Hernán Melgratti, Claudio Antares Mezzina, G. Michele Pinna, *A Petri net view of covalent bonds* in *Theoretical Computer* Science, Volume 908 (2022), Pages 89 - 119.

The publisher's version is available at:
https://doi.org/10.1016/j.tcs.2022.01.013

A Petri Net view of Covalent Bonds

Hernán Melgratti

ICC - Universidad de Buenos Aires - Conicet, Argentina

Claudio Antares Mezzina

Dipartimento di Scienze Pure e Applicate, Università di Urbino, Italy

G. Michele Pinna*

Dipartimento di Matematica e Informatica, Università di Cagliari, Italy

Abstract

In nature and chemistry the interactions among elements often form *bonds* and among them *covalent* bonds are relevant, involving the sharing of electrons. Another relevant and compelling facet of calculi modelling covalent bonds is that certain steps in reactions are the result of *concerting* different activities, possibly *reversing* some of them. Starting from a calculus for covalent bonds, we investigate on how it can be done in a compositional fashion and how it can be encoded in suitable Petri nets. The outcome gives us a compositional covalent bond calculus and a truly distributed implementation. On these results it is possible to build a behavioural equivalence among terms.

Keywords: Petri Nets, Bonds, Calculus of Covalent Bonding, Natural Computing

1. Introduction

The study of biochemical reactions as computational processes started with the seminal paper of the *Chemical Abstract Machine* (CHAM) [1]. In the sequel, several formal languages tailored to the modelling of biochemical reactions have been proposed in the literature [2, 3, 4, 5, 6, 7, 8, 9]. Typically, those languages allow for the description of processes concerning the formation of chemical bonds, i.e., the linkage between atoms, molecules and ions that produce chemical compounds. As an example, consider the catalysed reaction for the hydration of formaldehyde in water depicted in Figure 1 in which two molecules A and B bond together via a catalyser C. Firstly, each of the molecules A and B bonds with the catalyser C (depicted as arcs labelled by c and d). After that,

* *Corresponding author.* Dipartimento di Matematica e Informatica, Università di Cagliari, via Ospedale 72, 09124 Cagliari (Italy), e-mail: gmpinna@unica.it

A and B bond together (arc labelled by q) and, at the same, the existing bond c is broken. Finally, the bond d between B and C is broken; which releases C for serving as catalyser for another reaction.

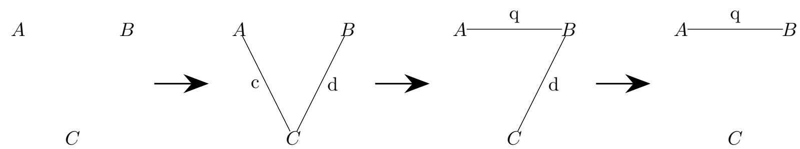

Figure 1: A catalytic reaction (borrowed from [8])

In computational terms, the transformations in fig. 1 can be thought of as different nature: the first one creates bonds, the last one undoes a previously created bond, and the second one mixes the creation of a new bond with the undoing of a previously created bond. Hence, the underlying computational model can be seen as a *reversible* model, i.e., one that features two flows of execution: a *forward* (normal) one and a *backward* one that undoes the effects of previously executed actions. Some models for biochemical processes are built upon this observation [10, 8]. In this paper we focus on the *Calculus of Covalent Bonding* (CCB) [8], which embodies a reversible model with very distinctive features. Differently from most of the reversible process calculi in the literature [11, 12, 13, 14], a computation step in CCB may combine forward and backward flows: Thanks to the so called *concerted actions*, a single reduction step in CCB can simultaneously create a bond and break an existing one, akin to the second transformation in fig. 1. Moreover, CCB enjoys some form of *out-of-causal* order reversibility [15], i.e., a process may exploit backward computation to reach states that cannot be reached by relying only on forward computation. As an example, consider the scenario introduced in fig. 1 and assume that A and B cannot bond together without the participation of C. Consequently, the final state in fig. 1 is only reachable because of the undoing of some previous forward steps (i.e., the breaking of the bonds c and d created in first step).

Interestingly, the distinguishing features of CCB are obtained via some not-so-standard ingredients in the operational semantics of the language; namely, the *lookahead* mechanism for concerted actions and the need of *assigning priority* to the reductions that exchange bonds within prefixes. Consequently, it becomes natural to ask whether the essential features of CCB can be explained by recasting to standard computational models. In this paper we provide an affirmative answer to that question by exploiting some recent results that show that Petri nets with inhibitor arcs can provide a unifying presentation for different reversible models [16, 17, 18]. As a by-product, our study sheds lights about some interesting aspects in the definition of CCB as, for instance, the fact that its semantics is non-compositional, and that it allows for the formation of bonds among an unbounded number of components.

Concretely, we introduce a variant of CCB that retains the intuitions, motivations and most of the capabilities of the original calculus, but has compositional semantics and avoids lookahead and priorities. For technical simplicity, we restrict our analysis to the fragment of binary bonds, i.e., we consider synchronisation algebras that only allow for bonds formed between exactly two components. Despite this choice reduces the synchronisation capabilities of the calculus, we remark that all case studies in [8] lay within this fragment.

Then, we show that each term of the calculus can be encoded into a behavioural equivalent (i.e., bisimilar) Petri net. Alike previous approaches aimed at expressing reversibility in Petri nets, our encoding needs to ensure that each marking of a net conveys sufficient information to enable exactly those admissible forward and backward computations, which is particularly challenging when dealing with out-of-causal order reversibility [19]. Here, we rely on [17] to accommodate out-of-causal order reversibility by representing causality in terms of inhibitor arcs. Another key ingredient of our encoding concerns the static representation of bonds, which are dynamically generated in `CCB`. We first note that bonds play the role of the communication keys present in some reversible calculi [11]. As done in [18], we represent dynamically generated communication keys with statically designated places of a net. This allows us to show that a proper subclass of Petri nets with inhibitor arcs, dubbed *covalent bond nets*, is expressive enough for encoding `CCB` terms.

Besides showing that the essential features of `CCB` can be obtained without appealing to mechanisms such as the fresh generation of keys, lookahead of computation steps or prioritised reductions, our encoding paves the way for the application of consolidated analysis techniques that have been developed for Petri nets for decades.

Our contribution is structured as follows: We introduce `CCB` and illustrates its applicability in Section 2; we revise the original semantics of `CCB` in Section 3 in order to have compositional semantics. After recalling the standard notion of Petri nets with inhibitor arcs (Section 4), we introduce *covalent bond* nets (cBN) (Section 5), which are a proper subclass of Petri nets with inhibitor arcs that corresponds to `CCB` terms. The encoding of `CCB` into cBN is presented in Section 6. The and finally state our main result in terms of an operational correspondence between the encoding and the compositional `CCB`.

Acknowledgment

This is our contribution to the Festschrift that celebrates Gabriel Ciobanu's 65th birthday. We tried to gather together just three, among many, topics in which Gabriel has been a pioneer, a prolific author and a great curious: models for biological systems (with a particular perspective on membrane systems) [20, 21, 22], reversibility [23, 24, 25, 26, 27] and Petri nets [28, 29, 30] just to cite a few of them.

Besides being an influential scientist, Gabriel has also been a mentor and a source of inspiration for many researchers. His curiosity and dedication to research will inspire generations. We hope it inspired us. Therefore we wish to thank Gabriel for being a friend and a guide for all of us.

Notation

We recall some useful notions that will be used throughout this paper. $\mathbb{N}$ denotes the set of natural numbers. A *multiset* over a set A is a function $m : A \to \mathbb{N}$. A multiset m will be often written by listing its elements separated by a comma, for instance the multiset m such that $m(a) = 2, m(b) = 1$ and $m(x) = 0$ for any other $x \in A$ different from a or b will be a, a, b. We assume multisets to be equipped with the usual operations of union (+) and difference (−), and write $m \subseteq m'$ if $m(a) \leq m'(a)$ for all $a \in A$. We often confuse a multiset m with the set $\{a \in A \mid m(a) \neq 0\}$ when $\forall a \in A. m(a) \leq 1$. In such cases, we write $a \in m$ instead of $m(a) \neq 0$, and $m \subseteq A$ if $m(a) = 1$ implies $a \in A$. Furthermore, we will use standard operations on sets, such as $\cap$, $\cup$ or $\setminus$. The set of all multisets over A is denoted by μA. The multiset m where for each $a \in A$ it holds that $m(a) = 0$ is the empty multiset and, with abuse of notation, we write $\emptyset$ for it. As usual, given $a \in A$, we write a also for the singleton $\{a\}$.

2. The Calculus of Covalent Bonding

In this section we report the finite part of the *Calculus of Covalent Bonding* [31, 8], `CCB` for short. `CCB` is a reversible, concurrent calculus tailored to the modeling of biochemical reactions. ~~As customary, concurrent processes can interact by performing complementary actions.~~ The set of `CCB` actions $\mathcal{A} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots\}$ is partitioned into *strong* and *weak actions* that respectively belongs to $\mathcal{S}$ and $\mathcal{W}$. In what follows we shall write some actions in *italics*, e.g., $a, b, c, \ldots$, to emphasise that they are weak. The syntax of finite `CCB` is in Figure 2.

$$\alpha ::= (r; b)$$
$$\mathtt{S} ::= \alpha.\mathtt{S} \mid \mathbf{0}$$
$$\mathtt{P} ::= \mathtt{S} \mid \mathtt{P} \parallel \mathtt{P} \mid \mathtt{P} \setminus L$$

Figure 2: `CCB` process syntax

All operators but the shape of prefixes α are standard. As usual, $\mathbf{0}$ represents the idle process, $\alpha.\mathtt{S}$ stands for a process prefixed by α, and $\mathtt{P} \parallel \mathtt{Q}$ represents the *parallel* composition of `P` and `Q`, i.e., the processes `P` and `Q` can either execute independently or synchronise with each other during computation. The *hiding* operator $\mathtt{P} \setminus L$ controls the scope of actions in a process, i.e., $\mathtt{P} \setminus L$ behaves as `P` except for the fact that it cannot perform any action belonging to the set of actions $L \subseteq \mathcal{A}$.

A prefix $(r; b)$ consists of a finite multiset of actions r and a weak action b, which are respectively called *strong* and *weak (part of the) prefix*. Intuitively, the optional weak action b becomes enabled only after all actions in r have been performed; the order in which actions in s are executed is irrelevant. It is assumed that r is not empty and at most one of its actions can be weak, i.e., $r \in \mathbb{N}^{\mathcal{A}} \cup \mathcal{W} - \emptyset$ and for all $b, c \in \mathcal{W}$ if $r(b) > 0$ and $r(c) > 0$ then $b = c$ and $r(b) = 1$. Additionally, b can sometimes be omitted, and we will write prefixes simply as $(r).\mathtt{S}$.

Example 1 (Weak and strong prefixes). *We now illustrate strong and weak prefixes. Suppose* $\mathtt{a}, \mathtt{b} \in \mathcal{S}$ *to be two strong actions and* $c \in \mathcal{W}$ *a weak one. Then, the prefix* $(\mathtt{a}, \mathtt{b}; c)$ *is such that the multiset* $\mathtt{a}, \mathtt{b}$ *is the strong (part of the) prefix and the action* c *is the weak (part of the) prefix. Intuitively, the actions in the strong prefix* $\mathtt{a}, \mathtt{b}$ *can be performed in any order, while the weak prefix* c *can occur only after* $\mathtt{a}$ *and* $\mathtt{b}$*. The prefix* $(\mathtt{a}, \mathtt{b}, c)$ *has instead the multiset* $\mathtt{a}, \mathtt{b}, c$ *as its strong part and its weak part has been omitted. In this case, the prefix does not impose any order on the execution of the actions: the weak action* c *in the strong prefix can be performed even before the strong actions* $\mathtt{a}$ *and* $\mathtt{b}$*. The distinction between weak and strong actions concerns to the capability of moving bonds between actions, which will be made clear when presentingt the semantics of the language.*

A distinctive feature of the introduced prefixes $(r; b)$ is that the execution of the weak prefix b forces the process to undo some previously executed action.

The reversibility mechanism of CCB is modeled via communication keys [11]. Let $\mathcal{K} = \{k, l, m, n, \ldots\}$ be the set of communication keys. Then, the set of performed actions is defined as $\mathcal{A} \times \mathcal{K}$. Hereafter, we write $\mathtt{a}[k]$ for the performed action $(\mathtt{a}, k)$.

The runtime syntax of the calculus is obtained by extending prefixes as follows

$$\alpha ::= \ldots \mid (s; \beta)$$

with $s \in \mathbb{N}^{(\mathcal{A} \times \mathcal{K}) \cup \mathcal{A}}$. We let $t, t', \ldots$ range over $\mathbb{N}^{\mathcal{A} \times \mathcal{K}}$; and β, β' to range over $\mathbb{N}^{(\mathcal{W} \times \mathcal{K}) \cup \mathcal{W}}$.

Before defining the set of free names and bound names of a process, we need to define the set of names of a multiset possibly marked with keys. We indicate such a set with $n(s)$ and we define it as follows:

$$n(\mathtt{a}, s) = n(\mathtt{a}[k], s) = \{\mathtt{a}\} \cup n(s) \qquad\qquad n(\emptyset) = \emptyset$$

The set of *free names* and *bound names* of a process P, denoted respectively as $\mathit{fn}(\mathtt{P})$ and $\mathit{bn}(\mathtt{P})$, are inductively defined as follows

$$\begin{aligned} \mathit{fn}((s;\beta).\mathtt{P}) &= n(s,\beta) \cup \mathit{fn}(\mathtt{P}) & \qquad \mathit{fn}(\mathtt{P} \parallel \mathtt{Q}) &= \mathit{fn}(\mathtt{P}) \cup \mathit{fn}(\mathtt{Q}) \\ \mathit{fn}(\mathtt{P} \setminus L) &= \mathit{fn}(\mathtt{P}) \setminus L & \mathit{fn}(\mathbf{0}) &= \emptyset \end{aligned}$$

$$\begin{aligned} \mathit{bn}((s;\beta).\mathtt{P}) &= \mathit{bn}(\mathtt{P}) & \qquad \mathit{bn}(\mathtt{P} \parallel \mathtt{Q}) &= \mathit{bn}(\mathtt{P}) \cup \mathit{bn}(\mathtt{Q}) \\ \mathit{bn}(\mathtt{P} \setminus L) &= \mathit{bn}(\mathtt{P}) \cup L & \mathit{bn}(\mathbf{0}) &= \emptyset \end{aligned}$$

The following notions are instrumental to the definition of the operational semantics of the calculus.

