

Truely concurrent structure with current state

Our Approach

πIH

$\downarrow$ E

πIK

$\downarrow$ Denotational semantics

Event Structure

πIK-calculus

CCSK uses keys to denote past actions and which actions they have synchronised with.

(Medic et al., 18) used keys to denote past actions in π-calculus, but due to only dealing with πI-calculus, our semantics are simpler.

$$
\begin{aligned}
& \overline{a}(b).\overline{b}(c) \mid a(x).x(y) \\
\xrightarrow{\overline{a}(b)[n_1]}\ & \overline{a}(b)[n_1].\overline{b}(c) \mid a(x).x(y) \\
\xrightarrow{a(b)[n_2]}\ & \overline{a}(b)[n_1].\overline{b}(c) \mid a(b)[n_2].b_{[n_2]}(y) \\
\xrightarrow{\tau[n_3]}\ & (\nu\ c)(\overline{a}(b)[n_1].\overline{b}(c)[n_3] \mid a(b)[n_2].b_{[n_2]}(c)[n_3])
\end{aligned}
$$

Operational Correspondence

We can construct a πIK process from a πIH process:

- Use locations as keys
- Use a second copy of the location to determine where the action originated
- Iteratively add actions from the extrusion history back onto the process
- Keep another copy of the process where the actions are instead reversed
- Use the state of the locations in the second process to determine which extrusions should be added to the process next.

We want to find the πIK process corresponding to

$$(\{(\overline{b}(c), u_2)\}, \varnothing, \{(b(a), \overline{b}(a), \langle 0u_0, 1u_1\rangle)\}) \vdash a(x) \mid 0$$

with locations $u_0 = [b(y).y(x)][a(x)]$, $u_1 = [\overline{b}(a)][0]$, and $u_2 = [\overline{b}(c).(b(y).y(x) \mid \overline{b}(a)][b(y).y(x) \mid \overline{b}(a)]$. We perform

$$E(\mathrm{lcopy}((\{(\overline{b}(c), u_2)\}, \varnothing, \{(b(a), \overline{b}(a), \langle 0u_0, 1u_1\rangle)\})) \vdash a(x) \mid 0, a(x) \mid 0)$$

We want to find the πIK process corresponding to

$$(\{(\overline{b}(c), u_2)\}, \varnothing, \{(b(a), \overline{b}(a), \langle 0u_0, 1u_1\rangle)\}) \vdash a(x) \mid 0$$

with locations $u_0 = [b(y).y(x)][a(x)]$, $u_1 = [\overline{b}(a)][0]$, and $u_2 = [\overline{b}(c).(b(y).y(x) \mid \overline{b}(a)][b(y).y(x) \mid \overline{b}(a)]$. We perform

$$E(\mathrm{lcopy}((\{(\overline{b}(c), u_2)\}, \varnothing, \{(b(a), \overline{b}(a), \langle 0u_0, 1u_1\rangle)\})) \vdash a(x) \mid 0, a(x) \mid 0)$$

$$E((\{(\overline{b}(c), u_2, u_2)\}, \varnothing, \varnothing) \vdash P_0 \mid P_1, b(y).y(x) \mid \overline{b}(a)) \quad \text{where}$$

$$P_0 = E((\varnothing, \{(b(a), u_0, \langle 0u_0, 1u_1\rangle)\}, \varnothing) \vdash a(x), a(x)) \text{ and}$$

$$P_1 = E((\{(\overline{b}(a), u_1, \langle 0u_0, 1u_1\rangle)\}, \varnothing, \varnothing) \vdash 0, 0)$$

We want to find the πIK process corresponding to

$$(\{(\overline{b}(c), u_2)\}, \varnothing, \{(b(a), \overline{b}(a), \langle 0u_0, 1u_1\rangle)\}) \vdash a(x) \mid 0$$

with locations $u_0 = [b(y).y(x)][a(x)]$, $u_1 = [\overline{b}(a)][0]$, and $u_2 = [\overline{b}(c).(b(y).y(x) \mid \overline{b}(a)][b(y).y(x) \mid \overline{b}(a)]$. We perform

$$E(\mathsf{lcopy}((\{(\overline{b}(c), u_2)\}, \varnothing, \{(b(a), \overline{b}(a), \langle 0u_0, 1u_1\rangle)\})) \vdash a(x) \mid 0, a(x) \mid 0)$$

$$E((\{(\overline{b}(c), u_2, u_2)\}, \varnothing, \varnothing) \vdash P_0 \mid P_1, b(y).y(x) \mid \overline{b}(a)) \quad \text{where}$$

