Local reversibility in a Calculus of Covalent Bonding

Stefan Kuhn, Irek Ulidowski

Department of Informatics, University of Leicester, Leicester, LE1 7RH, United Kingdom

Abstract

We introduce a process calculus with a new prefixing operator that allows us to model locally controlled reversibility. Actions can be undone spontaneously, as in other reversible process calculi, or as pairs of concerted actions, where performing a weak action forces undoing of another action. The new operator in its full generality allows us to model out-of-causal order computation, where causes are undone before their effects are undone, which goes beyond what typical reversible calculi can express. However, the core calculus, which uses only the reduced form of the new operator, is well behaved as it satisfied causal consistency. We demonstrate the usefulness of the calculus by modelling the hydration of formaldehyde in water into methanediol, an industrially important reaction, where the creation and breaking of some bonds are examples of locally controlled out-of-causal order computation.

Keywords: Reversible process calculi, local reversibility, modelling of biochemical reactions

1. Introduction

There are many different computation tasks which involve undoing of previously performed steps or actions. Consider a computation where the action a causes the action b, written $a < b$, and where the action c occurs independently of a and b. There are three executions of this computation that preserve *causality*, namely abc, acb and cab. We note that a always comes before b. There are several conceptually different ways of undoing these actions [38]. *Backtracking* is undoing in precisely the reverse order in which they happened. So, undo b undo c undo a is a backtrack of the execution acb. *Reversing* is a more general form of undoing: here actions can be undone in any order provided causality is preserved (meaning that causes cannot be undone before effects). For example, undo c undo b undo a is a reversal of acb for the events a, b and c above.

There are networks of reactions in biochemistry, however, where actions are undone seemingly *out-of-causal order*. The creation and breaking of molecular bonds between the proteins involved in the ERK signalling pathway is a good example of this phenomenon [29]. Let us assume for simplicity that the creation of molecular bonds is represented by actions a, b, c where, as above, $a < b$ and c is independent of a and b. In the ERK pathway, the molecular bonds are broken in the following order: undo a, undo b, undo c, which seems to undo the cause a before the effect b.

We introduced informally a novel and purely local in character mechanism for undoing of computation in short papers [16, 17]. Here, we build a process calculus around this mechanism and give it operational semantics. We then discuss various properties that hold in the calculus. Most importantly, we show that out-of-causal order computation can be modelled in the calculus. Hence, in general, the *causal consistency* property [7] does not hold. There are reachable states that can only be arrived at by a mixture of forward and reverse steps. However, we argue that causal consistency holds in a restricted version of our calculus, thus the full calculus is in effect a “conceptual” extension of a causally consistent reversible process calculus. The benefits of the calculus are shown by modelling hydration of formaldehyde in water. The molecules of formaldehyde and water are modelled as compositions of carbon, oxygen and hydrogen atoms. When composed in parallel, the molecules react and the reactions are represented by sequences of transitions of *concerted actions*. We are able to represent different forms of reversibility, including out-of-causal order reversibility, and computation can proceed in any directions without external control.

The novel features of our calculus are introduced via an example of a simple catalytic reaction. Consider two molecules A and B that are only able to bond if assisted by a catalyst C. Once A and B are bonded with the catalyst C, A bonds with B and, at the same time, the bond between A and C is broken. Finally, the bond between B and C is broken. This is illustrated below.

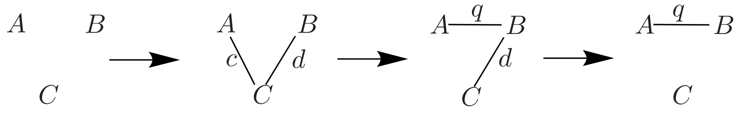

Figure 1: A catalytic reaction.

We assume $A \stackrel{def}{=} (a;p).A'$, $B \stackrel{def}{=} (b,p).B'$ and $C \stackrel{def}{=} (a,b).C'$, where A', B' and C' represent further potential behaviour of the molecules A, B and C. We use a new prefix operator $(s;p).P$ where s is a sequence of actions or executed actions and p is a *weak* action. Initially the actions in s take place, and then p takes place. The molecules A, B and C can bond by performing synchronously the matching actions according to the communication function $\gamma(a,a) = c$, $\gamma(b,b) = d$ and $\gamma(p,p) = q$, producing thus new actions c, d and q respectively. A weak action p can be left out in $(s;p)$ resulting in the simple prefix $(s).P$ (as in B and C above). In general, the actions of s in $(s;p).P$ can take place in any order, very much like in [8, 29], and the new feature is that p can happen only if all actions in s have already taken place. Once p takes place, one of the executed actions in s must be undone immediately: this is our new mechanism for triggering reverse computation. We shall model these two almost simultaneous events as a transition of concerted actions. This is a realistic representation of the mechanism of covalent bonding, the most common type of chemical bonding between atoms, hence we call our calculus a *Calculus of Covalent Bonding*.

Returning to our example, we represent the system of molecules A, B and C as $((a;p).A' \mid (b,p).B' \mid (a,b).C')\setminus\{a,b,p\}$, where '$\mid$' is the parallel composition and '$\setminus$' the restriction as in ACP [1]. We note that A and B cannot interact initially since $\gamma(a,b)$ is not defined. They can however both interact with C:

$$(a;p).A' \mid (b,p).B' \mid (a,b).C' \xrightarrow{c[1]} (a[1];p).A' \mid (b,p).B' \mid (a[1],b).C' \xrightarrow{d[2]}$$
$$(a[1];p).A' \mid (b[2],p).B' \mid (a[1],b[2]).C'$$

Numbers 1 and 2 are the *communication keys* [25, 26]: they indicate which pairs of actions have bonded. Molecules A and B can now bond on p (with the key 3), producing the action $q[3]$. This causes immediately the breaking of the bond $c[1]$, which means undoing of the action a in A and action a in C (and still leaving A and B bonded). We model such an event of creating a bond and simultaneously breaking another bond by a pair of *concerted actions*:

$$(a[1];p).A' \mid (b[2],p).B' \mid (a[1],b[2]).C' \xrightarrow{\{q[3],\underline{c}[1]\}}$$
$$(a;p[3]).A' \mid (b[2],p[3]).B' \mid (a,b[2]).C'$$

The bond with the key 3 on the weak action p in A is unstable, and thus gets *promoted* to a stable and stronger bond on a and p, which is modelled by the following rewrite:

$$(a;p[3]).A' \mid (b[2],p[3]).B' \mid (a,b[2]).C' \Rightarrow (a[3];p).A' \mid (b[2],p[3]).B' \mid (a,b[2]).C'$$

Finally, the catalyst dissolves the bond with B:

$$(a[3];p).A' \mid (b[2],p[3]).B' \mid (a,b[2]).C' \xrightarrow{\underline{d}[2]} (a[3];p).A' \mid (b,p[3]).B' \mid (a,b).C'$$

We note that A and B are now bonded although the synchronisation function did not allow it to happen initially. The main consequence of this is that the bond between $a[3]$ and $p[3]$ is *irreversible*, namely it cannot be undone. Looking at the pattern of doing and undoing of bonds we obtain $c[1]d[2]q[3]\underline{c}[1]\underline{d}[2]$. Since creation of bonds c and d causes the bond q, we have here an example of an out-of-causal order computation.

The calculus CCB is given *Structural Operational Semantics* (SOS for short) style semantics. This includes novel SOS rules for concerted actions and three rewrite rules that prescribe when bonds on weak actions can be promoted to strong action bonds. We show that CCB is a well behaved calculus by proving a number of useful properties. For example, the sub-calculus with the simple prefixing operator $(s).P$ satisfies causal consistency. We show that the full calculus allows us to represent out-of-causal order computation patterns via the hydration of formaldehyde in water case study.

Next we summarise the main items of related work.

1.1. Related Work

Scientists started to investigate the speed of chemical reactions and the rates achieved as soon as the concept of chemical reactions was first developed. The behaviour of a system of compounds over time can be modelled using a set of ordinary differential equations (ODEs). The fast calculations of such ODEs were popularised by Gillespie [13] in order to show the dynamic behaviour of systems of chemical compounds. When biochemical processes, which involve not only small molecules but also macromolecules, cells and membranes, were modelled attention turned to how the individual objects were represented and how they behaved, and the usefulness of computer science methods was demonstrated in [12].

Process calculi are very successful formalisms for representing concurrent and distributed systems. Each component of a system has its definition which specifies what it does and how it interacts with other components. The behaviour of the system emerges then from the independent actions of the components and from the interactions between them. Calculus of Communicating Systems (CCS) [21], Communicating Sequential Processes (CSP) [14], and the π-calculus [22] are examples of process calculi. Starting with Regev et al. [32, 35, 34] process calculi, specifically the π-calculus, were used to model biochemical systems. The biochemical compounds are represented as processes, and how they react is modelled by communication on ports. A creation of a bond or a dissolution of a bond is represented as establishing or breaking of a communication between ports. So there is a natural analogy between concurrent processes and biochemical entities in natural systems, and between communication among processes and reactions between biochemical entities. The aim of this work, as stated in [35], was to represent suitably biological knowledge in processes and to enable computer-based analysis of this representation. This approach was extended shortly afterwards to include reaction rates [31] using the stochastic π-calculus, which was introduced previously in [30].

Various other calculi, including Bio-PEPA ([3]), the biochemical abstract machine (BIOCHAM, [10]), P systems ([24]), BioAmbients ([33]), the kappa calculus ([9]) and Brane Calculus ([2]), followed aiming to capture various other aspects of biological systems such as, for example, compartments and membranes.

Most of biochemical reactions are reversible, and creating bonds is as important in biochemical processes as breaking of bonds. Hence, it became useful to be able to represent directly reversibility in process calculi. This realisation lead to the development of a number of reversible process calculi [7, 8, 25, 26, 19, 18, 20, 6], which have application far beyond biochemistry. Out-of-causal order reversibility, an important aspect of biochemical system which is typically not captured in traditional reversible process calculi, was first proposed in [29], where the calculus CCSK [26] is extended with an *execution control mechanism* for managing the pattern and the direction of computation. The control mechanism for reversibility is external to the processes it controls, and it can have a global scope. The calculus introduced in this paper has in contrast no global control and the behaviour of a biochemical systems emerges from the behaviour of its components. Out-of-causal order computation was also studied in [28, 27].

2. A Calculus of Covalent Bonding

In this section we define the calculus introduced informally in the Introduction. First, we introduce some preliminary notions and notations.

Let $\mathcal{A}$ be the set of (forward) action labels, ranged over by a, b, c, d, e, f. We partition $\mathcal{A}$ into the set of *strong actions*, written as $\mathcal{SA}$, and the set of *weak actions*, written as $\mathcal{WA}$. Reverse (or past) action labels are members of $\underline{\mathcal{A}}$, with typical members $\underline{a}, \underline{b}, \underline{c}, \underline{d}, \underline{e}, \underline{f}$, and represent undoing of actions. The set $\mathcal{P}(\mathcal{A} \cup \underline{\mathcal{A}})$ is ranged over by L.

Let $\mathcal{K}$ be an infinite set of *communication keys* (or *keys* for short) [25, 26], ranged over by k, l, m, n. The Cartesian product $\mathcal{A} \times \mathcal{K}$, denoted by $\mathcal{AK}$, represents past actions, which are written as $a[k]$ for $a \in \mathcal{A}$ and $k \in \mathcal{K}$. Correspondingly, we have the set $\underline{\mathcal{AK}}$ that represents undoing of past actions. We use α, β to identify actions which are either from $\mathcal{A}$ or $\mathcal{AK}$. It would be useful to consider sequences of actions or past actions, namely the elements of $(\mathcal{A} \cup \mathcal{AK})^*$, which are ranged over by s, s' and sequences of purely past actions, namely the elements of $\mathcal{AK}^*$, which are ranged over by t, t'. The empty sequence is denoted by ϵ. We use the notation α, s and s, s' to denote a concatenation of elements, which can be strings or single actions.

We shall also use two sets of auxiliary action labels, namely the set $(\mathcal{A}) = \{(a) \mid a \in \mathcal{A}\}$, and its product with the set of keys, namely $(\mathcal{A})\mathcal{K}$. These labels will be used in the auxiliary rules when defining the semantics of CCB.

We now define the Calculus of Covalent Bonding, or CCB for short. The syntax of CCB is given below where P is a process term:

$$P ::= S \ \mid \ (s; b).P \ \mid \ P \mid Q \ \mid \ P \backslash L$$

The set of process identifiers (constants) $\mathcal{PI}$ contains typical elements S and T. A process identifier S has normally a defining equation $S \overset{def}{=} P$ where P contains only forward actions (and no past actions). There is also a special identifier $\mathbf{0}$, denoting the deadlocked process, which has no defining equation.

We have a general prefixing operator $(s; b).P$, where s is a non-empty sequence of actions or past actions. This operator extends the prefixing operator in [29]. The action b is a weak action and it can be omitted, in which case the prefixing is written as $(s).P$ and is called the *simple prefix*. The simple prefix is the prefixing operator in [29]. One of the actions in s in $(s).P$ may be a weak action from $\mathcal{WA}$. If s is a sequence that contains a single action, then the action is a strong action and the operator is the prefixing operator of CCS [21]. We omit trailing $\mathbf{0}$s so, for example, $(s).\mathbf{0}$ is written as (s). The new feature of the operator $(s; b).P$ is the execution of the weak action b, which can happen only after all the actions in s have taken place. Performing b then forces undoing one of the past actions in s (by the concert rule in Figure 5).

$P \mid Q$ represents two systems P and Q which can perform actions or reverse actions on their own, or which can interact with each other according to a communication function γ. As in the calculus ACP [11], the communication function is a partial function $\gamma : \mathcal{A} \times \mathcal{A} \to \mathcal{A}$ which is commutative and associative. The function γ is used in the operational semantics to define when two processes can interact. Processes P and Q in $P \mid Q$ can also perform a pair of concerted actions, which is the new feature of our calculus. We also have the ACP-like restriction operator $\backslash L$, where L is a set of labels. It prevents actions from taking place and, due to the synchronisation algebra used, it also blocks communication. If $\gamma(a, b) = c$ then $a.P$ and $b.Q$ cannot communicate in $(a.P \mid b.Q) \setminus c$. Note that we do not use here the usual relabelling operator $[f]$, where $f : \mathcal{A} \to \mathcal{A}$, which could be easily added.

The example in the Introduction and our main example in Section 5 seem to indicate that only simple processes of the form $(s; b).\mathbf{0}$ are sufficient in the modelling of chemical reactions. However, there are examples where a nested prefix $(s; b).(s'; b').P$ is useful. Consider a base excision repair as in [15] where a protein "walks" along a strand of DNA and repairs faults which occurred in the DNA replication. The walking along a DNA strand could be modelled by actions in s, and, once a fault is found, the repair mechanism could be modelled by the actions in s'. Another example where the full calculus is useful is a model of long standing transactions with compensations in [29].

The set of *process terms* is ranged over by P, Q and R and is denoted by Proc. In the setting of CCB these terms are called simply *processes*. A context $C[\;]$ is a process term containing a *hole*, represented by $[\;]$. Formally, contexts are defined by the following syntax: $C ::= [\;] \mid (s; b).C \mid P \mid C \mid C \mid P \mid C\backslash L$. The term $C[Q]$ denotes the result of filling the hole in the context $C[\;]$ with the process Q. We say that R is a *subprocess* of P if P is $C[R]$ for some context $C[\;]$.

We define the semantics of our calculus by a labelled transition system, LTS for short, which is a structure $(St, AL, \to :\subseteq St \times AL \times St)$ with St the set of states, AL the set of action labels and $\to :\subseteq St \times AL \times St$ the labelled transition relation. The set of states St is the set Proc. In practice, all our results and examples hold for *consistent* processes, namely processes reachable from standard processes (see Definition 4). The action labels are the forward actions $\mathcal{AK}$, the reverse actions $\underline{\mathcal{A}}\mathcal{K}$ and the *pairs of concerted actions* $\mathcal{AK} \times \underline{\mathcal{A}}\mathcal{K}$. The labelled transition relation is defined by SOS rules (Figures 3–6) and rewrite rules (Figure 7), where the rules in Figures 3–4 are influenced by [26]. Note that sequences s and t are members of $(\mathcal{A} \cup \mathcal{AK})^*$ and $\mathcal{AK}^*$ respectively in Figures 3–5.

We now introduce and explain the SOS rules before returning to the rewrite rules. Let r be an SOS rule for an operator f of CCB as in Figures 3–5. Transitions above the horizontal bar in r are called *premises*. The set of premises is written as $pre(r)$. The transition below the bar in r is the *conclusion* and is written as $con(r)$. We use two predicates $\mathsf{std}(P) : \mathsf{Proc}$ and $\mathsf{fsh}[m](P) : \mathcal{K} \times \mathsf{Proc}$ in our SOS rules. They are defined in Figure 2, and they use two auxiliary functions $\mathsf{k}(s) : (\mathcal{A} \cup \mathcal{AK})^* \to \mathcal{P}(\mathcal{K})$ and $\mathsf{keys}(P) : \mathsf{Proc} \to \mathcal{P}(\mathcal{K})$. The function

$$\frac{}{\mathsf{std}(\mathbf{0})} \qquad\qquad \frac{}{\mathsf{fsh}[m](\mathbf{0})}$$

$$\frac{\mathsf{std}(P)}{\mathsf{std}(S)}\ S \overset{def}{=} P \qquad\qquad \frac{\mathsf{fsh}[m](P)}{\mathsf{fsh}[m](S)}\ S \overset{def}{=} P$$

$$\frac{\mathsf{k}(s) = \emptyset \quad \mathsf{std}(P)}{\mathsf{std}((s;b).P)} \qquad\qquad \frac{m \notin \mathsf{k}(s) \quad \mathsf{fsh}[m](P)}{\mathsf{fsh}[m]((s;b).P)}$$

$$\frac{\mathsf{std}(P) \quad \mathsf{std}(Q)}{\mathsf{std}(P \mid Q)} \qquad\qquad \frac{m \notin \mathsf{k}(s) \quad m \neq n \quad \mathsf{fsh}[m](P)}{\mathsf{fsh}[m]((s;b[n]).P)}$$

$$\frac{\mathsf{std}(P)}{\mathsf{std}(P \setminus L)} \qquad\qquad \frac{\mathsf{fsh}[m](P) \quad \mathsf{fsh}[m](Q)}{\mathsf{fsh}[m](P \mid Q)} \qquad \frac{\mathsf{fsh}[m](P)}{\mathsf{fsh}[m](P \setminus L)}$$

Figure 2: Predicates std and fsh.

$$\mathsf{act1}\ \frac{\mathsf{std}(P) \quad \mathsf{fsh}[k](s,s')}{(s,a,s';b).P \xrightarrow{a[k]} (s,a[k],s';b).P} \qquad\qquad \mathsf{act2}\ \frac{P \xrightarrow{a[k]} P' \quad \mathsf{fsh}[k](t)}{(t;b).P \xrightarrow{a[k]} (t;b).P'}$$

$$\mathsf{par}\ \frac{P \xrightarrow{a[k]} P' \quad \mathsf{fsh}[k](Q)}{P \mid Q \xrightarrow{a[k]} P' \mid Q} \qquad\qquad \mathsf{com}\ \frac{P \xrightarrow{a[k]} P' \quad Q \xrightarrow{d[k]} Q'}{P \mid Q \xrightarrow{c[k]} P' \mid Q'}\ (*)$$

$$\mathsf{res}\ \frac{P \xrightarrow{a[k]} P'}{P\setminus L \xrightarrow{a[k]} P'\setminus L}\ a \notin L \qquad\qquad \mathsf{con}\ \frac{P \xrightarrow{a[k]} P'}{S \xrightarrow{a[k]} P'}\ S \overset{def}{=} P$$

Figure 3: Forward SOS rules. The condition (*) is $\gamma(a,d) = c$, and $b \in \mathcal{WA}$.

$\mathsf{k}()$ is defined as follows: $\mathsf{k}(\epsilon) = \emptyset$, $\mathsf{k}(\alpha : s) = \{l\} \cup \mathsf{k}(s)$ if $\alpha = a[l]$, for $a \in \mathcal{A}$ and $l \in \mathcal{K}$, and $\mathsf{k}(\alpha : s) = \mathsf{k}(s)$ if $\alpha \in \mathcal{A}$. The function $\mathsf{keys}()$ is given by $\mathsf{keys}(\mathbf{0}) = \emptyset$, $\mathsf{keys}(S) = \mathsf{keys}(P)$ if $S \overset{def}{=} P$, $\mathsf{keys}((s;b).P) = \mathsf{k}(s) \cup \mathsf{k}(b) \cup \mathsf{keys}(P)$, $\mathsf{keys}(P \mid Q) = \mathsf{keys}(P) \cup \mathsf{keys}(Q)$, and $\mathsf{keys}(P\setminus L) = \mathsf{keys}(P)$. Informally $\mathsf{keys}(P)$ associates with each term P the set of its keys. A process P is standard, written $\mathsf{std}(P)$, if it contains no past actions (hence no keys). A key n is fresh in Q, written $\mathsf{fsh}[n](Q)$, if Q contains no past action with the key n. We extend the notion of fresh keys to the sequences of actions and past actions s and t via the function $\mathsf{k}()$.