Definition 1. *The multiset of keys of a multiset of actions s, written $key(s)$ is defined by*

$$key(\emptyset) = \emptyset \qquad key(\mathtt{a}[k], s') = \{k\} + key(s') \qquad key(\mathtt{a}, s') = key(s')$$

The set of keys of a process $\mathtt{P}$*, written* $key(\mathtt{P})$*, is inductively defined as follows:*

$$\begin{array}{rclcrcl} key((s;\beta).\mathtt{P}) &=& key(s,\beta) \cup key(\mathtt{P}) & & key(\mathbf{0}) &=& \emptyset \\ key(\mathtt{P} \parallel \mathtt{Q}) &=& key(\mathtt{P}) \cup key(\mathtt{Q}) & & key(\mathtt{P} \setminus L) &=& key(\mathtt{P}) \end{array}$$

Definition 2. *A key k is* fresh *in a process* $\mathtt{P}$ *(resp., in a multiset of actions s) written* $fresh(k, \mathtt{P})$ *(resp.* $fresh(k, s)$*) if* $k \cap key(\mathtt{P}) = \emptyset$ *(resp.,* $k \cap key(s) = \emptyset$*).*

Terms are considered up-to the structural congruence $\equiv$, i.e, the least congruence defined such that $\parallel$ is associative and commutative with identity $\mathbf{0}$ and satisfies the following rule

$$\mathtt{P} \setminus L \parallel \mathtt{Q} \equiv (\mathtt{P} \parallel \mathtt{Q}) \setminus L \qquad \text{if} \qquad fn(\mathtt{Q}) \cap L = \emptyset$$

The operational semantics of the forward flow of the computation is defined by the inference rules in Figure 3.

$$(\text{ACT1})\ \frac{std(\mathtt{S}) \qquad fresh(k, s)}{(\mathtt{a}, s; b).\mathtt{S} \xrightarrow{\mathtt{a}[k]} (\mathtt{a}[k], s; b).\mathtt{S}} \qquad (\text{ACT2})\ \frac{\mathtt{S} \xrightarrow{\mathtt{a}[k]} \mathtt{S}' \qquad fresh(k, t)}{(t; b).\mathtt{S} \xrightarrow{\mathtt{a}[k]} (t; b).\mathtt{S}'}$$

$$(\text{W-ACT1})\ \frac{std(\mathtt{S}) \qquad fresh(k, t)}{(t; b).\mathtt{S} \xrightarrow{(b)[k]} (t; b[k]).\mathtt{S}} \qquad (\text{W-ACT2})\ \frac{\mathtt{S} \xrightarrow{(b)[k]} \mathtt{S}' \qquad fresh(k, t)}{(t; a).\mathtt{S}' \xrightarrow{(b)[k]} (t; a).\mathtt{S}'}$$

$$(\text{PAR})\ \frac{\mathtt{P} \xrightarrow{\mathtt{a}[k]} \mathtt{P}' \qquad fresh(k, \mathtt{Q})}{\mathtt{P} \parallel \mathtt{Q} \xrightarrow{\mathtt{a}[k]} \mathtt{P}' \parallel \mathtt{Q}} \qquad (\text{COM})\ \frac{\mathtt{P} \xrightarrow{\mathtt{a}[k]} \mathtt{P}' \qquad \mathtt{Q} \xrightarrow{\mathtt{b}[k]} \mathtt{Q}'}{\mathtt{P} \parallel \mathtt{Q} \xrightarrow{\gamma(\mathtt{a},\mathtt{b})[k]} \mathtt{P}' \parallel \mathtt{Q}'}$$

$$(\text{RES})\ \frac{\mathtt{P} \xrightarrow{\mathtt{a}[k]} \mathtt{P}'}{\mathtt{P} \setminus L \xrightarrow{\mathtt{a}[k]} \mathtt{P}' \setminus L}\ \mathtt{a} \notin L$$

Figure 3: Forward SOS rules

Rule ACT1 mimics forward rules of reversible calculi based on communication keys: if a prefix contains an unperformed action $\mathtt{a}$, then such action is executed by assigning a fresh key k to it. The chosen key is reflected in the label $\mathtt{a}[k]$ and the continuation records that $\mathtt{a}$ has been already executed with key k. Rule ACT2 states that a computation can proceed under a prefix only when the multiset of actions in the prefix has been already executed (recall that $t \in \mathbb{N}^{\mathcal{A} \times \mathcal{K}}$). Rule PAR accounts for the execution of a forward action by one of the processes in a parallel composition; this is possible only when the assigned key k has not been used in $\mathtt{Q}$. The synchronisation of actions in concurrent processes is described by Rule COM. The communication model follows ACP [32], in which a symmetric, binary, partial function $\gamma : \mathcal{A} \times \mathcal{A} \to \mathcal{A}$ defines the allowed interactions, e.g., two actions $\mathtt{a}$ and $\mathtt{b}$ synchronise if and only if $\gamma(\mathtt{a}, \mathtt{b})$ is defined. Moreover, their interaction is described by the action $\gamma(\mathtt{a}, \mathtt{b})$. Hence, the rule COM states that two concurrent processes interact if: (i) they execute synchronisable actions according to γ and (ii) the processes agree on the communication key k assigned to the actions. Rule RES states that a process cannot perform a restricted action; this also includes the cases in which the restricted action is consequence of the synchronisation of two actions. For instance, the process $((\mathtt{a}; b).\mathbf{0} \parallel (\mathtt{a}; b).\mathbf{0}) \setminus \{\mathtt{c}\}$ is blocked if $\gamma(\mathtt{a}, \mathtt{a}) = \mathtt{c}$. The two rules W-AUX1 and W-AUX2 describe the executions of weak prefixes. Rule W-AUX1 accounts for the execution of the weak prefix b, which is allowed only if the continuation P has not been started. The execution of the action b is associated with a fresh key (as for any action in a prefix), however the parenthesis in the label (i.e., (b) instead of b) reflects that b has been executed as part of a weak prefix. Rule AUX2 accounts for the execution of a weak prefix below a prefix, which is analogous to the rule ACT2. Note that the label (b) forbids the application of rule COM, hence synchronisations involving labels like (b) are prohibited.

Note that the operational rules allow for non-binary interactions. Consider $(\mathtt{a}; b).\mathtt{S}_1 \parallel (\mathtt{a}; b).\mathtt{S}_2 \parallel (\mathtt{a}; b).\mathtt{S}_3$ and γ defined such that $\gamma(\mathtt{a}, \mathtt{a}) = \mathtt{a}$. Then, a 3-way synchronisation is possible as shown below:

$$\dfrac{\dfrac{\dfrac{std(\mathtt{S}_1)}{(\mathtt{a}; b).\mathtt{S}_1 \xrightarrow{\mathtt{a}[k]} (\mathtt{a}[k]; b).\mathtt{S}_1} \qquad \dfrac{std(\mathtt{S}_2)}{(\mathtt{a}; b).\mathtt{S}_2 \xrightarrow{\mathtt{a}[k]} (\mathtt{a}[k]; b).\mathtt{S}_2}}{(\mathtt{a}; b).\mathtt{S}_1 \parallel (\mathtt{a}; b).\mathtt{S}_2 \xrightarrow{\mathtt{a}[k]} (\mathtt{a}[k]; b).\mathtt{S}_1 \parallel (\mathtt{a}[k]; b).\mathtt{S}_2} \qquad \dfrac{std(\mathtt{S}_3)}{(\mathtt{a}; b).\mathtt{S}_3 \xrightarrow{\mathtt{a}[k]} (\mathtt{a}[k]; b).\mathtt{S}_3}}{(\mathtt{a}; b).\mathtt{S}_1 \parallel (\mathtt{a}; b).\mathtt{S}_2 \parallel (\mathtt{a}; b).\mathtt{S}_3 \xrightarrow{\mathtt{a}[k]} (\mathtt{a}[k]; b).\mathtt{S}_1 \parallel (\mathtt{a}[k]; b).\mathtt{S}_2 \parallel (\mathtt{a}[k]; b).\mathtt{S}_3}$$

For technical simplicity in the following sections, we restrict our attention to binary synchronisations, i.e., hereafter we just consider synchronisation functions γ defined such that an action in the image of γ does not enable further synchronisations.

$$\forall \mathtt{a}, \mathtt{b}, \mathtt{c} \in \mathcal{A}. \gamma(\mathtt{a}, \mathtt{b}) = \mathtt{c} \quad \textit{implies} \quad \forall \mathtt{d} \in \mathcal{A}. \gamma(\mathtt{c}, \mathtt{d}) \ \textit{undefined}$$

Despite our development does not depend on the following assumption, we will only consider synchronisation functions such that $dom(\gamma) \subseteq (\mathcal{W} \times \mathcal{W}) \cup (\mathcal{S} \times \mathcal{S})$ to forbid mixed synchronisations between strong and weak actions. It should be noted that the forward flow does not allow for synchronisation of a weak action b in a prefix of the form $(t; b)$. Such weak action will come to play in combination with the reversing mechanism of actions. To that aim, we assume that every action $\mathtt{a}$ is associated with a reversing action $\underline{\mathtt{a}}$ and write $\underline{\mathcal{A}}$ for the set of reversible actions, i.e., $\underline{\mathcal{A}} = \{\underline{\mathtt{a}} \mid \mathtt{a} \in \mathcal{A}\}$. Then, a backward/reversing step in a computation will be represented by a label of the form $\underline{\mathcal{A}} \times \mathcal{K}$, which will be written as $\underline{\mathtt{a}}[k]$ instead of $(\underline{\mathtt{a}}, k)$. Backward transitions are given by the inference rules in Figure 4. It should be notice that each forward rule in Figure 3 is paired with a backward version that undoes the computation step. The rules are self-explanatory. Differently from the original presentation of CCB, we write rule $\underline{\text{ACT1}}$ to make explicit that it can be applied regardless of whether the weak prefix has been already executed or not, i.e., we write $\beta \in \{b, b[k]\}$ instead of just b (disallowing $b[k]$ makes concerted rules inapplicable as discussed in the following).

$$(\underline{\text{ACT1}})\ \frac{std(\mathtt{S})}{(\mathtt{a}[k], s; \beta).\mathtt{S} \xrightarrow{\underline{\mathtt{a}}[k]} (\mathtt{a}, s; \beta).\mathtt{S}}\ \beta \in \{b, b[l]\} \qquad (\underline{\text{ACT2}})\ \frac{\mathtt{S} \xrightarrow{\underline{\mathtt{a}}[k]} \mathtt{S}'}{(t; b).\mathtt{S}' \xrightarrow{\underline{\mathtt{a}}[k]} (t; b).\mathtt{S}'}$$

$$(\underline{\text{PAR}})\ \frac{\mathtt{P} \xrightarrow{\underline{\mathtt{a}}[k]} \mathtt{P}' \qquad fresh(k, \mathtt{Q})}{\mathtt{P} \parallel \mathtt{Q} \xrightarrow{\underline{\mathtt{a}}[k]} \mathtt{P}' \parallel \mathtt{Q}} \qquad (\underline{\text{COM}})\ \frac{\mathtt{P} \xrightarrow{\underline{\mathtt{a}}[k]} \mathtt{P}' \qquad \mathtt{Q} \xrightarrow{\underline{\mathtt{b}}[k]} \mathtt{Q}'}{\mathtt{P} \parallel \mathtt{Q} \xrightarrow{\underline{\gamma(\mathtt{a},\mathtt{b})}[k]} \mathtt{P}' \parallel \mathtt{Q}'}$$

$$(\underline{\text{RES}})\ \frac{\mathtt{P} \xrightarrow{\underline{\mathtt{a}}[k]} \mathtt{P}'}{\mathtt{P} \setminus L \xrightarrow{\underline{\mathtt{a}}[k]} \mathtt{P}' \setminus L}\ \mathtt{a} \notin L$$

Figure 4: Backward SOS rules

The characteristic feature of CCB concerns the execution of the weak prefix, which forces the reversing of an executed action of the same prefix. Such behaviour is described by the concert rules in Figure 5. The interactions of a process through a weak prefix are given by rule Concert. On the one hand, the process P aiming at synchronising over a weak prefix should also reverse a previously executed action: the first premise accounts for the execution of the weak prefix (b) while the second one stands for the reversal of a. On the other hand, the remaining parallel component Q should execute a forward action that synchronises with b and reverse an action d that synchronises with a. Interestingly, Q can execute the matching forward action c that synchronises with b either as part of a strong prefix or as its weak prefix. For this reason, the third premise requires a reduction labelled by $\alpha \in \{c, (c)\}$. The label of a concerted action records both the forward synchronisation $\gamma(b, \mathtt{c})$ and the reverse one $\underline{\gamma(\mathtt{a}, \mathtt{d})}$. Note that the premises of the rule Concert have a *look-a-head* mechanism, meaning that the right-end side of a premise may occur in the left-hand side of the premise [33]. If we look at the premises of the Concert we have that this is the case for P′ and Q′. The remaining rules stand for the contextual cases for concerted transitions.

It should be noted that there are no contextual rules for transitions corre-

$$(\textsc{concert})\ \frac{\mathtt{P} \xrightarrow{(b)[k]} \mathtt{P}' \qquad \mathtt{P}' \xrightarrow{\underline{\mathtt{a}}[l]} \mathtt{P}'' \qquad \mathtt{Q} \xrightarrow{\alpha[k]} \mathtt{Q}' \qquad \mathtt{Q}' \xrightarrow{\underline{\mathtt{d}}[l]} \mathtt{Q}''}{\mathtt{P} \parallel \mathtt{Q} \xrightarrow{\{\gamma(b,c)[k],\underline{\gamma(\mathtt{a},\mathtt{d})}[l]\}} \mathtt{P}'' \parallel \mathtt{Q}''}\ \alpha \in \{c,(c)\}$$

$$(\textsc{concert act})\ \frac{\mathtt{P} \xrightarrow{\{c[k],\underline{\mathtt{a}}[l]\}} \mathtt{P}' \qquad \mathit{fresh}(k,t)}{(t;b).\mathtt{P} \xrightarrow{\{c[k],\underline{\mathtt{a}}[l]\}} (t;b).\mathtt{P}'}$$

$$(\textsc{concert par})\ \frac{\mathtt{P} \xrightarrow{\{c[k],\underline{\mathtt{a}}[l]\}} \mathtt{P}' \qquad \mathit{fresh}(k,\mathtt{Q}) \qquad \mathit{fresh}(l,\mathtt{Q})}{\mathtt{P} \parallel \mathtt{Q} \xrightarrow{\{c[k],\underline{\mathtt{a}}[l]\}} \mathtt{P}' \parallel \mathtt{Q}}$$

$$(\textsc{concert res})\ \frac{\mathtt{P} \xrightarrow{\{c[k],\underline{\mathtt{a}}[l]\}} \mathtt{P}'}{\mathtt{P} \setminus L \xrightarrow{\{c[k],\underline{\mathtt{a}}[l]\}} \mathtt{P}' \setminus L}\ \mathtt{a}, c \notin L$$

Figure 5: Concerted SOS rules

sponding to the execution of weak prefixes, e.g., we cannot derive $\mathtt{P} \parallel \mathtt{Q} \xrightarrow{(b)[k]} \mathtt{P}' \parallel \mathtt{Q}$ from $\mathtt{P} \xrightarrow{(b)[k]} \mathtt{P}'$. Consequently, the first two premises in rule CONCERT should be interpreted as requiring the same agent to execute the forward action (b) and the reverse one $\underline{\mathtt{a}}$. Moreover, the premises in ACT1 and $\underline{\text{ACT1}}$ imply that such actions corresponds to the same prefix. Contrastingly, the third and fourth premises in CONCERT allow the forward action to be executed by some sequential agent, while the reverse action can be be performed by other agent, as illustrated by the following derivation

$$\frac{\begin{array}{c}(\mathtt{a}[l];b).\mathbf{0} \xrightarrow{(b)[k]} (\mathtt{a}[l];b[k]).\mathbf{0} \qquad (\mathtt{a}[l];b[k]).\mathbf{0} \xrightarrow{\underline{\mathtt{a}}[l]} (\mathtt{a};b[k]).\mathbf{0} \\ (c;e).\mathbf{0} \parallel (\mathtt{d}[l];e).\mathbf{0} \xrightarrow{c[k]} (c[k];e).\mathbf{0} \parallel (\mathtt{d}[l];e).\mathbf{0} \\ (c[k];e).\mathbf{0} \parallel (\mathtt{d}[l];e).\mathbf{0} \xrightarrow{\underline{\mathtt{d}}[l]} (c[k];e).\mathbf{0} \parallel (\mathtt{d};e).\mathbf{0}\end{array}}{(\mathtt{a}[l];b).\mathbf{0} \parallel (c;e).\mathbf{0} \parallel (\mathtt{d}[l];e).\mathbf{0} \xrightarrow{\{\gamma(b,c)[k],\underline{\gamma(\mathtt{a},\mathtt{d})}[l]\}} (\mathtt{a};b[k]).\mathbf{0} \parallel (c[k];e).\mathbf{0} \parallel (\mathtt{d};e).\mathbf{0}}$$

in which the forward weak action synchronises the first and second agent while the reverse one synchronises the first and the third one. In order to be able to apply rule CONCERT, it is essential to derive $(\mathtt{a}[l];b[k]).\mathbf{0} \xrightarrow{\underline{\mathtt{a}}[l]} (\mathtt{a};b[k]).\mathbf{0}$, i.e., to be able to apply $\underline{\text{ACT1}}$ also when the weak prefix corresponds to an already executed action.

We now are in place to model the example depicted in Figure 1.

Example 2. *Let* $A = (\mathtt{c};q)$*,* $B = (\mathtt{d},q)$ *and* $C = (\mathtt{c},\mathtt{d})$ *the molecules of Figure 1. For simplicity we omit the trailing* $\mathbf{0}$*s. Supposing* $\gamma(\mathtt{c},\mathtt{c}) = \mathtt{f}$*,* $\gamma(\mathtt{d},\mathtt{d}) = \mathtt{e}$

and $\gamma(q,q) = a$*, we can derive the following synchronisations:*

$$A \parallel B \parallel C \xrightarrow{\mathtt{f}[1]} (\mathtt{c}[1];q) \parallel B \parallel (\mathtt{c}[1],\mathtt{d}) \xrightarrow{\mathtt{e}[2]} (\mathtt{c}[1];q) \parallel (\mathtt{d}[2],q) \parallel (\mathtt{c}[1],\mathtt{d}[2])$$

Thanks to a concerted action, the process $(\mathtt{c}[1];q)$ *may break (i.e., undo) the bond on c and create a bond on q. We have:*

$$(\mathtt{c}[1];q) \parallel (\mathtt{d}[2],q) \parallel (\mathtt{c}[1],d[2]) \xrightarrow{\{a[3],\underline{\mathtt{f}}[1]\}} (\mathtt{c};q[3]) \parallel (\mathtt{d}[2],q[3]) \parallel (\mathtt{c},\mathtt{d}[2])$$

We remark that the above transition cannot be derived with the original semantics of `CCB` *[8], because the rule* <u>ACT1</u> *requires the weak prefix to be a non-executed action (as discussed in the description of rule* CONCERT*).*

Finally, the bond on d can be break and C gets free, i.e.,

$$(\mathtt{c};q[3]) \parallel (\mathtt{d}[2],q[3]) \parallel (\mathtt{c},\mathtt{d}[2]) \xrightarrow{\underline{\mathtt{e}}[2]} (\mathtt{c};q[3]) \parallel (\mathtt{d},q[3]) \parallel (\mathtt{c},\mathtt{d})$$

We remark that there are no reverse rules for weak prefixes. However, a bond on a weak action is aimed at modelling a weak bond in a chemical reaction, which should be interpreted as a temporary bond that should be "passed" to a strong action as soon as possible to release the bond on the weak action, *committing* it. This is modelled in `CCB` via the rules in Figure 6. The transitions are labelled with τ, which stands for internal, silent moves. Rule Prom deals with the transfer of the key from the weak prefix to some strong action $\mathtt{a}$ that has not been executed yet. Rule Move transfers a key from a weak action in a strong prefix to some strong action $\mathtt{a}$ that has not been executed yet. Also here our presentation deviates from the original one, since Move allows to transfer a key only in absence of a weak prefix. For uniformity in the presentation, we assume all prefixes to carry its strong and weak components. Then, a process like $(s).\mathtt{P}$ in the original presentation, which does not provide a synchronisation via a weak prefix can be written simple as $((s;b).\mathtt{P}) \setminus b$ for some b not appearing in $\mathtt{P}$. According to [8], pre-congruence rules have priority over the transition rules. That is, they has to be applied as soon as possible.

