Journal of Logical and Algebraic Methods in Programming 84 (2015) 781–805

Contents lists available at ScienceDirect

Journal of Logical and Algebraic Methods in Programming

www.elsevier.com/locate/jlamp

Reversibility and asymmetric conflict in event structures

Iain Phillips [a,*], Irek Ulidowski [b]

[a] Department of Computing, Imperial College London, England, United Kingdom
[b] Department of Computer Science, University of Leicester, England, United Kingdom

ARTICLE INFO

Article history:
Received 30 November 2014
Received in revised form 22 May 2015
Accepted 10 July 2015
Available online 17 July 2015

Keywords:
Reversible computation
Event structure
Asymmetric conflict

ABSTRACT

Reversible computation has attracted increasing interest in recent years, with applications in hardware, software and biochemistry. We introduce reversible forms of prime event structures and asymmetric event structures. In order to control the manner in which events are reversed, we use asymmetric conflict on events. We prove a number of results about reachable configurations; for instance, we show under what conditions reachable configurations which are finite are reachable by purely finite means. We discuss, with examples, reversing in causal order, where an event is only reversed once all events it caused have been reversed, as well as forms of non-causal reversing.

1. Introduction

Causal reversibility in concurrent systems means that events that cause other events can only be undone after the caused events are undone first, and that events which are independent of each other can be reversed in an arbitrary order. The last decade has produced a good understanding of how causal reversibility can be achieved in the settings of operational semantics and process calculi. Research on reversing process calculi can be traced back perhaps to Berry and Boudol's Chemical Abstract Machine [3]. Danos and Krivine reversed CCS [6,7] and then together with Sobociński took a more abstract approach with an application to Petri nets [8]. A general method for reversing process calculi was proposed in [16], and reversible structures that compute forwards and backwards asynchronously were developed by Cardelli and Laneve [5]. Mechanisms for controlling reversibility based on a rollback construct were devised by Lanese, Mezzina, Schmitt and Stefani [12] for a reversible higher-order π calculus [13], and an alternative mechanism based on the execution control operator was proposed in [20].

Perhaps with the exception of [20,21,24], other common forms of reversibility, such as *inverse causal* reversibility, have not been studied yet. In [21] we presented an initial study of a form of reversible event structure based on a generalisation of Winskel's enabling relation [25]. In this paper we propose reversible event structures which are strongly contrasted to those of [21], as we here focus on analysing conflict and causation as first-class notions in the setting of reversible computation, rather than maximising expressive power.

We here take the view that reversing an event a means that a is removed from the current configuration (a set of events which have occurred and have not been reversed), and it is as if a had never occurred, apart possibly from indirect effects, such as a having caused another event b before a was reversed.

* Corresponding author.
E-mail addresses: i.phillips@imperial.ac.uk (I. Phillips), iu3@leicester.ac.uk (I. Ulidowski).

http://dx.doi.org/10.1016/j.jlamp.2015.07.004
2352-2208/

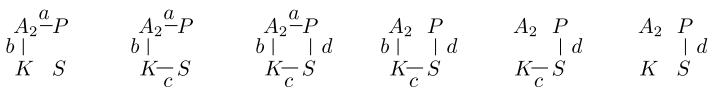

Fig. 1. Basic catalytic cycle for substrate phosphorylation by a kinase.

Our motivating example is the basic catalytic cycle for protein substrate phosphorylation by a kinase. We describe how bonds are created and dissolved in the cycle as presented in [23, Fig. 1a]. A kinase K aims to transfer a phosphate group P from a nucleotide Adenosine TriPhospate (ATP), which has three phosphate groups, to a protein substrate S. After the transfer ATP will become Adenosine DiPhosphate (ADP), and so we denote ATP as $A_2{-}P$, where the bond between A_2 and P is a, and ADP as A_2. Firstly, $A_2{-}P$ and then S bind to the active site of K. We denote the bonds thus created as b and c respectively; see Fig. 1, which should be read from left to right. Then phosphorylation takes place: P is transferred from $A_2{-}P$ to a Ser, Thr or Tyr residue of S by creating the bond d and then dissolving a. Finally A_2 and then S is released from the active site of K, so b and then c is broken. We note that the order in which bonds are created and broken differs for different kinases in such catalytic cycles [23]; hence we seek a general method for reversing events in an arbitrary order.