$$P_0 = E((\varnothing, \{(b(a), u_0, \langle 0u_0, 1u_1\rangle)\}, \varnothing) \vdash a(x), a(x)) \text{ and}$$

$$P_1 = E((\{(\overline{b}(a), u_1, \langle 0u_0, 1u_1\rangle)\}, \varnothing, \varnothing) \vdash 0, 0)$$

$$P_0 = E((\varnothing, \varnothing, \varnothing) \vdash$$
$$b(a)[\langle 0u_0, 1u_1\rangle].S(a(x), y(x), [\langle 0u_0, 1u_1\rangle], y), b(y).y(x))$$

$$P_0 = b(a)[\langle 0u_0, 1u_1\rangle].a_{[\langle 0u_0, 1u_1\rangle]}(x)$$

We want to find the πIK process corresponding to

$$(\{(\overline{b}(c), u_2)\}, \emptyset, \{(b(a), \overline{b}(a), \langle 0u_0, 1u_1\rangle)\}) \vdash a(x) \mid 0$$

with locations $u_0 = [b(y).y(x)][a(x)]$, $u_1 = [\overline{b}(a)][0]$, and $u_2 = [\overline{b}(c).(b(y).y(x) \mid \overline{b}(a)][b(y).y(x) \mid \overline{b}(a)]$. We perform

$$E(\mathrm{lcopy}((\{(\overline{b}(c), u_2)\}, \emptyset, \{(b(a), \overline{b}(a), \langle 0u_0, 1u_1\rangle)\})) \vdash a(x) \mid 0, a(x) \mid 0)$$

$$E((\{(\overline{b}(c), u_2, u_2)\}, \emptyset, \emptyset) \vdash P_0 \mid P_1, b(y).y(x) \mid \overline{b}(a)) \quad \text{where}$$

$$P_0 = E((\emptyset, \{(b(a), u_0, \langle 0u_0, 1u_1\rangle)\}, \emptyset) \vdash a(x), a(x)) \text{ and}$$

$$P_1 = E((\{(\overline{b}(a), u_1, \langle 0u_0, 1u_1\rangle)\}, \emptyset, \emptyset) \vdash 0, 0)$$

$$P_0 = E((\emptyset, \emptyset, \emptyset) \vdash$$
$$b(a)[\langle 0u_0, 1u_1\rangle].S(a(x), y(x), [\langle 0u_0, 1u_1\rangle], y), b(y).y(x))$$

$$P_0 = b(a)[\langle 0u_0, 1u_1\rangle].a_{[\langle 0u_0, 1u_1\rangle]}(x)$$

$$P_1 = \overline{b}(a)[\langle 0u_0, 1u_1\rangle].0$$

We want to find the πIK process corresponding to

$$(\{(\overline{b}(c), u_2)\}, \varnothing, \{(b(a), \overline{b}(a), \langle 0u_0, 1u_1\rangle)\}) \vdash a(x) \mid 0$$

with locations $u_0 = [b(y).y(x)][a(x)]$, $u_1 = [\overline{b}(a)][0]$, and $u_2 = [\overline{b}(c).(b(y).y(x) \mid \overline{b}(a)][b(y).y(x) \mid \overline{b}(a)]$. We perform

$$E(\mathrm{lcopy}((\{(\overline{b}(c), u_2)\}, \varnothing, \{(b(a), \overline{b}(a), \langle 0u_0, 1u_1\rangle)\})) \vdash a(x) \mid 0, a(x) \mid 0)$$

$$E((\{(\overline{b}(c), u_2, u_2)\}, \varnothing, \varnothing) \vdash P_0 \mid P_1, b(y).y(x) \mid \overline{b}(a)) \quad \text{where}$$

$$P_0 = E((\varnothing, \{(b(a), u_0, \langle 0u_0, 1u_1\rangle)\}, \varnothing) \vdash a(x), a(x)) \text{ and}$$

$$P_1 = E((\{(\overline{b}(a), u_1, \langle 0u_0, 1u_1\rangle)\}, \varnothing, \varnothing) \vdash 0, 0)$$

$$P_0 = E((\varnothing, \varnothing, \varnothing) \vdash b(a)[\langle 0u_0, 1u_1\rangle].S(a(x), y(x), [\langle 0u_0, 1u_1\rangle], y), b(y).y(x))$$

$$P_0 = b(a)[\langle 0u_0, 1u_1\rangle].a_{[\langle 0u_0, 1u_1\rangle]}(x)$$

$$P_1 = \overline{b}(a)[\langle 0u_0, 1u_1\rangle].0$$

$$E((\{(\overline{b}(c), u_2, u_2)\}, \varnothing, \varnothing) \vdash P_0 \mid P_1, b(y).y(x) \mid \overline{b}(a)) =$$

$$\overline{b}(c)[u_2].(b(a)[\langle 0u_0, 1u_1\rangle].a_{[\langle 0u_0, 1u_1\rangle]}(x) \mid \overline{b}(a)[\langle 0u_0, 1u_1\rangle].0)$$

