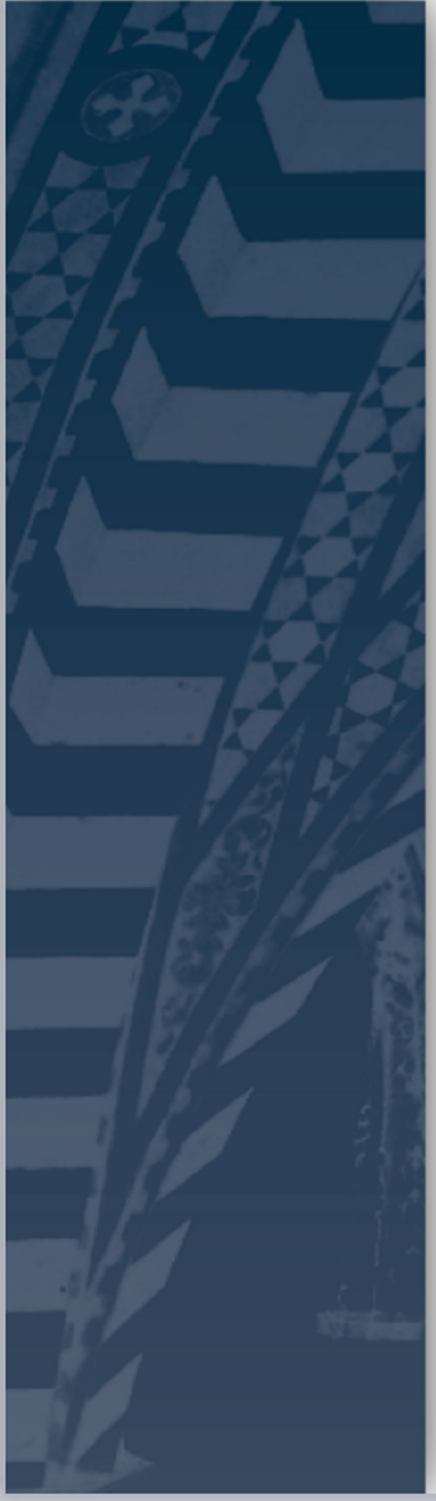

IMT SCHOOL FOR ADVANCED STUDIES LUCCA

cost EUROPEAN COOPERATION IN SCIENCE AND TECHNOLOGY

Static VS Dynamic Reversibility in CCS

Research partly supported by the EU COST Action IC1405.

Doriana Medić in collaboration with
Ivan Lanese and Claudio Antares Mezzina

March 29, 2017

Introduction

Two reversible variant of CCS:

- RCCS[1], introduced by Danos and Krivine and
- CCSK[2], proposed by Phillips and Ulidowski

We provide encodings between CCSK and RCCS and show their correctness in terms of strong back and forth bisimulation

[1] J. Krivine. A verification technique for reversible process algebra. In Reversible Computation, 4th International Workshop, RC 2012, Copenhagen, Denmark, July 2-3, 2012.

[2] I. C. C. Phillips and I. Ulidowski. Reversing algebraic process calculi. J. Log. Algebr. Program., 2007.

Reversible CCS

Syntax of RCCS is:

$$\begin{aligned}
\text{(CCS Processes)} \quad & P, Q ::= \mathbf{0} \mid \alpha.P \mid (P|Q) \mid \sum \alpha_i.P \mid P\backslash A \\
\text{(RCCS Processes)} \quad & R, S ::= m \triangleright P \mid (R|S) \mid R\backslash A \\
\text{(Memories)} \quad & m ::= \langle\rangle \mid \langle i, \alpha, P\rangle \cdot m \mid \langle\uparrow\rangle \cdot m
\end{aligned}$$

where

- i is an identifier
- $\langle\uparrow\rangle$ represents a splitting event

Extract of semantic rules:

$$\text{(R-ACT)} \qquad m \triangleright \alpha.P + Q \rightarrow^{i}_{\alpha}$$

Extract of semantic rules:

$$\text{(R-ACT)} \qquad m \triangleright \alpha.P + Q \rightarrow^{i}_{\alpha} \langle i, \alpha, Q \rangle \cdot m \triangleright P$$