As previously discussed, the operational semantics is defined up-to structural congruence of terms, formally, by the following rule:

$$\text{STRUCT}\ \frac{\mathtt{P} \equiv \mathtt{P}' \qquad \mathtt{P}' \xrightarrow{\mu} \mathtt{Q}' \qquad \mathtt{Q}' \equiv \mathtt{Q}}{\mathtt{P} \xrightarrow{\mu} \mathtt{Q}}$$

We now show an example of how rules in Figure 6 can be used.

Example 3. *Consider the process* $(\mathtt{a};b) \parallel \mathtt{a} \parallel b$ *with* $\gamma(\mathtt{a},\mathtt{a}) = \mathtt{c}$ *and* $\gamma(b,b) = d$*. We have the following reduction:*

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

Now, the weak bond on b in process $(\mathtt{a};b[2]) \parallel \mathtt{a} \parallel b[2]$ *can be promoted to a*

$$\textsc{Prom} \quad (s, \mathtt{a}; b[k]).\mathsf{S} \xrightarrow{\tau} (s, \mathtt{a}[k]; b).\mathsf{S} \quad \mathtt{a} \in \mathcal{S}$$

$$\textsc{Move} \quad (s, \mathtt{a}, b[k]; c).\mathsf{S} \xrightarrow{\tau} (s, \mathtt{a}[k], b; c).\mathsf{S} \quad \mathtt{a} \in \mathcal{S}$$

Figure 6: Pre-congruence rules for the promotion of a weak bond

strong bond; this is achieved by applying rule PROM *as follows:*

$$(\mathtt{a}; b[2]) \parallel \mathtt{a} \parallel b[2] \xrightarrow{\tau} (\mathtt{a}[2]; b) \parallel \mathtt{a} \parallel b[2]$$

As a result, the bond between $\mathtt{a}[2]$ *and* $b[2]$ *is irreversible since* $\gamma(a, b)$ *is undefined.*

Remark. The interaction between restriction in CCB and concerted rules is unexpected. Consider the following variant of the process introduced in Example 3, defined as follows

$$(((\mathtt{a}; b).\mathtt{e}.\mathtt{f} \parallel \mathtt{a}.\mathtt{e}.\mathtt{f}) \setminus \{\mathtt{e}\} \parallel (b.\mathtt{e}.\mathtt{f}) \setminus \{\mathtt{f}\}) \setminus \{\mathtt{e}, \mathtt{f}\}$$

with $\gamma(\mathtt{a}, \mathtt{a}) = \mathtt{c}$ and $\gamma(b, b) = d$ (i.e., there are no synchronisations for e and f). One would expect this process to be equivalent to the one in Example 3, since the added continuations e and f do not have any chance to be executed because there are no synchronisations for e and f; and, moreover, they are restricted names. However, the behaviour of the two processes differs, because of the concerted reductions. Note that modified version can mimic the first reduction, i.e.,

$$\begin{array}{l}(((\mathtt{a}; b).\mathtt{e}.\mathtt{f} \parallel \mathtt{a}.\mathtt{e}.\mathtt{f}) \setminus \{\mathtt{e}\} \parallel (b.\mathtt{e}.\mathtt{f}) \setminus \{\mathtt{f}\}) \setminus \{\mathtt{e}, \mathtt{f}\} \xrightarrow{\mathtt{c}[1]} \\ (((\mathtt{a}[1]; b).\mathtt{e}.\mathtt{f} \parallel \mathtt{a}[1].\mathtt{e}.\mathtt{f}) \setminus \{\mathtt{e}\} \parallel (b.\mathtt{e}.\mathtt{f}) \setminus \{\mathtt{f}\}) \setminus \{\mathtt{e}, \mathtt{f}\}\end{array}$$

At this point, rule (CONCERT) cannot be applied. First note that restrictions cannot be rearranged because e occurs free in $(b.\mathtt{e}.\mathtt{f}) \setminus \{\mathtt{f}\}$ and f occurs free in $((\mathtt{a}; b).\mathtt{e}.\mathtt{f} \parallel \mathtt{a}.\mathtt{e}.\mathtt{f}) \setminus \{\mathtt{e}\}$. Hence, the only possibility for the application of (CONCERT) would require: (1) $((\mathtt{a}[1]; b).\mathtt{e}.\mathtt{f} \parallel \mathtt{a}[1].\mathtt{e}.\mathtt{f}) \setminus \{\mathtt{e}\} \xrightarrow{(b)[2]}\xrightarrow{\underline{\mathtt{a}[1]}}$ and (2) $(b.\mathtt{e}.\mathtt{f}) \setminus \{\mathtt{f}\} \xrightarrow{b[2]}\xrightarrow{\underline{\mathtt{a}[1]}}$. While (1) holds, (2) clearly does not. The phenomenon is basically due to the fact that the definition of rule (CONCERT) is not compositional.

3. The Compositional Calculus of Covalent Bonding

In this section we revise the semantics of CCB presented in section 2. This will help us to better state the correspondence between CCB and *covalent* bond nets (CBNs). We remodel CCB semantics following these three ideas: (i) we get rid of priority of pre-congruence rules (PROM and MOVE) by applying them directly in the transition rules; (ii) we avoid the look-a-head of the CONCERT rule by using a more compact and standard way of gathering labels thorough the parallel operator, more in the line with a synchronisation algebra [34]; and (iii) when a key/bond is passed from a weak action to a strong one (mimicking PROM and MOVE) this key is deemed as *invalid* (e.g., decorated with †) so to avoid unwanted backward de-synchronisations. On the one hand, these modifications make the semantics compositional. On the other hand, the calculus retains the flavour of the original semantics and is consistent with it. The syntax of processes and all the definitions remain the same, when not explicitly specified.

The set of labels is defined below. We extend the set of labels used in the original presentation to account for moves compositionally. Our transition system considers a extended set of label. In addition to the original presentation, we have (i) triples of actions $(\underline{\mathtt{a}}[l], b[k], \underline{c}[l])$ that represents, e.g., the moves of a parallel process in which one part starts the concert with $\underline{\mathtt{a}}[l], b[k]$ and the other contributes with just the complementary reverse $\underline{c}[l]$; and (ii) $\langle b \rangle$ that stands for the execution of a weak action in a strong position (i.e., in the left-hand side of ';') for contributing to a concert rule.

$$\begin{aligned} \lambda ::= &\quad \mathtt{a}[k] \mid (b)[k] \mid \underline{\mathtt{a}}[k] \mid \langle b \rangle[k] \\ \mu := &\quad \lambda \mid \lambda, \lambda \mid \lambda, \lambda, \lambda \end{aligned}$$

Define $key(\mu)$ for the set of keys of a label, we use $fresh(\mu, s)$ for any key fresh in $key(\mu)$. We consider runtime processes in which actions can be marked with a key of the form $\dagger l$, which means a bond that has been promoted or moved. In this way the key l cannot be anymore used for backward synchronisations. When we write k, l, we denote keys different from $\dagger l$.

The rules are reported in fig. 7 and fig. 8. Let us briefly comment on them. Rules STR_s and STR_w allow for the execution of respectively a strong action and a weak action in a *strong position* (e.g., at the left part of ';'). Note that STR_w can be applied only when all actions in the prefix but c have been already executed (recall that t ranges over $\mathbb{N}^{\mathcal{A}\times\mathcal{K}}$). STR_m combines the original rules ACT1 and MOVE: the key k associated with the execution of the weak action c is moved to the strong action $\mathtt{a}$. We remark that rule STR_m differs from STR_w, which considers the cases in which every strong action in the prefix has been already executed, and consequently the bond created by the execution of the weak action cannot be moved. Rule WK accounts for the execution of a weak prefix as in standard CCB. Rule $\underline{\text{STR}}$ allows for the undoing of a strong action provided no weak action in a strong position is marked (i.e., the reverse action cannot be followed by a move). Rule A-$\underline{\text{STR}}_p$ undoes a strong action and directly *promotes* the weak bond. Analogously, rule A-$\underline{\text{STR}}_m$ combines the reverse of a strong action with a move. The other rules are defined as in CCB.

The rules for concerted actions are reported in fig. 8. Rule A-WK_1 accounts for the initiation of a concert, which consists of the execution of the weak prefix b and the reversal of the strong action $\underline{\mathtt{a}}$ (within the same prefix), which is followed by the move of the weak bond l created by the execution of b. Note that the weak bond l is invalidated after the move. Rule A-WK_2 accounts for the transitions in which a prefix contributes to a concert with both a forward weak

$$(\text{STR}_s)\ \frac{\mathsf{a}\in\mathcal{S} \qquad std(\mathsf{S}) \qquad fresh(k,s)}{(\mathsf{a},s;b).\mathsf{S} \xrightarrow{\mathsf{a}[k]} (\mathsf{a}[k],s;b).\mathsf{S}} \qquad (\text{STR}_w)\ \frac{std(\mathsf{S}) \qquad fresh(k,t)}{(c,t;b).\mathsf{S} \xrightarrow{c[k]} (c[k],t;b).\mathsf{S}}$$

$$(\text{STR}_m)\ \frac{std(\mathsf{S}) \qquad fresh(k,s)}{(c,\mathsf{a},s;b).\mathsf{S} \xrightarrow{c[k]} (c,\mathsf{a}[\dagger k],s;b).\mathsf{S}} \qquad (\text{WK})\ \frac{std(\mathsf{S}) \qquad fresh(k,t)}{(t;b).\mathsf{S} \xrightarrow{(b)[k]} (t;b[k]).\mathsf{S}}$$

$$(\underline{\text{STR}})\ \frac{std(\mathsf{S}) \qquad s\cap\mathcal{W}\times\mathcal{K}=\emptyset}{(\mathsf{a}[k],s;b).\mathsf{S} \xrightarrow{\underline{\mathsf{a}}[k]} (\mathsf{a},s;b).\mathsf{S}} \qquad (\text{A-}\underline{\text{STR}}_p)\ \frac{std(\mathsf{S})}{(\mathsf{a}[k],s;b[l]).\mathsf{S} \xrightarrow{\underline{\mathsf{a}}[k]} (\mathsf{a}[\dagger l],s;b).\mathsf{S}}$$

$$(\text{A-}\underline{\text{STR}}_m)\ \frac{std(\mathsf{S})}{(\mathsf{a}[k],c[l],s;b).\mathsf{S} \xrightarrow{\underline{\mathsf{a}}[k]} (\mathsf{a}[\dagger l],c,s;b).\mathsf{S}}$$

$$(\text{COM})\ \frac{\mathsf{P} \xrightarrow{\mathsf{a}[k]} \mathsf{P}' \qquad \mathsf{Q} \xrightarrow{\mathsf{c}[k]} \mathsf{Q}'}{\mathsf{P}\parallel\mathsf{Q} \xrightarrow{\gamma(\mathsf{a},\mathsf{c})[k]} \mathsf{P}'\parallel\mathsf{Q}'} \qquad (\underline{\text{COM}})\ \frac{\mathsf{P} \xrightarrow{\underline{\mathsf{a}}[k]} \mathsf{P}' \qquad \mathsf{Q} \xrightarrow{\underline{\mathsf{c}}[k]} \mathsf{Q}'}{\mathsf{P}\parallel\mathsf{Q} \xrightarrow{\underline{\gamma(\mathsf{a},\mathsf{c})}[k]} \mathsf{P}'\parallel\mathsf{Q}}$$

$$(\text{CONT})\ \frac{\mathsf{S} \xrightarrow{\mu} \mathsf{S}' \qquad fresh(\mu,t)}{(t;b).\mathsf{S} \xrightarrow{\mu} (t;b).\mathsf{S}'} \qquad (\text{PAR})\ \frac{\mathsf{P} \xrightarrow{\mu} \mathsf{P}' \qquad fresh(\mu,\mathsf{Q})}{\mathsf{P}\parallel\mathsf{Q} \xrightarrow{\mu} \mathsf{P}'\parallel\mathsf{Q}}$$

$$(\text{RES})\ \frac{\mathsf{P} \xrightarrow{\mu} \mathsf{P}'}{\mathsf{P}\setminus L \xrightarrow{\mu} \mathsf{P}'\setminus L}\ n(\mu)\cap L=\emptyset$$

Figure 7: Compositional rules for CCB

action and a reversal of a strong action. As in A-Wk$_1$, rule A-Wk$_2$ also reflects the move of the weak bond from the weak action to the strong action that is being reversed. Rule A-Wk$_3$ is analogous to the previous one, but considers the cases in which the created bond l is moved to another strong action in the same prefix. Finally Conc rule deals with concert rule and relies on the definition of the following commutative operation $\oplus$, which combines labels as follows

$$\mu_1\oplus\mu_2 = \begin{cases} \underline{\mathsf{a}}[k], \langle b\rangle[l] & \text{if } \mu_1=\underline{\mathsf{a}}[k]\wedge\mu_2=b[l] \\ \mu_1,\mu_2 & \text{if } \mu_1=\{\underline{\mathsf{a}}[k],(b)[l]\}\wedge\mu_2=c[l]\wedge\gamma(b,c)\downarrow \\ \mu_1,\mu_2 & \text{if } \mu_1=\{\underline{\mathsf{a}}[k],(b)[l]\}\wedge\mu_2=\underline{\mathsf{c}}[k]\wedge\gamma(\mathsf{a},\mathsf{c})\downarrow \\ \underline{\gamma(\mathsf{a},\mathsf{c})}[k],\gamma(b,d)[l] & \text{if } \mu_1,\mu_2=\{\underline{\mathsf{a}}[k],\underline{\mathsf{c}}[k],(b)[l],\beta[l]\}\wedge\beta\in\{c,(c),\langle c\rangle\} \end{cases}$$

The first case allows for the combination of parallel process that contribute with a reverse action $\underline{\mathsf{a}}[k]$ and a forward weak action in a strong position $b[l]$. Note that $\underline{\mathsf{a}}[k]\oplus b[l] = \underline{\mathsf{a}}[k]\oplus\langle b\rangle[l]$, where $\langle b\rangle[l]$ reflects the fact that both actions are intended to contribute to a concert. The following two cases account

$$(\text{A-WK}_1)\ \frac{std(\mathtt{S}) \qquad fresh(l,\{k,t\})}{(\mathtt{a}[k],t;b).\mathtt{S} \xrightarrow{(b)[l],\underline{\mathtt{a}}[k]} (\mathtt{a}[\dagger l],t;b).\mathtt{S}}$$

$$(\text{A-WK}_2)\ \frac{std(\mathtt{S}) \qquad fresh(l,\{k,t\})}{(\mathtt{a}[k],c,s;b).\mathtt{S} \xrightarrow{\langle c\rangle[l],\underline{\mathtt{a}}[k]} (\mathtt{a}[\dagger l],c,s;b).\mathtt{S}}$$

$$(\text{A-WK}_3)\ \frac{std(\mathtt{S}) \qquad fresh(l,\{k,s\})}{(\mathtt{a}[k],c,\mathtt{d},s;b).\mathtt{S} \xrightarrow{\langle c\rangle[l],\underline{\mathtt{a}}[k]} (\mathtt{a},c,\mathtt{d}[\dagger l],s;b).\mathtt{S}}$$

$$(\text{CONC})\ \frac{\mathtt{P} \xrightarrow{\mu_1} \mathtt{P}' \qquad \mathtt{Q} \xrightarrow{\mu_2} \mathtt{Q}'}{\mathtt{P} \parallel \mathtt{Q} \xrightarrow{\mu_1 \oplus \mu_2} \mathtt{P}' \parallel \mathtt{Q}}$$

Figure 8: Compositional concert rules for CCB

for the situations in which one parallel branch contributes with the label that indicates the starting of a concert, i.e., $\{\underline{\mathtt{a}}[k],(b)[l]\}$, and the other with the complementary forward ($c[l]$) or backward ($\underline{\mathtt{c}}[k]$) action. The last case deals with a concerted move in which the labels of the different branches contribute with all needed actions.