Let events a, b, c, d represent the bonds a, b, c, d. The order in which bonds are created can be defined by the *causality* relation $<$ of *prime event structures* (PES) [15,25]: $a < b < c < d$. To express undoing of events we shall add to PES a new *reverse causality* relation $\prec$: here $a \prec \underline{a}$, $b \prec \underline{b}$ and $c \prec \underline{c}$ mean that a, b, c can be reversed (notation $\underline{a}, \underline{b}, \underline{c}$) as long as they have happened, and $d \prec \underline{a}$, $d \prec \underline{b}$, $d \prec \underline{c}$ force undoing of a, b, c only after d. We do not include $d \prec \underline{d}$, since d is irreversible here. We force that a is undone before b is undone by extending PES further with a *prevention* relation $\triangleright$: $a \triangleright \underline{b}$ prevents undoing of b while a is present; similarly $b \triangleright \underline{c}$. Thus, we obtain a *reversible* PES (RPES). The resulting forward transitions between configurations are $(\emptyset \rightarrow)\{a\} \rightarrow \{a, b\} \rightarrow \{a, b, c\} \rightarrow \{a, b, c, d\}$ and reverse transitions are $\{a, b, c, d\} \rightarrow \{b, c, d\} \rightarrow \{c, d\} \rightarrow \{d\}$. This is an example of *inverse* causal reversibility: a is reversed before undoing b even though a causes b, similarly for b and c. See [20,21] for other examples of non-causal reversibility.

There is a deficiency in the RPES solution in that, for example, a can occur again (so to speak) in configurations $\{b, c, d\}, \{c, d\}, \{d\}$. A general remedy is to add forwards prevention $e \triangleright e'$ to the reverse prevention $e \triangleright \underline{e'}$ already present in RPESs to obtain *reversible* asymmetric event structures (RAES). These are a reversible version of the *asymmetric event structures* (AES) of Baldan, Corradini and Montanari [1]. Prevention $e \triangleright e'$ is *asymmetric* conflict, where both e and e' can happen, but only if e' occurs before e. This generalises the symmetric conflict relation $e \sharp e'$ of PESs. If we add $d \triangleright a$ (d prevents a from taking place) to our example then this disallows a in $\{b, c, d\}, \{c, d\}$ and $\{d\}$.

The second example illustrates another limitation of using PESs in the reversible setting. Consider a long running transaction with a compensation. The full solution is given in Example 4.31; we are only concerned here with the three particular stages of a transaction. The events start, error and comp denote the start of the transaction, an error taking place, which is followed by reversing the computation all the way to the beginning including the start step, and the compensation stage of the transaction respectively. Events start, error are reversible and comp is not. We have start $<$ error $<$ comp but we do not intend start $<$ comp since start and comp are in conflict. This makes sense in the reversible setting as the computation goes through these configurations: $\emptyset$, {start}, {start, error}, {error}, {error, comp} and {comp}.

There are two standard ways of explaining causation. Event a *causes* event b ($a < b$) means either (1) in any run (computation), if b occurs then a occurs earlier or (2) if b is enabled at configuration X then we must have $a \in X$. The two views are equivalent if there is no reversing. Suppose that we have three events with $a < b < c$. On view (1) we deduce that $a < c$. On view (2) we also deduce that $a < c$, provided that X is left-closed (downwards closed under $<$), which will be the case for forward-only computation. Thus causation is transitive, as is the case in PESs and AESs.

In the context of reversible computation the second view of causation is simpler, and that is the one that we adopt. If all reversing is causal then all configurations will still be left-closed, and so it is still natural to require $<$ to be transitive. However once we admit the possibility of non-causal reversing, which leads to non-left-closed configurations (such as $\{b, c, d\}$ and $\{c, d\}$ in our example), it is no longer reasonable to insist on $<$ being transitive; if $a < b < c$ then a may have been reversed after b occurs, and before c occurs. Therefore, when defining RAESs we allow causation to be non-transitive. This extension is somewhat orthogonal to the move from symmetric to asymmetric conflict. We introduce the concept of *sustained causation*, where $a \ll b$ means that a causes b and a cannot reverse until b reverses. This is the analogue of standard causation for forwards computation, and we take sustained causation to be transitive.

We also consider the issue of conflict inheritance (if $a < b$ and $a \sharp c$ then $b \sharp c$) in the reversible setting. If $a < b$ and $a \sharp c$ and a is reversible, then we can undo a in $\{a, b\}$ to reach $\{b\}$. Now there is nothing in $\{b\}$ to prevent c from taking place, and so we expect that $\{b, c\}$ is a configuration, and b and c are not in conflict. Hence, there is no conflict inheritance with $<$. However, we still have conflict inheritance with respect to sustained causality $a \ll b$.

We assign meaning to the structures we consider by defining *configuration systems*, which are transition systems with configurations as states and sets of concurrent events as labels. It is natural to allow *mixed* transitions, which perform both forward and reverse moves. We are not aware of models with mixed transitions having been considered previously.

Our main contributions are as follows. We present an account of conflict and causation in the reversible, and not necessarily causal, setting. We define RPESs and RAESs, and relate them to their respective forward-only counterparts. We prove a number of results about reachable configurations, and, more generally, secured configurations, which are limits of non-monotone sequences. We show under what conditions reachable configurations which are finite are reachable by purely