Extract of semantic rules:

$$\text{(R-ACT)} \qquad m \triangleright \alpha.P + Q \rightarrow^{i}_{\alpha} \langle i, \alpha, Q \rangle \cdot m \triangleright P$$

$$\langle i, \alpha, Q \rangle \cdot m \triangleright P \rightsquigarrow^{i}_{\alpha} m \triangleright \alpha.P + Q$$

Extract of semantic rules:

$$\text{(R-ACT)} \qquad m \triangleright \alpha.P + Q \rightarrow^{i}_{\alpha} \langle i, \alpha, Q \rangle \cdot m \triangleright P$$

$$\langle i, \alpha, Q \rangle \cdot m \triangleright P \rightsquigarrow^{i}_{\alpha} m \triangleright \alpha.P + Q$$

Extract of structural laws:

$$\text{(SPLIT)} \qquad m \triangleright (P|Q) \equiv \langle \uparrow \rangle \cdot m \triangleright P \mid \langle \uparrow \rangle \cdot m \triangleright Q$$

CCS with communication keys (CCSK)

- Syntax of CCSK is:

$$\text{(CCSK Processes)} \quad X, Y ::= \mathbf{0} \mid \sum_{i \in I} \pi_i.X_i \mid (X|Y) \mid X\backslash A$$

$$\text{(CCSK Prefixes)} \quad \pi ::= \alpha \mid \alpha[i]$$

where i is a key.

Definition (Reachable Process)

A CCSK process X is reachable if it can be derived from an CCS process P, by using semantics rules.

Extract of semantics rules:

$$\text{(K-ACT1)} \qquad \alpha.P \xrightarrow{\alpha[i]}$$

Extract of semantics rules:

$$\text{(K-ACT1)} \qquad \alpha.P \xrightarrow{\alpha[i]} \alpha[i].P$$

Extract of semantics rules:

(K-ACT1) $\alpha.P \xrightarrow{\alpha[i]} \alpha[i].P$

$\alpha[i].P \overset{\alpha[i]}{\rightsquigarrow} \alpha.P$

Extract of semantics rules:

(K-ACT1) $\alpha.P \xrightarrow{\alpha[i]} \alpha[i].P$

$\alpha[i].P \overset{\alpha[i]}{\rightsquigarrow} \alpha.P$

(K-SUM-L) $\alpha[i].\beta.P + Q \xrightarrow{\beta[j]}$

Extract of semantics rules:

(K-ACT1) $\alpha.P \xrightarrow{\alpha[i]} \alpha[i].P$

$\alpha[i].P \overset{\alpha[i]}{\rightsquigarrow} \alpha.P$

(K-SUM-L) $\alpha[i].\beta.P + Q \xrightarrow{\beta[j]} \alpha[i].\beta[j].P + Q$

Extract of semantics rules:

(K-ACT1) $\alpha.P \xrightarrow{\alpha[i]} \alpha[i].P$

$\alpha[i].P \overset{\alpha[i]}{\rightsquigarrow} \alpha.P$

(K-SUM-L) $\alpha[i].\beta.P + Q \xrightarrow{\beta[j]} \alpha[i].\beta[j].P + Q$

$\alpha[i].\beta[j].P + Q \overset{\beta[j]}{\rightsquigarrow} \alpha[i].\beta.P + Q$

Let's consider the CCS process $a.(b|c) + d$

- RCCS:

Let's consider the CCS process $a.(b|c) + d$

- RCCS:

$$\langle\rangle \triangleright a.(b|c) + d \rightarrow_a^i \langle i, a, d\rangle \cdot \langle\rangle \triangleright (b|c)$$

Let's consider the CCS process $a.(b|c) + d$

- RCCS:

$$\langle\rangle \triangleright a.(b|c) + d \rightarrow_a^i \langle i, a, d\rangle \cdot \langle\rangle \triangleright (b|c)$$
$$\langle i, a, d\rangle \cdot \langle\rangle \triangleright (b|c) \equiv \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright b \mid \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright c$$