Example 4. *Consider the molecules $A = (\mathtt{c};q)$, $B = (\mathtt{d},q)$ and $C = (\mathtt{c},\mathtt{d})$ with $\gamma(\mathtt{c},\mathtt{c}) = \mathtt{f}$, $\gamma(\mathtt{d},\mathtt{d}) = \mathtt{e}$ and $\gamma(q,q) = a$ defined in example 2. We have the following reductions:*

$$A \parallel B \parallel C \xrightarrow{\mathtt{f}[1]} (\mathtt{c}[1];q) \parallel B \parallel (\mathtt{c}[1],\mathtt{d}) \xrightarrow{\mathtt{e}[2]} (\mathtt{c}[1];q) \parallel (\mathtt{d}[2],q) \parallel (\mathtt{c}[1],\mathtt{d}[2])$$

Then, we have the following derivation:

$$\frac{(\mathtt{c}[1];q) \xrightarrow{\langle q\rangle[3],\underline{\mathtt{c}}[1]} (\mathtt{c}[\dagger 3];q) \qquad \dfrac{(\mathtt{d}[2],q) \xrightarrow{q[3]} (\mathtt{d}[2],q[3]) \qquad (\mathtt{c}[1],\mathtt{d}[2]) \xrightarrow{\underline{\mathtt{c}}[1]} (\mathtt{c},\mathtt{d}[2])}{(\mathtt{d}[2],q) \parallel (\mathtt{c}[1],\mathtt{d}[2]) \xrightarrow{\langle q\rangle[3],\underline{\mathtt{c}}[1]} (\mathtt{d}[2],q[3]) \parallel (\mathtt{c},\mathtt{d}[2])}}{(\mathtt{c}[1];q) \parallel (\mathtt{d}[2],q) \parallel (\mathtt{c}[1],\mathtt{d}[2]) \xrightarrow{\{a[3],\underline{\mathtt{f}}[1]\}} (\mathtt{c}[\dagger 3];q) \parallel (\mathtt{d}[2],q[3]) \parallel (\mathtt{c},\mathtt{d}[2])}$$

Let us note that the label $\langle q\rangle[3],\underline{\mathtt{c}}[1]$ is obtained as the result of $q[3] \oplus \underline{\mathtt{c}}[1]$. This example is borrowed from [8]. Now, the bond between the actions d *identified with the key 2 can be undone.*

$$(\mathtt{c}[\dagger 3];q) \parallel (\mathtt{d}[2],q[3]) \parallel (\mathtt{c},\mathtt{d}[2]) \xrightarrow{\underline{\mathtt{e}}[2]} (\mathtt{c}[\dagger 3];q) \parallel (\mathtt{d}[\dagger 3],q) \parallel (\mathtt{c},\mathtt{d})$$

Note that the final configuration differs from the final one in example 2 because promotion and move of weak bonds take place instantaneously. Nonetheless, the *original semantics of* CCB *requires the application of pre-congruence rules before deriving new transitions from the final configuration of example 2, which would produce the similar configuration* $(\mathtt{c}[3]; q) \parallel (\mathtt{d}[3], q) \parallel (\mathtt{c}, \mathtt{d})$ *(if we disregard* $\dagger$*). Our semantics deviates from the original definition of* CCB *in the treatment of promoted bonds, which are not reversible in our case. Note that* $(\mathtt{c}[\dagger 3]; q) \parallel (\mathtt{d}[\dagger 3], q) \parallel (\mathtt{c}, \mathtt{d})$ *would be unable to break the bond* $\dagger 3$ *between* c *and* q *even if* c *and* q *were synchronisable. After noting that none of the examples in [8] exploit such a mechanism, we adopted this constraint that simplifies our encoding.*

Example 5. *Consider the process* $(\mathtt{a}; b) \parallel \mathtt{a} \parallel b$ *with* $\gamma(\mathtt{a}, \mathtt{a}) = \mathtt{c}$ *and* $\gamma(b, b) = d$ *defined in example 3. We have the following reductions:*

$$(\mathtt{a}; b) \parallel \mathtt{a} \parallel b \xrightarrow{\gamma(\mathtt{a},\mathtt{a})[1]} (\mathtt{a}[1]; b) \parallel \mathtt{a}[1] \parallel b \xrightarrow{\mu_1 \oplus \mu_2} (\mathtt{a}[\dagger 2]; b) \parallel \mathtt{a} \parallel b[2]$$

where $\mu_1 = (b)[2], \underline{\mathtt{a}}[1]$ *and* $\mu_2 = \langle b \rangle[2], \underline{\mathtt{a}}[1]$*. Let us note that* μ_2 *is the result of two actions in parallel performed by* $\mathtt{a}[1]$ *and* b*. With respect to the example 3 (and in general with the original* CCB*) we can notice that the promotion of a weak bond to a strong one is done automatically in the rule, and not via auxiliary pre-congruence rules. In this way we get rid of priority on rules. Also, we use* $\dagger$ *to indicate a promoted (or moved) bond, which cannot participate anymore in any synchronisation.*

Definition 3 (Process Context). *A context* $\mathtt{C}$ *is a process with a hole* $\bullet$*, which is defined by the following grammar:*

$$\mathtt{C}[\bullet] ::= (\bullet, s; \beta).\mathtt{C} \mid (s; \bullet).\mathtt{C} \mid \mathtt{C} \parallel \mathtt{Q} \mid \mathtt{Q} \parallel \mathtt{C} \mid \mathtt{C} \setminus L$$

We now give the definition of well-formedness for both prefixes and processes. Intuitively a prefix is well-formed if its keys are pair-wise different (e.g., there are no repetitions); invalid keys are associated only with strong actions; and a weak action is marked only if all other strong actions are marked. Condition for a process $\mathtt{P}$ to be well-formed are a bit more involved. Let us comment on them. The first condition tells that if the process $\mathtt{P}$ is a prefix of the form $(s; \beta).\mathtt{S}$ then they key used by the prefix $(s; \beta)$ are disjoint from the keys of its continuation $\mathtt{S}$ and if $\mathtt{S}$ has some executed actions then the weak action of the prefix does not bear any key. A parallel composition $\mathtt{P}_1 \parallel \mathtt{P}_2$ is well-formed if the two processes are well-formed and they do not have marked weak prefixed. Also, the shared keys have to be used once per part, meaning that there has been a syncrhonization. To this end we have three sub-cases: (i) either a shared key k marks two action which are not restricted by their respective contextes; (ii) the key is used to invalidate (via $\dagger$) a strong action in one part of the parallel and in the other part it is used to mark a weak action; or (iii) in both parts the key is used to invalidate a strong action. Finally, the process $\mathtt{P} \setminus L$ is well-formed if $\mathtt{P}$ is well-formed.

We now give the definition of well-formedness for both prefixes and processes.

Definition 4 (Well-formedness). *A prefix* $(s; \beta)$ *is well-formed if*

1. $key(s;\beta)$ *is a set; and*

2. $\forall c \in \mathcal{W}, k \in \mathcal{K}$, $s(c[\dagger k]) = 0$ *and* $\beta \neq c[\dagger k]$.

3. $\forall c \in \mathcal{W}, k \in \mathcal{K}$, *if* $s, \beta(c[k]) = 1$ *then* $s \in \mathcal{A} \times \mathcal{K}$.

A process $\mathtt{P}$ *is well-formedness if:*

1. $\mathtt{P} = (s;\beta).\mathtt{S}$, *and* $(s;\beta)$ *and* $\mathtt{S}$ *are well-formed, and* $key(s;\beta) \cap key(\mathtt{S}) = \emptyset$, *and* $key(\mathtt{S}) \neq \emptyset$ *implies* $s \in \mathbb{N}^{\mathcal{A}\times\mathcal{K}}$ *and* $\beta \in \mathcal{A}$.

2. $\mathtt{P} = \mathtt{P}_1 \| \mathtt{P}_2$, *and* $\mathtt{P}_1$ *and* $\mathtt{P}_2$ *are well-formed and* free from executed weak prefixes, *and for all* $k \in \mathcal{K}$ *if* $k \in key(\mathtt{P}_1) \cap key(\mathtt{P}_2)$ *then* $key(\mathtt{P}_1)(k) = key(\mathtt{P}_2)(k) = 1$ *and one of the following holds*

   i $\mathtt{P}_1 = \mathtt{C}_1[\mathtt{a}[k]]$ *and* $\mathtt{P}_2 = \mathtt{C}_2[\mathtt{b}[k]]$ *and* $\gamma(\mathtt{a},\mathtt{b}) \downarrow$, $\mathtt{a} \notin bn(\mathtt{C}_1[])$ *and* $\mathtt{b} \notin bn(\mathtt{C}_2[])$.

   ii $\mathtt{P}_1 = \mathtt{C}_1[\mathtt{a}[\dagger k]]$ *and* $\mathtt{P}_1 = \mathtt{C}_2[\mathtt{b}[k]]$ *and* $\mathtt{a} \in \mathcal{S}$, $\mathtt{b} \in \mathcal{W}$, $\mathtt{a} \notin bn(\mathtt{C}_1[])$ *and* $\mathtt{b} \notin bn(\mathtt{C}_2[])$.

   iii $\mathtt{P}_1 = \mathtt{C}_1[\mathtt{a}[\dagger k]]$ *and* $\mathtt{P}_1 = \mathtt{C}_2[\mathtt{b}[\dagger k]]$ *and* $\mathtt{a}, \mathtt{b} \in \mathcal{S}$, $\mathtt{a} \notin bn(\mathtt{C}_1[])$ *and* $\mathtt{b} \notin bn(\mathtt{C}_2[])$.

3. $\mathtt{P} = \mathtt{P}_1 \setminus L$ *and* $\mathtt{P}_1$ *is well-formed.*

Lemma 1. *If* $\mathtt{P}$ *is well-formed then* $\forall k \in \mathcal{K}.key(\mathtt{P})(k) \leq 2$.

Proof. By a straightforward induction on the structure of $\mathtt{P}$. □

Lemma 2 (Well-formedness preservation). *If* $\mathtt{P}$ *is well-formed and* $\mathtt{P} \xrightarrow{\mu} \mathtt{P}'$, *then* $\mathtt{P}'$ *is well-formed.*

Proof. By induction on the derivation and with a case analysis on the last applied rule. All the cases are simple. The main idea is that all keys are freshly generated, and through Com or Conc keys are forced to match if a synchronisation can happen. It is here the only place in which two instances of the same key k can be generated, and these instances depending on the rule used to derive the label can be used to mark a strong action, to invalidate a strong action or to mark a weak action. □

4. Nets

We summarise the basics of Petri net with inhibitor arcs along the lines of [35, 36]. A net is the triple $\langle S, T, F\rangle$ where S is a set of *places*, T is a set of *transitions* such that $S \cap T = \emptyset$ and $F \subseteq (S \times T) \cup (T \times S)$ is the *flow* relation.

Definition 5. *A* Petri net *is a pair* $N = (\langle S, T, F\rangle, \mathsf{m})$ *where* $\langle S, T, F\rangle$ *is a net and* $\mathsf{m} \in \mu S$ *is the* initial marking.

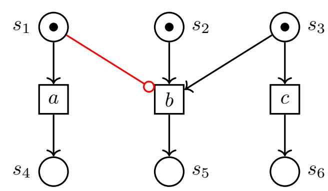

Figure 9: An IPT N

Definition 6. *A* Petri net with inhibitor arcs *(*IPT *for short) is the pair* $N = (\langle S, T, F, I\rangle, \mathsf{m})$*, where* $(\langle S, T, F\rangle, \mathsf{m})$ *is a Petri net and* $I \subseteq S \times T$ *is the* inhibiting *relation.*

Given an IPT $N = (\langle S, T, F, I\rangle, \mathsf{m})$ and $x \in S \cup T$, the *pre-* and *postset* of x are respectively defined as the (multi)sets ${}^\bullet x = \{y \mid (y, x) \in F\}$ and $x^\bullet = \{y \mid (x, y) \in F\}$. If $x \in S$ then ${}^\bullet x \in \mu T$ and $x^\bullet \in \mu T$; analogously, if $x \in T$ then ${}^\bullet x \in \mu S$ and $x^\bullet \in \mu S$. The *inhibitor set* of a transition t is the (multi)set ${}^\circ t = \{s \mid (s, t) \in I\}$. The definition of ${}^\bullet\cdot, \cdot^\bullet, {}^\circ\cdot$ generalise straightforwardly to multisets of transitions.

Example 6. *Consider the* IPT N *depicted in Figure 9, which consists of six places depicted as circles and three transitions drawn as boxes. The flow relation is represented by black arrows while the inhibitor relation is shown by red lines ended with a small circle. The initial marking* $\mathsf{m} = \{s_1, s_2, s_3\}$ *is represented by the bullets drawn within the corresponding places. Consider, for instance, the transition* b*: it consumes tokens from* s_2 *and* s_3*, produces a token in* s_5*, and is inhibited by* s_1*. Hence, its pre-, post- and inhibiting sets are respectively* ${}^\bullet b = \{s_2, s_3\}$*,* $b^\bullet = \{s_5\}$*, and* ${}^\circ b = \{s_1\}$*.*

A (multiset of) transition(s) $A \in \mu T$ is *enabled at a marking* $m \in \mu S$, written $m\,[A\rangle$, whenever ${}^\bullet A \subseteq m$ and $\forall s \in {}^\circ A.\ m(s) = 0 \ \wedge\ A^\bullet(s) = 0$. The last condition requires the absence of tokens in all places connected via inhibitor arcs to the transitions in A. Observe that the multiset $\emptyset$ is enabled at every marking. A (multiset of) transition(s) A enabled at a marking m can *fire* and its firing produces the marking $m' = m - {}^\bullet A + A^\bullet$. The firing of A at a marking m is denoted by $m\,[A\rangle\,m'$. We assume that each transition t of an IPT N is defined such that ${}^\bullet t \neq \emptyset$, i.e., it cannot fire *spontaneously* in an uncontrolled manner without consuming tokens. Moreover, we assume that there are no isolated places, i.e., for all place s, there exists a transition t such that $s \in {}^\bullet t \cup t^\bullet \cup {}^\circ t$.

Example 7. *Consider the* IPT *introduced in Example 6 and note that both* a *and* c *are enabled at the initial marking* m*. On the contrary,* b *is not enabled because its inhibitor place* s_1 *contains a token. The firing of* a *produces the marking* $m' = \{s_2, s_3, s_4\}$*, i.e.* $\mathsf{m}\,[a\rangle\,m'$*, at which* b *becomes enabled because its preset* s_2 *and* s_3 *is marked while its inhibitor place* s_1 *is not.*

A marking m is *reachable* in N if there exists a firing sequence $\sigma = \mathsf{m}\,[A_0\rangle\, m_1 \cdots m_n\,[A_n\rangle\, m$ from the initial marking m to m. We write $\mathcal{M}_N$ for the set of all reachable markings of N.

Example 8. *The set of reachable markings of the* IPT *in Example 6 is* $\mathcal{M}_{N_1} = \{\mathsf{m}, \{s_2, s_3, s_4\}, \{s_1, s_2, s_6\}, \{s_4, s_5\}, \{s_2, s_4, s_6\}\}$.

An IPT N is *safe* if each reachable marking is a set. Hereafter, we will consider only safe IPT.

Given a net $N = (\langle S, T, F, I\rangle, \mathsf{m})$ and a subset $T' \subseteq T$ of transitions the subnet generated by T' is the net $(\langle S', T', F', I'\rangle, \mathsf{m}')$ where $S' = {}^\bullet T' \cup T'^\bullet \cup {}^\circ T'$, $F' = F \cap ((S' \times T') \cup (T' \times S'))$, $I' = I \cap (S' \times T')$ and m' is the restriction of m to the places in S'.

