An Axiomatic Approach to Reversibility

Irek Ulidowski

University of Leicester, Leicester, England,

AGH University of Science and Technology, Kraków, Poland

OPCT 2023, Bertinoro

Joint work with Ivan Lanese (Bologna) and Iain Phillips (Imperial College)

Outline

Reversible computation

Axiomatic approach

Case studies

Open problems

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 discrete event simulation (Carothers et al 1999)
- debugging (GDB since 2009, Undo UDB)
- error recovery in robot assembly operations (Laursen et al 2015)
- modelling of bio-chemical reactions

Reversible calculi and languages

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

Reversible extensions of concurrent formalisms and languages:

- RCCS (Danos & Krivine 2004) is a reversible form of CCS (Milner 1980) using memory stacks,
- CCSK (Phillips & Ulidowski Algebraic Process Algebras: the first 25 years and beyond, Bertinoro, 2005; FoSSaCS 2006) uses communication keys (identifiers) not stacks,
- Axiomatic approach (Phillips & Ulidowski 2007) rather than ad hoc properties.

Reversible extensions of π-calculus, event structures, Petri nets, Erlang and others exist.

Main idea for reversing a language: add some form of memory so that computation can be reversed.

Outline

Reversible computation

Axiomatic approach

Case studies

Open problems

Our research question

Observation: in all the settings, similar properties are proved. Techniques are similar but ad hoc.

Can we develop a general theory and then instantiate it on different formalisms?

Advantages:

- prove results once and for all
    - encourages automatic proof checking
- highlight similarities and differences among approaches

Abstraction

We abstract away the syntax of the formalisms

We just consider their (reversible) labelled transition system (LTS):

Forward transition: $P \xrightarrow{a} Q$

Backward transition: $Q \xrightarrow{\underline{a}} P$

Forward or backward transition: $t : P \xrightarrow{\alpha} Q$

Inverse of t always exists (Loop Lemma): $\underline{t} : Q \xrightarrow{\underline{\alpha}} P$

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 order of actions is less relevant:

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 cannot reverse a before b.

But if $P \xrightarrow{a} Q$ and $Q \xrightarrow{b} R$ are independent (concurrent) we can have

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

Here Q' was not visited going forwards, but could have been:

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

Causal equivalence

We need some preliminary notions.

Paths r, s are sequences of transitions $t_1 t_2 \ldots t_n$.

Causal equivalence on paths: $r \approx s$ iff s can be obtained from r by

1. swapping adjacent independent transitions
2. adding/removing pairs of do/undo or undo/redo: $t\underline{t} = \underline{t}t = \epsilon$

To do this, one has to fix a notion of independence.

Causal Consistency

Causal-consistent reversibility has been mostly charaterised in terms of the following property

Causal Consistency (CC - Danos & Krivine 2004)

If r and s are coinitial and cofinal paths then $r \approx s$.

It captures the fact that the information stored in memory is compatible with the notion of causal equivalence.

Proofs of CC are quite lengthy but mostly take a similar approach.

The relationship of CC with the intuitive definition on page 8 is not clear, and has not been studied much in the literature.

Our approach

Our idea

We want to show that properties such as CC follow from a small set of axioms. Proving the axioms should be easier than proving the properties directly.

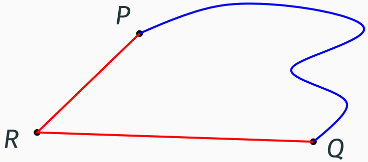
We use abstract labelled transition systems with independence (LTSIs). Related to LTSIs of Sassone et al (1996).

- We treat reverse transitions as first-class citizens;
- We adopt a minimal set of axioms and add more as needed.

Classical proof of CC

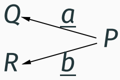
A typical proof of CC uses the Parabolic Lemma (PL):

Every path is causal equivalent to a backward path followed by a forward path



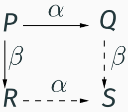
Basic axioms

1. Coinitial backward transitions are independent (BTI):



(generalizes backward determinism from sequential reversibility)

2. Square property (SP): If transitions are coinitial and independent then we can close the diamond:



3. Well-foundedness (WF): no infinite reverse path

$$\cdots \xrightarrow{a_{n+1}} P_n \xrightarrow{a_n} \cdots \xrightarrow{a_2} P_1 \xrightarrow{a_1} P_0$$

Cannot reverse to before starting point.

First results

Theorem

If BTI and SP then PL.

Theorem

If WF and PL then CC.

