Forward-Reverse Observational Equivalences in CCSK

Ivan Lanese

University of Bologna, Italy/INRIA, France

Joint work with Iain Phillips (Imperial College)

Outline

Behavioural equivalences for reversible systems

CCSK

Our insights and results

Conclusion

Reversible computation

Reversible computation allows computation to proceed not only in the standard, forward direction, but also backwards, recovering past states.

Applications in different areas:

- low-power computing (Landauer 1961)
- optimistic parallel simulation (Carothers et al 1999)
- error recovery in robot assembly operations (Laursen et al 2015)
- debugging (GDB since 2009, WinDbg)
- ...

Reversible models of concurrent systems

In many of these areas, concurrent systems are of interest.

Reversible extensions of concurrent models and languages have been proposed

Seminal one, RCCS (Danos & Krivine 2004) is a reversible form of CCS (Milner 1980)

Another reversible CCS, CCSK, has been proposed by Phillips & Ulidowski in 2006

Reversible extensions of π-calculus, Petri Nets, Erlang and others exist

Main idea: add memories so that computation can be reversed

Reversibility and concurrency

In a sequential setting actions are undone in reverse order:

$$P \xrightarrow{a} Q \xrightarrow{b} R \qquad\qquad R \xrightarrow{b} Q \xrightarrow{a} P$$

In concurrent systems, the total order of actions is not relevant and may not even exist.

Causal-consistent reversibility (Danos & Krivine 2004)

An action can be reversed iff all its consequences (if any) have been already reversed.

If $P \xrightarrow{a} Q$ causes $Q \xrightarrow{b} R$ then we cannot reverse a before b.

But if $P \xrightarrow{a} Q$ and $Q \xrightarrow{b} R$ are concurrent then we can reverse them in any order:

$$P \xrightarrow{a} Q \xrightarrow{b} R \qquad\qquad R \xrightarrow{a} Q' \xrightarrow{b} P$$

The need for analysis techniques

(Reversible) models allow one to describe systems

We also want to reason on such systems

Many analysis techniques in the literature: (behavioural) types, model checking, behavioural equivalences, ...

Behavioural equivalences

Equivalence relations on processes

Equivalent processes are not distinguishable by some form of observation

Barbed congruence: relation closed under reductions (that is, internal steps), basic observations (called barbs), and contexts

Bisimulation: relation closed under transitions (that is interactions with the environment)

From the concurrency theory community:

- Barbed congruence is more natural and straightforward to define;
- It is difficult to work with barbed congruence due to the universal quantification over contexts;
- Bisimulation is frequently used as a tool to prove barbed congruence.

Framing the problem

We want to find suitable behavioural equivalences:

for causal-consistent reversibility, since we are interested in concurrent systems (and behavioural equivalences are tailored for them);

direction sensitive, that is distinguishing forward from backward steps;

for uncontrolled reversibility: no policy on whether to go forward or backward, or which action to take if many are enabled;

strong equivalences: distinguish processes that produce the same observation after a different number of internal steps.

Controlled reversibility and weak equivalences are interesting, but first the more basic setting we consider needs to be understood.

Selecting the target language

We will work on CCSK

CCS is a simple starting point, yet it is very used

A number of works already tackled this setting (Phillips & Ulidowski 2006, ...)

RCCS has too much redundancy, making axioms very difficult to write (we will come back to this)

Outline

Behavioural equivalences for reversible systems

CCSK

Our insights and results

Conclusion

CCSK

CCSK syntax

$$X, Y := \pi.X \mid X + Y \mid (X \mid Y) \mid (\nu a)X \mid 0$$

$$\pi := \alpha \mid \alpha[k]$$

$$\alpha =: a \mid \overline{a} \mid \tau$$

$a, b, \cdots$: communication channels
$k, m, \cdots$: keys

Keys highlight when the corresponding prefix has been executed, processes without keys are CCS processes.