Example 1. We illustrate how processes compute forwards using the new prefixing operator. Consider a standard process $(a;b).(c) \mid (a,d,c)$ and the communication function γ given by $\gamma(a,a) = a$ and $\gamma(c,c) = c$. We have

$$(a;b).(c) \mid (a,d,c) \xrightarrow{a[1]} (a[1];b).(c) \mid (a[1],d,c)$$

$$\textsf{rev act1}\ \frac{\mathsf{std}(P)}{(s, a[k], s'; b).P \xrightarrow{a[k]} (s, a, s'; b).P} \qquad \textsf{rev act2}\ \frac{P \xrightarrow{a[k]} P'}{(t; b).P \xrightarrow{a[k]} (t; b).P'}$$

$$\textsf{rev par}\ \frac{P \xrightarrow{a[k]} P' \quad \mathsf{fsh}[k](Q)}{P \mid Q \xrightarrow{a[k]} P' \mid Q} \qquad \textsf{rev com}\ \frac{P \xrightarrow{a[k]} P' \quad Q \xrightarrow{d[k]} Q'}{P \mid Q \xrightarrow{c[k]} P' \mid Q'}\ (*)$$

$$\textsf{rev res}\ \frac{P \xrightarrow{a[k]} P'}{P \backslash L \xrightarrow{a[k]} P' \backslash L}\ a \notin L \qquad \textsf{rev con}\ \frac{P \xrightarrow{a[k]} P'}{P \xrightarrow{a[k]} S}\ S \overset{def}{=} P'$$

Figure 4: Reverse SOS rules. The condition (*) is $\gamma(a, d) = c$, and and $b \in \mathcal{WA}$.

by the SOS rules act1 and com from Figure 3. This is because (c) is standard and the key 1 is fresh in ε. The next step of computation involves a communication of the actions c, which we obtain by rules act2 and com:

$$(a[1]; b).(c) \mid (a[1], d, c) \xrightarrow{c[2]} (a[1]; b).(c[2]) \mid (a[1], d, c[2])$$

We note that the key 2 is fresh in $a[1]$. Finally, the action d takes place by act1 and, informally, the symmetric version of par.

$$(a[1]; b).(c[2]) \mid (a[1], d, c[2]) \xrightarrow{d[3]} (a[1]; b).(c[2]) \mid (a[1], d[3], c[2])$$

Formally, we use par, the structural congruence rule sc in Figure 6 and the reduction rule red1 in Figure 7.

The next example illustrates how some of the reverse SOS rules work.

Example 2. Consider $(a[1], b).(c).S$ where $S \overset{def}{=} (a, b).(c).S$. We have

$$(a[1], b).(c).S \xrightarrow{a[1]} (a, b).(c).S$$

by rev act1 since $(c).S$ is standard. Since $(a, b).(c).S$ is the definition of S we obtain by rule rev con $(a[1], b).(c).S \xrightarrow{a[1]} S$.

Figure 5 contains the SOS rules that define the new concerted actions transitions. The main rule is the rule concert that defines when a pair of concerted actions takes place. We also have two auxiliary rules aux1 and aux2 which define only an auxiliary transition relation needed in the concert rule. Note that the concert rule uses *lookahead* [36]. Also note that transitions in aux1 and aux2 use the auxiliary labels $(b)[k]$ for all $b \in \mathcal{WA}$ and $k \in \mathcal{K}$. The rule concert par requires that k is fresh in Q, correspondingly as in par. Moreover, we need to ensure that when we reverse h with the key l in P we do not leave out any

$$\mathsf{aux1}\ \frac{\mathsf{std}(P) \quad \mathsf{fsh}[k](t)}{(t;b).P \xrightarrow{(b)[k]} (t;b[k]).P} \qquad \mathsf{aux2}\ \frac{P \xrightarrow{(b)[k]} P' \quad \mathsf{fsh}[k](t)}{(t;b').P \xrightarrow{(b)[k]} (t;b').P'}$$

$$\mathsf{concert}\ \frac{P \xrightarrow{(b)[k]} P' \quad P' \xrightarrow{\underline{a}[l]} P'' \qquad Q \xrightarrow{\alpha[k]} Q' \quad Q' \xrightarrow{\underline{d}[l]} Q''}{P \mid Q \xrightarrow{\{e[k],\underline{f}[l]\}} P'' \mid Q''}(*)$$

$$\mathsf{concert\ act}\ \frac{P \xrightarrow{\{a[k],\underline{h}[l]\}} P' \quad \mathsf{fsh}[k](t)}{(t;b).P \xrightarrow{\{a[k],\underline{h}[l]\}} (t;b).P'}$$

$$\mathsf{concert\ par}\ \frac{P \xrightarrow{\{a[k],\underline{h}[l]\}} P' \quad \mathsf{fsh}[k](Q) \quad \mathsf{fsh}[l](Q)}{P \mid Q \xrightarrow{\{a[k],\underline{h}[l]\}} P' \mid Q}$$

$$\mathsf{concert\ res}\ \frac{P \xrightarrow{\{a[k],\underline{h}[l]\}} P'}{P\backslash L \xrightarrow{\{a[k],\underline{h}[l]\}} P'\backslash L}(**)$$

Figure 5: SOS rules for concerted actions. The condition (*) is 1. α is c or (c) and $\gamma(b,c) = e$ for some $c \in \mathcal{A}$, and 2. $\gamma(a,d) = f$. The condition (**) is $a, \underline{h} \notin L \cup (L)$. Recall that $t \in \mathcal{AK}^*$.

$$\frac{P \Rrightarrow Q \quad Q \xrightarrow{\mu} Q' \quad Q' \Rrightarrow P'}{P \xrightarrow{\mu} P'}$$

Figure 6: Structural congruence rule sc when $\mu \in \mathcal{AK} \cup (\mathcal{AK} \times \underline{\mathcal{A}}\mathcal{K})$, and rev sc when $\mu \in \underline{\mathcal{A}}\mathcal{K}$.

actions with the key l in Q which make up a multiaction communication with the key l. Hence, we also include the premise $\mathsf{fsh}[l](Q)$ in concert par. The rule concert act requires, correspondingly as act, that k is fresh in t. Our operational semantics guarantees that if a standard process evolves to $(t;b).P$, for some P, and P reverses an action with the key l, then l is fresh in t. Hence, we do not include $\mathsf{fsh}[l](t)$ in the premises of concert act. Next, we illustrate how concerted actions transitions work.

Example 3. Consider the process $(a;b) \mid a \mid b$ with $\gamma(a,a) = c$ and $\gamma(b,b) = d$. After the initial synchronisation of actions a, which produces the transition $c[1]$, we have a transition with a pair of concerted actions by rule concert in Figure 5

$$(a[1];b) \mid a[1] \mid b \xrightarrow{\{d[2],\underline{c}[1]\}} (a;b[2]) \mid a \mid b[2]$$

since $(a[1];b) \xrightarrow{(b)[2]} (a[1];b[2])$ by aux1, $(a[1];b[2]) \xrightarrow{\underline{a}[1]} (a;b[2])$ by rev act1, and since $a[1] \mid b \xrightarrow{b[2]} a[1] \mid b[2] \xrightarrow{\underline{a}[1]} a \mid b[2]$ by par and rev par.

$$\begin{array}{lll}
\mathsf{red1}: & P \mid Q \Rightarrow Q \mid P & \\
\mathsf{red2}: & P \mid (Q \mid R) \Rightarrow (P \mid Q) \mid R & \\
\mathsf{red3}: & (P \mid Q) \mid R \Rightarrow P \mid (Q \mid R) & \\
\mathsf{red4}: & P \mid \mathbf{0} \Rightarrow P & \\
\mathsf{red5}: & (P \mid Q)\backslash L \Rightarrow P\backslash L \mid Q & \text{if } \mathrm{fn(Q)} \cap L = \emptyset \\
\mathsf{red6}: & P\backslash L \mid Q \Rightarrow (P \mid Q)\backslash L & \text{if } \mathrm{fn(Q)} \cap L = \emptyset \\
\mathsf{prom}: & (s, a, s'; b[k]).P \Rightarrow (s, a[k], s'; b).P & \text{if } a \in \mathcal{SA}, b \in \mathcal{WA} \\
\mathsf{move\text{-}r}: & (s, a, s', b[k], s'').P \Rightarrow (s, a[k], s', b, s'').P & \text{if } a \in \mathcal{SA}, b \in \mathcal{WA} \\
\mathsf{move\text{-}l}: & (s, b[k], s', a, s'').P \Rightarrow (s, b, s', a[k], s'').P & \text{if } a \in \mathcal{SA}, b \in \mathcal{WA}
\end{array}$$

Figure 7: Reduction rules. Sequences s, s', s'' are members of $(\mathcal{A} \cup \mathcal{AK})^*$.

Example 4. Consider $(a[1]; b) \mid (a[1]; b) \mid e$ with $\gamma(a, a) = c$ and $\gamma(b, b) = d$. We clearly have the following pair of concerted actions

$$(a[1]; b) \mid (a[1]; b) \mid e \xrightarrow{\{d[2], \underline{c}[1]\}} (a; b[2]) \mid (a; b[2]) \mid e.$$

There are processes with weak actions that can potentially communicate but there are no concerted actions transitions due to our SOS rules:

Example 5. Consider $(a[1]; b) \mid (e[2]; b) \mid (a[1], e[2])$ with $\gamma(a, a) = c$ and $\gamma(b, b) = d$. The process cannot perform any concerted actions: Although $(a[1]; b) \xrightarrow{(b)[l]}\xrightarrow{\underline{a}[1]} (a; b[l])$, for any l different from 1 and 2, but $(e[2]; b) \mid (a[1], e[2])$ cannot perform the auxiliary $(b[l])$ transition since there are no SOS rules for parallel composition and auxiliary actions (b). This forces us to treat $(a[1]; b)$ and $(e[2]; b)$ as P and Q in the concert rule, respectively, and we notice that we cannot undo a communication on a or e.

Overall, the transitions in Figures 3–5 are labelled with $a[k] \in \mathcal{AK}$, or with $\underline{c}[l] \in \underline{\mathcal{A}}\mathcal{K}$, or with concerted actions $(a[k], \underline{c}[l])$.

We also have the usual structural congruence rules sc and rev sc in Figure 6, which potentially combine reductions (defined below) with transitions.

Next, we introduce our reduction relation which is given by the reduction (rewrite) rules in Figure 7. The reduction relation is needed to define *promotion* of actions. First we define the function fn for *free names* of processes.

Definition 1. The function $\mathsf{fn} : \mathsf{Proc} \to \mathcal{P}(\mathcal{K})$ is given as follows: $\mathsf{fn}(\mathbf{0}) = \emptyset$, $\mathsf{fn}(S) = \mathsf{fn}(P)$ if $S \overset{def}{=} P$, $\mathsf{fn}((\alpha : s; b).P) = \{\alpha\} \cup \mathsf{fn}((s; b).P)$, $\mathsf{fn}((a; b).P) = \{a, b\} \cup \mathsf{fn}(P)$, $\mathsf{fn}(P \mid Q) = \mathsf{fn}(P) \cup \mathsf{fn}(Q)$, and $\mathsf{fn}(P\backslash L) = \mathsf{fn}(P)\backslash L$.

Our reduction rules have names such as, for example, red and we write red: $P \Rightarrow Q$ to indicate that the reduction rule $P \Rightarrow Q$ is called red. The process P in the rule $P \Rightarrow Q$ is called a *redex*, and the process Q is called a *contractum*. A reduction rule $P \Rightarrow Q$ can be seen as a prescription for deriving rewrites $C[P] \Rightarrow C[Q]$ for arbitrary context $C[\ ]$. A P redex may be replaced by its contractum Q in an arbitrary context $C[\ ]$ giving rise to a reduction step: $C[P] \Rightarrow C[Q]$.

Definition 2. The reduction relation $\Rightarrow$ is the smallest reflexive and transitive relation on CCB processes that is preserved by all contexts, and that satisfies the rules in Figure 7.

Note that we do not want $\Rightarrow$ to be symmetric as we wish to apply prom only from left to right.

The rewrite rules in Figure 7 include prom, move-r, and move-l which promote weak bonds (here b) to strong bonds (here a). The rule prom applies to the full version of our prefix operator (with the ; construct), and move-r and move-l apply only to the simple prefix. These three rules are here to model what happens in chemical systems: a bond on a weak action is temporary and as soon as there is a strong action that can accommodate that bond (as the result of concerted actions) the bond establishes itself on the strong action thus releasing the weak action. In order to align the use of these three rules to what happens in chemical reactions, we insist that they are used as soon as they becomes applicable: this is made precise in Definition 3. We could have used the idea of ordering on SOS rules and rewrite rules [37, 23] to specify that the rewrite rules prom, move-r and move-r are higher in the ordering than all SOS rules and the remaining rewrite rules, implying that they should be applied first when deriving transitions. Alternatively, we could have tried to employ some of the techniques presented in [5] to define our transition relation. This would require the use of negative information in the premisses, and the definitions in the style as those in [37, 23]. However, since we combine SOS rules with rewrite rules, we opted for a directly defined transition relation.

We now define the transition relation for the labelled transition system for CCB. Recall that the states of the LTS are processes in Proc and the labels are members of $\mathcal{A}$, $\mathcal{AK}$, $(\mathcal{A})\mathcal{K}$ and the concerted actions labels in $\mathcal{AK} \times \underline{\mathcal{A}}\mathcal{K}$. Let $d : \mathsf{Proc} \to \mathbb{N}$ be the operator depth function defined by $d(P) = 0$ if P is a constant, and $d(f(P_1, \ldots, P_n)) = 1 + max\{d(P_i) | 1 \leq i \leq n\}$ otherwise, where f is an operator of CCB. The transition relation is given as follows:

Definition 3. We associate to Proc and $\mathcal{AK} \cup \underline{\mathcal{A}}\mathcal{K} \cup (\mathcal{A})\mathcal{K} \cup (\mathcal{AK} \times \underline{\mathcal{A}}\mathcal{K})$ a transition relation $\rightarrow$ given by $\bigcup_{l<\omega} \rightarrow^l$, where transition relations $\rightarrow^l \subseteq \mathsf{Proc} \times \mathcal{AK} \cup \underline{\mathcal{A}}\mathcal{K} \cup (\mathcal{A})\mathcal{K} \cup (\mathcal{AK} \times \underline{\mathcal{A}}\mathcal{K}) \times \mathsf{Proc}$ are as follows, with $b \in \mathcal{AK}$ and $\mu \in \mathcal{AK} \cup \underline{\mathcal{A}}\mathcal{K} \cup (\mathcal{AK} \times \underline{\mathcal{A}}\mathcal{K})$:

1. $P \xrightarrow{(b)[k]} P' \in \rightarrow^l$ if $d(P) = l$, $P \xrightarrow{(b)[k]} P' = \rho(con(r))$, where r is either aux1 or aux2, and each premise in $pre(r)$ is a valid transition in $\bigcup_{k<l} \rightarrow^k$ or a valid predicate.

2. $P \xrightarrow{\mu} P' \in \rightarrow^l$ if $d(P) = l$, $P \Rightarrow Q$, for some Q such that Q does not contain any prom, $\mathsf{move\text{-}r}$ and $\mathsf{move\text{-}l}$ redex, $Q \xrightarrow{\mu} Q' = con(r)$, for some rule r where each member of $pre(r)$ is either a valid transition in $\bigcup_{k<l} \rightarrow^k$, a valid rewrite or a valid predicate, and $Q' \Rightarrow P'$.

The first part of the definition specifies the auxiliary transitions using rules aux1 and aux2. The second part tells us how to use the remaining rules to define transitions. If P has no prom, move-r and move-l redex, then we apply our rules in a standard way. Otherwise, we are required to reduce P to Q with prom, move-r and move-l first, then we define a transition of Q to Q' in a standard way, and finally we reduce Q' to P' (if needed). This implies that if P has a prom, move-r or move-l redex, then we must use one of the structural congruence rules in Figure 6. And, if we use any of these rules, then the reduced process Q must no longer have any prom, move-r and move-l redex.

The next example illustrates the application of the promotion rewrite rule.

Example 6. The transition $(a[1]; b) \mid a[1] \mid b \xrightarrow{\{d[2], \underline{c}[1]\}} (a; b[2]) \mid a \mid b[2]$ from Example 3 cannot be followed by a communication of actions a because there is a prom redex $(a; b[2])$ in $(a; b[2]) \mid a \mid b[2]$. The rewrite of this redex takes priority: the bond 2 moves from the weak b to the strong a by prom:

$$(a; b[2]) \mid a \mid b[2] \Rightarrow (a[2]; b) \mid a \mid b[2]$$

As a result, we can bond on the weak b again and, importantly, the $a[2]$ to $b[2]$ bond is irreversible as $\gamma(a, b)$ is undefined. Note that reaching this bond by computing forwards alone is not possible.

We shall call henceforth the transitions derived by the forward SOS rules as the *forward transitions* and, the the transitions derived by the reverse SOS rules as the *reverse transitions*. Correspondingly, there are the *concerted (action)* transitions.

3. Properties of CCB

In this section we establish several properties of the LTS for CCB. We start by showing the expected properties of keys, namely that when an action takes place it uses a fresh key, and when a past action is undone its key is removed from the resulting process. We also show that the reverse transitions invert the corresponding forward transitions, and vice versa.

Definition 4. A process P is *consistent* if $Q \rightarrow^* P$ for some process Q such that $\mathsf{std}(Q)$.

Proposition 1. *Let P be consistent. Then*
1. If $P \xrightarrow{a[k]} Q$ then $k \notin \mathsf{keys}(P)$ and $\mathsf{keys}(Q) = \mathsf{keys}(P) \cup \{k\}$ for all Q.
2. If $P \xrightarrow{\underline{a}[k]} Q$ then $k \in \mathsf{keys}(P)$ and $\mathsf{keys}(Q) = \mathsf{keys}(P) \setminus \{k\}$ for all Q.
3. If $P \xrightarrow{a[k]} P'$ then $P' \xrightarrow{\underline{a}[k]} P$. If $P' \xrightarrow{\underline{a}[k]} P$ and P' has no move-r *or* move-l *redexes, then $P \xrightarrow{a[k]} P'$.*

Proof. By induction on the depth of the inference tree of transitions $P \xrightarrow{a[k]} Q$ or $P \xrightarrow{\underline{a}[k]} Q$.

Next, we introduce some notation. We define a new transition relation $\longmapsto$ by $P \xmapsto{a[k]} Q$ if $P \xrightarrow{a[k]} Q$ or $P \xrightarrow{\underline{a}[k]} Q$. Process P is called the *source* and Q the *target* of $P \xmapsto{a[k]} Q$. We will use $t, t', t_1, \ldots$ to denote transitions, for example $t : P \xmapsto{a[k]} Q$. Two $\longmapsto$ transitions are *coinitial* if they have the same source, and they are *cofinal* if their targets are identical.

We define when two transitions are concurrent.

Definition 5. Two coinitial transitions $P \xmapsto{a[k]} P'$ and $P \xmapsto{b[l]} P''$ are *concurrent* if there exists $M \neq P$ such that $P' \xmapsto{b[l]} M$ and $P'' \xmapsto{a[k]} M$.

Note that two concurrent transitions are coinitial and, together with the two transitions (with the target M) required by Definition 5, they form a “diamond” structure with the nodes P, P', P'' and M.

When transitions in Definition 5 are forward, we may not be able to complete the diamond as the following example shows. In such case, we say that the transitions are in *conflict*. Consider $(a) \mid (b) \mid (b)$ with $\gamma(a, b) = c$. The two coinitial transitions below are in conflict:

$$\begin{array}{lcl} (a) \mid (b) \mid (b) & \xrightarrow{c[1]} & (a[1]) \mid (b[1]) \mid (b) \\ (a) \mid (b) \mid (b) & \xrightarrow{c[2]} & (a[2]) \mid (b) \mid (b[2]) \end{array}$$

However, coinitial reverse transitions are concurrent. We shall denote the syntactical equality of process expressions by $\equiv$.