Let's consider the CCS process $a.(b|c) + d$

- RCCS:

$$\langle\rangle \triangleright a.(b|c) + d \rightarrow_a^i \langle i, a, d\rangle \cdot \langle\rangle \triangleright (b|c)$$

$$\langle i, a, d\rangle \cdot \langle\rangle \triangleright (b|c) \equiv \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright b \mid \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright c$$

$$\langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright b \mid \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright c \rightarrow_b^j$$

$$\langle j, b, \mathbf{0}\rangle \cdot \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright \mathbf{0} \mid \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright c$$

Let's consider the CCS process $a.(b|c) + d$

- RCCS:

$$\langle\rangle \triangleright a.(b|c) + d \rightarrow_a^i \langle i, a, d\rangle \cdot \langle\rangle \triangleright (b|c)$$
$$\langle i, a, d\rangle \cdot \langle\rangle \triangleright (b|c) \equiv \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright b \mid \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright c$$
$$\langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright b \mid \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright c \rightarrow_b^j$$
$$\langle j, b, \mathbf{0}\rangle \cdot \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright \mathbf{0} \mid \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright c$$

- CCSK:

Let's consider the CCS process $a.(b|c) + d$

- RCCS:

$$\langle\rangle \triangleright a.(b|c) + d \rightarrow_a^i \langle i, a, d\rangle \cdot \langle\rangle \triangleright (b|c)$$
$$\langle i, a, d\rangle \cdot \langle\rangle \triangleright (b|c) \equiv \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright b \mid \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright c$$
$$\langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright b \mid \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright c \rightarrow_b^j$$
$$\langle j, b, \mathbf{0}\rangle \cdot \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright \mathbf{0} \mid \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright c$$

- CCSK:

$$a.(b|c) + d \xrightarrow{a[i]} a[i].(b|c) + d$$

Let's consider the CCS process $a.(b|c) + d$

- RCCS:

$$\langle\rangle \triangleright a.(b|c) + d \rightarrow_a^i \langle i, a, d\rangle \cdot \langle\rangle \triangleright (b|c)$$
$$\langle i, a, d\rangle \cdot \langle\rangle \triangleright (b|c) \equiv \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright b \mid \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright c$$
$$\langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright b \mid \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright c \rightarrow_b^j$$
$$\langle j, b, \mathbf{0}\rangle \cdot \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright \mathbf{0} \mid \langle\uparrow\rangle \cdot \langle i, a, d\rangle \cdot \langle\rangle \triangleright c$$

- CCSK:

$$a.(b|c) + d \xrightarrow{a[i]} a[i].(b|c) + d$$
$$a[i].(b|c) + d \xrightarrow{b[j]} a[i].(b[j]|c) + d$$

Encoding CCSK in RCCS

If $\mathcal{P}_K$ and $\mathcal{P}_R$ are the sets of processes from CCSK and RCCS, respectively, and $\mathcal{M}$ is the set of all the memories, the encoding function $[\![\cdot]\!] : \mathcal{P}_K \times \mathcal{M} \to \mathcal{P}_R$, is inductively defined as follows:

Encoding CCSK in RCCS

If $\mathcal{P}_K$ and $\mathcal{P}_R$ are the sets of processes from CCSK and RCCS, respectively, and $\mathcal{M}$ is the set of all the memories, the encoding function $[\![\cdot]\!] : \mathcal{P}_K \times \mathcal{M} \to \mathcal{P}_R$, is inductively defined as follows:

$$
\begin{aligned}
[\![X]\!] &= [\![X, \langle\rangle]\!] \\
[\![P, m]\!] &= m \triangleright P \\
[\![\alpha[i].X + \sum_{j \in J} \alpha_j.P_j, m]\!] &= [\![X, \langle i, \alpha, \sum_{j \in J} \alpha_j.P_j \rangle \cdot m]\!] \\
[\![X \backslash A, m]\!] &= [\![X, m]\!] \backslash A \\
[\![X | Y, m]\!] &= [\![X, \langle\uparrow\rangle \cdot m]\!] | [\![Y, \langle\uparrow\rangle \cdot m]\!]
\end{aligned}
$$