CCSK semantics: sample rules

Forward	Backward
$\dfrac{\mathtt{std}(X)}{\alpha.X \xrightarrow{\alpha[m]}_f \alpha[m].X}$	$\dfrac{\mathtt{std}(X)}{\alpha[m].X \xrightarrow{\alpha[m]}_r \alpha.X}$
$\dfrac{X \xrightarrow{\beta[n]}_f X'}{\alpha[m].X \xrightarrow{\beta[n]}_f \alpha[m].X'}\ m \neq n$	$\dfrac{X \xrightarrow{\beta[n]}_r X'}{\alpha[m].X \xrightarrow{\beta[n]}_r \alpha[m].X'}\ m \neq n$
$\dfrac{X \xrightarrow{\alpha[m]}_f X' \quad Y \xrightarrow{\overline{\alpha}[m]}_f Y'}{X \mid Y \xrightarrow{\tau[m]}_f X' \mid Y'}\ \alpha \neq \tau$	$\dfrac{X \xrightarrow{\alpha[m]}_r X' \quad Y \xrightarrow{\overline{\alpha}[m]}_r Y'}{X \mid Y \xrightarrow{\tau[m]}_r X' \mid Y'}\ \alpha \neq \tau$

Reachable processes

In reversible calculi only processes which have consistent history information are of interest.

Definition (Reachable process)

A process is reachable if there is a derivation leading to it from a process with no keys (standard process).

Side result

In the paper you can find a correct and complete syntactic characterisation of reachable processes. We are not aware of similar characterisations in the literature.

Starting point for bisimulation definition

We start from the definition in [Phillips & Ulidowski, 2007]:

Definition (Forward-reverse bisimulation)

A symmetric relation $\mathcal{R}$ is a *forward-reverse bisimulation* if whenever $X \mathcal{R} Y$:

1. $\mathtt{keys}(X) = \mathtt{keys}(Y)$;
2. if $X \xrightarrow{\mu}_f X'$ then there is Y' such that $Y \xrightarrow{\mu}_f Y'$ and $X' \mathcal{R} Y'$;
3. if $X \xrightarrow{\mu}_r X'$ then there is Y' such that $Y \xrightarrow{\mu}_r Y'$ and $X' \mathcal{R} Y'$.

Outline

On keys

Keys serve two purposes:

- to distinguish executed from non-executed prefixes;
- to link actions which have synchronised.

A key may be free in a process (one occurrence, not attached to a τ) or bound (two occurrences, or one attached to a τ).

If a key is free then the other occurrence should be in the context.

Key insight

Identity of free keys matters, identity of bound keys does not. E.g., we want:

$$\overline{a}[n] \mid a[n] \;\mathcal{R}\; \overline{a}[m] \mid a[m]$$

$$\overline{a}[n] \;\not\mathcal{R}\; \overline{a}[m]$$

The processes in the latter will behave differently in a context $\cdot \mid a[n]$.

Our solution

We add rules for α-conversion of bound keys.

$$X \equiv X[n/m] \qquad m \text{ bound in } X, n \notin \texttt{keys}(X)$$

$$\frac{Y \equiv X \quad X \xrightarrow{\alpha[m]}_f X' \quad X' \equiv Y'}{Y \xrightarrow{\alpha[m]}_f Y'} \qquad \frac{Y \equiv X \quad X \xrightarrow{\alpha[m]}_r X' \quad X' \equiv Y'}{Y \xrightarrow{\alpha[m]}_r Y'}$$

Without changing the semantics we would have:

$$\overline{a}[n] \mid a[n] \mid b \not\mathcal{R}\ \overline{a}[m] \mid a[m] \mid b$$

since the former could choose m as new key to execute b.

Revised FR bisimulation

Definition (Revised forward-reverse bisimulation)

A symmetric relation $\mathcal{R}$ is a *revised forward-reverse bisimulation* if whenever $X \mathrel{\mathcal{R}} Y$:

1. if $X \xrightarrow{\mu}_f X'$ then there is Y' such that $Y \xrightarrow{\mu}_f Y'$ and $X' \mathrel{\mathcal{R}} Y'$;
2. if $X \xrightarrow{\mu}_r X'$ then there is Y' such that $Y \xrightarrow{\mu}_r Y'$ and $X' \mathrel{\mathcal{R}} Y'$.

Revised FR bisimilarity, written $\sim$, is the largest revised FR bisimulation.

Matching bound keys is irrelevant thanks to α-conversion.

Barbed congruence

Definition (Forward-reverse barbed congruence)

A symmetric relation $\mathcal{R}$ is a *forward-reverse (FR) barbed bisimulation* if whenever $X \mathcal{R} Y$:

- $X \downarrow_{\overline{a}}$ implies $Y \downarrow_{\overline{a}}$;
- $X \uparrow_{\alpha[n]}$ implies $Y \uparrow_{\alpha[n]}$;
- if $X \xrightarrow{\tau[n]}_f X'$ then there is Y' such that $Y \xrightarrow{\tau[n]}_f Y'$ and $X' \mathcal{R} Y'$;
- if $X \xrightarrow{\tau[n]}_r X'$ then there is Y' such that $Y \xrightarrow{\tau[n]}_r Y'$ and $X' \mathcal{R} Y'$.