- Proof much shorter than existing proofs
- Success for the axiomatic approach
- Shows that CC is not much stronger than PL

Beyond CC

If CC is weaker than thought, how should we characterise causal-consistent reversibility (An action can be reversed iff all its consequences have been already reversed)?

Split its informal definition into:

- Causal Safety: if we can reverse t, then all consequences of t have been undone (all events after t are independent of t);
- Causal Liveness: if all events after t are not consequences of t (namely are independent of t), then we can reverse t.

Formalising CS and CL

We give three definitions of CS/CL:

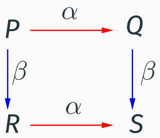
- via independence of transitions ($P \xrightarrow{a} Q \ \iota \ Q_1 \xrightarrow{c} Q_2$)
- via independence of events ($[P \xrightarrow{a} Q] \ \mathrm{ci} \ [Q_1 \xrightarrow{c} Q_2]$)
- via ordering of events ($[P \xrightarrow{a} Q] \not< [Q_1 \xrightarrow{c} Q_2]$)

With minimal axioms these are all different, but with our full set of axioms they become equivalent.

But what are events?

Events are equivalence classes of transitions.

Equate transitions representing the same action executed at different points in the computation.



If coinitial transitions in the square are independent then we let

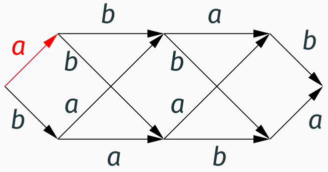
$$P \xrightarrow{\alpha} Q \sim R \xrightarrow{\alpha} S \qquad P \xrightarrow{\beta} R \sim Q \xrightarrow{\beta} S$$

Get two events $[P \xrightarrow{\alpha} Q]$ and $[P \xrightarrow{\beta} R]$ as equivalence classes.

Lift independence to events: $[t_1]$ ci $[t_2]$ if have representatives t_1' and t_2' which are coinitial and independent.

CC does not imply CS or CL

Satisfies all axioms so far, hence PL+CC, but not CS and not CL:



Independence: BTI + leftmost a is independent on all the b.

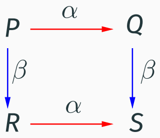
CS fails on *ba*$\underline{b}$: we reverse *b* but *a*, which a consequence of *b*, has not been undone yet.

CL fails on *abb*: *a* is not a consequence of *b*, but it cannot be undone after *abb*.

We provide further axioms from which CS and CL can be deduced.

Axioms

1. BTI: backward transitions are independent
2. SP: square property
3. WF: well-founded
4. PCI: propagation of coinitial independence (around a square)



5. IRE: independence respects events (if $t \sim t' \iota u$ then $t \iota u$)
6. CIRE: Coinitial IRE
7. IEC: independence of events is coinitial (if $t \iota u$ then $[t]$ ci $[u]$)

If in adition to 1-3 some of 4-7 hold, then versions of CS and CL hold.

Structural axioms CLG: coinitial label generated, and LG: label generated. Independence is defined purely by reference to labels.

Allows to derive some axioms, for example PCI, CIRE and IRE.

Outline

Reversible computation

Axiomatic approach

Case studies

Open problems

RCCS

Independence coincides with concurrency.

All the axioms are satisfied. Mostly proved in the original paper or trivial. CPI and IRE follow easily by CLG (since independence is defined on labels).

We get for free PL, CC, three forms of CS and CL (and other minor results).

HOπ and Erlang

Similar to RCCS, the main difference is that they have reduction semantics. However, richer labels have been defined using the memories involved in transitions, and CLG holds.

We also get IRE by extending coinitial independence along events.

We obtain for free PL, CC, and forms of CS and CL (and other minor results).

Reversible occurrence nets

Global notion of concurrency.

SP proved in the original paper. BTI, WF and PCI can easily be shown.

Since concurrency is global, IRE follows, which implies CIRE.

As all axioms are satisfied, we get for free PL, CC, CS and CL.

Summary

- We presented basic axioms which are satisfied by RCCS and other reversible formalisms.
- Verifying these axioms is easier than verifying the properties directly.
- Causal Consistency provides limited information, and should be supplemented by Causal Safety and Causal Liveness.
- Our abstract proofs are relatively easy to formalise in a proof assistant.

Outline

Reversible computation

Axiomatic approach

Case studies

Open problems

Open problems

- Equational axiom systems for reversible PAs.
- Reexamine testing semantics of PAs in presense of reversibility of processes and tests.
- Fully symmetric calculus for forward/reverse computation.
- Understand better out-of-causal order reversibility.