Example 9. *Consider again the* IPT *N of Figure 9, and take $T' = \{a\}$. The places of the subnet identified by $\{a\}$ are s_1 and s_4 and the flow arcs are (s_1, a) and (a, s_4), there are no inhibitor arcs and only s_1 is initially marked. It the subset of transitions were $\{b\}$ then the subnet would have s_1, s_2, s_3 and s_5 as places, $(s_2, b), (s_3, b)$ and (b, s_5) as flow arcs, (s_1, b) as inhibitor arcs. Its initial marking would assign tokens to s_1, s_2, s_3 (but not to s_5).*

Definition 7. *Let $N = (\langle S, T, F, I\rangle, \mathsf{m})$ be an* IPT*. N can be partitioned into k components if there exists a partition ~~of transitions $T_1, \ldots, T_k$~~ $\{T_1, \ldots, T_k\}$ of the set of transitions T such that*

- $N_i = (\langle S_i, T_i, F_i, I_i\rangle, \mathsf{m}_i)$ ~~*where $S_i = {}^\bullet T_i \cup T_i^\bullet \cup {}^\circ T_i$, F_i is the restriction of F to S_i and T_i, I_i is the restriction of I to S_i and T_i and m_i is the restriction of m to S_i is an* IPT~~ *is the subnet of N generated by T_i; and*
- ~~$(\langle \bigcup_{i\in\{1,\ldots,k\}} S_i, \bigcup_{i\in\{1,\ldots,k\}} T_i, \bigcup_{i\in\{1,\ldots,k\}} F_i, \bigcup_{i\in\{1,\ldots,k\}} I_i\rangle, \bigcup_{i\in\{1,\ldots,k\}} \mathsf{m}_i) = N$.~~ $\{S_1, \ldots, S_k\}$ *is a partition of the set of places S.*

A net can be partitioned if it consists of a set of ~~independent~~disjoint subnets, ~~where independent means that in N~~i.e., there is no arc connecting a place in S_i with a transition in T_j, if $i \neq j$. It is worth observing that the IPT N is $(\langle \bigcup_{i\in\{1,\ldots,k\}} S_i, \bigcup_{i\in\{1,\ldots,k\}} T_i, \bigcup_{i\in\{1,\ldots,k\}} F_i, \bigcup_{i\in\{1,\ldots,k\}} I_i\rangle, \bigcup_{i\in\{1,\ldots,k\}} \mathsf{m}_i)$ because we consider nets without isolated places

Example 10. *Consider the net N in Figure 10. It can be partitioned into two subnets, namely N_1 (with places s_1, s_2, s_4 and s_5 and transitions a and b) and N_2 (with places s_3 and s_6 and transition c).*

~~In this paper~~When establishing the correspondence between nets and terms of `CCB` the *name* of the executed transitions will play a fundamental role, hence we consider labeled nets, where the labeling mapping is defined as a total mapping of transitions into a set of labels L.

Definition 8. *A* labeled *Petri net with inhibitor arcs (*LIPT *for short) is a pair $N = (\langle S, T, F, I, \ell\rangle, \mathsf{m})$, where $(\langle S, T, F, I\rangle, \mathsf{m})$ is an* IPT *and $\ell : T \to \mathsf{L}$ is a labeling function.*

The notion of partition is lifted to labeled nets in the obvious way.

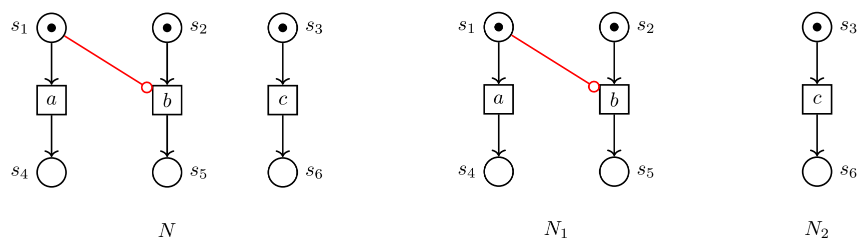

Figure 10: A net N with two partitions (N_1 and N_2)

5. Covalent Bond nets

We restrict our attention to a class of nets in which dependencies are represented through inhibitor arcs, along the lines of [17]. We introduce a class of contextual Petri nets, baptized *bond nets*, which consists of several components in which (i) dependencies are *mainly* recovered from the inhibiting relation instead of the usual flow relation, and (ii) *synchronisations* (bonds) among components are allowed under suitable assumptions.

We consider a set of labels L, the precise structure of which will be made explicit later. For now, we assume that L contains a non empty subset $\mathsf{L}_{\mathcal{W}} \subset \mathsf{L}$ of weak labels, and a non empty subset $\mathsf{L}_{\mathcal{S}} \subset \mathsf{L}$ of strong labels. For the sake of the simplicity, we omit the marking of LIPT in the following definitions because the class of *bond nets* is defined only in terms of the structure of the net.

Definition 9. *Let $N = \langle S, T, F, I, \ell \rangle$ be an* LIPT. *N is a* basic *net if the following conditions are satisfied:*

1. $\forall t, t' \in T.\ {}^{\bullet}t' \cap t^{\bullet} = \emptyset$;
2. $\forall t \in T.\ {}^{\bullet}t \neq \emptyset$ *and* $t^{\bullet} \neq \emptyset$; *and*
3. $\forall t \in T.\ {}^{\circ}t$ *is finite.*

The conditions imposed on basic nets share motivations with *occurrence* nets [37], *unravel* nets [38, 39, 40] and *flow* nets [41], in which computations are explained without resorting to firing sequences, as it will be clearer when discussing the notion of configuration. For now it is enough to observe that the execution of some transition can be inferred by looking at the postset of it, though this could not be the unique condition. The first condition implies that dependencies in basic nets do not arise because of the flow relation. The second condition ensures that each transition is not allowed to fire *spontaneously* (non empty preset) and that some effect of its firing can be observed (non empty postset). The third condition implies that each transition has a finite set of inhibiting places, which suggests that the dependencies of each transition are finite.

The notion of basic net is further refined into the one of *pre-bond* net, where the labels play a major role.

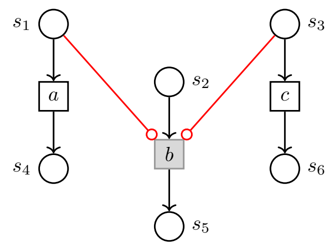

Figure 11: A simple pBN.

Definition 10. *Let* $N = \langle S, T, F, I, \ell\rangle$ *be a basic net,* $T_{\mathcal{S}} = \{t \in T \mid \ell(t) \in \mathsf{L}_{\mathcal{S}}\}$ *and* $T_{\mathcal{W}} = \{t \in T \mid \ell(t) \in \mathsf{L}_{\mathcal{W}}\}$ *be subsets of transitions.* N *is a* pre-bond *net (*pBN *for short) if the following further conditions are satisfied:*

1. $T_{\mathcal{S}} \neq \emptyset \neq T_{\mathcal{W}}$;
2. $\forall t \in T_{\mathcal{S}} \cup T_{\mathcal{W}}.\ |{}^{\bullet}t| = 1$;
3. ~~$\forall t, t' \in T_{\mathcal{S}} \cup T_{\mathcal{W}}.\ {}^{\bullet}t \cap {}^{\bullet}t' \neq \emptyset \Rightarrow \ell(t) = \ell(t')$;~~
3. $\forall t \in T_{\mathcal{S}}.\ {}^{\circ}t = \emptyset$; *and*
4. $\exists! t \in T_{\mathcal{W}}.\ {}^{\circ}t = {}^{\bullet}T_{\mathcal{S}} \cup {}^{\bullet}(T_{\mathcal{W}} \setminus \{t\})$.

In a pBN there must be transitions labeled in $\mathsf{L}_{\mathcal{S}}$ and in $\mathsf{L}_{\mathcal{W}}$, and these transitions have just a single place in their preset. Transitions with the same label in $\mathsf{L}_{\mathcal{S}} \cup \mathsf{L}_{\mathcal{W}}$ may share the same preset. Transitions with labels in $\mathsf{L}_{\mathcal{S}}$ are such that their inhibitor set is empty, and there exists a unique transitions labeled in $\mathsf{L}_{\mathcal{W}}$ which is inhibited by all the other transitions with labels in $\mathsf{L}_{\mathcal{S}}$ and in $\mathsf{L}_{\mathcal{W}}$. This transition is called ~~a~~ *weak* ~~transition,~~, since it can be fired only when all the other transitions with labels in $\mathsf{L}_{\mathcal{S}}$ and $\mathsf{L}_{\mathcal{W}}$ have been executed. The weak transition can be identified and denoted as $\mathsf{weak}(N)$, provided that N is a pBN. Thus $\mathsf{weak}(N) = \{t \in T \mid \ell(t) \in \mathsf{L}_{\mathcal{W}} \ \wedge\ {}^{\circ}t = {}^{\bullet}T_{\mathcal{S}} \cup {}^{\bullet}(T_{\mathcal{W}} \setminus t)\}$ is the set containing the weak transition and $\mathsf{strong}(N) = T_{\mathcal{S}} \cup T_{\mathcal{W}} \setminus \mathsf{weak}(N)$ are all the transitions in a pBN that are not weak and have a label in $\mathsf{L}_{\mathcal{S}} \cup \mathsf{L}_{\mathcal{W}}$, which we call *strong* transition. It should be stressed that a strong transition may have a weak label, but a weak transition should have a weak label.

Example 11. *The net in Figure 11 is a* pBN *where the set* $\mathsf{weak}(N)$ *contains the transition* b *and the set of strong transitions is* $\mathsf{strong}(N) = \{a, c\}$.

Proposition 1. *Let* $N = \langle S, T, F, I, \ell\rangle$ *be a* pBN. *Then* $\mathsf{weak}(N)$ *is a singleton.*

Proof. It follows from the definition of pBN. □

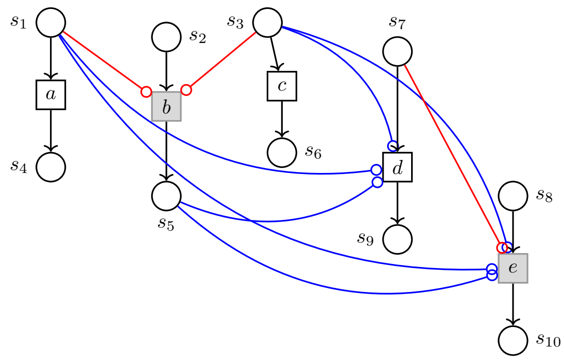

Figure 12: A stratified pBN.

Definition 11. *Let* $N = \langle S, T, F, I, \ell \rangle$ *be a* pBN. *N is* stratified *if there exists a partition of places and transitions indexed by* $\mathcal{I}$ *such that* $S = \bigcup_{i \in \mathcal{I}} S_i$, $T = \bigcup_{i \in \mathcal{I}} T_i$, $F = \bigcup_{i \in \mathcal{I}} F_i$, $\mathsf{m} = \bigcup_{i \in \mathcal{I}} \mathsf{m}_i$, ℓ_i *is* ℓ *restricted to the transition in* T_i, *and*

1. $\forall i \in \mathcal{I}.\ N_i = (\langle S_i, T_i, F_i, I \cap S_i \times T_i, \ell_i \rangle, \mathsf{m}_i)$ *is a pre-bond net; and*
2. $\forall t \in T_i$ *either* ${}^{\circ}\mathsf{strong}(N_i) = \emptyset$ *and* ${}^{\circ}\mathsf{weak}(N_i) = {}^{\bullet}\mathsf{strong}(N_i)$ *or there exists a unique* $j \in \mathcal{I}.\ {}^{\circ}t \setminus S_i = {}^{\bullet}\mathsf{strong}(N_j) \cup \mathsf{weak}(N_j)^{\bullet}$.

For each $i \in \mathcal{I}$ we say that N_i is a *part* of the stratified pre-bond net N, and it is denoted by $\mathcal{C}(N, i)$. A stratified pre-bond net is a net *formed* by pre-bond nets that are only connected by inhibitor arcs, and the transitions of one of the components which are inhibited by places in another component, are all inhibited by the same subset of places. The weak transitions of a stratified pBN can be identified, as they are the union of the weak transitions of each component. With abuse of notation, we write $\mathsf{weak}(N) = \bigcup_{i \in \mathcal{I}} \mathsf{weak}(\mathcal{C}(N, i))$ for the set of weak transitions.

Example 12. *The net in Figure 12 is a stratified one. It has two parts* $\mathcal{C}(N, 1)$ *and* $\mathcal{C}(N, 2)$. *The net* $\mathcal{C}(N, 1)$ *is the one depicted in Figure 11, whereas* $\mathcal{C}(N, 2)$ *is the net with places* $\{s_7, s_8, s_9, s_{10}\}$, *transitions d and e. The weak transition e is inhibited by the place* s_7 *(depicted with a red inhibitor arc in Figure 12) and the two transitions are inhibited by some places belonging to the simple net* $\mathcal{C}(N, 1)$, *namely the places* s_1, s_3 *and* s_5.

Definition 12. *Let* $N = \langle S, T, F, I, \ell \rangle$ *be a stratified* pBN. *We say that it is* well-stratified *if the set of index* $\mathcal{I}$ *can be totally ordered and* $\forall i \in \mathcal{I}.\ \forall t \in T_i$ *if* ${}^{\circ}t \not\subseteq S_i$ *then* ${}^{\circ}t \subseteq S_j$ *with j being the immediate predecessor of i.*

The fact that $\mathcal{I}$ can be totally ordered implies that there is a bijection between $\mathcal{I}$ and $\{1, \ldots, n\}$ with $n = |\mathcal{I}|$, where the notion of immediate predecessor, if needed, is just minus one. If the set of indexes $\mathcal{I}$ can be totally ordered, we write $\mathsf{fst}(\mathcal{I})$ for the minimum of $\mathcal{I}$.

Example 13. *The net in Figure 12 is well-stratified. Indeed, its two components are $\mathcal{C}(N, 1)$ and $\mathcal{C}(N, 2)$ and the set of indexes is $\{1, 2\}$ ordered as natural numbers.*

Proposition 2. *Let $N = \langle S, T, F, I, \ell \rangle$ be a well-stratified* pBN *with respect to the set of index $\mathcal{I}$. Then*

- $\mathcal{C}(N, \mathsf{fst}(\mathcal{I})) = \langle S_{\mathsf{fst}(\mathcal{I})}, T_{\mathsf{fst}(\mathcal{I})}, F_{\mathsf{fst}(\mathcal{I})}, I \cap S_{\mathsf{fst}(\mathcal{I})} \times T_{\mathsf{fst}(\mathcal{I})}, \ell_{\mathsf{fst}(\mathcal{I})} \rangle$ *is a simple* pBN*; and*
- *if $|\mathcal{I}| > 1$ then $N' = \langle S \setminus S_{\mathsf{fst}(\mathcal{I})}, T \setminus T_{\mathsf{fst}(\mathcal{I})}, F \setminus F_{\mathsf{fst}(\mathcal{I})}, I \setminus (I \cap S_{\mathsf{fst}(\mathcal{I})} \times T_{\mathsf{fst}(\mathcal{I})}), \ell' \rangle$ with ℓ' being ℓ restricted to transitions in $T \setminus T_{\mathsf{fst}(\mathcal{I})}$, is a well-stratified* pBN *with respect to the set of indexes $\mathcal{I} \setminus \{\mathsf{fst}(\mathcal{I})\}$.*

Proof. $\mathcal{C}(N, \mathsf{fst}(\mathcal{I})) = \langle S_{\mathsf{fst}(\mathcal{I})}, T_{\mathsf{fst}(\mathcal{I})}, F_{\mathsf{fst}(\mathcal{I})}, I \cap S_{\mathsf{fst}(\mathcal{I})} \times T_{\mathsf{fst}(\mathcal{I})}, \ell_{\mathsf{fst}(\mathcal{I})} \rangle$ is a pBN, as it is prescribed by ~~the definition of well-stratifiedness~~definition 12.

Assume now that N' is not well-stratified. This implies that the set of indexes $\mathcal{I} \setminus \{1\}$ cannot be totally ordered. Hence, $\mathcal{I}$ is not totally ordered, giving a contradiction. □

We introduce now the notion of *key* for a transition in a net. The idea is to have one or more places whose unique purpose is to signal that ~~either~~ a given transition has been executed~~ or some activity involving the name of suitable transitions has happened.~~. These places will become handy when dealing with *reversing* transitions, similarly to the approach pursued in [18], or when some activity depends on the execution of other ones~~.~~, but each transition having key places must at least produce a token in a place which is not categorized as a key.

Definition 13. *Let $N = \langle S, T, F, I, \ell \rangle$ be an* LIPT*. Let $t \in T$, we say that $S' \subseteq t^\bullet$ is a set of* keys *for the transition t, written as $\mathsf{k}(t)$, if $|t^\bullet \setminus S'| \neq \emptyset$ and $\forall s \in S'.\ s^\bullet = \emptyset$ and $s \notin {}^\circ T$.*

~~**Definition 14.** *Let $N = \langle S, T, F, I, \ell \rangle$ be an* LIPT*. We say that N is* keys-complete *if for each $t \in T$ either $\mathsf{k}(t) \neq \emptyset$ or $\exists! t' \in T.\ {}^\bullet t \cap {}^\bullet t' \neq \emptyset$, $\mathsf{k}(t') \neq \emptyset \wedge \ell(t) = \ell(t')$.*~~

Example 14. *The net in Figure 13 is a* pBN *where the transitions a, b and c have key, which are those in green. More precisely s_7 is the key place of a, s_8 is the one of b and finally s_9 is the key place of c. Each transition, when fired, produces a token in a place which is not a key place, and those places are s_4, s_5 and s_6. Observe that in this net the choice of which places are key places is arbitrary, but it cannot happen that both places in the postset of a transition are categorized as key places, as Definition 13 forbids it.*

In most cases we will consider nets where each transition has just one key place, but in some peculiar cases we do need more key places, hence the key places for

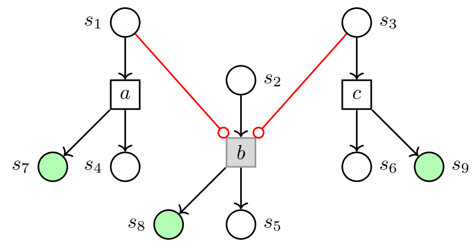

Figure 13: A ~~key complete~~ pBN where each transition has a key place.

a transitions are in general a set. The set of keys in the net N is denoted with $\mathsf{keys}(N) = \bigcup_{t \in T} \mathsf{k}(t)$.