Proposition 2 (Reverse Diamond). *Let P be a consistent process and let $t' : P \xrightarrow{\underline{a}[k]} P'$ and $t'' : P \xrightarrow{\underline{b}[l]} P''$ with $l \neq k$. Then t' and t'' are concurrent.*

Proof. We prove Proposition 2 by induction on the depth of the inference tree for transition of P.

1. Base case: Processes with an inference tree of depth 0 have no reverse transitions, so the proposition is valid.
2. Inductive hypothesis: We assume that for all subprocesses R of P and all $\underline{c}[m], \underline{d}[n]$, if R is a consistent process, $R \xrightarrow{\underline{c}[m]} R'$ and $R \xrightarrow{\underline{d}[n]} R''$, with $m \neq n$, then there is an N so that $R' \xrightarrow{\underline{d}[n]} N$ and $R'' \xrightarrow{\underline{c}[m]} N$.
3. Induction step: We consider cases depending on the structure of P:
   (a) $P \equiv (s; b).R$ with s containing two or more past actions: This includes the the case $P \equiv (t; b).R$. This is by rule $\mathsf{rev\ act1}$. With s' being the sequence obtained from s by removing $a[k]$ and $b[l]$ with $k, l \notin \mathsf{keys}(s')$ we have $(a[k], b[l], s'; c).R \xrightarrow{\underline{a}[k]} (a, b[l], ts; c).R \xrightarrow{\underline{b}[l]} (a, b, s'; c).R$ or $(a[k], b[l], s'; c).R \xrightarrow{\underline{b}[l]} (a[k], b, s'; c).R \xrightarrow{\underline{a}[k]} (a, b, s'; c).R$. Let $M \equiv (a, b, s'; c).U \mid T$ as required.

(b) $P \equiv (s; b).R$ with s containing one or none past action: We cannot deduce the required transitions $P \xrightarrow{\underline{a}[k]} P'$ and $P \xrightarrow{\underline{b}[l]} P''$ for any a, b, k, l and $l \neq k$ by any SOS rule. Hence the proposition is vacuously valid.

(c) $P \equiv Q \mid R$: There are three cases:

i. $P \xrightarrow{\underline{a}[k]} P'$ by rule rev par and $P \xrightarrow{\underline{b}[l]} P'$ by rule rev par. There are two subcases here:

A. Transitions in the same subprocess: Assume without loss of generality $Q \xrightarrow{\underline{a}[k]} Q'$ and $Q \xrightarrow{\underline{b}[l]} Q''$. By the inductive hypothesis there is an N so that $Q' \xrightarrow{\underline{b}[l]} N$ and $Q'' \xrightarrow{\underline{a}[k]} N$. We can conclude by using rule rev par that $Q' \mid R \xrightarrow{\underline{b}[l]} N \mid R$ and $Q'' \mid R \xrightarrow{\underline{a}[k]} N \mid R$. With $M \equiv N \mid R$ we get the result.

B. Transitions in different subprocesses: Assume without loss of generality that $Q \xrightarrow{\underline{a}[k]} Q'$ and $R \xrightarrow{\underline{b}[l]} R'$. By rule rev par $Q \mid R \xrightarrow{\underline{a}[k]} Q' \mid R \xrightarrow{\underline{b}[l]} Q' \mid R'$ and $Q \mid R \xrightarrow{\underline{b}[l]} Q \mid R' \xrightarrow{\underline{a}[k]} Q' \mid R'$ are valid. These form the required reversal diamond with $M \equiv Q' \mid R'$.

ii. $P \xrightarrow{\underline{a}[k]} P'$ by rule rev com and $P \xrightarrow{\underline{b}[l]} P'$ by rule rev par: Without loss of generality this covers all cases with one rev par and one rev com transition. We assume that $\underline{a}[k]$ is by rule rev com, that $\gamma(a_1, a_2) = a$ and that $\underline{b}[l]$ is by rev par. We also assume that b happens in Q, so that $Q \xrightarrow{\underline{b}[l]} Q'$ and $\mathsf{fsh}[l](R)$, and that $Q \xrightarrow{\underline{a_1}[k]} Q''$ and $R \xrightarrow{\underline{a_2}[k]} R'$. We know that $l \neq k$ because $\mathsf{fsh}[l](R)$ and $R \xrightarrow{\underline{a_2}[k]} R''$, a transition which could not happen if $l = k$, since according to Proposition 1.2 a key cannot be fresh for a reverse transition to happen with this key. By the inductive hypothesis there is an N so that $Q' \xrightarrow{\underline{a_1}[k]} N$ and $Q'' \xrightarrow{\underline{b}[l]} N$. Using the rev com rule we can deduce $P \xrightarrow{\underline{a}[k]} Q'' \mid R'$, $P \xrightarrow{\underline{b}[l]} Q' \mid R$, $Q'' \mid R' \xrightarrow{\underline{b}[l]} N \mid R'$ and $Q' \mid R \xrightarrow{\underline{a}[k]} N \mid R'$. Taking $M \equiv N \mid R'$ we get the result.

iii. $P \xrightarrow{\underline{a}[k]} P'$ by rev com and $P \xrightarrow{\underline{b}[l]} P''$ by rev com: Without loss of generality this covers all cases with two rev com transitions. We assume $\gamma(a_1, a_2) = a$ and $\gamma(b_1, b_2) = b$. Also $Q \xrightarrow{\underline{a_1}[k]} Q'$, $Q \xrightarrow{\underline{b_1}[l]} Q''$, $R \xrightarrow{\underline{a_2}[k]} R'$ and $R \xrightarrow{\underline{b_2}[l]} R''$. Since $Q \xrightarrow{\underline{a_1}[k]} Q'$ and $Q \xrightarrow{\underline{b_1}[l]} Q''$ by the inductive hypothesis it follows that there is an N so that $Q' \xrightarrow{\underline{b_1}[l]} N$ and $Q'' \xrightarrow{\underline{a_1}[k]} N$ and since $R \xrightarrow{\underline{a_2}[k]} R'$ and $R \xrightarrow{\underline{b_2}[l]} R''$ there is an N' so that $R' \xrightarrow{\underline{b_2}[l]} N'$ and $R'' \xrightarrow{\underline{a_2}[k]} N'$.

By rule rev par it follows that $P \xrightarrow{a[k]} Q' \mid R' \xrightarrow{b[l]} N \mid N'$ and $P \xrightarrow{b[l]} Q'' \mid R'' \xrightarrow{a[k]} N \mid N'$. Let $M \equiv N \mid N'$ as required.

(d) cases $P \equiv R \setminus L$ and $P \equiv S$ with $S \overset{def}{=} R$ follow in a standard way by using rules rev res and rev con in Figure 4, respectively, and the inductive hypothesis.

Before we show that coinitial forward transitions are concurrent if they result in cofinal computations, we introduce *traces*. A trace is a sequence of composable forward and reverse transitions over CCB. Traces are ranged over by $\sigma, \sigma', \sigma_1, \ldots$. Two transitions are composable if the target of the first transition is the source of the second transition. The composition of transitions and traces is denoted by ';'. The *source* of a trace is the source of the first transition of the trace, and the *target* of a trace is the target of the last transition in the trace. As with transitions, two traces are *coinitial* if they have the same source, and they are *cofinal* if their targets are identical. The syntactical equality of transitions is also denoted by $\equiv$.

Proposition 3 (Forward Diamond). *If P is a consistent process and $t_1 \equiv P \xrightarrow{a[k]} P'$, $t_2 \equiv P \xrightarrow{b[l]} P''$, with $l \neq k$, and $P' \rightarrow^* R$ and $P'' \rightarrow^* R$, for some R, then there is $M \not\equiv P$ such that $P' \xrightarrow{b[l]} M$, $P'' \xrightarrow{a[k]} M$ and $M \rightarrow^* R$.*

Proof. By induction on the depth of the inference tree for transition of P.

1. Base case: obvious.
2. Inductive hypothesis: We assume that Proposition 3 holds for all subprocesses R of P and all $c[m], d[n]$, namely if R is a consistent process, $t'_1 \equiv R \xrightarrow{c[m]} R'$ and $t'_2 \equiv R \xrightarrow{d[n]} R''$ with $m \neq n$, and $t'_1; \sigma'_1$ and $t'_2; \sigma'_2$, for some σ'_1 and σ'_2, are cofinal then there is an N so that $R' \xrightarrow{d[n]} N$, $R'' \xrightarrow{c[m]} N$ and $N \rightarrow^* N'$ is cofinal with σ'_1 and σ'_2.
3. Induction step: We assume $t_1 \equiv P \xrightarrow{a[k]} P'$ and $t_2 \equiv P \xrightarrow{b[l]} P''$, and consider cases depending on the structure of P:

   (a) $P \equiv (s; b).R$ with s containing two or more fresh actions: This happens by rule act1. s is of the structure a, b, s' with $k, l \notin \mathsf{keys}(s')$ so that $P \equiv (a, b, s'; b).R$. So we have $t_3 \equiv P' \xrightarrow{b[l]} (a[k], b[l], s'; c).R$ and $t_4 \equiv P'' \xrightarrow{a[k]} (a[k], b[l], s'; c).R$. With $M \equiv (a[k], b[l], s'; c).R$ this gives us the result.

   (b) $P \equiv Q \mid R$:

      i. $P \xrightarrow{a[k]} P'$ by rule par and $P \xrightarrow{b[l]} P'$ by rule par.

         A. Transitions in the same subprocess: Assume without loss of generality $Q \xrightarrow{a[k]} Q'$ and $Q \xrightarrow{b[l]} Q''$. By the inductive hypothesis there is an N so that $Q' \xrightarrow{b[l]} N$ and $Q'' \xrightarrow{a[k]} N$ and there is a T so that $Q' \rightarrow^* T$ and $Q'' \rightarrow^* T$ implying that $a_1[k]$ and $b[l]$ do not exclude the execution of each other in Q. We can conclude, by rule par, that $Q' \mid R \xrightarrow{b[l]} N \mid R$ and $Q'' \mid R \xrightarrow{a[k]} N \mid R$. Taking $M \equiv N \mid R$ gives the result.

B. Transitions in different subprocesses: Assume without loss of generality that $Q \xrightarrow{a[k]} Q'$ and $R \xrightarrow{b[l]} R'$. By rule par $Q \mid R \xrightarrow{a[k]} Q' \mid R \xrightarrow{b[l]} Q' \mid R'$ and $Q \mid R \xrightarrow{b[l]} Q \mid R' \xrightarrow{a[k]} Q' \mid R'$ are valid. These form the required forward diamond with $M \equiv Q' \mid R'$.

ii. $P \xrightarrow{a[k]} P'$ by rule com and $P \xrightarrow{b[l]} P'$ by rule par: Without loss of generality this covers all cases with one par and one com transition. We assume that $a[k]$ happens by rule com, that $\gamma(a_1, a_2) = a$ and that $b[l]$ happens by rule par. We also assume that b happens in Q, so that $Q \xrightarrow{b[l]} Q'$ and $\mathsf{fsh}[l](R)$, and that $Q \xrightarrow{a_1[k]} Q''$ and $R \xrightarrow{a_2[k]} R'$. Also there is a T so that $Q' \to^* T$ and $Q'' \to^* T$ implying that $a_1[k]$ and $b[l]$ do not exclude the execution of each other in Q. By the inductive hypothesis there is an N so that $Q' \xrightarrow{a_1[k]} N$ and $Q'' \xrightarrow{b[l]} N$. By com we deduce that $P \xrightarrow{a[k]} Q'' \mid R'$, $P \xrightarrow{b[l]} Q' \mid R$, $Q'' \mid R' \xrightarrow{b[l]} N \mid R'$ and $Q' \mid R \xrightarrow{a[k]} N \mid R'$. With $M \equiv N \mid R'$ this gives us the result.

iii. $P \xrightarrow{a[k]} P'$ by com and $P \xrightarrow{b[l]} P''$ by com: Without loss of generality this covers all cases with two com transitions. We assume that $\gamma(a_1, a_2) = a$ and $\gamma(b_1, b_2) = b$. Also $Q \xrightarrow{a_1[k]} Q'$, $Q \xrightarrow{b_1[l]} Q''$, $R \xrightarrow{a_2[k]} R'$ and $R \xrightarrow{b_2[l]} R''$. Since $Q \xrightarrow{a_1[k]} Q'$ and $Q \xrightarrow{b_1[l]} Q''$ by the inductive hypothesis it follows that there is an N so that $Q' \xrightarrow{b_1[l]} N$ and $Q'' \xrightarrow{a_1[k]} N$ and since $R \xrightarrow{a_2[k]} R'$ and $R \xrightarrow{b_2[l]} R''$ there must be an N' so that $R' \xrightarrow{b_2[l]} N'$ and $R'' \xrightarrow{a_2[k]} N'$. Also there is a T so that $Q' \to^* T$ and $Q'' \to^* T$ implying that $a_1[k]$ and $b_1[l]$ do not exclude the execution of each other in Q and there is a T' so that $R' \to^* T'$ and $R'' \to^* T'$ implying that $a_2[k]$ and $b_2[l]$ do not exclude the execution of each other in R. By par $P \xrightarrow{a[k]} Q' \mid R' \xrightarrow{b[l]} N \mid N'$ and $P \xrightarrow{b[l]} Q'' \mid R'' \xrightarrow{a[k]} N \mid N'$. We let $M \equiv N \mid N'$ as required.

(c) cases $P \equiv R \setminus L$ and $P \equiv S$ with $S \overset{def}{=} R$ follow a standard way.

The next subsection explores some properties of concerted transitions.

3.1. Concerted transitions

The properties of keys corresponding to those in parts 1 and 2 of Proposition 1 hold also for the concerted transitions in CCB.

Proposition 4. *Let P be consistent. If $P \xrightarrow{\{\mu[k],\underline{\nu}[l]\}} Q$ then $k \notin \mathsf{keys}(P)$, $l \in \mathsf{keys}(P)$ and $\mathsf{keys}(Q) = (\mathsf{keys}(P) \cup \{k\}) \setminus \{l\}$ for all Q.*

Proof. See Appendix.

The property corresponding to part 3 of Proposition 1, namely $P \xrightarrow{\{\mu[k],\underline{\nu}[l]\}} P'$ if and only if $P' \xrightarrow{\{\nu[l],\underline{\mu}[k]}\}P$ does not hold in general but only in certain circumstances, which we now describe. Consider $(a[k];b).Q \mid R$ and c, d, for any Q, R, such that $\gamma(a,c) = d = \gamma(b,c)$ with $R \xrightarrow{c[l]} R'$ and $R' \xrightarrow{\underline{c}[k]} R''$. We obtain, by $\mathsf{concert}$ and prom rules,

$$(a[k];b).Q \mid R \xrightarrow{\{d[l],\underline{d}[k]\}} (a;b[l]).Q \mid R'' \Rightarrow (a[l];b).Q \mid R''$$

Since $R'' \xrightarrow{c[k]} R' \xrightarrow{\underline{c}[l]} R$, we get, again by $\mathsf{concert}$ and prom rules

$$(a[l];b).Q \mid R'' \xrightarrow{\{d[k],\underline{d}[l]\}} (a;b[l]).Q \mid R \Rightarrow (a[k];b).Q \mid R$$

Assume that R is $(c, c[k]).R_1$ for some R_1. If we take S as $(a[l];b).Q \mid R''$, then the following result could be seen as corresponding to part 3 of Proposition 1:

Proposition 5. *Consider $(a[k];b).Q$ for any Q and c, d such that $\gamma(a,c) = d = \gamma(b,c)$. There exist R, S and l such that $(a[k];b).Q \mid R \xrightarrow{\{d[l],\underline{d}[k]\}} S$ if and only if $S \xrightarrow{\{d[k],\underline{d}[l]\}} (a[k];b).Q \mid R$.*

4. CCB without weak actions

In this section we discuss the main properties of the sub-calculus of CCB that uses no weak actions. Our prefix operator is thus $(s).P$, where s contains only strong actions. We call this calculus CCB$_s$. Its SOS rules are as for CCB except that the rules in Figure 5 do not apply as there are no weak actions. The congruence rules prom, $\mathsf{move\text{-}r}$ and $\mathsf{move\text{-}l}$ also do not apply since there are no weak actions. We shall use μ, ν to denote strong actions in this section.

We shall also consider the forward-only version of CCB$_s$ called CCB$_f$. The syntax of CCB$_f$ is

$$P ::= S \;\mid\; (s).P \;\mid\; P \mid Q \;\mid\; P \backslash L$$

and the SOS rules are given in Figure 8; we also have the reduction rules from Figure 7 (without prom, $\mathsf{move\text{-}r}$ and $\mathsf{move\text{-}l}$) which, together with rules in Figure 8, generate the transition relation $\rightarrow_f$ for CCB$_f$. Note that we do not record past actions (act_f rule); hence CCB$_f$ is similar to the core of ACP. We note that CCB$_s$ is different from CCSK [25, 26] as it uses multiset prefixing (very much like in [8, 29]) and ACP-like communication.

We show firstly that $\rightarrow$ for CCB$_s$ is essentially conservative over $\rightarrow_f$. A process of CCB$_s$ is converted to a CCB$_f$ process by "pruning" past actions:

$$act_f \; \frac{}{(s,a,s').P \xrightarrow{a}_f (s,s').P} \qquad par_f \; \frac{P \xrightarrow{a}_f P'}{P \mid Q \xrightarrow{a}_f P' \mid Q}$$

$$com_f \; \frac{P \xrightarrow{a}_f P' \quad Q \xrightarrow{b}_f Q'}{P \mid Q \xrightarrow{c}_f P' \mid Q'} (*) \qquad res_f \; \frac{P \xrightarrow{a}_f P'}{P\backslash L \xrightarrow{a}_f P'\backslash L}\; a \notin L$$

$$con_f \; \frac{P \xrightarrow{a}_f P'}{S \xrightarrow{a}_f P'}\; S \overset{def}{=} P \qquad sc \; \frac{P \Rightarrow^* Q \quad Q \xrightarrow{a}_f Q' \quad Q' \Rightarrow^* P'}{P \xrightarrow{a}_f P'}$$

Figure 8: SOS rules for CCB$_f$. We have $a, b, c \in \mathcal{SA}$ and (*) is $\gamma(a,b) = c$.

Definition 6. The pruning map $\pi : \mathsf{Proc}_{\mathrm{CCB}_s} \to \mathsf{Proc}_{\mathrm{CCB}_f}$ is defined as follows, where $t \in \mathcal{AK}^*$ and $s, s' \in \mathcal{A}^*$:

$$\pi(\mathbf{0}) = \mathbf{0} \qquad \pi((s,t,s').P) = (s,s').\pi(P) \qquad \pi((t).P) = \pi(P)$$
$$\pi(P \mid Q) = \pi(P) \mid \pi(Q) \qquad \pi(P \setminus L) = \pi(P) \setminus L \qquad \pi(S) = \pi(P) \text{ if } S \overset{def}{=} P$$

Theorem 1 (Conservation). *Let $P \in \mathsf{Proc}_{\mathrm{CCB}_s}$.*
1. If $P \xrightarrow{\mu[k]} Q$ then $\pi(P) \xrightarrow{\mu}_f \pi(Q)$.
2. If $\pi(P) \xrightarrow{\mu}_f Q$, then for any $k \in \mathcal{K}\backslash\mathsf{keys}(P)$ there is Q' such that $P \xrightarrow{\mu[k]} Q'$ and $\pi(Q') = Q$.

Proof. 1. We use induction on the depth of the inference tree of $P \xrightarrow{\mu[k]} Q$.

(a) Base case: obvious.

(b) Inductive hypothesis: We assume that for all subprocesses R of P and all $\nu[l]$, if R is a consistent process and if $R \xrightarrow{\nu[l]} R'$ for some R' then $\pi(R) \xrightarrow{\nu}_f \pi(R')$.

(c) Induction step: We consider cases depending on the structure of P. Assume that P is consistent and $P \xrightarrow{\mu[k]} P'$ in all cases.

i. $P \equiv (s; b).R$: There are two cases:

A. $P \xrightarrow{\mu[k]} P'$ by $\mathsf{act1}$ in Figure 3: Assume $s = \mu : s', t$ where $s' \in \mathcal{A}^*$ and $t \in \mathcal{AK}^*$. Then $(\mu : s', t; b).R \xrightarrow{\mu[k]} (\mu[k] : s', t; b).R$. We can apply π to the processes: $\pi((\mu : s', t; b).R) = (\mu : s').R = \pi(P)$ and $\pi((\mu[k] : s', t; b).R) = (s').R = \pi(P')$. The transition $(\mu : s').R \xrightarrow{\mu}_f (s').R$ is by $\mathsf{act1}$ and we get $\pi(P) \xrightarrow{\mu}_f \pi(P')$.