A *forward-reverse (FR) barbed congruence* is a FR barbed bisimulation such that $X \mathcal{R} Y$ implies $C[X] \mathcal{R} C[Y]$ for each C such that $C[X]$ and $C[Y]$ are both reachable.

On barbs

Definition (Barbs)

Forward output barb at a: $\downarrow_{\overline{a}}$ iff $X \xrightarrow{\overline{a}[n]}_f X'$ for some n and X'.

Backward barb at $\alpha[n]$: $\uparrow_{\alpha[n]}$ iff $X \xrightarrow{\alpha[n]}_r X'$ for some X' ($\alpha \neq \tau$).

Having forward barbs as detailed as the backward ones will not change the equivalence.

Why do we need so detailed backward barbs?

We would like $a[n] \mathcal{R} b[n]$.

If a context $C[\bullet]$ is able to interact with $a[n]$ then $C[b[n]]$ is not reachable, since occurences of the same key should be attached to complementary prefixes.

Main results

Theorem

Revised FR bisimilarity is a congruence (provided that the compositions are reachable).

Theorem

Revised FR bisimilarity coincides with the largest FR barbed congruence.

Theorem

Revised FR bisimilarity on standard processes is strictly finer than CCS bisimilarity.

Indeed $a.b + b.a$ and $a \mid b$ are equivalent in CCS (this is an instance of the Expansion Law) but not for revised FR bisimilarity.

Sound axioms

A number of axioms can be easily proved sound, e.g.:

Sound axioms

$$X \mid Y \sim Y \mid X$$

$$X \mid 0 \sim X$$

$$(\nu a)(\nu b)X \sim (\nu b)(\nu a)X$$

$$(\nu a)(X \mid Y) \sim X \mid (\nu a)Y \qquad \text{iff } a \notin \mathtt{fn}(X)$$

$$(\ldots)$$

$$X + P \sim X \qquad \text{iff } \mathtt{toStd}(X) = P$$

$$(\nu a)(\overline{a}.P \mid a.Q) \sim \tau.(\nu a)(P \mid Q)$$

$$(\nu a)(\overline{a}[n].X \mid a[n].Y) \sim \tau[n].(\nu a)(X \mid Y)$$

$$\tau \mid \tau \sim \tau.\tau$$

Why not RCCS?

Example (In CCSK)

$$a.(P \mid Q) + a.(P \mid Q) \xrightarrow{a[n]}_f a[n].(P \mid Q) + a.(P \mid Q)$$

The simple axiom $X + P \sim X$ (if $\texttt{toStd}(X) = P$) allows us to prove:

$$a[n].(P \mid Q) + a.(P \mid Q) \sim a[n].(P \mid Q)$$

Example (In RCCS)

$$\emptyset \triangleright a.(P \mid Q) + a.(P \mid Q) \xrightarrow{\emptyset:a}{}_f^{RCCS} [*, a, a.(P \mid Q)] \cdot \emptyset \triangleright (P \mid Q) \equiv$$
$$(< 1 > \cdot[*, a, a.(P \mid Q)] \cdot \emptyset \triangleright P) \mid (< 2 > \cdot[*, a, a.(P \mid Q)] \cdot \emptyset \triangleright Q)$$

You are welcome to try to write down axiom(s) to prove:

$$(< 1 > \cdot[*, a, a.(P \mid Q)] \cdot \emptyset \triangleright P) \mid (< 2 > \cdot[*, a, a.(P \mid Q)] \cdot \emptyset \triangleright Q) \sim$$
$$(< 1 > \cdot[*, a, 0] \cdot \emptyset \triangleright P) \mid (< 2 > \cdot[*, a, 0] \cdot \emptyset \triangleright Q)$$

Outline

Summary

- We defined a new form of bisimulation for causal-consistent systems.
- We proved it equivalent to a form of barbed congruence.
- We proved correct a number of axioms.

Future work

- Consider weak equivalences and controlled reversibility.
- Tackle more complex calculi (it requires modelling them with no redundancy).
- Characterising the equivalence induced on CCS (hereditary history-preserving bisimilarity?).