~~In a keys-complete net if a transition does not have a key, then it shares its preset with an equally labeled transition which has a key. The set of keys in the net N is denoted with $\mathsf{keys}(N) = \bigcup_{t \in T} \mathsf{k}(t)$. A keys-complete net is a net such that removing the places representing the *keys* (which are not connected with any other transitions), the result is a net with the same firing sequences.~~ The places in $\mathsf{k}(t)$, when marked, simply implies that the transition t has been executed. As there is no transition removing that token, this information remains. ~~To each $s \in \mathsf{keys}(N)$ it is possible to assign a unique identifier which we write $\mathfrak{k}(s)$.~~

So far we have considered nets where a transition that could be interpreted as a synchronisation is not present. We characterize the transitions that, beside the key places, have the same effect of the execution of other two transitions, and that may be executed whenever the two transitions can be executed.

Definition 14. *Let $N = \langle S, T, F, I, \mathsf{m}, \ell\rangle$ be a ~~key complete~~* LIPT. *We say that $t \in T$ is a* bond *transition if there exists $t_1, t_2 \in T$ with ${}^\bullet t_1 \cap {}^\bullet t_2 = \emptyset$ and ${t_1}^\bullet \cap {t_2}^\bullet = \emptyset$ such that*

- ${}^\bullet t = {}^\bullet t_1 \cup {}^\bullet t_2$*;*
- $t^\bullet \setminus \mathsf{k}(t) = {t_1}^\bullet \setminus \mathsf{k}(t_1) \cup {t_2}^\bullet \setminus \mathsf{k}(t_2)$*; and*
- ${}^\circ t = {}^\circ t_1 \cup {}^\circ t_2$.

The transition t is a bond between the transitions t_1 and t_2, and it conveys the idea that the execution of t has the same effects as the execution of both beside the key places, which in this case will record that the bond transition t has been executed and not the two ones.

Definition 15. *Let $N = \langle S, T, F, I, \mathsf{m}, \ell\rangle$ be a ~~key complete~~* LIPT. *We say that the subset of transition $T' \subseteq T$ such that*

- *$\forall t \in T'$. t is a bond transition; and*
- ${}^\bullet(T \setminus T') \cup (T \setminus T')^\bullet = S \setminus (\bigcup_{t \in T'} \mathsf{k}(t))$*;*

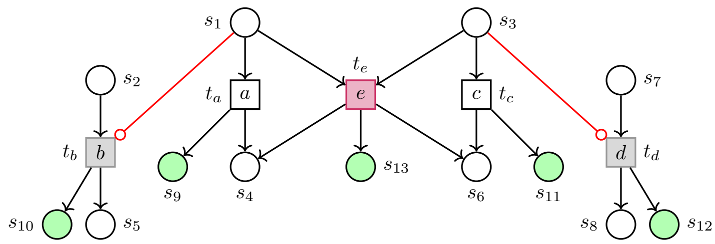

Figure 14: A ~~key-complete~~ net N with a bond transition.

is the set of bond *transitions of N and it is denoted with* $\mathsf{bond}(N)$*.*

A transition in $\mathsf{bond}(N)$ can be possibly removed without changing the reachable markings, beside the possible keys, which do not contribute to the enabling or disabling of any transition. A bond transition will be used to *bond* two independent subnets.

Example 15. *In the net depicted in Figure 14 the transition e bonds the two transition a and c: beside the key places of these two transitions, e has the same effect (without its key place). In the net N the set $\mathsf{bond}(N)$ contains just the transition e.*

We are now ready to define what a bond net is, capturing the intuition that a bond net is made by some well-stratified components which are connected using bonds.

Definition 16. *Let $N = \langle S, T, F, I, \ell\rangle$ be a ~~key-complete~~ LIPT, let $\mathsf{bond}(N) \subseteq T$ be the set of bond transitions of N, and let $S_{\mathsf{bond}(N)}$ be the set of keys of the bond transitions $\{s \in S \mid \exists t \in \mathsf{bond}(N).\ s \in \mathsf{k}(t)\}$. We say that N is a* bond *nets (BN for short) if there exists a set of indexes $\mathcal{Y}$ such that $\{T_i \mid i \in \mathcal{Y}\}$ is a partition of the set of transitions $T \setminus \mathsf{bond}(N)$ and*

- *for each $i \in \mathcal{Y}$ the net $N_i = \langle S_i, T_i, F_i, I_i, \ell_i\rangle$ is a well-stratified pBN, where $S_i = {}^\bullet T_i \cup T_i{}^\bullet \cup {}^\circ T_i$, F_i, I_i and ℓ_i are the restrictions of F, I, ℓ to transitions and places in S_i and T_i;*
- *$\langle \bigcup_{i\in\mathcal{Y}} S_i, \bigcup_{i\in\mathcal{Y}} T_i, \bigcup_{i\in\mathcal{Y}} F_i, \bigcup_{i\in\{1,\dots,k\}} I_i, \bigcup_{i\in\mathcal{Y}} \ell_i\rangle$ is the net $\langle S \setminus S_{\mathsf{bond}(N)}$, $T \setminus \mathsf{bond}(N), F', I', \ell\rangle$, where F', I' and ℓ' are F, I and ℓ restricted to transitions in $T \setminus \mathsf{bond}(N)$ and places in $S \setminus S_{\mathsf{bond}(N)}$; and*
- *for each $t \in \mathsf{bond}(N)$ there exists two indexes i, j with $i \neq j$ such that ${}^\bullet t \subseteq S_i \cup S_j$, $t^\bullet \setminus \mathsf{k}(t) \subseteq S_i \cup S_i$ and $|\,{}^\bullet t \cap S_h| = 1 = |t^\bullet \cap S_h|$, with $h \in \{i, j\}$.*

The first two conditions simply state that the net without the bond transitions and the associated keys places can be partitioned into well-stratified pBNs, and

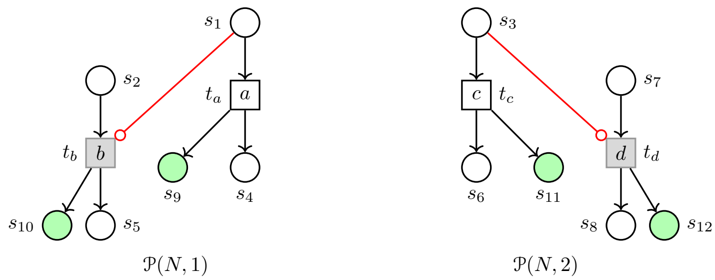

Figure 15: The two components $\mathcal{P}(N,1)$ and $\mathcal{P}(N,2)$ of the bond net N.

the last one requires that each transition in $\mathsf{bond}(N)$ bonds just two of these subnets. $\mathcal{Y}$ denotes the set of the indexes of the partition and each well-stratified subnet is identified with $\mathcal{P}(N,j)$ for $j \in \mathcal{Y}$.

Example 16. *The net in Figure 14 is a bond net whose two components are depicted in Figure 15.*

Given a bond net N, we can assume that the sets of indexes associate to the stratification of each well-stratified component $\mathcal{P}(N,j)$ is such that $\mathcal{I}_j \cap \mathcal{I}_l = \emptyset$ for each $j \neq l$~~, and with $N_{i,j}$ we denote the subnet $\langle S_{i,j}, T_{i,j}, F_{i,j}, I_{i,j}, \ell_{i,j}\rangle$ which is $\mathcal{C}(\mathcal{P}(N,j), i)$. This will ease the notation in the rest of the paper.~~.

We now turn our attention to markings.

Definition 17. *Let $N = \langle S, T, F, I, \ell\rangle$ be a* LIPT *and $\mathsf{m} : S \to \mathbb{N}$ be a marking, we say that m is feasible whenever* ~~$\mathsf{m}(s) = 1$ *then if* $s \in {}^\bullet t$ *then* $\forall s' \in t^\bullet.\ \mathsf{m}(s') = 0$ *and if* $s \in t^\bullet$ *then* $\forall s' \in {}^\bullet t.\ \mathsf{m}(s') = 0$.~~*if the place s is marked (i.e. $\mathsf{m}(s) = 1$) then the followings hold:*

- *if $s \in {}^\bullet t$ for some $t \in T$, then $\forall s' \in t^\bullet.\ \mathsf{m}(s') = 0$; and*
- *if $s \in t^\bullet$ for some $t \in T$, then $\forall s' \in {}^\bullet t.\ \mathsf{m}(s') = 0$.*

A feasible marking, which is not necessarily a reachable one, is such that if a marked place is in the preset of a transition t then the postset of the transition is unmarked and vice versa.

With abuse of notation, we will call N both the net $\langle S, T, F, I, \ell\rangle$ and the marked one $(\langle S, T, F, I, \ell\rangle, \mathsf{m})$.

Definition 18. *Let $N = (\langle S, T, F, I, \ell\rangle, \mathsf{m})$ be a marked* LIPT *where m is a feasible marking. We way that N is* acyclic *if each transition can be executed just once in each firing sequence.*

Observe that this notion, differently from what happens in nets like occurrence net, is not a syntactic one, but rather a semantic one.

Proposition 3. *Let $N = (\langle S, T, F, I, \ell\rangle, \mathsf{m})$ be a* BN *where* m *is a feasible marking. Then* N *is acyclic.*

Proof. If N is just a simple pBN the thesis follows, and acyclicty scales to well-stratified nets as each component of a well-stratified net is trivially inhibited by someone preceding it in the order. If the net N has k well-stratified components acyclicity follows observing that each component is acyclic and the transitions connecting them (the bond transitions) preserve the acyclicity. □

~~We now introduce a way to forbid that certain transitions bearing a label to happen.~~We now introduce a way to forbid the happening of transitions bearing a specific label.

Definition 19. *Let $N = (\langle S, T, F, I, \ell\rangle, \mathsf{m})$ be a* LIPT *and $L \subseteq \mathsf{L}$ is a subset of labels. Then $N \setminus L = (\langle S', T, F, I', \ell\rangle, \mathsf{m}')$ where $S' = S \uplus \{s_a \mid a \in L\}$, $I' = I \uplus \{(s_a, t) \mid \ell(t) = a\}$ and $\mathsf{m}' = \mathsf{m} \uplus \{s_a \mid a \in L\}$ is net N restricted to transitions not labeled in L.*

The idea is rather simple: instead of removing the transitions that have not to execute, as it is usually done, we inhibit them permanently. This is possible since there is no transition removing the token from one of these newly introduced and marked places, and these places are connected only with inhibitor arcs to suitable transitions.

Proposition 4. *Let $N = (\langle S, T, F, I, \ell\rangle, \mathsf{m})$ be a* LIPT*, $L \subseteq \mathsf{L}$ be a subset of labels, and let $t \in T$ be a transition such that $\ell(t) \in L$ and there exists a marking $m \in \mathcal{M}_N$ such that $m\,[t\rangle$. Consider the net $N \setminus L = (\langle S', T, F, I', \ell\rangle, , \mathsf{m}')$ as defined in Definition 19. Then $\forall m \in \mathcal{M}_{N\setminus L}$ it holds that $\neg(m\,[t\rangle\,)$.*

Proof. As already noticed, the transitions with labels in L are permanently inhibited in the net, hence the thesis follows. □

We have so far considered labeled nets with some characteristics, which we recap here. The nets have just *forward* transitions, meaning that when a token is produced in a place, it remains in that place forever, and the inhibitor arcs are used to enforce that the execution of transitions is (partially) ordered.

5.1. Covalent bond nets

~~Up to now we have considered nets which are essentially acyclic.~~ We now define *covalent* bond nets where specific transitions are allowed to (re)move keys. As keys are connected to the execution of a certain activities, these transitions either simply *undo* the activities these keys are connected with, or they undo some activities and coordinate some others.

Definition 20. *Let $N = \langle S, T, F, I, \ell\rangle$ be an* LIPT*. We say that it is a* covalent *bond net (*cBN*) if there exists a subset $U \subseteq T$ of transitions such that*

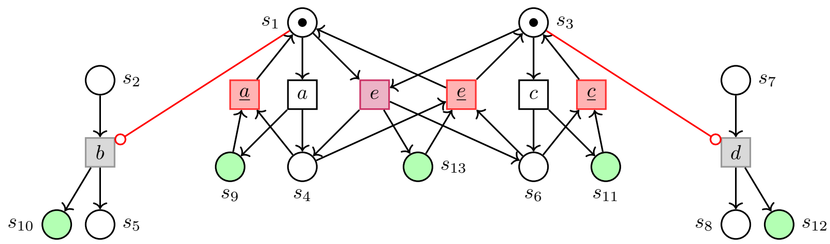

Figure 16: The net N in Figure 14 with the transitions reversing a, c and the bond e.

1. *$N' = \langle S', T', F', I', \ell' \rangle$, where $T' = T \setminus U$, $S' = {}^\bullet T' \cup T'^\bullet \cup {}^\circ T'$ and F', I' and ℓ' are the restrictions of F, I and ℓ to the transitions in T' and places in S', is a ~~key complete~~ bond net with $\mathcal{Y}$ components $\mathcal{P}(N', j) = \langle S_j, T_j, F_j, I_j, \ell_j \rangle$ with $j \in \mathcal{Y}$;*

2. *$\forall u \in U.\ {}^\bullet u \cap \mathsf{keys}(N') \neq \emptyset$;*

3. *$\forall j \in \mathcal{Y}$, $\forall t \in \mathsf{strong}(\mathcal{P}(N', j))$. $\exists! u \in U$ such that ${}^\bullet t = u^\bullet$, $t^\bullet = {}^\bullet u$ and $\ell(u) = \underline{\ell(t)}$; and*

4. *$\forall t \in \mathsf{bond}(N')$. $\exists! u \in U$ such that ${}^\bullet t = u^\bullet$, $t^\bullet = {}^\bullet u$ and $\ell(u) = \underline{\ell(t)}$.*

A covalent bond net is a bond net where the added transitions connecting the various components are either transitions reversing the effect of some forward transitions in the net, or these transitions somehow *concert* more complex activities, some of them has happened, henceforth these transitions use key-places. In Figure 16 the classical reversing transitions are depicted: these transitions simply undo either a bond or some strong transition.

The definition of covalent bond net is rather liberal but it conveys the two main features we are interested in:

- the net is formed by *sequential* components interacting together, and these components have a specific form (well-stratified nets); and
- the interaction between these components are via bonds and concerting transition, where forward interactions produce keys, and other interactions use the keys produced by these interactions.

6. CCB and Covalent Bond nets

This section presents the encoding of CCB terms as labelled cBNs, i.e., cBNs are equipped with a labelling function that maps transitions to labels. We consider the following set L of labels:

$$\begin{array}{rl} \mathsf{l} ::= & \mathsf{a} \mid \underline{\mathsf{a}} \mid (b) \mid \langle b \rangle \\ \mathsf{L} ::= & \mathsf{l} \mid \mathsf{l}, \mathsf{l} \mid \mathsf{l}, \mathsf{l}, \mathsf{l} \end{array}$$

They are basically the CCB labels, without keys., as information on keys is inferred from key places. The totally undefined labeling mapping is written as $\perp$.

For a process P, we write $\mathcal{N}(\mathtt{P})$ for the associated net $\langle S, T, F, I, \ell \rangle$ and $\mathcal{M}(\mathtt{P})$ for the associated marking. We first describe and discuss on how the net part of the encoding is defined, and then formalize the associated marking. The encoding is defined inductively on the structure of processes.

The terminated process 0 is encoded as the empty net.

Definition 21. *$\mathcal{N}(\mathtt{0}) = \langle \emptyset, \emptyset, \emptyset, \emptyset, \perp \rangle$ is the net associated to the term* 0*.*

Lemma 3. *The net $\mathcal{N}(\mathtt{0})$ is a* cBN*.*

Proof. Trivial. □

The marking associated to 0 is $\mathcal{M}(\mathtt{0}) = \emptyset$, hence the marked net associate to 0 is $(\mathcal{N}(\mathtt{0}), \mathcal{M}(\mathtt{0}))$. The marking $\emptyset$ is clearly a feasible marking.

The encoding of a prefixed process is defined in terms of the auxiliary encoding of prefixes defined below. Intuitively, the net associated with a prefix generates several transitions to encode the behaviour associated with each inference rule of the operational semantics.