B. $P \xrightarrow{\mu[k]} P'$ by $\mathsf{act2}$ in Figure 3: We deduce that $P \equiv (t; b).R$, $\mathsf{fsh}[k](t)$ and $P' \equiv (t; b).R'$. Recall that t contains only past actions. The transition must be $(t; b).R \xrightarrow{\mu[k]} (t; b).R'$, and we get by act2 $R \xrightarrow{\mu[k]} R'$ and $\mathsf{fsh}[k](t)$. Applying π to the processes we get $\pi((t;b).R) = \pi(R)$ and $\pi((t;b).R') = \pi(R')$. Since $R \xrightarrow{\mu[k]} R'$, by the inductive hypothesis, we obtain $\pi(R) \xrightarrow{\mu}_f \pi(R')$. The last implies $\pi((t;b).R) \xrightarrow{\mu}_f \pi((t;b).R')$ which is $\pi(P) \xrightarrow{\mu} \pi(P')$.

ii. $P \equiv R \mid Q$: There are two cases:

A. This happens by par in Figure 3. Since $P \xrightarrow{\mu[k]} P'$, by rule par, $R \xrightarrow{\mu[k]} R'$ and $\mathsf{fsh}[k](Q)$ must hold. By the inductive hypothesis $\pi(R) \xrightarrow{\mu}_f \pi(R')$ is true. By rule par $\pi(R) \mid \pi(Q) \xrightarrow{\mu}_f \pi(R') \mid \pi(Q)$ holds, which means $\pi(R \mid Q) \xrightarrow{\mu}_f \pi(R' \mid Q)$ and $\pi(P) \xrightarrow{\mu}_S \pi(P')$ as required.

B. This is by com in Figure 3. The inductive hypothesis in this case is not only true about R, but also about Q, so that if Q is consistent and $Q \xrightarrow{\mu_1[k]} Q'$ then $\pi(Q) \xrightarrow{\mu_1}_f \pi(Q')$. Assume P is consistent, $P \xrightarrow{\mu[k]} P'$ and $\gamma(\mu_1, \mu_2) = \mu$. We can calculate $\pi(P) = \pi(R \mid Q) = \pi(R) \mid \pi(Q)$ and $\pi(P') = \pi(R' \mid Q') = \pi(R') \mid \pi(Q')$. Since $\pi(R) \xrightarrow{\mu}_f \pi(R')$ and $\pi(Q) \xrightarrow{\mu}_f \pi(Q')$ it follows that according to rule com $\pi(R) \mid \pi(Q) \xrightarrow{\mu}_f \pi(R') \mid \pi(Q')$ which means $\pi(P) \xrightarrow{\mu}_f \pi(P')$ as required.

iii. Cases for $P \equiv R \backslash L$ and $P \equiv S$ with $S \overset{def}{=} R$ are straightforward.

2. We prove Theorem 1.2 by using induction on the definition of $\pi(P)$, which means on the structure of P.

(a) Base case: obvious.

(b) Inductive hypothesis: We assume that for all subprocesses R of P and all ν, if $\pi(R) \xrightarrow{\nu}_f R'$, then for any $l \in \mathcal{K} \setminus \mathsf{keys}(R)$ there is R'' such that $R \xrightarrow{\nu[l]} R''$ and $\pi(R'') = R'$.

(c) Induction step: We show the result for $\pi(P)$ as well. we consider cases depending on the structure of P. Assume that P is consistent and $\pi(P) \xrightarrow{\mu}_f Q$ for some Q in all cases.

i. $P \equiv (s, s'; b)$ and $\pi((s, s'; b).R) = (s').\pi(R)$: Here rule act_f applies with s' being of the structure ν, s''. The transition is $(\nu, s'').R \xrightarrow{\nu}_f (s'').R)$. Then with $k \in \mathcal{K} \setminus \mathsf{keys}(P)$ we have $(\nu, s'').R \xrightarrow{\nu[k]} (\nu[k], s'').R$ and we let $Q' \equiv (\nu[k], s'').R$ with $\pi(Q') \equiv (s'').R$.

ii. $P \equiv (t;b).R$ and $\pi((t;b).R) = \pi(R)$: Here the transition is in R by $\pi(R) \xrightarrow{\nu}_f R'$. By the inductive hypothesis $R \xrightarrow{\mu[k]} R''$ and $\pi(R'') = R'$ for some R''. We deduce that $\pi(P) \xrightarrow{\nu}_f R'$ and $P \xrightarrow{\mu[k]} R''$. Let $Q' \equiv R''$ with the required properties.

iii. $P \equiv R \mid T$ and $\pi(P) = \pi(R \mid T)$: There are two cases:

A. The transition is by par_f. Assume $\pi(R \mid T) \xrightarrow{\mu}_f Q$ for some Q. So $\pi(R \mid T) \xrightarrow{\mu}_f Q$ implies without loss of generality $\pi(R) \xrightarrow{\mu}_f R'$ for some R' and $Q \equiv R' \mid \pi(T)$ since $\pi(P) = \pi(R \mid T) = \pi(R) \mid \pi(T)$. Since $\pi(R) \xrightarrow{\mu}_f R'$ by the inductive hypothesis there exists a U and a $k \in \mathcal{K} \setminus (\mathsf{keys}(R) \cup \mathsf{keys}(T))$ such that $R \xrightarrow{\mu[k]} U$ and $\pi(U) = R'$. Since $R \xrightarrow{\mu[k]} U$ and since $\mathsf{fsh}[k](T)$ we obtain by rule par $R \mid T \xrightarrow{\mu[k]} U \mid T$. Now we calculate $\pi(U \mid T) = \pi(U) \mid \pi(T) = R' \mid \pi(T) = Q$. Let $Q' \equiv U \mid T$ with the required properties.

B. The transition is by com_f. Since $\pi(P) \xrightarrow{\mu} P'$, by rule com_f, $R \xrightarrow{\nu_1} R'$, $T \xrightarrow{\nu_1} T'$ and $\gamma(\nu_1, \nu_2) = \mu$ for some R', T' must hold without loss of generality. The inductive hypothesis in this case is not only true about R, so there is an R'' such that $R \xrightarrow{\nu_1[k]} R''$ and $\pi(R'') = R'$ and a T'' such that $T \xrightarrow{\nu_2[k]} T''$ and $\pi(T'') = T'$. We can deduce that $R \mid T \xrightarrow{\mu[k]} R'' \mid T''$. Also $\pi(R'' \mid T'') = \pi(R'') \mid \pi(T'') = R' \mid T' = P'$. Let $Q' \equiv R'' \mid T''$. This has the required properties.

iv. Cases for $P \equiv R \backslash L$ and $P \equiv S$ with $S \overset{def}{=} R$ are straightforward.

We now consider the causal consistency property, first defined and discussed in [7, 19], for CCB$_s$. We define when a set of keys is a cause for another key:

Definition 7. The set of causes of a key k in P is defined as follows:

$$cau(\mathbf{0}, k) = \emptyset \qquad cau(P \backslash L, k) = cau(P, k)$$

$$cau((s).P, k) = \mathsf{k}(s) \cup cau(P, k) \text{ if } k \in \mathsf{keys}(P) \qquad cau((\mu[k] : s).P, k) = \emptyset$$

$$cau((s).P, k) = \emptyset \text{ if } k \notin \mathsf{keys}(P) \qquad cau(S) = cau(P) \text{ if } S \overset{def}{=} P$$

$$cau(P \mid Q, k) = cau(P, k) \cup cau(Q, k)$$

If one of two coinitial transitions is forward and the other reverse, either they are concurrent or the forward transition depends causally on the reverse one. The following result holds by induction on the depth of the inference tree for transition of P: see Appendix.

Proposition 6. *If $t_1 \equiv P \xrightarrow{\mu[k]} P'$ and $t_2 \equiv P \xrightarrow{\nu[l]} P''$, then either t_1 and t_2 are concurrent or $k \in cau(P'', l)$.*

Next we introduce further notation on *traces*. We denote the reverse transition corresponding to a forward transition t (and the forward transition corresponding to a reverse transition t) as $t^\bullet$. Similarly to denoting the reverse transitions by $^\bullet$, we denote the reverse trace of σ as $\sigma^\bullet$. The empty trace with a process P is written as ϵ_P, and when the process is the source or the target of the transition t, we write $\epsilon_{source(t)}$ respectively $\epsilon_{target(t)}$. Similarly as with forward and reverse transitions, we shall use *forward traces* (traces composed of forward transitions only) and *reverse traces* (traces composed of reverse transitions only).

We can now define causal equivalence between traces.

Definition 8. *Causally equivalent* traces are defined by the least equivalence relation $\asymp$ which is closed under composition and obeys the following rules, where t_1 is $P \overset{a[k]}{\longmapsto} Q$, t_2 is $P \overset{b[l]}{\longmapsto} R$, d_1 is $Q \overset{b[l]}{\longmapsto} S$ and d_2 is $R \overset{a[k]}{\longmapsto} S$:

$$t_1; d_1 \asymp t_2; d_2 \qquad t; t^\bullet \asymp \epsilon_{source(t)} \qquad t^\bullet; t \asymp \epsilon_{target(t)}$$

The first relation states that the concurrent transitions t_1 and t_2 are causally independent, hence they can happen in any order. The trace $t_1; d_1$ forms a diamond with $t_2; d_2$, so the traces are causally equivalent. The remaining relations state that doing a transition and its reverse version is the same as doing nothing.

The next two results are needed to prove causal consistency for CCB$_s$; they follow closely [7, 19]. The first states that any computation has a causally equivalent version in which we first compute in reverse for a while and then we only compute forwards. The second result says that a trace which has a forward-only coinitial and cofinal and causally equivalent trace can always be shortened to a forward-only trace. The proofs are provided in Appendix.

Proposition 7 (Rearrangement). *If σ is a trace then there exist forward traces σ_1 and σ_2 such that $\sigma \asymp \sigma_1^\bullet; \sigma_2$.*

Proposition 8 (Shortening). *If σ_1 and σ_2 are coinitial and cofinal traces, with σ_2 a forward trace, then there exists a forward trace σ_1' of length at most that of σ_1 such that $\sigma_1' \asymp \sigma_2$.*

Next we have the second important result for CCB$_s$.

Theorem 2 (Causal consistency). *Let σ_1 and σ_2 be traces. Then $\sigma_1 \asymp \sigma_2$ if and only if σ_1 and σ_2 are coinitial and cofinal.*

Proof. The statement, if $\sigma_1 \asymp \sigma_2$ then σ_1 and σ_2 are coinitial and cofinal, follows directly from the construction of $\asymp$ in Definition 8.

Next we show that if σ_1 and σ_2 are coinitial and cofinal then $\sigma_1 \asymp \sigma_2$. We use induction on the sum of the lengths of σ_1 and σ_2 and on the distance between the end of σ_1 and the earliest pair of transitions t_1 in σ_1 and t_2 in σ_2 which are not equal. The coinitial and cofinal traces σ_1 and σ_2 are rewritten as compositions of reverse and forward traces by Proposition 7. There are three cases:

1. t_1 is forward and t_2 is reverse: We can assume that $\sigma_1 = \sigma^\bullet; t_1; \sigma'$ and $\sigma_2 = \sigma^\bullet; t_2^\bullet; \sigma''$ by Proposition 7. Since t_1 is forward $t_1; \sigma'$ must be forward, whereas $t_2^\bullet; \sigma''$ must start with at least one reverse transition. Since σ_1 and σ_2 are coinitial and cofinal $t_1; \sigma'$ and $t_2^\bullet; \sigma''$ are also coinitial and cofinal. We notice that the assumption of Proposition 8 is valid for $t_1; \sigma'$ and $t_2^\bullet; \sigma''$, hence there is a forward trace σ''' shorter than $t_2^\bullet; \sigma''$ such that $\sigma''' \asymp t_2^\bullet; \sigma''$. So the trace $\sigma^\bullet; \sigma'''$ is shorter than $\sigma_2 = \sigma^\bullet; t_2^\bullet; \sigma''$, and the result follows by induction.

2. t_1 and t_2 are forward: Assume $t_1 \equiv R \xrightarrow{\mu[k]} R'$ and $t_2 \equiv R \xrightarrow{\nu[l]} R''$ for some R, R', R''. Transitions t_1 and t_2 must be be concurrent according to Proposition 3. Since σ_1 and σ_2 are cofinal there must be transitions $t'_1 \equiv P \xrightarrow{\nu[l]} P'$ and $t'_2 \equiv Q \xrightarrow{\mu[k]} Q'$ for some P, P', Q, Q' at some later stage in σ_1 and σ_2. We look at the case of $t'_1 \equiv P \xrightarrow{\nu[l]} P'$ in σ_1 here, the other case follows in the corresponding way. All transitions from t_1 to t'_1 are concurrent since we can do $\mu[k]$ either as the first transition $R \xrightarrow{\mu[k]} R'$ of σ_1 or as the last transition in the sub-trace $t_2; t^*; t'_2$. Hence we can rearrange the transitions by the technique used in the proof of Proposition 7 so that $R \xrightarrow{\nu[l]} R'''$ comes first. This transition is identical to t_2. So we have moved the first non-matching pair of transitions in σ_1 and σ_2 to the right, and the result follows by induction.
3. $t_1^\bullet$ and $t_2^\bullet$ are reverse: Transitions $t_1^\bullet \equiv R \xrightarrow{\underline{\mu[k]}} R'$ and $t_2^\bullet \equiv R \xrightarrow{\underline{\nu[l]}} R''$, for some R, R' and R'', are undoing different actions. Since σ_1 and σ_2 are cofinal there is either a transition $t_1'^\bullet \equiv P'' \xrightarrow{\underline{\mu[k]}} P'''$, for some P'' and P''' in σ_2, or a transition $t'_1 \equiv P \xrightarrow{\mu[k]} P'$, for some P and P' in σ_1, redoing the action undone in $t_1^\bullet$. For $t_2^\bullet$ there is either a transition $t_2'^\bullet \equiv S'' \xrightarrow{\underline{\mu[k]}} S'''$, for some S'' and S''' in σ_1, or a transition $t'_2 \equiv S \xrightarrow{\mu[k]} S'$, for some S and S' in σ_2, redoing the action undone in $t_1^\bullet$. We treat these cases separately:
   (a) $t_1'^\bullet \equiv P'' \xrightarrow{\underline{\mu[k]}} P'''$ in σ_2 or $t_2'^\bullet \equiv S'' \xrightarrow{\underline{\nu[l]}} S'''$ in σ_1: We consider $t_1'^\bullet \equiv P'' \xrightarrow{\underline{\mu[k]}} P'''$ in σ_2, the other case follows correspondingly. The traces $t_1^\bullet$ and $t_2^\bullet$ are concurrent according to Proposition 2. All transitions from $t_2^\bullet$ to $t_1'^\bullet$ must be concurrent since from R we can do $\mu[k]$ either as the first transition as in σ_1 or as the last one in this sub-trace. So we can rearrange the transitions by a similar technique of "swaps" as used in the proof of Proposition 7 so that $R \xrightarrow{\underline{\mu[k]}} R'''$ comes first. This transition is identical to $t_1^\bullet$, and we have moved the first non-matching pair of transitions in σ_1 and σ_2 to the right, hence the case follows by induction.
   (b) $t'_1 \equiv P \xrightarrow{\mu[k]} P'$ in σ_1 and $t'_2 \equiv S \xrightarrow{\nu[l]} S'$ in σ_2 and $P'' \xrightarrow{\underline{\mu[k]}} P'''$ not in σ_2 and $S'' \xrightarrow{\underline{\nu[l]}} S'''$ not in σ_1: We consider $t'_1 \equiv P \xrightarrow{\mu[k]} P'$ in σ_1 only, the case $t'_2 \equiv S \xrightarrow{\nu[l]} S'$ in σ_2 follows correspondingly. We show that the transitions from $t_1^\bullet$ up to t'_1 are concurrent. In fact, we show that $t_1^\bullet$ and t'', the next transition on the way to t'_1, are concurrent. If they are concurrent we can swap them thus moving $t_1^\bullet$ to the right towards t'_1. If we follow this approach we will eventually have $t_1^\bullet$ immediately to the left of t'_1, and we can remove $t_1^\bullet; t'_1$ according to Definition 8. What remains to be done is to prove that $t_1^\bullet$ and t'' are concurrent. There are two cases: If t'' is forward then we have a situation as in the proof of Proposition 8, where we show that the transitions

are concurrent. The second case is for t'' being reverse. Assume $t'' \equiv R' \xrightarrow{\underline{\alpha[n]}} R''$. Hence by Proposition 1.3 we have $R' \xrightarrow{\mu[k]} R$. So we have $R' \xrightarrow{\underline{\alpha[n]}} R''$ and $R' \xrightarrow{\mu[k]} R$. This matches the assumptions of Proposition 6 (where R' is P). If we show that $n \notin cau(R, k)$ then they are concurrent. The transitions $R \xrightarrow{\underline{\mu[k]}} R' \xrightarrow{\underline{\alpha[n]}} R''$ show that $\mu[k]$ is undone before $\alpha[n]$, so n cannot be one of the causes of k in R. Hence the transitions $t_1^\bullet$ and t'' are concurrent, and the result follows by induction.

One of the consequences of causal consistency for CCB$_s$ concerns reachability: any state that can be reached from a standard process during an arbitrary computation can be reached by computing forwards alone. This property is not valid in the full calculus CCB as can be seen in the Introduction and in Example 6.

5. The hydration of formaldehyde in water

In this section we describe in detail the hydration of formaldehyde in water [1] Formaldehyde is a good preservative and is well known for its use in preserving specimen samples. It also serves as an important building block in industrial processes and is therefore produced in large quantities. At room temperature, formaldehyde forms a colourless and smelly gas. Formaldehyde reacts with water molecules to form methanediol. The reaction is shown in Figure 9. It is reversible but it is much faster towards the methanediol, so the equilibrium contains mostly methanediol. It should be noted that other reactions are taking place in a solution of formaldehyde in water, however we consider only the interaction of formaldehyde with water molecules. The carbon atom of formaldehyde is not shown in Figure 9 in line with a common convention. It resides at the point where the lines from the oxygen and the hydrogens meet; we follow this convention in all other reaction diagrams. We use single lines to represent single bonds in all our reactions, and we use double lines for double bonds, where two electron pairs are shared between atoms.

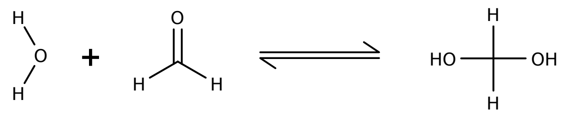

Figure 9: Hydration of formaldehyde in water into methanediol

[1]Detailed description of the reaction is given in [4]. The main path is on page 143, the base-catalysed reaction is on page 713, and the acid-catalysed reaction in on page 343 of [4].

5.1. The most common mechanism of the reaction

A closer look at the reaction in Figure 9 shows that the most common sequence of intermediate reactions leading to methanediol is the one given in Figure 10. We now describe carefully its steps. The oxygen atom in one of the water molecules attacks the central carbon in the formaldehyde to form the intermediate 2 in Figure 10. Then one of the hydrogens of the positively charged oxygen is taken away by another water molecule in a proton transfer. This gives the intermediate 3. Finally, a proton is donated to the other oxygen in the intermediate 3 to give the final configuration 4.

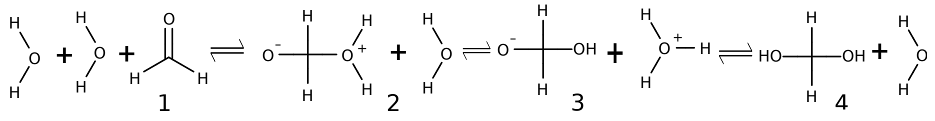

Figure 10: The most common path through the hydration of formaldehyde

In order to model this reaction we need to understand what it is that makes it happen. The main factor is that the oxygen in the water is nucleophilic: it tends to connect to another atomic nucleus. In the formaldehyde the oxygen attracts electrons towards itself, so the carbon has a positive charge. Now the oxygen in the water is attracted by this carbon and, due to its nucleophilicity, it forms a bond to the carbon. This bond is formed out of the electrons of one of the electrons in the oxygen so far not involved in bonding, so-called lone electron pairs. Since the carbon cannot have more than four bonds this reaction is compensated by the double bond in the formaldehyde becoming a single bond and the electrons from the double bond forming a lone pair on the oxygen, which now has three lone pairs. These movements are *concerted*, namely they happen together without a stable intermediate state and cannot be separated. So we have a continuous change between two tetracoordinate species through a pentacoordinate intermediate. We have now reached the intermediate 2 in Figure 10.