Example:

Let $X = a + c[i].(d[h]|P)$

Encoding is:

$$[\![X, \langle\rangle]\!] = [\![c[i].(d[h]|P) + a, \langle\rangle]\!]$$

Example:

Let $X = a + c[i].(d[h]|P)$

Encoding is:

$$\begin{aligned}[\![X, \langle\rangle]\!] &= [\![c[i].(d[h]|P) + a, \langle\rangle]\!] \\ &= [\![d[h]|P, \langle i, c, a\rangle \cdot \langle\rangle]\!]\end{aligned}$$

Example:

Let $X = a + c[i].(d[h]|P)$

Encoding is:

$$\begin{aligned}
[\![X, \langle\rangle]\!] &= [\![c[i].(d[h]|P) + a, \langle\rangle]\!] \\
&= [\![d[h]|P, \langle i, c, a\rangle \cdot \langle\rangle]\!] \\
&= [\![d[h], \langle\uparrow\rangle \cdot \langle i, c, a\rangle \cdot \langle\rangle]\!] \mid [\![P, \langle\uparrow\rangle \cdot \langle i, c, a\rangle \cdot \langle\rangle]\!]
\end{aligned}$$

Example:

Let $X = a + c[i].(d[h]|P)$

Encoding is:

$$\begin{aligned}
[\![X, \langle\rangle]\!] &= [\![c[i].(d[h]|P) + a, \langle\rangle]\!] \\
&= [\![d[h]|P, \langle i, c, a\rangle \cdot \langle\rangle]\!] \\
&= [\![d[h], \langle\uparrow\rangle \cdot \langle i, c, a\rangle \cdot \langle\rangle]\!] \mid [\![P, \langle\uparrow\rangle \cdot \langle i, c, a\rangle \cdot \langle\rangle]\!] \\
&= [\![\mathbf{0}, \langle h, d, \mathbf{0}\rangle \cdot \langle\uparrow\rangle \cdot \langle i, c, a\rangle \cdot \langle\rangle]\!] \mid \langle\uparrow\rangle \cdot \langle i, c, a\rangle \cdot \langle\rangle \triangleright P
\end{aligned}$$

Example:

Let $X = a + c[i].(d[h]|P)$

Encoding is:

$$\begin{aligned}
[\![X, \langle\rangle]\!] &= [\![c[i].(d[h]|P) + a, \langle\rangle]\!] \\
&= [\![d[h]|P, \langle i, c, a\rangle \cdot \langle\rangle]\!] \\
&= [\![d[h], \langle\uparrow\rangle \cdot \langle i, c, a\rangle \cdot \langle\rangle]\!] \mid [\![P, \langle\uparrow\rangle \cdot \langle i, c, a\rangle \cdot \langle\rangle]\!] \\
&= [\![\mathbf{0}, \langle h, d, \mathbf{0}\rangle \cdot \langle\uparrow\rangle \cdot \langle i, c, a\rangle \cdot \langle\rangle]\!] \mid \langle\uparrow\rangle \cdot \langle i, c, a\rangle \cdot \langle\rangle \triangleright P \\
&= \langle h, d, \mathbf{0}\rangle \cdot \langle\uparrow\rangle \cdot \langle i, c, a\rangle \cdot \langle\rangle \triangleright \mathbf{0} \mid \langle\uparrow\rangle \cdot \langle i, c, a\rangle \cdot \langle\rangle \triangleright P
\end{aligned}$$

Definition (Back and Forth (BF) Bisimulation)

Given a bijective function $\gamma : \mathcal{L}_{c_1} \to \mathcal{L}_{c_2}$, a relation ${}_{c_1}\mathcal{R}_{c_2} \subseteq \mathcal{P}_{c_1} \times \mathcal{P}_{c_2}$ is a strong back and forth simulation if whenever $P\,{}_{c_1}\mathcal{R}_{c_2}\,R$:

- $P \xrightarrow{\ell}_{c_1} Q$ implies $R \xrightarrow{\gamma(\ell)}_{c_2} S$ with $Q\,{}_{c_1}\mathcal{R}_{c_2}\,S$
- $P \overset{\ell}{\rightsquigarrow}_{c_1} Q$ implies $R \overset{\gamma(\ell)}{\rightsquigarrow}_{c_2} S$ with $Q\,{}_{c_1}\mathcal{R}_{c_2}\,S$

A relation ${}_{c_1}\mathcal{R}_{c_2} \subseteq \mathcal{P}_{c_1} \times \mathcal{P}_{c_2}$ is called a strong back and forth bisimulation if ${}_{c_1}\mathcal{R}_{c_2}$ and $({}_{c_1}\mathcal{R}_{c_2})^{-1}$ are strong back and forth simulations. We call strong bisimilarity, noted ${}_{c_1}\sim_{c_2}$, the largest bisimulation with respect to calculi c_1 and c_2.

Theorem (Bisimilarity)

For any reachable CCSK process X, $X \sim_{CCSK} \sim_{RCCS} [\![X, \langle\rangle]\!]$.

We show that the relation

$$\mathcal{R} = \{(X, R) \text{ with } X \text{ CCSK reachable } \wedge R \equiv [\![X, \langle\rangle]\!]\}$$

is a strong back and forth bisimulation.

Encoding RCCS to CCSK

In RCCS via structural congruence it is possible to split a parallel composition of processes sharing the same memory into a parallel composition of different monitored processes.

Encoding RCCS to CCSK

In RCCS via structural congruence it is possible to split a parallel composition of processes sharing the same memory into a parallel composition of different monitored processes.

Let us consider CCS process $\alpha.(\beta.P_1|\gamma.P_2) + Q$

Encoding RCCS to CCSK

In RCCS via structural congruence it is possible to split a parallel composition of processes sharing the same memory into a parallel composition of different monitored processes.

Let us consider CCS process $\alpha.(\beta.P_1|\gamma.P_2) + Q$

RCCS: $\langle\rangle \triangleright \alpha.(\beta.P_1|\gamma.P_2) + Q \rightarrow^{i}_{\alpha} \rightarrow^{j}_{\beta} R$, where

$$R = \langle j, \beta, \mathbf{0}\rangle \cdot \langle\uparrow\rangle \cdot \langle i, \alpha, Q\rangle \cdot \langle\rangle \triangleright P_1 \mid \langle\uparrow\rangle \cdot \langle i, \alpha, Q\rangle \cdot \langle\rangle \triangleright \gamma.P_2$$

Encoding RCCS to CCSK

In RCCS via structural congruence it is possible to split a parallel composition of processes sharing the same memory into a parallel composition of different monitored processes.

Let us consider CCS process $\alpha.(\beta.P_1|\gamma.P_2) + Q$

RCCS: $\langle\rangle \triangleright \alpha.(\beta.P_1|\gamma.P_2) + Q \rightarrow^{i}_{\alpha} \rightarrow^{j}_{\beta} R$, where

$$R = \langle j, \beta, \mathbf{0}\rangle \cdot \langle\uparrow\rangle \cdot \langle i, \alpha, Q\rangle \cdot \langle\rangle \triangleright P_1 \mid \langle\uparrow\rangle \cdot \langle i, \alpha, Q\rangle \cdot \langle\rangle \triangleright \gamma.P_2$$

CCSK: $\alpha.(\beta.P_1|\gamma.P_2) + Q \xrightarrow{\alpha[i]} \xrightarrow{\beta[j]} X$, where

$$X = \alpha[i].(\beta[j].P_1|\gamma.P_2) + Q$$

In the example

$$\alpha[i].(\beta[j].P_1 \mid \gamma.P_2) + Q$$

the encoding has to join together processes $\beta[j].P_1$ and $\gamma.P_2$ and put them in the context

$$\alpha[i].[\bullet] + Q$$

In this case encoding cannot be compositional .