Let $(s; b)$ be the prefix to be encoded, then the set of transitions associated with rule $\textsc{str}_s$ are the following:

$$T_{\textsc{str}_s}(s; b) = \{\langle \mathtt{a}, i \rangle \mid \mathtt{a} \in \mathcal{S} \wedge 1 \le i \le s(\mathtt{a})\}$$

Basically, the occurrence of a strong action a in s is mapped to a transition $\langle \mathtt{a}, i \rangle$, where the natural $i \in \mathbb{N}$ is used to disambiguate different occurrences of the same action a in s. A transition $\langle \mathtt{a}, i \rangle$ has the place $^*\langle \mathtt{a}, i \rangle$ as its pre set and the places $\langle \mathtt{a}, i \rangle^*$ and $\langle \underline{\mathtt{a}}, i \rangle^\dagger$ as its post set, i.e.,

$$^\bullet\langle \mathtt{a}, i \rangle = \{^*\langle \mathtt{a}, i \rangle\} \qquad \langle \mathtt{a}, i \rangle^\bullet = \{\langle \mathtt{a}, i \rangle^*, \langle \underline{\mathtt{a}}, i \rangle^\dagger\} \tag{1}$$

The place $\langle \mathtt{a}, i \rangle^*$ indicates that the i-th occurrence of a has been fired, while $\langle \underline{\mathtt{a}}, i \rangle^\dagger$ states that the key has been originated from the execution of a (in contraposition to keys obtained by promotion or moves, as will be discussed later). Besides, we do not associate any inhibitor arc with this transitions, i.e.,

$$^\circ\langle \mathtt{a}, i \rangle = \emptyset$$

The labels associated to those transitions are the associated actions, i.e.,

$$\ell\langle \mathtt{a}, i \rangle = \mathtt{a}$$

For the sake of simplicity, the arcs associated to these transitions are denoted as $F_{\textsc{str}_s}$ (the set of inhibitor arcs for these transition is empty), and the labeling mapping for these transitions $\ell_{\textsc{str}_s}$.

The set of transitions associated to the instances of the rule STR_w is as follows.

$$T_{\text{STR}_w}(s;b) = \{\langle c, 1\rangle \mid c \in \mathcal{W} \cap s\}$$

We remark that this set can have at most one transition, since at most one weak action can be in the strong part of a prefix. The pre and post sets of the transitions are defined analogously, i.e.,

$$^{\bullet}\langle c, 1\rangle = \{^{*}\langle c, 1\rangle\} \qquad \langle c, 1\rangle^{\bullet} = \{\langle c, 1\rangle^{*}, \langle \underline{c}, 1\rangle^{\dagger}\}$$

However, the set of inhibitor places is not empty, i.e., this transition will be enabled after every other strong action in the prefix have been already executed

$$^{\circ}\langle c, 1\rangle = \{^{*}\langle \mathtt{a}, i\rangle \mid \mathtt{a} \in \mathcal{S} \wedge 1 \leq j \leq s(\mathtt{a})\}$$

Also in this case the label is the action, i.e., $\ell\langle c, 1\rangle = \mathtt{c}$. F_{STR_w} and I_{STR_w} are the flow arcs and inhibitor arcs for these transitions, and the labeling is ℓ_{STR_w}.
The set of transitions for mimicking STR_m is

$$T_{\text{STR}_m}(s;b) = \{\langle c, \mathtt{a}, i\rangle \mid c \in \mathcal{W} \cap s \wedge \mathtt{a} \in \mathcal{S} \wedge 1 \leq i \leq s(\mathtt{a})\}$$

A transition $\langle c, \mathtt{a}, i\rangle$ represents the execution of the weak action c in the strong part of the prefix followed by a move of the generated bond to the strong action $\mathtt{a}$.

$$^{\bullet}\langle c, \mathtt{a}, i\rangle = \{^{*}\langle \mathtt{a}, i\rangle\} \qquad \langle c, \mathtt{a}, i\rangle^{\bullet} = \{\langle \mathtt{a}, i\rangle^{*}\}$$

In addition, we use inhibitor arcs to enforce the availability of c, i.e.,

$$^{\circ}\langle c, \mathtt{a}, i\rangle = \{\langle c, 1\rangle^{*}\}$$

The pre set and post set of $\langle c, \mathtt{a}, i\rangle$ resemble those of $\langle \mathtt{a}, i\rangle$; however, we do not produce a key, to recall that the bond is due to a move and hence invalidated, i.e., $\dagger k$. In this case, the label corresponds to the weak action, i.e., $\ell\langle c, \mathtt{a}, i\rangle = c$ and F_{STR_m} and I_{STR_m} are the flow arcs and inhibitor arcs for these transitions, whereas ℓ_{STR_m} is the labeling mapping.

The set of transitions for mimicking WK is

$$T_{\text{WK}}(s;b) = \{\langle b, 0\rangle\}$$

The pre, post and inhibitor sets are, as expected,

$$^{\bullet}\langle b, 0\rangle = \{^{*}\langle b, 0\rangle\} \qquad \langle b, 0\rangle^{\bullet} = \{\langle b, 0\rangle^{*}, \langle \underline{b}, 0\rangle^{\dagger}\}$$

and

$$^{\circ}\langle b, 0\rangle = \{^{*}\langle \mathtt{a}, i\rangle \mid \mathtt{a} \in \mathcal{A} \wedge 1 \leq j \leq s(\mathtt{a})\}$$

In this case, the label corresponds to the weak action, i.e., $\ell\langle b, 0\rangle = b$. F_{WK} and I_{WK} are the flow arcs and inhibitor arcs for these transitions, ℓ_{WK} the labeling mapping.

Observe that this is a weak transition.

Example 17. *Consider the prefix* $(\mathtt{a}, c; b)$. *The encoding of transition rules corresponding to forward transitions is shown in fig. 17. The transition* $\langle \mathtt{a}, 1\rangle$

$$T_{\text{STR}_s}(\mathtt{a}, c; b) = \{\langle \mathtt{a}, 1\rangle\}$$
$$T_{\text{STR}_w}(\mathtt{a}, c; b) = \{\langle c, 1\rangle\}$$
$$T_{\text{STR}_m}(\mathtt{a}, c; b) = \{\langle c, \mathtt{a}, 1\rangle\}$$
$$T_{\text{WK}}(\mathtt{a}, c; b) = \{\langle b, 0\rangle\}$$

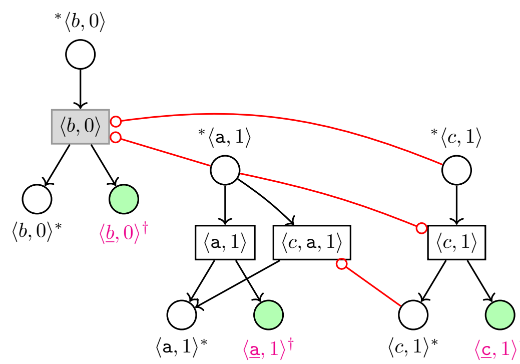

Figure 17: Forward transitions for the encoding of $(\mathtt{a}, c; b)$

accounts for the execution of the strong action $\mathtt{a}$ *in the prefix. The transition* $\langle \mathtt{b}, 0\rangle$ *corresponding to the execution of the weak action* b *is quite similar except for the inhibitor arc that prevents its firing if the strong action* $\mathtt{a}$ *has not be fired. The case in which* c *is fired when* $\mathtt{a}$ *has not, is modeled by the transition* $\langle c, \mathtt{a}, 1\rangle$, *that basically accounts for the execution of* c *followed by a move of the bond from the weak action* c *to* $\mathtt{a}$, *which produces effects analogous to the execution of* $\mathtt{a}$. *Finally, the transition* $\langle b, 0\rangle$ *corresponds to the execution of the weak prefix* b, *which is prevented if some of the actions in the strong part of the prefix (i.e.,* $\mathtt{a}$ *and* c*) has not yet been ~~already~~ executed.*

Lemma 4. *Let* $(s; b)$ *be a CCB prefix. Then the* LIPT $\langle S, T, F, I, \ell\rangle$ *where*

- $S = \{{}^*\langle \mathtt{a}, i\rangle, \langle \mathtt{a}, i\rangle^*, \langle \underline{\mathtt{a}}, i\rangle^\dagger \mid \mathtt{a} \in \mathcal{S} \wedge 1 \leq i \leq s(\mathtt{a})\} \cup \{{}^*\langle c, 1\rangle, \langle c, 1\rangle^*, \langle \underline{c}, 1\rangle^\dagger \mid c \in \mathcal{W} \cap s\} \cup \{{}^*\langle b, 0\rangle, \langle b, 0\rangle^*, \langle \underline{b}, 0\rangle^\dagger\}$;
- $T = T_{\text{STR}_s} \cup T_{\text{STR}_w} \cup T_{\text{STR}_m} \cup T_{\text{WK}}$;
- $F = F_{\text{STR}_s} \cup F_{\text{STR}_w} \cup F_{\text{STR}_m} \cup F_{\text{WK}}$;
- $I = I_{\text{STR}_w} \cup I_{\text{STR}_m} \cup I_{\text{WK}}$; *and*
- $\ell = \ell_{\text{STR}_s} \cup \ell_{\text{STR}_w} \cup \ell_{\text{STR}_m} \cup \ell_{\text{WK}}$

is a CBN.

Proof. To prove that the net the LIPT $N = \langle S, T, F, I, \ell\rangle$ is indeed a CBNit is enough to show that it is a ~~key-complete~~ PBN.

We first observe that $\mathsf{weak}(N) = \{\langle b, 0\rangle\}$ and its inhibitor set ${}^\circ\langle b, 0\rangle$ is $\{{}^*\langle \mathtt{a}, i\rangle \mid \mathtt{a} \in \mathcal{S} \wedge 1 \leq i \leq s(\mathtt{a})\} \cup \{{}^*\langle c, 1\rangle\}$. Each transition in $T_{\text{STR}_s} \cup T_{\text{STR}_w} \cup$

$T_{\text{STR}_m} \cup T_{\text{WK}}$ has just a place in their pre set, and two transition sharing the pre set are equally labeled. Each transition in T_{STR_s} (labeled in L_S) has an empty inhibitor set, and finally each transition $t \in$ ~~T~~$T_{\text{STR}_s} \cup T_{\text{STR}_w} \cup T_{\text{WK}}$ has a key place. Therefore N is indeed a ~~key-complete~~ pBN. □

Example 18. *Consider the* cBN *depicted in Figure 17, the initial marking has places* $^*\langle \mathsf{a}, 1\rangle, ^*\langle c, 1\rangle$ *and* $^*\langle b, 0\rangle$ *marked as in* $(\mathsf{a}, c; b)$ *none of the action has been executed. The execution of the strong action* a *leads to removing the token from place* $^*\langle \mathsf{a}, 1\rangle$*, marking the places* $\langle \mathsf{a}, 1\rangle^*$ *and* $\langle \mathsf{a}, 1\rangle^\dagger$*. This correspond to* $(\mathsf{a}[k], c; b)$ *where* k *correspond uniquely to* $\langle \mathsf{a}, 1\rangle^\dagger$*.*

So far we have considered just the transitions performing *forward* actions~~.~~, beside the move actions, where a weak action is executed and it is reversed and its key is moved to a strong action, which will result as performed though its key is invalidated. We now focus on the encoding of transitions corresponding to *reverse* actions. We start from rule $\underline{\text{STR}}$ obtaining the set of transitions

$$T_{\underline{\text{STR}}}(s; b) = \{\langle \underline{\mathsf{a}}, i\rangle \mid \mathsf{a} \in \mathcal{A} \wedge 1 \le i \le s(\mathsf{a})\}$$

The definitions of the pre and post sets are as expected.

$$^\bullet\langle \underline{\mathsf{a}}, i\rangle = \{\langle \mathsf{a}, i\rangle^*, \langle \underline{\mathsf{a}}, i\rangle^\dagger\} \quad \langle \underline{\mathsf{a}}, i\rangle^\bullet = \{^*\langle \mathsf{a}, i\rangle\}$$

The inhibitor arc just checks that the reversal without move can only proceed only if no weak action has been executed as part of either the strong or the weak prefix, i.e.,

$$^\circ\langle \underline{\mathsf{a}}, i\rangle = \{\langle c, 1\rangle^* \mid c \in \mathcal{W} \cap s\} \cup \{\langle b, 0\rangle^*\}$$

Labels keep track of the performed reverse action, i.e., $\ell\langle \underline{\mathsf{a}}, i\rangle = \underline{\mathsf{a}}$. $F_{\underline{\text{STR}}}$ and $I_{\underline{\text{STR}}}$ are the flow arcs and inhibitor arcs for these transitions, $\ell_{\underline{\text{STR}}}$ the labeling mapping.

The transitions of the reversal of a strong action combined with a promotion and a move discriminate situations in which the key that is being promoted or ~~the~~ moved is not bound ~~every~~anywhere else in the system from the cases in which the key appears within other parallel process.

The transitions associated with the rule $\underline{\text{STR}}_p$, i.e., a reversal followed by a promotion, are of the kind $\langle \underline{\mathsf{a}}, i, (b)\rangle$ and they stand for the cases in which the promoted key is free while (the other cases are handled by concert rules, which will be described when encoding the composition). Henceforth we have

$$T_{\underline{\text{STR}}_p}(s; b) = \{\langle \underline{\mathsf{a}}, i, (b)\rangle \mid 1 \le i \le s(\mathsf{a})\}$$

and the pre sets, post sets and inhibitor sets are as follows

$$^\bullet\langle \underline{\mathsf{a}}, i, (b)\rangle = \{\langle \underline{\mathsf{a}}, i\rangle^\dagger, \langle b, 0\rangle^*, \langle b, 0\rangle^\dagger\} \quad \langle \underline{\mathsf{a}}, i, (b)\rangle^\bullet = \{^*\langle b, 0\rangle\} \quad ^\circ\langle \underline{\mathsf{a}}, i, (b)\rangle = \emptyset$$

Recall that the key of b is promoted to an invalid key of a, and for this reason there is no key place in the post set of this transition, and the key in a is removed. In this case, labels record the reversed action, i.e., $\ell\langle\underline{\mathsf{a}}, i, (b)\rangle = \underline{\mathsf{a}}$. $F_{\underline{\mathrm{STR}}_p}$ and $I_{\underline{\mathrm{STR}}_p}$ are the flow arcs and inhibitor arcs for these transitions, $\ell_{\underline{\mathrm{STR}}_p}$ the labeling mapping.

The last case, in which the reversal of a strong action is combined with a move is defined analogously, the only difference being that the key moved is the one of c, which is in the pre set of these transitions.

$$T_{\underline{\mathrm{STR}}_m}(s; b) = \{\langle\underline{\mathsf{a}}, i, c\rangle \mid \mathsf{a} \in \mathcal{S} \wedge 1 \leq i \leq s(\mathsf{a}) \wedge c \in \mathcal{W} \cap s\}$$

The post sets, pre sets and inhibitor sets are

$$^{\bullet}\langle\underline{\mathsf{a}}, i, c\rangle = \{\langle\underline{\mathsf{a}}, i\rangle^{\dagger}, \langle c, 1\rangle^{*}, \langle c, 1\rangle^{\dagger}\} \quad \langle\underline{\mathsf{a}}, i, c\rangle^{\bullet} = \{^{*}\langle c, 1\rangle\} \quad ^{\circ}\langle\underline{\mathsf{a}}, i, c\rangle = \{\langle b, 0\rangle^{*}\}$$

The inhibitor arc is used to give priority to the promotion over the move. Also in this case, labels record the reversed action, i.e., $\langle\underline{c}, \underline{\mathsf{a}}, i\rangle = \underline{\mathsf{a}}$. $F_{\underline{\mathrm{STR}}_m}$ and $I_{\underline{\mathrm{STR}}_m}$ are the flow arcs and inhibitor arcs for these transitions, $\ell_{\underline{\mathrm{STR}}_m}$ the labeling mapping.