Looking at the intermediate 2, we can see that this has one oxygen which is negatively charged, because it has three lone pairs and a surplus of one electron. The other oxygen is positively charged, since it has three bonds and only one lone pair, therefore is missing an electron. The intermediate 2 reacts then with a water molecule. The water molecule is nucleophilic and has lone pairs for a bonding. It therefore can abstract one of the hydrogens on the positively charged oxygen. This leads to the intermediate 3 and a H_3O molecule, a water with an additional hydrogen and a positively charged oxygen.

Finally, a hydrogen can be re-donated to the negatively charged oxygen. Note that the re-donated hydrogen may not be the one which was originally attached to the oxygen. This transfer is possible since the oxygen in the H_30 is electron-deficient and the negative oxygen is rich in electrons. We then get the substrate 4 which contains the final product methanediol, and water.

5.2. *Other paths through the reaction*

There are other paths through the reaction of formaldehyde and water. We assume that we now have a mixture of three water molecules and one formaldehyde molecule. This shall allow us to explore all other water-formaldehyde interactions. Two water molecules can interact to form H_3O and HO. This is known as *autoprotolysis of water* and is described in detail in [16]. These two molecules can be considered an acid and a base respectively[2]. Both bases and acids can interact with formaldehyde and thus produce methanediol by different mechanisms than those in the previous section.

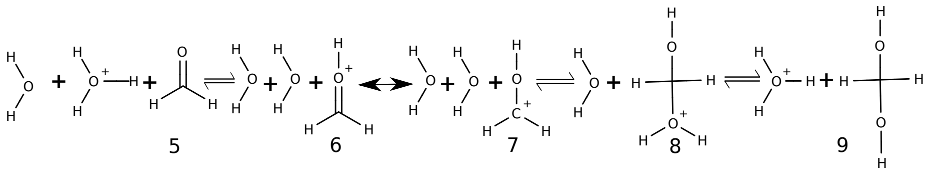

Figure 11: Acid-catalysed hydration of formaldehyde in water.

The acid-catalysed reaction is described in Figure 11. Here the H_3O molecule, which was formed during the autoprotolysis serves as the acid as it easily donates a proton. This proton can be donated to the oxygen in the formaldehyde, since this is slightly negatively charged, as we have seen. We then get the intermediate 6 in Figure 11. The charge on the oxygen can “move” to the carbon by the electrons forming the bonds a lone pair on the oxygen. The real charge distribution is somewhere in between. We shall assume that the intermediates 6 and 7 are somewhat different structures, and we shall model the transition from 6 to 8 as a pair of concerted actions. Once the intermediate 8 is reached one of the protons from the oxygen in the intermediate 8 can be abstracted by one of the water molecules, giving H_3O and making it a catalytic process.

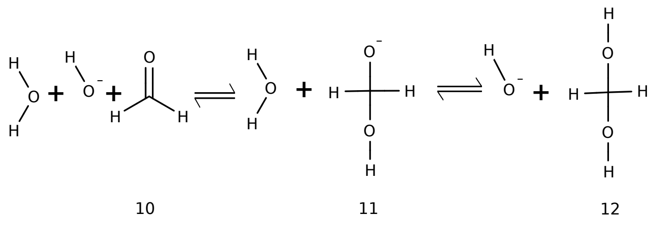

Figure 12: Base-catalysed hydration of formaldehyde in water.

The base-catalysed reaction in Figure 12 starts with a water and a HO^- molecule, the base, which tends to accept protons very strongly. It can therefore

[2]There are slightly different definitions of acids and bases in the literature, we use the general Bronsted-Lowry acid-base theory, which defines a base as a proton acceptor and an acid as a proton donor.

interact with the carbon in the formaldehyde, which is similar to the water attacking the carbon. One of the bonds of the carbon-oxygen double bond is broken and the oxygen becomes negatively charged. This gives the intermediate 11 which is in fact the same as the intermediate 3 in Figure 10. Then one of the protons of the water can be abstracted, and we obtain the methanediol and an HO molecule. Although this is the same structure as the original acid, it is made up from different atoms. The process is considered catalytic since the HO molecule is retrieved at the end of the process.

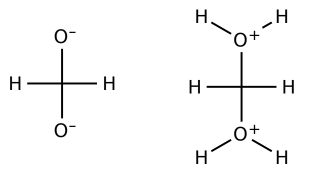

Figure 13: Other compounds possible in the reaction of formaldehyde with water.

There are other ways that several molecules of water can react with formaldehyde to produce methanediol and other byproducts. For example the base, HO, could directly interact with intermediate 7 to form methanediol and two water molecules. Also, the two compounds given in Figure 13 can be created as intermediate products in the reaction.

All possible reactions of a system of one formaldehyde and three water molecules are shown in Figure 14. Although the direction of reactions are denoted with arrows, all reactions are reversible; the arrows merely indicate the direction to the final product. We write FA for formaldehyde, W for water and MD for methanediol. Many of the intermediate compounds are denoted by iX where $X \in \{2, 3, 6, 7, 8, 13, 14\}$. Figure 14 includes all intermediates from Figure 11 and Figure 12. The compounds in Figure 13 have only one path leading to them, and we could only move away from them by reverting these actions. This is because these compounds are the "extreme" states where there is either no hydrogen on an oxygen or both oxygens are fully saturated with two hydrogens. The resonance between the intermediates i6 and i7 in Figure 11 is represented by a dashed arrow.

6. A CCB model of the hydration of formaldehyde in water

We are now ready to model our reaction in CCB. Figure 15 shows in more detail the three main paths through our reaction that we described in the last section; these paths have been highlighted in bold in Figure 14.

The initial state of the reaction is modelled by the composition of one formaldehyde FA three waters, written as FA | W | W | W. The final products are methanediol MD and two waters, written as MD | W | W. The path via the intermediate molecules 2 and 3 in Figure 10 is via the nodes denoted by i2 and

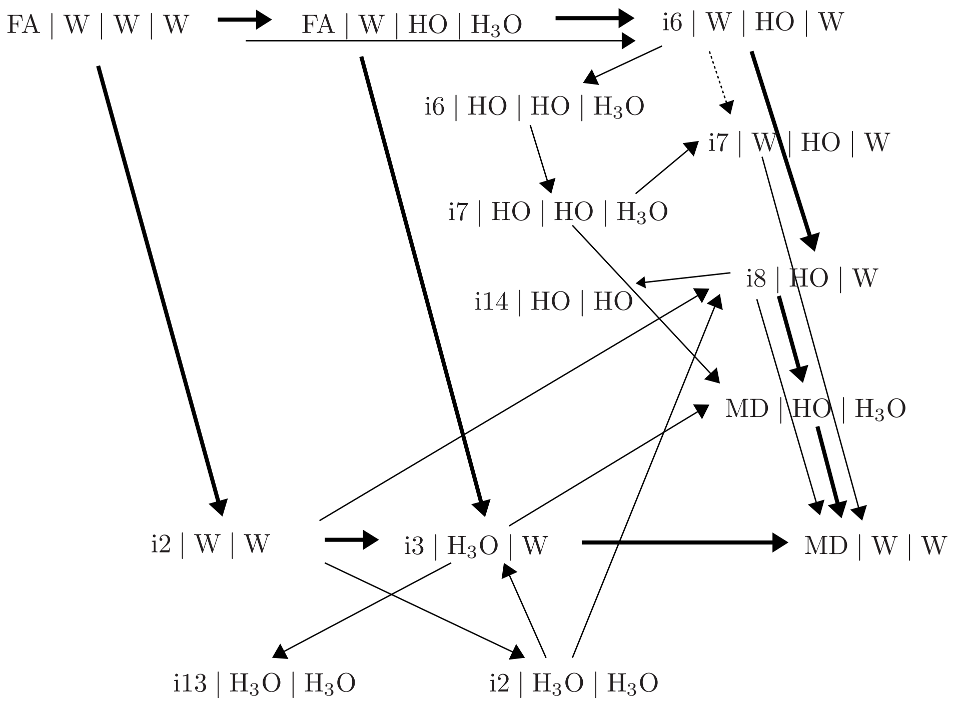

Figure 14: All possible reactions in a system of formaldehyde and three water molecules. The reactions displayed in more detail in Figure 15 are denoted here by bold arrows.

i3 in Figure 15. The FA | W | HO | H_3O node is the result of an autoprotolysis. The base-catalysed and acid-catalysed reactions are put into the same diagram, again the intermediates i6 and i8 correspond to molecules in Figure 11. As we can see in Figure 15 the reactions are driven completely by concerted actions.

We model the formaldehyde molecule CH_2O and the three water molecules H_2O as appropriate compositions of their atoms, namely hydrogen, oxygen and carbon. We use our general prefixing operator to define these atoms:

$$\begin{aligned} H &\overset{def}{=} (h;p).H' \\ O &\overset{def}{=} (o,o,n).O' \\ C &\overset{def}{=} (c,c,c,c;p).C' \end{aligned}$$

Carbon has four strong actions c, representing the potential for four covalent bonds, and a weak action p, standing for a positive partial charge. The oxygen can have up to three bonds. Normally is has two bonds, however, if a suitable reaction partner is close, an additional weak bond is available. We use therefore the simple prefixing operator to model it, with one weak action n standing for the potential for a negative partial charge. A partial charge means that a part or parts of a molecule have an electric charge, even though the molecule as a whole

$$\begin{tikzcd}
\mathrm{FA} \mid \mathrm{W} \mid \mathrm{W} \mid \mathrm{W} \arrow[r, "\{np[12]{,}\underline{h_3o_3}[5]\}"] \arrow[ddd, "\{np[11]{,}\underline{c_4o_2}[4]\}"] & \mathrm{FA} \mid \mathrm{W} \mid \mathrm{HO} \mid \mathrm{H_3O} \arrow[r, "\{np[13]{,}\underline{h_5o_5}[7]\}"] \arrow[ddd, "\{np[11]{,}\underline{c_4o_2}[4]\}"] & \mathrm{i6} \mid \mathrm{W} \mid \mathrm{HO} \mid \mathrm{W} \arrow[d, "\{np[11]{,}\underline{c_4o_2}[4]\}"'] \\
 & & \mathrm{i8} \mid \mathrm{HO} \mid \mathrm{W} \arrow[d, "\{np[14]{,}\underline{h_6o_6}[8]\}"'] \\
 & & \mathrm{MD} \mid \mathrm{HO} \mid \mathrm{H_3O} \arrow[d, "\{np[5]{,}\underline{h_8o_8}[14]\}"'] \\
\mathrm{i2} \mid \mathrm{W} \mid \mathrm{W} \arrow[r, "\{np[12]{,}\underline{h_3o_3}[5]\}"] & \mathrm{i3} \mid \mathrm{H_3O} \mid \mathrm{W} \arrow[r, "\{np[13]{,}\underline{h_5o_5}[12]\}"] & \mathrm{MD} \mid \mathrm{W} \mid \mathrm{W}
\end{tikzcd}$$

Figure 15: The three main reaction paths in the hydration of formaldehyde.

is neutrally charged. These uneven charge distributions enable many reactions by allowing another molecule to attack a partially charged part. The hydrogen has one strong bond, and we use additionally a weak action p to represent that it can become positively charged. Processes H', O', and C' represent unspecified further behaviour of the respective atoms.

Since our reaction involves multiple copies of H and O we shall adopt a subscript notation to denote distinct copies of the same process. Both action labels and process names will be subscripted. For example, $H_1 \overset{def}{=} (h_1; p).H_1'$ and $H_2 \overset{def}{=} (h_2; p).H_2'$ are two copies of hydrogen H. We shall abstract away these subscripts in Section 6.5, where we introduce informally chemical equivalence. Also, the copies of actions c of carbon will also be subscripted.

The synchronisation function γ is defined on subscripted actions as follows:

$$\begin{array}{lcll}
\gamma(c_i, h_j) & = & c_i h_j & i \in \{1, \ldots, 4\}, j \in \{1, \ldots, 8\} \\
\gamma(h_i, o_j) & = & h_i o_j & i \in \{1, \ldots, 8\}, j \in \{1, \ldots, 6\} \\
\gamma(c_i, o_j) & = & c_i o_j & i \in \{1, \ldots, 4\}, j \in \{1, \ldots, 6\} \\
\gamma(c_i, n) & = & c_i n & i \in \{1, \ldots, 4\} \\
\gamma(h_i, n) & = & h_i n & i \in \{1, \ldots, 8\} \\
\gamma(n, p) & = & np &
\end{array}$$

Now we are ready to model H_2O and CH_2O molecules. A water molecule is modelled simply as a composition of two copies of the hydrogen process with one copy of oxygen; see below. The restriction of actions h_i and o_i, for $i \in \{3, 4\}$, ensures that actions such as $h_3[5]$ cannot be undone alone but only together with their partners $o_3[5]$. This happens via undoing of $h_3o_3[5]$ bond. Correspondingly, when any of these actions is fresh they can only happen together with their partners (as prescribed by the communication function γ), and not alone. We also restrict np to stop self-bonding of water's hydrogens with its oxygen.

$$((h_3[5]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3[5], o_4[6], n).O_2') \setminus \{h_3, h_4, o_3, o_4, np\}$$

The formaldehyde molecule is modelled by

$$(((c_1[1], c_2[2], c_3[3], c_4[4]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2' \mid (o_1[3], o_2[4], n).O_1') \\ \setminus\{c_1, c_2, c_3, c_4, h_1, h_2, o_1, o_2, np, \underline{c_1h_1}, \underline{c_2h_2}, \}$$

We restrict the appropriate versions of the c_i, o_j and h_k actions, for all i, j and k, meaning that we do not allow the creation of new bonds (involving these actions) between different molecules as single acts of synchronisation. Such new bonds will be created via concerted actions and (almost) always via the weak np bonds which will then get promoted to strong bonds.

We also restrict $\underline{c_ih_i}$ for $i \in \{1, 2\}$. It prevents breaking any of the bonds between C_1 and its hydrogens H_1 and H_2. This serves two purposes. Firstly, it makes sure that once we have done the p action of the carbon, we will break one of the bonds between the carbon and the oxygen. This is justified since in reality it is one of the oxygen bonds which is broken, so this models the reality closely. Secondly, it also prevents O_2 or O_3 from abstracting H_1 or H_2 from the carbon. Note that $\underline{h_i}$, $\underline{o_j}$ and $\underline{n}$ are not restricted in FA and in W: this allows us to break bonds involving these actions as a part of concerted actions.

The four molecules of the reaction are now composed in parallel:

$$(\mathrm{CH_2O} \mid \mathrm{H_2O} \mid \mathrm{H_2O} \mid \mathrm{H_2O}) \setminus \{n, p\}$$

We restrict actions n and p, so that they cannot happen separately from each other but only together within this system of processes.

The main path through the reaction require only two copies of water so we start with the following initial state, where keys $1, \ldots, 8$ specify the existing initially bonds of formaldehyde and the two waters.

$$\begin{aligned}&(((c_1[1], c_2[2], c_3[3], c_4[4]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2' \mid (o_1[3], o_2[4], n).O_1') \\ &\qquad \setminus \{c_1, c_2, c_3, c_4, h_1, h_2, o_1, o_2, np, \underline{c_1h_1}, \underline{c_2h_2}\} \\ &\mid ((h_3[5]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3[5], o_4[6], n).O_2') \setminus \{h_3, h_4, o_3, o_4, np\} \\ &\mid ((h_5[7]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[7], o_6[8], n).O_3') \setminus \{h_5, h_6, o_5, o_6, np\}) \ \setminus \{n, p\}\end{aligned}$$

In order to simplify the display of transitions or rewrites of this process we omit the four restrictions and write the initial process simply as

$$\begin{aligned}&(c_1[1], c_2[2], c_3[3], c_4[4]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2' \mid (o_1[3], o_2[4], n).O_1' \\ &\mid (h_3[5]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3[5], o_4[6], n).O_2' \\ &\mid (h_5[7]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[7], o_6[8], n).O_3'\end{aligned}$$

remembering that the restrictions are still there.

6.1. The main path through the reaction

We first consider the reaction steps in Figure 10, following the path via i3 | $\mathrm{H_3O}$ | W and i2 | W | W in Figures 14 and 15. The first step is the n, p reaction between C_1 and O_2 or O_3. Note that there are other n, p reactions that are allowed by our model. We could have O_2 getting a hydrogen from O_3, or vice versa, which is the autoprotolysis of water which we have mentioned before. We shall discuss this further later on. Also O_1 could pull one of the hydrogens from one of the water molecules: again we shall discuss it later.

Returning to the first step of our reaction, we have O_2 bonding with C with the key 11. This is followed immediately by breaking of the bond 3 or 4 by the rule concert. Note that breaking of 1 or 2 is not possible because of the restriction of c_1h_1 and c_2h_2. We break the bond 4 and get a transition by concerted actions: we create the bond $np[11]$ and break the bond $\underline{c_4o_2}[4]$. Henceforth we shall write the name of the target of a transition below the transition, using the names that appear in Figures 14-15. Here, for example, the compound resulting from this transition is i2 | W but since in general we have an extra water W, we write i2 | W | W.

$$\xrightarrow{\{np[11],\underline{c_4o_2}[4]\}} (c_1[1], c_2[2], c_3[3], c_4; p[11]).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2' \mid \\ (o_1[3], o_2, n).O_1' \mid (h_3[5]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3[5], o_4[6], n[11]).O_2' \\ \mid (h_5[7]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[7], o_6[8], n).O_3'$$

i2 | W | W

Now the prom rewrite rule must be applied before we derive the next concerted transition: we promote the weak bond 11 of the carbon to a stronger bond on c_4, which has become available:

$$\Rightarrow ((c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2' \\ \mid (o_1[3], o_2, n).O_1' \mid (h_3[5]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3[5], o_4[6], n[11]).O_2' \\ \mid (h_5[7]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[7], o_6[8], n).O_3') \setminus L$$

i2 | W | W

The next step is to form a bond between O_3 and either H_3 or H_4. We bond with H_3 with the key 12 and break the bond 5, producing a pair of concerted actions. We then move a weak bond 11 on n in O_2 to a stronger bond on o_3 (which has become available) by rewrite rule move-r. We also promote the weak bond 12 in H_3 to a strong bond on h_3 by rewrite rule prom, giving overall this transition:

$$\xrightarrow{\{np[12],\underline{h_3o_3}[5]\}} (c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2' \mid \\ (o_1[3], o_2, n).O_1' \mid (h_3[12]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3[11], o_4[6], n).O_2' \\ \mid (h_5[7]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[7], o_6[8], n[12]).O_3'$$

i3 | H_3O | W

The next step is a proton transfer from O_3 to O_1. We transfer H_5 (but we could have used H_6 or H_3 since they all have the p action ready). Performing the transfer of H_5 from O_3 to O_1 (and breaking the bond 12), we obtain the following:

$$\xrightarrow{\{np[13],\underline{h_5o_5}[12]\}} (c_1[1],c_2[2],c_3[3],c_4[11];p).C' \mid (h_1[1];p).H_1' \mid (h_2[2];p).H_2' \mid$$
$$(o_1[3],o_2,n[13]).O_1' \mid (h_3;p[13]).H_3' \mid (h_4[6];p).H_4' \mid (o_3[11],o_4[6],n).O_2'$$
$$\mid (h_5[7];p).H_5' \mid (h_6[8];p).H_6' \mid (o_5[7],o_6[8],n).O_3'$$

MD | W | W

and promoting and moving the bond 13 in H_3 and O_1, respectively, to strong bonds we obtain the final product (methanediol $CH_2(OH)_2$) and two waters:

$$\Rightarrow\ (c_1[1],c_2[2],c_3[3],c_4[11];p).C' \mid (h_1[1];p).H_1' \mid (h_2[2];p).H_2'$$
$$\mid (o_1[3],o_2[13],n).O_1' \mid (h_3[13];p).H_3' \mid (h_4[6];p).H_4' \mid (o_3[11],o_4[6],n).O_2'$$
$$\mid (h_5[7];p).H_5' \mid (h_6[8];p).H_6' \mid (o_5[7],o_6[8],n).O_3'$$

MD | W | W

Note that the n, p actions are ready again and all the existing bonds are on strong actions. So we now can reverse the reaction by O_3 abstracting a hydrogen from H_4 or H_5, and all the way to the initial state. Moreover, let us inspect the bonds 4, 5 and 7 which are broken during the reaction. The bonds were formed during the initial bonding of the atoms. They are broken as a result of application of our new general prefixing operator. This operator enables the reaction to work forwards without external control. Indeed the original order of the formation of the bonds is completely irrelevant for the reaction to work.