Definition

The function δ of a memory m, is inductively defined as follows:

$$\delta(m \cdot \langle \uparrow \rangle) = \delta(m) \qquad \delta(\emptyset) = \langle\rangle$$
$$\delta(m \cdot \langle i, \alpha, Q \rangle) = m \cdot \langle i, \alpha, Q \rangle \qquad \delta(\langle \uparrow \rangle) = \langle\rangle$$

Let $\mathcal{P}_K$ and $\mathcal{P}_R$ be the set of processes from CCSK and RCCS, respectively. The encoding function $(\!|\cdot|\!) : \mathcal{P}_R \to \mathcal{P}_K$ is inductively defined as follows:

$$
\begin{aligned}
(\!|R \backslash A|\!) &= (\!|R|\!) \backslash A \\
(\!|(\textstyle\prod_{I \in I} m_I^0 \cdot \langle\uparrow\rangle \cdot m \triangleright P_I|\!) &= \wr m, \textstyle\prod_{I \in I} (\!|\delta(m_I^0) \triangleright P_I|\!) \int \quad \text{if } \langle\uparrow\rangle \notin m \\
(\!|m \triangleright P|\!) &= \wr m, P \int \\
\wr \langle i, \alpha, Q \rangle \cdot m, X \int &= \wr m, \alpha[i].X + Q \int \quad \text{if } Q \neq \mathbf{0} \\
\wr \langle i, \alpha, \mathbf{0} \rangle \cdot m, X \int &= \wr m, \alpha[i].X \int \\
\wr \langle\rangle, X \int &= X
\end{aligned}
$$

Let's get back on our example and encode process R.

$$(\!|R|\!) = (\!|\langle j, \beta, \mathbf{0}\rangle \cdot \langle\uparrow\rangle \cdot \langle i, \alpha, Q\rangle \cdot \langle\rangle \triangleright P_1 \mid \langle\uparrow\rangle \cdot \langle i, \alpha, Q\rangle \cdot \langle\rangle \triangleright \gamma.P_2|\!)$$

Let's get back on our example and encode process R.

$$
\begin{aligned}
(\!|R|\!) &= (\!|\langle j, \beta, \mathbf{0}\rangle \cdot \langle\uparrow\rangle \cdot \langle i, \alpha, Q\rangle \cdot \langle\rangle \triangleright P_1 \mid \langle\uparrow\rangle \cdot \langle i, \alpha, Q\rangle \cdot \langle\rangle \triangleright \gamma.P_2|\!) \\
&= \wr \langle i, \alpha, Q\rangle, (\!|\delta(\langle j, \beta, \mathbf{0}\rangle) \triangleright P_1|\!) \mid (\!|\delta(\emptyset) \triangleright \gamma.P_2|\!) \int
\end{aligned}
$$

Let's get back on our example and encode process R.

$$
\begin{aligned}
(\!|R|\!) &= (\!|\langle j, \beta, \mathbf{0}\rangle \cdot \langle\uparrow\rangle \cdot \langle i, \alpha, Q\rangle \cdot \langle\rangle \triangleright P_1 \mid \langle\uparrow\rangle \cdot \langle i, \alpha, Q\rangle \cdot \langle\rangle \triangleright \gamma.P_2|\!) \\
&= \wr\langle i, \alpha, Q\rangle, (\!|\delta(\langle j, \beta, \mathbf{0}\rangle) \triangleright P_1|\!) \mid (\!|\delta(\emptyset) \triangleright \gamma.P_2|\!)\int \\
&= \wr\langle i, \alpha, Q\rangle, (\!|\langle j, \beta, \mathbf{0}\rangle \triangleright P_1|\!) \mid (\!|\langle\rangle \triangleright \gamma.P_2|\!)\int
\end{aligned}
$$

Let's get back on our example and encode process R.