Event structure semantics of πIK-calculus

Bundle Event Structure (Langerak, 92)

Events a, b, c

Causation $\{a, b\} \mapsto c$

Conflict $a \sharp b$

$$\begin{tikzcd}
\{a, c\} & & \{b, c\} \\
\{a\} \arrow[u] & & \{b\} \arrow[u] \\
& \emptyset \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

Reversible Bundle Event Structure (Graversen et al, 18)

Events a, b, c

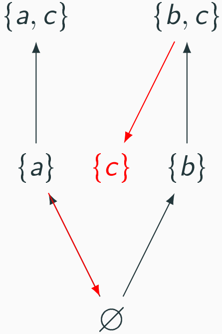
Reversible events a, b

Causation $\{a, b\} \mapsto c$

Conflict $a \sharp b$

Reverse causation $\{c\} \mapsto \underline{b}$

Prevention $c \triangleright \underline{a}$



Denotational Semantics

We generate an event structure inductively on the structure of the process, so $\{\!|P|\!\}_{\mathcal{N}} = (\mathcal{E}, \mathsf{Init}, k)$ where

- P is a process with all bound free names distinct
- $\mathcal{N} \supseteq n(P)$ is the set of names any input in the process can receive
- $\mathcal{E}$ is an event structure
- Init is the initial state of the structure, corresponding to past actions
- k assigns the keys of the past actions to the events of Init

Sample Rule

$$\{\!|a(x).P|\!\}_{\mathcal{N}} = \left\langle \begin{array}{l} \sum\limits_{n\in(\mathcal{N}\setminus\mathsf{sbn}(P))} a(n)(e_n).\mathcal{E}_{P_n}, \\ \bigcup\limits_{n\in(\mathcal{N}\setminus\mathsf{sbn}(P))} \{n\} \times \mathsf{Init}_{P_n}, \\ (n,e) \mapsto k_{P_n}(e) \end{array} \right\rangle$$

$$\text{for some fresh } e_n \notin E_n \text{ where}$$

$$\{\!|P[x := n]|\!\}_{\mathcal{N}} = \langle \mathcal{E}_{P_n}, \mathsf{Init}_{P_n}, k_{P_n} \rangle$$

Where $\mathcal{N}$ is the set of names the input can receive, $\mathcal{E}$ is an event structure, Init is the initial state of the structure, k assigns the keys of the past actions to the events of Init

Example

$$\{\!|\, a(x) \mid \overline{a}(b)[n] \,|\!\}_{\{a,b,x\}} =$$

Events

$a(a), a(b), a(x), \overline{a}(b), \tau$

Reversible events

$a(a), a(b), a(x), \overline{a}(b), \tau$

Causation $\{\overline{a}(b)\} \mapsto a(b)$

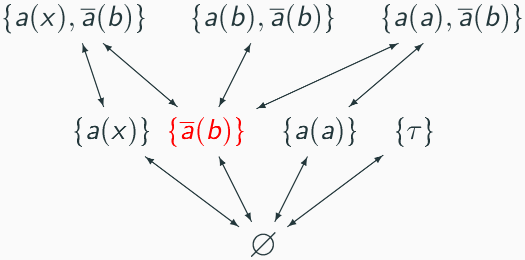
Conflict

$a(a) \sharp a(b),\ a(a) \sharp a(x)$

$a(b) \sharp a(x),\ a(a) \sharp \tau$

$a(b) \sharp \tau,\ a(c) \sharp \tau,\ \overline{a}(b) \sharp \tau$

Prevention $a(b) \rhd \underline{\overline{a}(b)}$



With initial state $\{\overline{a}(b)\}$

Conclusion

- We have defined the first early reversible π-calculus, πIH.
- We have used a statically reversible early calculus, πIK, as an intermediate stage to define true concurrency semantics.
- Unlike existing true concurrency semantics of dynamically reversible calculi, this does not require redoing every past action after mapping to an event structure.

Future Work

Event structure semantics of a full reversible π-calculus

- Handle more complex causations, e.g. $(\nu x)(\overline{a}\langle x\rangle \mid \overline{b}\langle x\rangle \mid x(y))$.
- Initial state will be harder due to actions causing bigger changes to process.

Full version available at arxiv.org/abs/2004.01211