6.2. The base-catalysed path

We now consider the base-catalysed path described in Figure 12. The path is via FA | W | HO | H_3O and i3 | H_3O | W to MD | W | W in Figures 14-15. The path needs three water molecules. We shall show the application of the promotion or move rules without explaining them in detail. We start with the initial system (remembering that restrictions are not shown):

$$(c_1[1],c_2[2],c_3[3],c_4[4];p).C' \mid (h_1[1];p).H_1' \mid (h_2[2];p).H_2' \mid (o_1[3],o_2[4],n).O_1'$$
$$\mid (h_3[5];p).H_3' \mid (h_4[6];p).H_4' \mid (o_3[5],o_4[6],n).O_2'$$
$$\mid (h_5[7];p).H_5' \mid (h_6[8];p).H_6' \mid (o_5[7],o_6[8],n).O_3'$$
$$\mid (h_7[9];p).H_7' \mid (h_8[10];p).H_8' \mid (o_7[9],o_8[10],n).O_4'$$

FA | W | W | W

$$\xrightarrow{\{np[12],\underline{h_3o_3}[5]\}} (c_1[1],c_2[2],c_3[3],c_4[4];p).C' \mid (h_1[1];p).H_1' \mid (h_2[2];p).H_2'$$
$$\mid (o_1[3],o_2[4],n).O_1' \mid (h_3;p[12]).H_3' \mid (h_4[6];p).H_4' \mid (o_3,o_4[6],n).O_2'$$
$$\mid (h_5[7];p).H_5' \mid (h_6[8];p).H_6' \mid (o_5[7],o_6[8],n[12]).O_3'$$
$$\mid (h_7[9];p).H_7' \mid (h_8[10];p).H_8' \mid (o_7[9],o_8[10],n).O_4'$$

$$\Rightarrow (c_1[1], c_2[2], c_3[3], c_4[4]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'$$
$$\mid (o_1[3], o_2[4], n).O_1' \mid (h_3[12]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3, o_4[6], n).O_2'$$
$$\mid (h_5[7]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[7], o_6[8], n[12]).O_3'$$
$$\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n).O_4'$$

FA | W | HO | H$_3$O

$$\xrightarrow{\{np[11], \underline{c_4 o_4}[4]\}} (c_1[1], c_2[2], c_3[3], c_4; p[11]).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'$$
$$\mid (o_1[3], o_2, n).O_1' \mid (h_3[12]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3, o_4[6], n[11]).O_2'$$
$$\mid (h_5[7]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[7], o_6[8], n[12]).O_3'$$
$$\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n).O_4'$$

$$\Rightarrow (c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'$$
$$\mid (o_1[3], o_2, n).O_1' \mid (h_3[12]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3[11], o_4[6], n).O_2'$$
$$\mid (h_5[7]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[7], o_6[8], n[12]).O_3'$$
$$\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n).O_4'$$

i3 | H$_3$O | W

$$\xrightarrow{\{np[13], \underline{h_3 n}[12]\}} (c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'$$
$$\mid (o_1[3], o_2, n[13]).O_1' \mid (h_3; p[13]).H_3' \mid (h_4[6]; p).H_4' \mid (o_3[11], o_4[6], n).O_2'$$
$$\mid (h_5[7]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[7], o_6[8], n).O_3'$$
$$\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n).O_4'$$

$$\Rightarrow (c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'$$
$$\mid (o_1[3], o_2[13], n).O_1' \mid (h_3[13]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3[11], o_4[6], n).O_2'$$
$$\mid (h_5[7]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[7], o_6[8], n).O_3'$$
$$\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n).O_4'$$

MD | W | W

We have reached the final state of the reaction: a methanediol with two waters. Notice that the methanediol process is identical (including keys) to the methanediol process in the main path.

6.3. The acid-catalysed path

Next we consider the acid-catalysed path described in Figure 11. The path is via i6 | W | HO | W and i8 | HO | W in Figures 14–15. The path uses three water molecules as in the base-catalysed path. We shall apply the promotion or move rules as needed. We start with the initial system:

$$\begin{aligned}&(c_1[1], c_2[2], c_3[3], c_4[4]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'\\ &\mid (o_1[3], o_2[4], n).O_1' \mid (h_3[12]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3, o_4[6], n).O_2'\\ &\mid (h_5[7]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[7], o_6[8], n[12]).O_3'\\ &\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n).O_4'\end{aligned}$$

FA | W | HO | H_3O

$$\begin{aligned}&\xrightarrow{\{np[13], \underline{h_5 o_5}[7]\}} (c_1[1], c_2[2], c_3[3], c_4[4]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'\\ &\mid (o_1[3], o_2[4], n[13]).O_1' \mid (h_3[12]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3, o_4[6], n).O_2'\\ &\mid (h_5; p[13]).H_5' \mid (h_6[8]; p).H_6' \mid (o_5, o_6[8], n[12]).O_3'\\ &\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n).O_4'\end{aligned}$$

$$\begin{aligned}&\Rightarrow (c_1[1], c_2[2], c_3[3], c_4[4]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'\\ &\mid (o_1[3], o_2[4], n[13]).O_1' \mid (h_3[12]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3, o_4[6], n).O_2'\\ &\mid (h_5[13]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[12], o_6[8], n).O_3'\\ &\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n).O_4'\end{aligned}$$

i6 | W | HO | W

The last rewrite is by rule move-r. The reaction continues as follows:

$$\begin{aligned}&\xrightarrow{\{np[11], \underline{c_4 o_2}[4]\}} (c_1[1], c_2[2], c_3[3], c_4; p[11]).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'\\ &\mid (o_1[3], o_2, n[13]).O_1' \mid (h_3[12]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3, o_4[6], n).O_2'\\ &\mid (h_5[13]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[12], o_6[8], n[11]).O_3'\\ &\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n).O_4'\end{aligned}$$

$$\begin{aligned}&\Rightarrow (c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'\\ &\mid (o_1[3], o_2[13], n).O_1' \mid (h_3[12]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3, o_4[6], n).O_2'\\ &\mid (h_5[13]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[12], o_6[8], n[11]).O_3'\\ &\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n).O_4')\end{aligned}$$

i8 | HO | W

We continue from i8 | HO | W via MD | HO | H_3O with concerted actions:

$$\begin{aligned}&\xrightarrow{\{np[14], \underline{h_6 o_6}[8]\}} (c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'\\ &\mid (o_1[3], o_2[13], n).O_1' \mid (h_3[12]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3, o_4[6], n).O_2'\\ &\mid (h_5[13]; p).H_5' \mid (h_6; p[14]).H_6' \mid (o_5[12], o_6, n[11]).O_3'\\ &\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n[14]).O_4'\end{aligned}$$

$$\Rightarrow (c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'$$
$$\mid (o_1[3], o_2[13], n).O_1' \mid (h_3[12]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3, o_4[6], n).O_2'$$
$$\mid (h_5[13]; p).H_5' \mid (h_6[14]; p).H_6' \mid (o_5[12], o_6[11], n).O_3'$$
$$\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n[14]).O_4'$$

MD | HO | H_3O

Finally we reach the final state of a methanediol and two waters:

$$\xrightarrow{\{np[5], \underline{h_8 o_8}[14]\}} (c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'$$
$$\mid (o_1[3], o_2[13], n).O_1' \mid (h_3[12]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3, o_4[6], n[5]).O_2'$$
$$\mid (h_5[13]; p).H_5' \mid (h_6[14]; p).H_6' \mid (o_5[12], o_6[11], n).O_3'$$
$$\mid (h_7[9]; p).H_7' \mid (h_8; p[5]).H_8' \mid (o_7[9], o_8[10], n[14]).O_4'$$

$$\Rightarrow (c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'$$
$$\mid (o_1[3], o_2[13], n).O_1' \mid (h_3[12]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3[5], o_4[6], n).O_2'$$
$$\mid (h_5[13]; p).H_5' \mid (h_6[10]; p).H_6' \mid (o_5[12], o_6[11], n).O_3'$$
$$\mid (h_7[9]; p).H_7' \mid (h_8[5]; p]).H_8' \mid (o_7[9], o_8[10], n).O_4'$$

MD | W | W

We return to the transition from the intermediate 6 to the intermediate 8 in Figure 11. Once the hydrogen is bound to the oxygen in the compound 6 we have a so-called *resonance*, where the positive charge can be on the oxygen or the carbon (or in between) and the structure resonates between the intermediate 6 and 7 (indicated by the dashed arrow in Figure 11). We can model this movement between the two intermediates as follows: starting from i6 | W | HO | W we break bond 4 without forming a new bond at the same time, and perform a move-r rewrite on O_1:

$$\xrightarrow{\underline{c_4 o_2}[4]} (c_1[1], c_2[2], c_3[3], c_4; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'$$
$$\mid (o_1[3], o_2[13], n).O_1' \mid (h_3[12]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3, o_4[6], n).O_2'$$
$$\mid (h_5[13]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[12], o_6[8], n).O_3'$$
$$\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n).O_4'$$

i7 | W | HO | W

The movement from i7 to i6 is gotten by creating the bond on c_4 and n of O_1 with the key 4:

$$\xrightarrow{c_4 n[4]} (c_1[1], c_2[2], c_3[3], c_4[4]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'$$
$$\mid (o_1[3], o_2[13], n[4]).O_1' \mid (h_3[12]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3, o_4[6], n).O_2'$$
$$\mid (h_5[13]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[12], o_6[8], n).O_3'$$
$$\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n).O_4'$$

Note that the last compound is equivalent chemically to i6 | W | HO | W, and it is also identical syntactically to i6 | W | HO | W except the keys 4 and 13 are swapped in O_1.

We have noted before that the main path and the base-catalysed path form a diamond: we can get from FA | W | W | W to i3 | H_3O | W either via i2 | W | W or via FA | W | HO | H_3O. We underline that the resulting methanediol processes are syntactically identical. However, there is no such tight correspondence between methanediol in the main path and methanediol in the acid-catalysed path. In the main path we bind a H_2O to the carbon, and then use another H_2O as a proton shuttle to move a hydrogen. The H_2O which is used as a proton shuttle is unchanged. In the acid-catalysed path we bind a hydrogen to the formaldehyde first, bind a H_2O to it and then remove one of these hydrogens and put it back on a water. Hence, the two methanediol processes are not identical but they represent chemically the same compound. We shall explain later how this equivalence might be formally defined.

6.4. Other paths

We now discuss the remaining reactions in Figure 14.

The compounds 13 and 14 in Figure 13 are of particular interest since they have only one path leading to and out of them. Firstly we can get from i3 | H_3O | W to i13 | H_3O | H_3O as follows:

$$\begin{aligned}&(c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2' \mid (o_1[3], o_2, n).O_1'\\ &\mid (h_3[12]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3[11], o_4[6], n).O_2'\\ &\mid (h_5[7]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[7], o_6[8], n[12]).O_3'\\ &\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n).O_4'\end{aligned}$$

i3 | H_3O | W

$$\begin{aligned}&\xrightarrow{\{np[13], \underline{h_4 o_4}[6]\}} (c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'\\ &\mid (o_1[3], o_2, n).O_1' \mid (h_3[12]; p).H_3' \mid (h_4; p[13]).H_4' \mid (o_3[11], o_4, n).O_2'\\ &\mid (h_5[7]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[7], o_6[8], n[12]).O_3'\\ &\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n[13]).O_4'\end{aligned}$$

$$\begin{aligned}&\Rightarrow (c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'\\ &\mid (o_1[3], o_2, n).O_1' \mid (h_3[12]; p).H_3' \mid (h_4[13]; p).H_4' \mid (o_3[11], o_4, n).O_2'\\ &\mid (h_5[7]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[7], o_6[8], n[12]).O_3'\\ &\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n[13]).O_4'\end{aligned}$$

i13 | H_3O | H_3O

We can also reverse back to i3 | H_3O | W by performing concerted actions followed by a promotion and a move:

$$\xrightarrow{\{np[6],\underline{h_4 n}[13]}\}(c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'$$
$$\mid (o_1[3], o_2, n).O_1' \mid (h_3[12]; p).H_3' \mid (h_4; p[6]).H_4' \mid (o_3[11], o_4, n[6]).O_2'$$
$$\mid (h_5[7]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[7], o_6[8], n[12]).O_3'$$
$$\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n).O_4'$$

$$\Rightarrow (c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'$$
$$\mid (o_1[3], o_2, n).O_1' \mid (h_3[12]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3[11], o_4[6], n).O_2'$$
$$\mid (h_5[7]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[7], o_6[8], n[12]).O_3'$$
$$\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_3[9], o_8[10], n).O_4'$$

i3 | H_3O | W

Furthermore can also get from i8 | HO | W to i14 | HO | HO:

$$(c_1[1], c_2[2], c_3[3], c_4[1]; p]).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'$$
$$\mid (o_1[3], o_2[13], n).O_1' \mid (h_3[12]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3, o_4[6], n).O_2'$$
$$\mid (h_5[13]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[12], o_6[8], n[11]).O_3'$$
$$\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n).O_4'$$

i8 | HO | W

$$\xrightarrow{\{np[14],\underline{h_7 o_7}[9]\}} (c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'$$
$$\mid (o_1[3], o_2[13], n[14]).O_1' \mid (h_3[12]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3, o_4[6], n).O_2'$$
$$\mid (h_5[13]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[12], o_6[8], n[11]).O_3'$$
$$\mid (h_7; p[14]).H_7' \mid (h_8[10]; p).H_8' \mid (o_7, o_8[10], n).O_4'$$

$$\Rightarrow (c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2'$$
$$\mid (o_1[3], o_2[13], n[14]).O_1' \mid (h_3[12]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3, o_4[6], n).O_2'$$
$$\mid (h_5[13]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[12], o_6[8], n[11]).O_3'$$
$$\mid (h_7[14]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7, o_8[10], n).O_4'$$

i14 | HO | HO

And we can reverse back to i8 | HO | W in a very much the same way as from i13 | H_3O | H_3O to i3 | H_3O | W.

The reaction from i6 | W | HO | W to i6 | HO | HO | H_3O and its inverse is the autoprotolysis of water, and work independently of the rest. The corresponding applies to the reaction from i7 | HO | HO | H_3O to i7 | W | HO | W and its reversal.

The step from i6 | HO | HO | H_3O to i7 | HO | HO | H_3O involves the intermediate i6 and i8 only (as in Figure 11) and no other compounds, so the reaction is as described before.

The reaction from i2 | W | W to i2 | HO | H_3OW is also an autoprotolysis of water.

There are seven more reactions possible, all via concerted actions. We only describe one of them, namely from FA | W | W | W to i6 | W | HO | W:

$$\begin{aligned}
&(c_1[1], c_2[2], c_3[3], c_4[4]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2' \\
&\mid (o_1[3], o_2[4], n).O_1' \mid (h_3[5]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3[5], o_4[6], n).O_2' \\
&\mid (h_5[7]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[7], o_6[8], n).O_3' \\
&\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n).O_4'
\end{aligned}$$

$$\begin{aligned}
&\xrightarrow{\{np[13], \underline{h_5 o_5}[7]\}} (c_1[1], c_2[2], c_3[3], c_4[4]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2' \\
&\mid (o_1[3], o_2[4], n[13]).O_1' \mid (h_3[5]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3[5], o_4[6], n).O_2' \\
&\mid (h_5; p[13]).H_5' \mid (h_6[8]; p).H_6' \mid (o_5, o_6[8], n).O_3' \\
&\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n).O_4'
\end{aligned}$$

$$\begin{aligned}
&\Rightarrow (c_1[1], c_2[2], c_3[3], c_4[4]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2' \\
&\mid (o_1[3], o_2[4], n[13]).O_1' \mid (h_3[5]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3[5], o_4[6], n).O_2' \\
&\mid (h_5[13]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5, o_6[8], n).O_3' \\
&\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n).O_4'
\end{aligned}$$

i6 | W | HO | W

6.5. Chemical equivalence

We have seen that the system of FA with three W evolves to MA with two W via different sequences of concerted actions. Some paths end up at the same process, others lead to somewhat different syntactically processes. We have explained that these different processes represent the same chemical compounds, and we have explained that the difference originates as a side effect of using subscripted copies of atoms, where both the action names and process names are uniformly subscripted, and of using keys. Now we present informally our notion of *chemical equivalence* of systems of subscripted copies of processes. This essentially amounts to performing alpha conversion on subscripts and keys in processes. We leave the precise formulation for future research. Here, we only demonstrate how the methanediol obtained via the main path is equivalent chemically to the methanediol gotten via the acid-catalysed path by performing a number of atom swaps or key changes.

It is clear that a copy H_3 of hydrogen can be replaced in a compound by H_5 without changing the chemical meaning of the compound provided that H_5 does not appear in the compound in the first place. Also, it is irrelevant what precise keys are used in a process as we can replace a key by another fresh key without changing the meaning of the process. Hence, using alpha conversion, we can define a *swap* operation of subscripted versions of the same process and a *swap* operation of keys, and employ these swap operations to prove processes chemically equivalent.

Recall the MD process produced via the main path:

$$\begin{aligned}&c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2' \mid (o_1[3], o_2[13], n).O_1' \\ &\mid (h_3[13]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3[11], o_4[6], n).O_2' \\ &\mid (h_5[7]; p).H_5' \mid (h_6[8]; p).H_6' \mid (o_5[7], o_6[8], n).O_3' \\ &\mid (h_7[9]; p).H_7' \mid (h_8[10]; p).H_8' \mid (o_7[9], o_8[10], n).O_4'\end{aligned}$$

and the MD process produced via acid-catalysed path:

$$\begin{aligned}&c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2' \mid (o_1[3], o_2[13], n).O_1' \\ &\mid (h_3[12]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3[5], o_4[6], n).O_2' \\ &\mid (h_5[13]; p).H_5' \mid (h_6[10]; p).H_6' \mid (o_5[12], o_6[11], n).O_3' \\ &\mid (h_7[9]; p).H_7' \mid (h_8[5]; p]).H_8' \mid (o_7[9], o_8[10], n).O_4'\end{aligned}$$

We take the second MD and we swap H_3 with H_5. This results in H_3 becoming bonded to O_1 with key 13, and H_5 becoming bonded to O_3 with key 12. Note that the same result is obtained by simply swapping keys 12 and 13 in H_3 with H_5. This gives us:

$$\begin{aligned}&c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2' \mid (o_1[3], o_2[13], n).O_1' \\ &\mid (h_3[13]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3[5], o_4[6], n).O_2' \\ &\mid (h_5[12]; p).H_5' \mid (h_6[10]; p).H_6' \mid (o_5[12], o_6[11], n).O_3' \\ &\mid (h_7[9]; p).H_7' \mid (h_8[5]; p]).H_8' \mid (o_7[9], o_8[10], n).O_4'\end{aligned}$$

Next, we swap O_3 (together with H_5 which is bonded to O_3) with O_2 (and H_4 which is bonded to O$_2$). This is done by swapping the key 11 on o_6 of O_3 with the key 5 on o_3 of O_2. This gives us

$$\begin{aligned}&c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2' \mid (o_1[3], o_2[13], n).O_1' \\ &\mid (h_3[13]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3[11], o_4[6], n).O_2' \\ &\mid (h_5[12]; p).H_5' \mid (h_6[10]; p).H_6' \mid (o_5[12], o_6[5], n).O_3' \\ &\mid (h_7[9]; p).H_7' \mid (h_8[5]; p]).H_8' \mid (o_7[9], o_8[10], n).O_4'\end{aligned}$$

Finally we swap H_6 and H_8, which is the same as swapping keys 5 and 10 in H_6 and H_8, producing the following:

$$\begin{aligned}&c_1[1], c_2[2], c_3[3], c_4[11]; p).C' \mid (h_1[1]; p).H_1' \mid (h_2[2]; p).H_2' \mid (o_1[3], o_2[13], n).O_1' \\ &\mid (h_3[13]; p).H_3' \mid (h_4[6]; p).H_4' \mid (o_3[11], o_4[6], n).O_2' \\ &\mid (h_5[12]; p).H_5' \mid (h_6[5]; p).H_6' \mid (o_5[12], o_6[5], n).O_3' \\ &\mid (h_7[9]; p).H_7' \mid (h_8[10]; p]).H_8' \mid (o_7[9], o_8[10], n).O_4'\end{aligned}$$

This is identical to the MD obtained obtained via the main path if we replace also all the keys 12 and 5 by the keys 7 and 8, respectively, which are fresh.