$$
\begin{aligned}
(\!|R|\!) &= (\!|\langle j, \beta, \mathbf{0}\rangle \cdot \langle\uparrow\rangle \cdot \langle i, \alpha, Q\rangle \cdot \langle\rangle \triangleright P_1 \mid \langle\uparrow\rangle \cdot \langle i, \alpha, Q\rangle \cdot \langle\rangle \triangleright \gamma.P_2|\!) \\
&= \wr\langle i, \alpha, Q\rangle, (\!|\delta(\langle j, \beta, \mathbf{0}\rangle) \triangleright P_1|\!) \mid (\!|\delta(\emptyset) \triangleright \gamma.P_2|\!)\int \\
&= \wr\langle i, \alpha, Q\rangle, (\!|\langle j, \beta, \mathbf{0}\rangle \triangleright P_1|\!) \mid (\!|\langle\rangle \triangleright \gamma.P_2|\!)\int \\
&= \wr\langle i, \alpha, Q\rangle, \wr\langle j, \beta, \mathbf{0}\rangle, P_1\int \mid \wr\langle\rangle, \gamma.P_2\int\int
\end{aligned}
$$

Let's get back on our example and encode process R.

$$
\begin{aligned}
(\!|R|\!) &= (\!|\langle j, \beta, \mathbf{0}\rangle \cdot \langle\uparrow\rangle \cdot \langle i, \alpha, Q\rangle \cdot \langle\rangle \triangleright P_1 \mid \langle\uparrow\rangle \cdot \langle i, \alpha, Q\rangle \cdot \langle\rangle \triangleright \gamma.P_2|\!) \\
&= \wr\langle i, \alpha, Q\rangle, (\!|\delta(\langle j, \beta, \mathbf{0}\rangle) \triangleright P_1|\!) \mid (\!|\delta(\emptyset) \triangleright \gamma.P_2|\!)\int \\
&= \wr\langle i, \alpha, Q\rangle, (\!|\langle j, \beta, \mathbf{0}\rangle \triangleright P_1|\!) \mid (\!|\langle\rangle \triangleright \gamma.P_2|\!)\int \\
&= \wr\langle i, \alpha, Q\rangle, \wr\langle j, \beta, \mathbf{0}\rangle, P_1\int \mid \wr\langle\rangle, \gamma.P_2\int\int \\
&= \wr\langle i, \alpha, Q\rangle, \wr\langle\rangle, \beta[j].P_1\int \mid \gamma.P_2\int
\end{aligned}
$$

Let's get back on our example and encode process R.

$$
\begin{aligned}
(\!|R|\!) &= (\!|\langle j, \beta, \mathbf{0}\rangle \cdot \langle\uparrow\rangle \cdot \langle i, \alpha, Q\rangle \cdot \langle\rangle \triangleright P_1 \mid \langle\uparrow\rangle \cdot \langle i, \alpha, Q\rangle \cdot \langle\rangle \triangleright \gamma.P_2|\!) \\
&= \wr\langle i, \alpha, Q\rangle, (\!|\delta(\langle j, \beta, \mathbf{0}\rangle) \triangleright P_1|\!) \mid (\!|\delta(\emptyset) \triangleright \gamma.P_2|\!)\int \\
&= \wr\langle i, \alpha, Q\rangle, (\!|\langle j, \beta, \mathbf{0}\rangle \triangleright P_1|\!) \mid (\!|\langle\rangle \triangleright \gamma.P_2|\!)\int \\
&= \wr\langle i, \alpha, Q\rangle, \wr\langle j, \beta, \mathbf{0}\rangle, P_1\int \mid \wr\langle\rangle, \gamma.P_2\int\int \\
&= \wr\langle i, \alpha, Q\rangle, \wr\langle\rangle, \beta[j].P_1\int \mid \gamma.P_2\int \\
&= \wr\langle i, \alpha, Q\rangle, \beta[j].P_1 \mid \gamma.P_2\int
\end{aligned}
$$

Let's get back on our example and encode process R.