6.6. Evaluation

In this subsection we evaluate the suitability of CCB in the modelling of covalent reactions as exemplified by the hydration of formaldehyde in water. We first consider if all chemically valid interactions between the compounds of the reaction can be represented well in our calculus. We have seen in Figure 11 the resonance between the intermediates i6 and i7. We are able to model appropriately the movement between i6 and i7 by transitions that break a bond or create a bond on its own. What is less clear is why this bond break should happen at this point and not elsewhere. We are unable to represent the movement from i7 to i8, however, we model appropriately the evolution from i6 to i8 via a concerted actions transition. Apart from the reaction steps described above, all other steps can be represented well in our calculus.

The other way of assessing the suitability of CCB for this type of reactions is to ask if our calculus enables transitions which do not occur in reality. We notice that some bonds can break on their own as, for example, in water molecules. Although this does not happen in reality an alternative process exists which produces the same products H and HO. This is the autoprotolysis of water which we have described earlier and in [16]. So the breaking of water molecules could be treated as harmless. Also a bond between carbon and oxygen in methanediol can break on its own, without being accompanied by a new bond formation. This is a consequence of allowing the break of the double bond in the compound i6 on its own. The difference between these two cases is not properly modelled in our calculus. However, if the carbon-oxygen bond in MD breaks the carbon could bond immediately afterwards to the water, and we would get the compound i8, which is a valid intermediate, so this break of the carbon-oxygen bond does not not cause a problem. Our model also allows the interaction of the methanediol with a water molecule in the final system MD | W | W. Since the carbon C has the action p ready, and the oxygen O_3 in the water has the action n ready, and n, p is a part of our synchronisation function, a bond between C and O_3 can be created[3]. The reason why this type of interaction does not occur in reality is that the four groups around the carbon shield it from any interaction (as opposed to the three groups on the formaldehyde, of which two are relatively small hydrogens). This is a *steric* effect, the effect due to atoms occupying space and preventing other atoms from moving. Our calculus does not model spacial arrangement of atoms well enough; we aim to address this in the future.

Finally we discuss how CCB compares to other calculi used for the modelling biochemical reactions. We start with the process calculi based on CCS such as CCS-R [7, 8] or CCSK [25, 26]. One of the advantages of CCB are the concerted actions transitions. This permits us to represent faithfully that in covalent reactions some compounds are unstable and cannot exist independently of the reaction. A double bonded hydrogen is an example: it exists only "during" concerted actions transition. Other calculi cannot represent this point; a double

[3]We have used a similar mechanism of C interacting with O to get from the compound 1 to the compound 2 in Figure 10.

bonded hydrogen is a valid compound in those calculi and two transitions are used instead of our concerted transition.

Graph-based formalisms such as the κ-calculus [9] use rewrite rules to encode which bonds can be created, which bonds can be broken, and in which states. So the designer of the rules influences what reactions can occur and how they occur. In CCB, however, the processes contain all the information needed to drive reactions, of course in accordance with the given SOS, without any external control or guidance.

7. Conclusion

We have introduced a reversible process calculus CCB with a novel prefixing operator which is inspired by the mechanism of covalent bonding. This mechanism allows us to model locally controlled reversibility. We have given the calculus operational semantics. The new operator permits us to perform pairs of concerted actions, where the first element of the pair is a creation of a (weak) bond and the second element is breaking one of the existing bonds. We have also proposed rewrite rules that model promotion of weak bonds to strong bonds. Our prefixing operator provides a purely local control of computation; there is no need for an extensive memory or global control.

We have shown that the sub-calculus $\mathrm{CCB_s}$ satisfies conservation and causal consistency, and the full calculus satisfies several diamond properties. CCB is more expressive than other reversible calculi as it can also model out-of-causal order computation. We have shown that biochemical reactions with covalent bonding can be represented naturally and faithfully thanks to our new prefixing operator, and via concerted actions transitions and promotion rules. We have presented a detailed CCB model of the hydration of formaldehyde in water into methanediol, an industrially important reaction, where the creation and breaking of some bonds are examples of locally controlled out-of-causal order computation.

Acknowledgements

We are grateful to the referees for their detailed and helpful comments and suggestions. The authors acknowledge partial support of COST Action IC1405 on Reversible Computation - extending horizons of computing. The second author acknowledges the support by the University of Leicester in granting him Academic Study Leave, and thanks Nagoya University for support during the study leave.

References

[1] J. C. M. Baeten, W. P. Weijland, Process Algebra (Cambridge Tracts in Theoretical Computer Science 18), Cambridge University Press, 1990.

[2] L. Cardelli, Brane Calculi, in: V. Danos, V. Schachter (Eds.), Computational Methods in Systems Biology, vol. 3082 of *LNCS*, Springer, 257–278, 2005.

[3] F. Ciocchetta, J. Hillston, Bio-PEPA: A Framework for the Modelling and Analysis of Biological Systems, Theoretical Computer Science 410 (33-34) (2009) 3065–3084.

[4] J. Claydon, N. Greeves, S. Warren, P. Wothers, Organic Chemistry, Oxford University Press, 2001.

[5] R. Cleaveland, G. Löttgen, V. Natarajan, Priority in Process Algebra, in: J. Bergstra, A. Ponse, S. Smolka (Eds.), Handbook of Process Algebra, chap. 12, Elsevier Science, Amsterdam, 711–765, 2001.

[6] I. Cristescu, J. Krivine, D. Varacca, A Compositional Semantics for the Reversible π-Calculus, in: Proceedings of LICS 2013, IEEE Computer Society, 388–397, 2013.

[7] V. Danos, J. Krivine, Reversible Communicating Systems, in: P. Gardner, N. Yoshida (Eds.), CONCUR 2004 - Concurrency Theory, vol. 3170 of *LNCS*, Springer, 292–307, 2004.

[8] V. Danos, J. Krivine, Formal Molecular Biology Done in CCS-R, Electronic Notes in Theoretical Computer Science 180 (3) (2007) 31 − 49, Proceedings of the First Workshop on Concurrent Models in Molecular Biology (BioConcur 2003).

[9] V. Danos, C. Laneve, Formal molecular biology, Theoretical Computer Science 325 (1) (2004) 69–110.

[10] F. Fages, S. Soliman, N. Chabrier-Rivier, Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM, Journal of Biological Physics and Chemistry 4 (2004) 64–73.

[11] W. Fokkink, Introduction to Process Algebra (Texts in Theoretical Computer Science. An EATCS Series), Springer, 2000.

[12] W. Fontana, L. Buss, The Barrier of Objects: From Dynamical Systems to Bounded Organizations, IIASA Working Paper, International Institute for Applied Systems Analysis, IIASA, Austria, 1996.

[13] D. T. Gillespie, Exact stochastic simulation of coupled chemical reactions, The Journal of Physical Chemistry 81 (25) (1977) 2340–2361.

[14] C. A. R. Hoare, Communicating Sequential Processes (Prentice Hall International Series in Computing Science), Prentice-Hall, 1985.

[15] A. Köhler, J. Krivine, J. Vidmar, A Rule-Based Model of Base Excision Repair, in: P. Mendes, J. O. Dada, K. Smallbone (Eds.), Computational Methods in Systems Biology, vol. 8859 of *Lecture Notes in Bioinformatics*, Springer, 173–195, 2014.

[16] S. Kuhn, I. Ulidowski, Towards Modelling of Local Reversibility, in: J. Krivine, J.-B. Stefani (Eds.), Reversible Computation, vol. 9138 of *LNCS*, Springer, 279–284, 2015.

[17] S. Kuhn, I. Ulidowski, A Calculus for Local Reversibility, in: S. GDevitt, I. Lanese (Eds.), Reversible Computation, vol. 9720 of *LNCS*, Springer, 20–35, 2016.

[18] I. Lanese, C. Mezzina, A. Schmitt, J.-B. Stefani, Controlling Reversibility in Higher-Order Pi, in: J.-P. Katoen, B. König (Eds.), CONCUR 2011 Concurrency Theory, vol. 6901 of *LNCS*, Springer, 297–311, 2011.

[19] I. Lanese, C. Mezzina, J.-B. Stefani, Reversing Higher-Order Pi, in: Proceedings of the 21st International Conference on Concurrency Theory CONCUR 2010, vol. 6269 of *LNCS*, Springer, 478–493, 2010.

[20] I. Lanese, C. A. Mezzina, J.-B. Stefani, Controlled Reversibility and Compensations, in: R. Glück, T. Yokoyama (Eds.), Reversible Computation, vol. 7581 of *LNCS*, Springer, 233–240, 2013.

[21] R. Milner, A Calculus of Communicating Systems, vol. 92 of *LNCS*, Springer, 1980.

[22] R. Milner, Communicating and Mobile Systems: The π-calculus, Cambridge University Press, New York, NY, USA, 1999.

[23] M. Mousavi, I. Phillips, M. A. Reniers, I. Ulidowski, Semantics and expressiveness of ordered SOS, Information and Computation 207 (2) (2009) 85–119.

[24] G. Păun, Computing with Membranes, Journal of Computer and System Sciences 61 (1) (2000) 108–143.

[25] I. Phillips, I. Ulidowski, Reversing algebraic process calculi, in: Proceedings of 9th International Conference on Foundations of Software Science and Computation Structures, FOSSACS 2006, vol. 3921 of *LNCS*, Springer, 246–260, 2006.

[26] I. Phillips, I. Ulidowski, Reversing algebraic process calculi, The Journal of Logic and Algebraic Programming 73 (12) (2007) 70–96.

[27] I. Phillips, I. Ulidowski, Reversibility and Asymmetric Conflict in Event Structures, in: P. D'Argenio, H. Melgratti (Eds.), CONCUR 2013 Concurrency Theory, vol. 8052 of *LNCS*, Springer, 303–318, 2013.

[28] I. Phillips, I. Ulidowski, S. Yuen, Modelling of Bonding with Processes and Events, in: G. W. Dueck, D. M. Miller (Eds.), Reversible Computation, vol. 7948 of *LNCS*, Springer, 141–154, 2013.

[29] I. Phillips, I. Ulidowski, S. Yuen, A Reversible Process Calculus and the Modelling of the ERK Signalling Pathway, in: R. Glück, T. Yokoyama (Eds.), Reversible Computation, vol. 7581 of *LNCS*, Springer, 218–232, 2013.

[30] C. Priami, Stochastic π-Calculus, The Computer Journal 38 (7) (1995) 578–589.

[31] C. Priami, A. Regev, E. Shapiro, W. Silverman, Application of a stochastic name-passing calculus to representation and simulation of molecular processes, Information Processing Letters 80 (1) (2001) 25–31.

[32] A. Regev, Representation and simulation of molecular pathways in the stochastic π-calculus, in: U. K. R. Gauges, C. van Gend (Ed.), 2nd Workshop on Computation of Biochemical Pathways and Genetic Networks, Logos, Berlin, 109–115, 2001.

[33] A. Regev, E. M. Panina, W. Silverman, L. Cardelli, E. Shapiro, BioAmbients: an abstraction for biological compartments, Theoretical Computer Science 325 (1) (2004) 141–167.

[34] A. Regev, E. Shapiro, The π-calculus as an Abstraction for Biomolecular Systems, in: G. Ciobanu, G. Rozenberg (Eds.), Modelling in Molecular Biology, Springer, 219–266, 2004.

[35] A. Regev, W. Silverman, E. Shapiro, Representation and simulation of biochemical processes using the π-calculus process algebra, Pacific Symposium on Biocomputing (2001) 459–470.

[36] I. Ulidowski, Equivalences on Observable Processes, in: Proceedings of LICS 1992, IEEE Computer Society, 148–159, 1992.

[37] I. Ulidowski, I. Phillips, Ordered SOS Process Languages for Branching and Eager Bisimulations, Information and Computation 178 (1) (2002) 180 – 213.

[38] I. Ulidowski, I. Phillips, S. Yuen, Concurrency and Reversibility, in: S. Yamashita, S.-I. Minato (Eds.), Reversible Computation, vol. 8507 of *LNCS*, Springer, 1–14, 2014.

Appendix

Proposition 4. *Let P be consistent. If $P \xrightarrow{\{\mu[k],\underline{\nu}[l]\}} P'$ then $k \notin \mathsf{keys}(P)$, $l \in \mathsf{keys}(P)$ and $\mathsf{keys}(P') = (\mathsf{keys}(P) \cup \{k\}) \setminus \{l\}$ for all P'.*

Proof. By induction on the depth of the inference tree of $P \xrightarrow{\{\mu[k],\underline{\nu}[l]\}} Q$.

1. Base case follows for processes with the inference tree of depth 0.
2. Inductive hypothesis: We assume that Proposition 4 holds for all subprocesses R of P and all $\nu[l]$ and all $\mu[k]$, namely if R is a consistent process and $R \xrightarrow{\{\mu[k],\underline{\nu}[l]\}} R'$ for some R' then $k \notin \mathsf{keys}(R)$, $l \in \mathsf{keys}(R)$ and $\mathsf{keys}(R') = (\mathsf{keys}(R) \cup \{k\}) \setminus \{l\}$.
3. Induction step: We show that Proposition 4 holds for P. For this we consider cases depending on the structure of P:
   (a) $P \equiv R \mid Q$: There are two cases here:
      i. The transition is by the rule $\mathsf{concert}$ in Figure 5: Assume P is consistent and $P \xrightarrow{\{\mu[k],\underline{\nu}[l]\}} P'$ for some P'. Since $P \xrightarrow{\{\mu[k],\underline{\nu}[l]\}} P'$, by rule $\mathsf{concert}$, we have $R \xrightarrow{(\mu_1)[k]} R'$, $Q \xrightarrow{(\mu_2)[k]} Q'$ or $Q \xrightarrow{\mu_2[k]} Q'$, $R' \xrightarrow{\underline{\nu_1}[l]} R''$, $Q' \xrightarrow{\underline{\nu_2}[l]} Q''$, $\gamma(\nu_1, \nu_2) = \nu$ and $\gamma(\mu_1, \mu_2) = \mu$ must hold without loss of generality. Also, $P' \equiv R'' \mid Q''$. We can calculate $\mathsf{keys}(P)$: $\mathsf{keys}(P) = \mathsf{keys}(R) \cup \mathsf{keys}(Q)$. Since R and Q can both perform a transition with k we know, by argument similar to that used in the proof of Proposition 1.1, that $k \notin \mathsf{keys}(R)$ and $k \notin \mathsf{keys}(Q)$, and therefore $k \notin \mathsf{keys}(P)$ as required. Since R' and Q' can perform undoing of actions with the key l, we know that $l \in \mathsf{keys}(R')$ and $l \in \mathsf{keys}(Q')$ by Proposition 1.2. Transitions of R and Q to R' and Q' respectively do not involve l, so $l \in \mathsf{keys}(R)$ and $l \in \mathsf{keys}(Q)$ most hold. Since $\mathsf{keys}(P) = \mathsf{keys}(R \cup Q)$ we deduce that $l \in \mathsf{keys}(P)$ as required. Because of Proposition 1.1 and Proposition 1.2 we can calculate $\mathsf{keys}(Q'')$: $\mathsf{keys}(Q'') = \mathsf{keys}(Q') \setminus \{l\} = (\mathsf{keys}(Q) \cup \{k\}) \setminus \{l\}$, $\mathsf{keys}(R'')$: $\mathsf{keys}(R'') = \mathsf{keys}(R') \setminus \{l\} = (\mathsf{keys}(R) \cup \{k\}) \setminus \{l\}$, and $\mathsf{keys}(P')$: $\mathsf{keys}(P') = \mathsf{keys}(R'') \cup \mathsf{keys}(Q'') = (\mathsf{keys}(R) \cup \{k\}) \setminus \{l\} \cup (\mathsf{keys}(Q) \cup \{k\}) \setminus \{l\} = (\mathsf{keys}(R) \cup \mathsf{keys}(Q) \cup \{k\}) \setminus \{l\} = (\mathsf{keys}(P) \cup \{k\}) \setminus \{l\}$ as required.
      ii. The transition is by $\mathsf{concert\ par}$ rule in Figure 5: We assume without loss of generality $R \xrightarrow{\{\mu[k],\underline{\nu}[l]\}} R'$ and $\mathsf{fsh}[k](Q)$, and $P' \equiv R' \mid Q$. By the inductive hypothesis we have $k \notin \mathsf{keys}(R)$, $l \in \mathsf{keys}(R)$ and $\mathsf{keys}(R') = (\mathsf{keys}(R) \cup \{k\}) \setminus \{l\}$. Since $\mathsf{keys}(P) = \mathsf{keys}(R \mid Q) = \mathsf{keys}(R) \cup \mathsf{keys}(Q)$ and $\mathsf{fsh}[k](Q)$ and $k \notin \mathsf{keys}(R)$ we deduce that $k \notin \mathsf{keys}(P)$. Also since $l \in \mathsf{keys}(R)$ and $\mathsf{keys}(P) = \mathsf{keys}(R \mid Q) = \mathsf{keys}(R) \cup \mathsf{keys}(Q)$ and we deduce that $l \in \mathsf{keys}(P)$. Hence, we have $\mathsf{keys}(P') = \mathsf{keys}(R') \cup \mathsf{keys}(Q) = (\mathsf{keys}(R) \cup \{k\}) \setminus \{l\} \cup \mathsf{keys}(Q)$. Since $l \notin \mathsf{keys}(Q)$ by $\mathsf{concert\ par}$, we have

$(\mathsf{keys}(R) \cup \{k\}) \setminus \{l\} \cup \mathsf{keys}(Q) = (\mathsf{keys}(R \mid Q) \cup \{k\}) \setminus \{l\} = (\mathsf{keys}(P) \cup \{k\}) \setminus \{l\}$ as required.

(b) $P \equiv (t; b).R$: The transition is by $\mathsf{concert\ act}$ rule in Figure 5. Assume P is consistent and $P \xrightarrow{\{\mu[k], \underline{\nu}[l]\}} P'$. We deduce that $P' \equiv (t; b).R'$ for some R'. Recall that t denotes a sequence of past actions, namely an element from $\mathcal{AK}^*$. The premises of $\mathsf{concert\ act}$ ensure that $\mathsf{fsh}[k](t)$ and $R \xrightarrow{\{\mu[k], \underline{\nu}[l]\}} R'$. The process R is consistent since P is consistent. Since R is consistent and $R \xrightarrow{\{\mu[k], \underline{\nu}[l]\}} R'$ then, by the inductive hypothesis, $k \notin \mathsf{keys}(R)$, $l \in \mathsf{keys}(R)$ and $\mathsf{keys}(R') = (\mathsf{keys}(R) \cup \{k\}) \setminus \{l\}$. Since $k \notin \mathsf{keys}(R)$ and $\mathsf{fsh}[k](t)$ we obtain $k \notin \mathsf{keys}((t; b).R) = \mathsf{keys}(P)$ as required. Since $l \in \mathsf{keys}(R)$ we obtain $l \in \mathsf{keys}((t; b).R) = \mathsf{keys}(P)$ as required. It is clear by the rules $\mathsf{act2}$ and $\mathsf{concert\ act}$ that since $l \in \mathsf{keys}(R)$ the key l is not among the keys in t in $(t; b).R$. We now calculate $\mathsf{keys}(P')$: $\mathsf{keys}(P') = \mathsf{keys}((t; b).R') = \mathsf{k}(t) \cup \mathsf{k}(b) \cup \mathsf{keys}(R') = \mathsf{k}(t) \cup \mathsf{k}(b) \cup (\mathsf{keys}(R) \cup \{k\}) \setminus \{l\} = ((\mathsf{k}(t) \cup \mathsf{k}(b) \cup \mathsf{keys}(R)) \cup \{k\}) \setminus \{l\} = (\mathsf{keys}((t; b).R) \cup \{k\}) \setminus \{l\} = (\mathsf{keys}(P) \cup \{k\}) \setminus \{l\}$ as required.

(c) $P \equiv R \setminus L$: The transition must be by $\mathsf{concert\ res}$ from Figure 5. Assume P is consistent and $P \xrightarrow{\{\mu[k], \underline{\nu}[l]\}} P'$. Since $P \xrightarrow{\{\mu[k], \underline{\nu}[l]\}} P'$ by rule $\mathsf{concert\ res}$ we obtain $R \xrightarrow{\{\mu[k], \underline{\nu}[l]\}} R'$ and $\mu, \nu \notin L \cup \overline{(L)}$. Since $k \notin \mathsf{keys}(R)$ by the inductive hypothesis it follows that $k \notin \mathsf{keys}(R \setminus L)$ since restriction does not change the keys of the process by definition of keys. Since $l \in \mathsf{keys}(R)$ by the inductive hypothesis it follows that $l \in \mathsf{keys}(R \setminus L)$. Also $\mathsf{keys}(P) = \mathsf{keys}(R)$ and $\mathsf{keys}(P') = \mathsf{keys}(R')$ according to the definition of keys. Since by the inductive hypothesis $\mathsf{keys}(R') = \mathsf{keys}(R) \cup \{k\}$ we can calculate $\mathsf{keys}(P')$: $\mathsf{keys}(P') = \mathsf{keys}(R' \setminus L) = \mathsf{keys}(R') = (\mathsf{keys}(R) \cup \{k\}) \setminus \{l\} = (\mathsf{keys}(P) \cup \{k\}) \setminus \{l\}$ as required.