$$
\begin{aligned}
(\!|R|\!) &= (\!|\langle j, \beta, \mathbf{0}\rangle \cdot \langle\uparrow\rangle \cdot \langle i, \alpha, Q\rangle \cdot \langle\rangle \triangleright P_1 \mid \langle\uparrow\rangle \cdot \langle i, \alpha, Q\rangle \cdot \langle\rangle \triangleright \gamma.P_2|\!) \\
&= \wr\langle i, \alpha, Q\rangle, (\!|\delta(\langle j, \beta, \mathbf{0}\rangle) \triangleright P_1|\!) \mid (\!|\delta(\emptyset) \triangleright \gamma.P_2|\!)\wr \\
&= \wr\langle i, \alpha, Q\rangle, (\!|\langle j, \beta, \mathbf{0}\rangle \triangleright P_1|\!) \mid (\!|\langle\rangle \triangleright \gamma.P_2|\!)\wr \\
&= \wr\langle i, \alpha, Q\rangle, \wr\langle j, \beta, \mathbf{0}\rangle, P_1\wr \mid \wr\langle\rangle, \gamma.P_2\wr\wr \\
&= \wr\langle i, \alpha, Q\rangle, \wr\langle\rangle, \beta[j].P_1\wr \mid \gamma.P_2\wr \\
&= \wr\langle i, \alpha, Q\rangle, \beta[j].P_1 \mid \gamma.P_2\wr \\
&= \wr\langle\rangle, \alpha[i].(\beta[j].P_1 \mid \gamma.P_2) + Q\wr
\end{aligned}
$$

Let's get back on our example and encode process R.

$$
\begin{aligned}
(\!|R|\!) &= (\!|\langle j, \beta, \mathbf{0}\rangle \cdot \langle\uparrow\rangle \cdot \langle i, \alpha, Q\rangle \cdot \langle\rangle \triangleright P_1 \mid \langle\uparrow\rangle \cdot \langle i, \alpha, Q\rangle \cdot \langle\rangle \triangleright \gamma.P_2|\!) \\
&= \wr\langle i, \alpha, Q\rangle, (\!|\delta(\langle j, \beta, \mathbf{0}\rangle) \triangleright P_1|\!) \mid (\!|\delta(\emptyset) \triangleright \gamma.P_2|\!)\wr \\
&= \wr\langle i, \alpha, Q\rangle, (\!|\langle j, \beta, \mathbf{0}\rangle \triangleright P_1|\!) \mid (\!|\langle\rangle \triangleright \gamma.P_2|\!)\wr \\
&= \wr\langle i, \alpha, Q\rangle, \wr\langle j, \beta, \mathbf{0}\rangle, P_1\wr \mid \wr\langle\rangle, \gamma.P_2\wr\wr \\
&= \wr\langle i, \alpha, Q\rangle, \wr\langle\rangle, \beta[j].P_1\wr \mid \gamma.P_2\wr \\
&= \wr\langle i, \alpha, Q\rangle, \beta[j].P_1 \mid \gamma.P_2\wr \\
&= \wr\langle\rangle, \alpha[i].(\beta[j].P_1 \mid \gamma.P_2) + Q\wr \\
&= \alpha[i].(\beta[j].P_1 \mid \gamma.P_2) + Q
\end{aligned}
$$

Theorem

Let R to be a RCCS reachable process. Then $[\![(\!|R|\!), \langle\rangle]\!] = R$.

Cross-fertilization results

- RCCS is causally consistent $\Longrightarrow$ CCSK is causally consistent
- CCSK has Reverse Diamond Property (Reverse transitions are confluent) $\Longrightarrow$ Reverse Diamond Property holds in RCCS
- We can bring transactions from RCCS to CCSK

Conclusion

We show that RCCS and CCSK are equivalent in terms of strong back and forth bisimulation

Future work:

- show that back and forth bisimulation is a congruence
  - e.g. Forward-Reverse (FR) Bisimulation is a BF Bisimulation
- show that two bisimilar CCSK terms are translated into two bisimimilar RCCS terms.

References

- J. Krivine. A verification technique for reversible process algebra. In Reversible Computation, 4th International Workshop, RC 2012, Copenhagen, Denmark, July 2-3, 2012.
- I. C. C. Phillips and I. Ulidowski. Reversing algebraic process calculi. J. Log. Algebr. Program., 2007.
- V. Danos and J. Krivine. Reversible communicating systems. In CONCUR 2004 - Concurrency Theory, 15th International Conference, London, UK, August 31 - September 3, 2004
- D. Medic and C. A. Mezzina. Static VS dynamic reversibility in CCS. In S. J. Devitt and I. Lanese, editors, Reversible Computation - 8th International Conference, RC 2016, Bologna, Italy