(d) There are no concerted transitions for $P \equiv (s; c).R$ and $P \equiv S$ with $S \overset{def}{=} R$.

Proposition 6. If $t_1 \equiv P \xrightarrow{\underline{\mu}[k]} P'$ and $t_2 \equiv P \xrightarrow{\nu[l]} P''$, then either t_1 and t_2 are concurrent or $k \in cau(P'', l)$.

PROOF. By induction on the depth of inference trees for transition of P.

1. Base case: Obvious.
2. Inductive hypothesis: We assume that for all subprocesses R of P and all $\underline{c}[m], d[n]$, if R is a consistent process, $t_1 \equiv R \xrightarrow{\underline{c}[m]} R'$ and $t_2 \equiv R \xrightarrow{d[n]} R''$ then either t_1 and t_2 are concurrent or $m \in cau(P'', n)$.
3. Induction step: We show Proposition 6 for P. By Definition 5 there is either an M such that $M \neq P$, $P' \xrightarrow{\nu[l]} M$ and $P'' \xrightarrow{\underline{\mu}[k]} M$, or $k \in cau(P'', l)$ holds. We consider cases based on the structure of P:

(a) $P \equiv (s;b).R$ with s containing at least one past action and at least one fresh action. The transitions t_1 and t_2 are by rev act1 respectively act1. Since there is nothing in the rules giving any of them precedence, the transitions are concurrent as required. With s' being the sequence obtained from s by removing actions ν and $\mu[k]$ we have $t_1 \equiv (s', \nu, \mu[k]; b).R \xrightarrow{\mu[k]} (s', \nu, \mu, ; b).R$ and $t_2 \equiv (s', \nu, \mu[k]; b).R \xrightarrow{\nu[l]} (s', \nu[l], \mu[k]; b).R$. We now deduce $M \equiv (s', \nu[l], \mu; b).R$, and the properties required for ν and $\mu[k]$ being concurrent hold: namely $(s', \nu, \mu, ; b).R \xrightarrow{\nu[l]} M$ and $(s', \nu[l], \mu[k]; b).R \xrightarrow{\mu[k]} M$.

(b) $P \equiv (s;b).R$ where s contains only fresh actions: We cannot have the required transition t_2. Hence Proposition 6 is vacuously valid.

(c) $P \equiv (s;b).R$ where s contains only past actions: There is $R \xrightarrow{\nu[l]} R'$, and a $\mu[k] \in s$ so that $t_2 \equiv (s;b).R \xrightarrow{\nu[l]} (s;b).R'$ and $t_1 \equiv (s;b).R \xrightarrow{\mu[k]} (s';b).R$ for some s'. These transitions are not concurrent since doing any action in R prevents undoing of actions in s and vice versa. Hence $k \in cau(P'', l)$ holds with $P'' \equiv (s;b).R'$.

(d) $P \equiv Q \mid R$: There are three cases:

i. t_1 by rule rev par and t_2 by rule par. There are two sub-cases:

A. Transitions in the same subprocess: Assume without loss of generality $Q \xrightarrow{\mu[k]} Q'$ and $Q \xrightarrow{\nu[l]} Q''$. By the inductive hypothesis these transitions are either concurrent or $k \in cau(Q'', l)$.
- concurrent transitions: By the inductive hypothesis there is an N so that $Q' \xrightarrow{\nu[l]} N$ and $Q'' \xrightarrow{mu[k]} N$. We can conclude by using rule rev par that $Q' \mid R \xrightarrow{\nu[l]} N \mid R$ and $Q'' \mid R \xrightarrow{\mu[k]} N \mid R$. Letting $M \equiv N \mid R$ we get the required result.
- $k \in cau(Q'', l)$: Since $cau(P \mid Q, k) = cau(P, k) \cup cau(Q, k)$ according to Definition 7 we get $cau(Q'' \mid R, l) = cau(Q'', l) \cup cau(R, l)$. If $k \in cau(Q'', l)$ then $k \in cau(Q'', l) \cup cau(R, l)$ must be true as well and, hence, $k \in cau(Q'' \mid R)$. Since $P'' \equiv Q'' \mid R$ it follows that $k \in cau(P'', l)$ as required.

B. Transitions in different subprocesses: Assume without loss of generality that $Q \xrightarrow{\mu[k]} Q'$ and $R \xrightarrow{\nu[l]} R'$. These transitions are concurrent. By rule rev par $Q \mid R \xrightarrow{\mu[k]} Q' \mid R \xrightarrow{\nu[l]} Q' \mid R'$ and $Q \mid R \xrightarrow{\nu[l]} Q \mid R' \xrightarrow{\mu[k]} Q' \mid R'$ are valid. These form the diamond required for concurrent transitions with $M \equiv Q' \mid R'$.

ii. $P \xrightarrow{\mu[k]} P'$ by rule rev com and $P \xrightarrow{\nu[l]} P''$ by rule par: Without loss of generality this covers all cases with one par and one rev com or one rev par and one com transition. We assume that $P \xrightarrow{\mu[k]} P'$

happens by rule rev com, where $\gamma(\mu_1, \mu_2) = \mu$, and that $P \xrightarrow{\nu[l]} P''$ happens by rule par. We also assume that ν happens in Q, so that $Q \xrightarrow{\nu[l]} Q'$, and that $Q \xrightarrow{\mu_1[k]} Q''$ and $R \xrightarrow{\mu_2[k]} R'$. We know that $\mathsf{fsh}[l](R)$ because of the preconditions of the par rule and that $l \neq k$ because $\mathsf{fsh}[l](R)$ and $R \xrightarrow{\mu_2[k]} R''$, a transition which could not happen if $l = k$, since according to Proposition 1.2 a key cannot be fresh for a reverse transition to happen with this key. By the inductive hypothesis transition $Q \xrightarrow{\nu[l]} Q'$ and $Q \xrightarrow{\mu_1[k]} Q''$ are either concurrent or $k \in cau(Q', l)$.

A. concurrent transitions: By the inductive hypothesis there is an N so that $Q' \xrightarrow{\mu_1[k]} N$ and $Q'' \xrightarrow{\nu[l]} N$. Using the rev com rule we can deduce that $P \xrightarrow{\mu[k]} Q'' \mid R'$, $P \xrightarrow{\nu[l]} Q' \mid R$, $Q'' \mid R' \xrightarrow{\nu[l]} N \mid R'$ and $Q' \mid R \xrightarrow{\mu[k]} N \mid R'$. Letting $M \equiv N \mid R'$ we obtain the result.

B. $k \in cau(Q', l)$: $k \in cau(Q', l)$ implies $k \in (cau(Q', l) \cup cau(R, l))$ according to Definition 7, which is $k \in cau(Q' \mid R, l)$. Since $P'' = Q' \mid R$ we have $k \in cau(P'', l)$ as required.

iii. $P \xrightarrow{\mu[k]} P'$ by rev com and $P \xrightarrow{\nu[l]} P''$ by com: Without loss of generality this covers all cases with one com and one rev com transition. We assume that $\gamma(\mu_1, \mu_2) = \mu$ and $\gamma(\nu_1, \nu_2) = \nu$. Also $Q \xrightarrow{\mu_1[k]} Q'$, $Q \xrightarrow{\nu_1[l]} Q''$, $R \xrightarrow{\mu_2[k]} R'$ and $R \xrightarrow{\nu_2[l]} R''$. By the inductive hypothesis the transitions μ_1 and ν_1 must be either concurrent or $k \in cau(Q'', l)$. For transitions μ_2 and ν_2 the same holds. So we distinguish three cases:

A. two concurrent transitions: Since $Q \xrightarrow{\mu_1[k]} Q'$ and $Q \xrightarrow{\nu_1[l]} Q''$ by the inductive hypothesis it follows that there is an N so that $Q' \xrightarrow{\nu_1[l]} N$, and $Q'' \xrightarrow{\mu_1[k]} N$ and since $R \xrightarrow{\mu_2[k]} R'$ and $R \xrightarrow{\nu_2[l]} R''$ there is an N' so that $R' \xrightarrow{\nu_2[l]} N'$ and $R'' \xrightarrow{\mu_2[k]} N'$. By rev par and par it follows that $P \xrightarrow{\mu[k]} Q' \mid R' \xrightarrow{\nu[l]} N \mid N'$ and $P \xrightarrow{\nu[l]} Q'' \mid R'' \xrightarrow{\mu[k]} N \mid N'$. Letting $M \equiv N \mid N'$ gives the result.

B. $k \in cau(Q'', l)$ and $k \in cau(R'', l)$: Since $P'' \equiv Q'' \mid R''$ we can calculate $cau(P'', l) = cau(Q'' \mid R'', l) = cau(Q'', l) \cup cau(R'', l)$. Since $k \in cau(Q'', l)$ and $k \in cau(R'', l)$ it follows that $k \in cau(Q'', l) \cup cau(R'', l)$ and that $k \in cau(P'', l)$ as required.

C. $k \in cau(Q'', l)$, and $R \xrightarrow{\mu_2[k]} R'$ and $R \xrightarrow{\nu_2[l]} R''$ are concurrent: Since $P'' \equiv Q'' \mid R''$ we get $cau(P'', l) = cau(Q'' \mid R'', l) = cau(Q'', l) \cup cau(R'', l)$. Since $k \in cau(Q'', l)$ it follows that

$k \in cau(Q'', l) \cup cau(R'', l)$ and that $k \in cau(P'', l)$ as required. This also applies when $k \in cau(R'', l)$, so $Q \xrightarrow{\mu_1[k]} Q'$ and $Q \xrightarrow{\nu_1[l]} Q''$ are concurrent.

(e) $P \equiv R \setminus L$: The transitions $R \setminus L \xrightarrow{\underline{\mu}[k]} R' \setminus L$ and $R \setminus L \xrightarrow{\nu[l]} R'' \setminus L$ are by rule rev res in Figure 4 respectively res in Figure 3. We assume without loss of generality that $R \xrightarrow{\underline{\mu}[k]} R'$, $R \xrightarrow{\nu[l]} R''$ and $\mu, \nu \notin L$. By the inductive hypothesis $R \xrightarrow{\underline{\mu}[k]} R'$ and $R \xrightarrow{\nu[l]} R''$ are either concurrent or $k \in cau(R'', l)$.

   i. concurrent transitions: By the inductive hypothesis there is an N so that $R' \xrightarrow{\underline{\mu}[l]} N$ and $R'' \xrightarrow{\nu[k]} N$. Consider $M \equiv N \setminus L$. Since $\mu, \nu \notin L$ by rule rev res respectively res we deduce $P \xrightarrow{\underline{\mu}[k]} R' \setminus L \xrightarrow{\nu[l]} M$ and $P \xrightarrow{\nu[l]} R'' \setminus L \xrightarrow{\underline{\mu}[k]} M$.

   ii. $k \in cau(R'', l)$: Since $cau(P \setminus L, k) = cau(P, k)$ according to Definition 7 we calculate $cau(R'' \setminus L, l) = cau(R'', l)$. If $k \in cau(R'', l)$ it follows that $k \in cau(R'' \setminus L, l)$ as required.

(f) $P \equiv S$ with $S \overset{def}{=} R$: similar to the $P \equiv R \setminus L$ case.

Proposition 7 (Rearrangement). *If σ is a trace then there exist forward traces σ_1 and σ_2 such that $\sigma \asymp \sigma_1^\bullet; \sigma_2$.*

Proof. We give a constructive proof. We show that any trace can be transformed to the form required by Proposition7. Any trace must be either σ, $\sigma^\bullet$ or $\sigma_1^\bullet; \sigma_2$ or it must be of the form $\sigma_1^\bullet; \sigma_2^*; t_1; t_2^\bullet; \sigma_3$ where σ, σ_1 and σ_2 are forward traces and σ_3 is composed of any number of forward and reverse transitions. In other words this means that we can identify the earliest pair of forward-reverse transitions. The instructions for the transformation to the required form are in the algorithm in Figure 16.

We show that the algorithm terminates in all cases with the required result. Executing the inner while loop (lines 9 to 13) once decreases the length of the resulting σ_2 by 1 and increases the length of the resulting σ_3 by 1.

After the inner while loop terminates we have $length(\sigma_2') = length(\sigma_2)$. The trace $\sigma_1^\bullet; t^\bullet$ forms a new reverse only trace whose length is increased by 1 compared to $\sigma_1^\bullet$.

Executing the outer while loop once decreases the length of the sequence $t_2^\bullet; \sigma_3$, which is the "unprocessed" part of $\sigma_1^\bullet; \sigma_2; t_1; t_2^\bullet; \sigma_3$, and changes the structure of σ_{input} so it is no longer as required by Proposition 7. The sequence $\sigma_1^\bullet; \sigma_2; t_1$ is of the form required by Proposition 7 and its length is increased by the outer while loop. This is because at the end of the outer while loop we have $\sigma_1^\bullet; t^\bullet; \sigma_2'; \sigma_3$. The part $\sigma_1^\bullet; t^\bullet; \sigma_2'$ of this trace is in the correct form and its length is increased by one, since $t^\bullet$ is added, $\sigma_1^\bullet$ is unchanged, and $length(\sigma_2') = length(\sigma_2)$. The sequence σ_3 is the "unprocessed" part, which has been shortened by one transition, in comparison to $t_2^\bullet; \sigma_3$ since σ_3 is unchanged.

Hence the algorithm terminates once σ_3 has been completely processed and the final σ_{input} has the required form with σ_3 being empty.

Proposition 8 (Shortening). *If σ_1, σ_2 are coinitial and cofinal traces, with σ_2 forward, then there exists a forward trace σ_1' of length at most that of σ_1 such that $\sigma_1' \asymp \sigma_2$.*

Proof. By induction on the length of σ_1. If σ_1 is a forward trace then the proposition holds. If not, then by Proposition 7 we assume σ_1 to be $\sigma'^{\bullet};\sigma$ for some forward sequences σ and σ'. There is only one sub-trace $t_1^{\bullet};t_2$ in $\sigma'^{\bullet};\sigma$ where the first transition $t_1^{\bullet}$ is reverse and the second transition t_2 is forward. We assume $t_1^{\bullet} \equiv P \xrightarrow{\underline{\mu}[k]} P'$ and $t_2 \equiv P' \xrightarrow{\nu[l]} P''$. There is a transition $t' \equiv R \xrightarrow{\mu[k]} R'$ in σ_1 for some R, R', otherwise σ_1 could not be cofinal with σ_2. Proposition 1.3 implies that there is a transition $t_1 \equiv P' \xrightarrow{\mu[k]} P$. Transitions t_1 and t_2 are either concurrent, in conflict, or $l \in cau(P, k)$ or $k \in cau(P'', l)$. The possibility of t_1 and t_2 being in conflict is excluded since we have t_1 and t_2 in a valid trace, namely $P' \xrightarrow{\nu[l]} P'' \rightarrow^* R \xrightarrow{\mu[k]} R'$. Also $k \in cau(P'', l)$ is impossible since we perform $\nu[l]$ before $\mu[k]$ in the trace $P \xrightarrow{\underline{\mu}[k]} P' \xrightarrow{\nu[l]} P'' \rightarrow^* R \xrightarrow{\mu[k]} R'$. Finally, $l \in cau(P, k)$ cannot hold because otherwise, since $P' \xrightarrow{\mu[k]} P$, some $\alpha[l]$ transition must have happened in P' before $P' \xrightarrow{\mu[k]} P$. This contradicts $P' \xrightarrow{\nu[l]} P''$ (key l is not fresh in P' due to the $\alpha[l]$ action, hence $P' \xrightarrow{\nu[l]} P'''$ is not possible for any P'''). Hence $l \notin cau(P, k)$ and, overall, the only possibility is that t_1 and t_2 are concurrent.

The transitions $t_1^{\bullet}$ and t_2 form a diamond with two transitions $t_3 \equiv P \xrightarrow{\nu[l]} P'''$ and $t_4^{\bullet} \equiv P''' \xrightarrow{\underline{\mu}[k]} P''$ for some P''' according to Proposition 2. The trace $t_3;t_4^{\bullet}$ can replace $t_1^{\bullet};t_2$ in $\sigma'^{\bullet};\sigma$ since the traces are coinitial and cofinal. We can repeat this process of "moving" an equivalent version of $t_4^{\bullet}$ to the right (by following the steps described above) until the resulting $t^{\bullet}$ (with the label $\underline{\mu}[k]$) is directly to the left of $t' \equiv R \xrightarrow{\mu[k]} R'$. These transitions are then $Q \xrightarrow{\underline{\mu}[k]} R \xrightarrow{\mu[k]} R'$ for some Q (where $Q = R$). Using Definition 8 we can remove them. The resulting trace is shorter than σ_1 and we can repeat the process until the trace is forwards only.

```
1   Let $\sigma_{input}$ be our trace
2   while ($\sigma_{input} \neq \sigma \wedge \sigma_{input} \neq \sigma^{\bullet} \wedge \sigma_{input} \neq \sigma_1^{\bullet};\sigma_2$ for all forward traces
    $\sigma, \sigma_1, \sigma_2$)
3           $\sigma_{input}$ is of the form $\sigma_1^{\bullet};\sigma_2;t_1;t_2^{\bullet};\sigma_3$ for some (possibly new)
            $\sigma_1, \sigma_2, t_1, t_2, \sigma_3$, where $\sigma_1, \sigma_2, \sigma_3$ are forward traces, $t_1$, $t_2$ are for-
            ward transitions (any of $\sigma_1, \sigma_2, \sigma_3$ could be empty sequences $\epsilon$)
            Let $length(\sigma_1) = n$, $length(\sigma_2) = k$, $length(\sigma_3) = l$. Distance
            from start to the pair $t_1;t_2^{\bullet}$ is $n + k$
4           Let $t_1 \equiv P \xrightarrow{\mu[m]} P'$, $t_2^{\bullet} \equiv P' \xrightarrow{\underline{\nu[n]}} P''$ (so $t_2 \equiv P'' \xrightarrow{\nu[n]} P'$)
5           if ($\mu[m] = \nu[n]$) then
6                   Since $\mu[m] = \nu[k]$ we get $P \equiv P''$ and, by Definition 8,
                    transitions $t_1;t_2^{\bullet}$ are replaced by $\epsilon$ in $\sigma_{input}$
7                   $\sigma_{input}$ is now $\sigma_1^{\bullet};\sigma_2;\sigma_3$
8           else
9                   while ($\sigma_2 \neq \epsilon$)
10                          $\sigma_{input}$ is $\sigma_1^{\bullet};\sigma_2;t_1;t_2^{\bullet};\sigma_3$ for some (possibly new) $\sigma_2$,
                            $t_1$, $t_2$ and $\sigma_3$, with $\sigma_1$ and $\sigma_2$ being forward traces
                            and $t_1$ and $t_2$ being forward transitions
11                          According to Proposition 1.3 there must be a transi-
                            tion $t_1^{\bullet} \equiv P' \xrightarrow{\underline{\mu[m]}} P$. $t_1^{\bullet}$ and $t_2^{\bullet}$ form a diamond with
                            two transitions $t_3^{\bullet} \equiv P \xrightarrow{\underline{\nu[n]}} M$ and $t_4^{\bullet} \equiv P'' \xrightarrow{\underline{\mu[m]}} M$
                            for some $M$ according to Proposition 2. Accord-
                            ing to Proposition 1.3 there must be a transition
                            $t_4 \equiv M \xrightarrow{\mu[m]} P''$. The trace $t_3^{\bullet};t_4$ replaces $t_1;t_2^{\bullet}$
                            in $\sigma_{input}$ since the traces are coinitial and cofinal
12                          $\sigma_{input}$ is now $\sigma_1^{\bullet};\sigma_2;t_3^{\bullet};t_4;\sigma_3$
13                  end while (at this point $\sigma_2$ is $\epsilon$)
14                  $\sigma_{input}$ is now $\sigma_1^{\bullet};t^{\bullet};\sigma_2';\sigma_3$ for some $t, \sigma_2'$ where $\sigma_2'$ is for-
                    wards only
15          end if
16  end while (at this point $\sigma_{input} = \sigma \vee \sigma_{input} = \sigma^{\bullet} \vee \sigma_{input} = \sigma_1^{\bullet};\sigma_2$, for
    some forward traces $\sigma$, $\sigma_1$ and $\sigma_2$)
```

Figure 16: Rearrangement algorithm.