Example 19. *Consider the prefix* $(\mathsf{a}, c; b)$ *introduced in example 17. The transitions corresponding to reverse actions are shown in fig. 18. The reversal of*

$$\begin{aligned} T_{\underline{\mathrm{STR}}}(\mathsf{a}, c; b) &= \{\langle\underline{\mathsf{a}}, 1\rangle, \langle\underline{c}, 1\rangle\} \\ T_{\underline{\mathrm{STR}}_p}(\mathsf{a}, c; b) &= \{\langle\underline{\mathsf{a}}, i, (b)\rangle\} \\ T_{\underline{\mathrm{STR}}_m}(\mathsf{a}, c; b) &= \{\langle\underline{\mathsf{a}}, i, c\rangle\} \end{aligned}$$

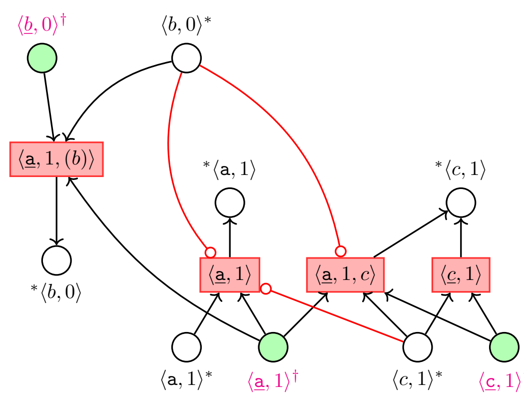

Figure 18: Backward transitions for the encoding of $(\mathsf{a}, c; b)$

the weak action c *in the strong part of a prefix is described by the transition* $\langle\underline{c}, 1\rangle$ *which consumes tokens from the places* $\langle c, 1\rangle^{*}$ *and* $\langle c, 1\rangle^{\dagger}$ *that respectively represent the execution of* c *and its key, and produce a token in the place* $^{*}\langle c, 1\rangle$ *that enables the execution of* c*. The remaining three transitions account for the reversal of the strong action* a*: the transition* $\langle\underline{\mathsf{a}}, 1\rangle$ *models the reversal of* a *when no successive promotion or move from a weak action is available (for this reason the transition is inhibited by the places* $\langle b, 0\rangle^{*}$ *and* $\langle c, 1\rangle^{*}$*), the transition* $\langle\underline{\mathsf{a}}, 1, (b)\rangle$ *represents the reversal of* a *followed by a promotion, and* $\langle\underline{\mathsf{a}}, 1, c\rangle$ *the reversal of* a *followed by a move. In the case the promotion or the move is done,* *it should be noticed that despite the action* $\mathtt{a}$ *is undone, a promotion moves the key from the weak* b *to the action* $\mathtt{a}$ *(or from the weak* c *to the action* $\mathtt{a}$*), consequently, after reversing* $\mathtt{a}$*,* b *(or* c*) becomes enabled and* $\mathtt{a}$ *remains as executed. The reason why the transition consumes and reads the key* $\langle a, 1\rangle^{\dagger}$ *is instrumental for the encoding of the parallel composition, as it will be clear later on. In the case of* $\langle \underline{\mathtt{a}}, 1, c\rangle$*, the inhibitor arc originated in* $\langle b, 0\rangle^{*}$ *states that promotion has priority over moves.*[1]

Lemma 5. *Let* $(s; b)$ *be a* CCB *prefix. Then the* LIPT $\langle S, T, F, I, \ell\rangle$ *where*

- $S = \{{}^{*}\langle \mathtt{a}, i\rangle, \langle \mathtt{a}, i\rangle^{*}, \langle \mathtt{a}, i\rangle^{\dagger} \mid \mathtt{a} \in \mathcal{S} \wedge 1 \leq i \leq s(\mathtt{a})\} \cup \{{}^{*}\langle c, 1\rangle, \langle c, 1\rangle^{*}, \langle c, 1\rangle^{\dagger} \mid c \in \mathcal{W} \cap s\} \cup \{{}^{*}\langle b, 0\rangle, \langle b, 0\rangle^{*}, \langle b, 0\rangle^{\dagger}\}$;
- $T = T_{\text{STR}_s} \cup T_{\text{STR}_w} \cup T_{\text{STR}_m} \cup T_{\text{WK}} \cup T_{\underline{\text{STR}}} \cup T_{\underline{\text{STR}}_p} \cup T_{\underline{\text{STR}}_m}$;
- $F = F_{\text{STR}_s} \cup F_{\text{STR}_p} \cup F_{\text{STR}_m} \cup F_{\text{WK}} \cup F_{\underline{\text{STR}}} \cup F_{\underline{\text{STR}}_p} \cup F_{\underline{\text{STR}}_m}$;
- $I = I_{\text{STR}_w} \cup I_{\text{STR}_m} \cup I_{\text{WK}} \cup I_{\underline{\text{STR}}} \cup I_{\underline{\text{STR}}_m}$, *and*
- $\ell = \ell_{\text{STR}_s} \cup \ell_{\text{STR}_w} \cup \ell_{\text{STR}_m} \cup \ell_{\text{WK}} \cup \ell_{\underline{\text{STR}}} \cup \ell_{\underline{\text{STR}}_p} \cup \ell_{\underline{\text{STR}}_m}$

is a cBN.

Proof. We have already seen in Lemma 4 that if we focus on the transitions in $T = T_{\text{STR}_s} \cup T_{\text{STR}_w} \cup T_{\text{STR}_m} \cup T_{\text{WK}}$ we have an cBN. The transitions in $T_{\underline{\text{STR}}} \cup T_{\underline{\text{STR}}_p} \cup T_{\underline{\text{STR}}_m}$ are such that they consume tokens from at least a key place, hence these transitions are in U. For each transition in $t \in T_{\text{STR}_s} \cup T_{\text{STR}_w}$ there is just a transition in $u \in T_{\underline{\text{STR}}_s} \cup T_{\underline{\text{STR}}_w}$ such that ${}^{\bullet}u = t^{\bullet}$, $u^{\bullet} = {}^{\bullet}t$ and $\underline{\ell(t)} = \ell(u)$ as required. Hence the net is a cBN. □

Example 20. *In the Figure 19 the ~~ones~~nets depicted in Figure 17 and Figure 18 are put together. At the beginning the places* ${}^{*}\langle \mathtt{a}, 1\rangle, {}^{*}\langle c, 1\rangle$ *and* ${}^{*}\langle b, 0\rangle$ *are marked. The execution of* $\langle \mathtt{a}, 1\rangle$ *will remove the token from* ${}^{*}\langle \mathtt{a}, 1\rangle$ *putting it in* $\langle \mathtt{a}, 1\rangle^{*}$ *and* $\langle \mathtt{a}, 1\rangle^{\dagger}$*. At this feasible marking we can reverse the last executed transition obtaining the initial marking or we can execute the action* c *(*$\langle c, 1\rangle$*) marking* $\langle c, 1\rangle^{*}$ *and* $\langle c, 1\rangle^{\dagger}$ *and after a* move *~~the key (⟨c, a, 1⟩)~~executing* $\langle \underline{\mathtt{a}}, 1, c\rangle$ *~~which~~ has the effect of moving the key of* c *to* $\mathtt{a}$*, which will have a invalid key (the place* $\langle \mathtt{a}, 1\rangle^{*}$ *is marked but* $\langle \mathtt{a}, 1\rangle^{\dagger}$ *~~it~~ is not).*

In what follows we discuss the set of transitions corresponding to concert rules A-WK$_1$, A-WK$_2$ and A-WK$_3$.

The set of transitions induced by the rule A-WK$_1$ is

$$T_{\text{A-WK}_1}(s; b) = \{\langle (b), \underline{\mathtt{a}}, i\rangle \mid \mathtt{a} \in \mathcal{S} \wedge 1 \leq i \leq s(\mathtt{a})\}$$

[1] In original CCB moves are not allowed if a prefix has a weak part.

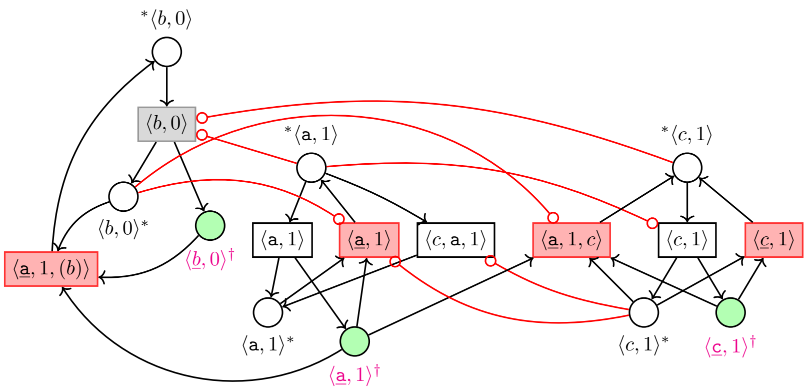

Figure 19: The net corresponding to the prefix $(\mathtt{a}, c; b)$ with forward and reverse transitions.

The inhibitor arcs for these transitions are as follows

$$^\circ\langle(b), \underline{\mathtt{a}}, i\rangle = \{^*\langle \mathtt{c}, j\rangle \mid 1 \le j \le s(\mathtt{c})\} \cup \{\langle b, 0\rangle^*\}$$

to account for the fact that the weak prefix cannot be executed until all actions in the strong part of a prefix have been executed. The pre and post sets capture the idea that the original key associated with $\mathtt{a}$ is released and $\mathtt{a}$ becomes now associated with the bond generated by the weak prefix, i.e.,

$$^\bullet\langle(b), \underline{\mathtt{a}}, i\rangle = \{\langle \underline{\mathtt{a}}, i\rangle^\dagger\} \qquad \langle(b), \underline{\mathtt{a}}, i\rangle^\bullet = \emptyset$$

The associated label records both the forward and backward actions, $\ell\langle(b), \underline{\mathtt{a}}, i\rangle = (b), \underline{\mathtt{a}}$. Again we denote these sets and the labeling mapping with $F_{\text{A-WK}_1}$, $I_{\text{A-WK}_1}$ and $\ell_{\text{A-WK}_1}$.

The transitions associated with A-WK$_2$ are

$$T_{\text{A-WK}_2}(s; b) = \{\langle c, \underline{\mathtt{a}}, i\rangle \mid \mathtt{a} \in \mathcal{S} \wedge 1 \le i \le s(\mathtt{a}) \wedge c \in \mathcal{W} \cap s\}$$

and the pre and post set are analogous to the previous case

$$^\bullet\langle c, \underline{\mathtt{a}}, i\rangle = \{\langle \underline{\mathtt{a}}, i\rangle^\dagger\} \quad \langle c, \underline{\mathtt{a}}, i\rangle^\bullet = \emptyset \quad ^\circ\langle c, \underline{\mathtt{a}}, i\rangle = \{\langle b, 1\rangle^*\}$$

The inhibitor arc for each transition just checks that c has not been executed. In this case labels record that the execution of the weak forward action and the reverse strong action in the strong part of a prefix , i.e., $\ell\langle c, \underline{\mathtt{a}}, i\rangle = \langle c\rangle, \underline{\mathtt{a}}$. $F_{\text{A-WK}_1}$, $I_{\text{A-WK}_1}$ and $\ell_{\text{A-WK}_1}$ are the flow and inhibitor arcs as well as the labeling mapping.

Finally the transitions for A-WK$_3$ do the same work, but involve three actions.

They are define as

$$T_{\text{A-WK}_3}(s;b) = \{\langle c, \underline{\mathtt{a}}, i, \mathtt{d}, j\rangle \mid \mathtt{a}, \mathtt{d} \in \mathcal{S} \land 1 \leq i \leq s(\mathtt{a}) \land 1 \leq j \leq s(\mathtt{d}) \land c \in \mathcal{W} \cap s\}$$

The inhibitor arc for each transition ensures that c has not been executed.

$$^{\circ}\langle c, \underline{\mathtt{a}}, i, \mathtt{d}, j\rangle = \{\langle c, 1\rangle^*\}$$

The pre and post set rearrange the markings to reverse $\mathtt{a}$ and invalidate $\mathtt{d}$ by associating the key borrowed from c.

$$^{\bullet}\langle c, \underline{\mathtt{a}}, i, \mathtt{d}, j\rangle = \{\langle \mathtt{a}, i\rangle^*, \langle \underline{\mathtt{a}}, i\rangle^{\dagger}, {}^*\langle \mathtt{d}, j\rangle\} \qquad \langle c, \underline{\mathtt{a}}, i, \mathtt{d}, j\rangle^{\bullet} = \{{}^*\langle \mathtt{a}, i\rangle, \langle \mathtt{d}, j\rangle^*\}$$

Labels are as before $\ell\langle c, \underline{\mathtt{a}}, i, \mathtt{d}, j\rangle = \langle c\rangle, \underline{\mathtt{a}}$. Arcs and labeling are $F_{\text{A-WK}_3}, I_{\text{A-WK}_3}$ and $\ell_{\text{A-WK}_3}$.

$T_{\text{A-WK}_1}(\mathtt{a}, c; b) = \{\langle (b), \underline{\mathtt{a}}, i\rangle\}$
$T_{\text{A-WK}_2}(\mathtt{a}, c; b) = \{\langle c, \underline{\mathtt{a}}, i\rangle\}$
$T_{\text{A-WK}_3}(\mathtt{a}, c; b) = \emptyset$

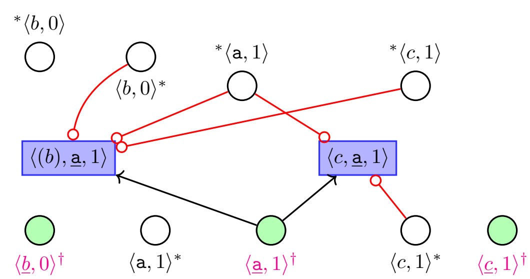

Figure 20: Concerted transitions for the encoding of $(\mathtt{a}, c; b)$

Example 21. *Consider the prefix $(\mathtt{a}, c; b)$ introduced in example 17. The transitions corresponding to concert actions are shown in fig. 20. Transition $\langle (b), \underline{\mathtt{a}}, i\rangle$ accounts for a prefix contributing with the reversal of $\mathtt{a}$ and the forward weak action in a weak prefix, i.e., (b). It should be noted that this transition can be applied only if the weak prefix has not been executed (i.e., inhibitor arc from $\langle b, 0\rangle^*$) and no action in the strong part of a prefix is still unexecuted (i.e., the inhibitor arcs ${}^*\langle \mathtt{a}, 1\rangle$ and ${}^*\langle c, 1\rangle$). Note that the bond created by the weak action will be promoted to the just reversed action. For this reason, the firing of the transition just consumes the key $\langle \underline{a}, i\rangle^{\dagger}$ the marking; as before the apparently, superfluous consume/produce loop on the key is instrumental for the encoding of parallel composition. The transition $\langle c, \underline{\mathtt{a}}, i\rangle$ corresponds to a prefix contributing with a forward weak action in strong position (i.e., $\langle c\rangle$) and the reverse of $\mathtt{a}$. The reversal of the weak action c in the strong part of a prefix is described by the transition $\langle \underline{c}, 1\rangle$ and its definition is analogous.*

Lemma 6. *Let $(s;b)$ be a CCB prefix. Then the* LIPT $\langle S, T, F, I, \ell\rangle$ *where*

- $S = \{{}^*\langle \mathtt{a}, i\rangle, \langle \mathtt{a}, i\rangle^*, \langle \mathtt{a}, i\rangle^{\dagger} \mid \mathtt{a} \in \mathcal{S} \land 1 \leq i \leq s(\mathtt{a})\} \cup \{{}^*\langle c, 1\rangle, \langle c, 1\rangle^*, \langle c, 1\rangle^{\dagger} \mid c \in \mathcal{W} \cap s\} \cup \{{}^*\langle b, 0\rangle, \langle b, 0\rangle^*, \langle b, 0\rangle^{\dagger}\}$;

- $T = T_{\text{STR}_s} \cup T_{\text{STR}_w} \cup T_{\text{STR}_m} \cup T_{\text{WK}} \cup T_{\underline{\text{STR}}} \cup T_{\underline{\text{STR}}_p} \cup T_{\underline{\text{STR}}_m} \cup T_{\text{A-WK}_1} \cup T_{\text{A-WK}_2} \cup T_{\text{A-WK}_3}$;
- $F = F_{\text{STR}_s} \cup F_{\text{STR}_w} \cup F_{\text{STR}_m} \cup F_{\text{WK}} \cup F_{\underline{\text{STR}}} \cup F_{\underline{\text{STR}}_p} \cup F_{\underline{\text{STR}}_m} \cup F_{\text{A-WK}_1} \cup F_{\text{A-WK}_2} \cup F_{\text{A-WK}_3}$;
- $I = I_{\text{STR}_w} \cup I_{\text{STR}_m} \cup I_{\text{WK}} \cup I_{\underline{\text{STR}}} \cup I_{\underline{\text{STR}}_m} \cup I_{\text{A-WK}_1} \cup I_{\text{A-WK}_2} \cup I_{\text{A-WK}_3}$, *and*
- $\ell = \ell_{\text{STR}_s} \cup \ell_{\text{STR}_w} \cup \ell_{\text{STR}_m} \cup \ell_{\text{WK}} \cup \ell_{\underline{\text{STR}}} \cup \ell_{\underline{\text{STR}}_p} \cup \ell_{\underline{\text{STR}}_m} \cup \ell_{\text{A-WK}_1} \cup \ell_{\text{A-WK}_3} \cup \ell_{\text{A-WK}_3}$

is a cBN.

Proof. We have already seen in Lemma 5 that if we focus on the transitions in $T_{\text{STR}_s} \cup T_{\text{STR}_w} \cup T_{\text{STR}_m} \cup T_{\text{WK}} \cup T_{\underline{\text{STR}}} \cup T_{\underline{\text{STR}}_p} \cup T_{\underline{\text{STR}}_m}$ we have an cBN. The new added transitions manipulate basically keys, hence are in U and are no counterpart of any strong or weak transition. The thesis follows. □

We now put together all the pieces we have collected so far. We write Rules for the set of the names of rules associated with prefixes, i.e.,

$$\mathsf{Rules} = \{\text{STR}_s, \text{STR}_w, \text{STR}_m, \text{WK}, \underline{\text{STR}}, \underline{\text{STR}}_p, \underline{\text{STR}}_m, \text{A-WK}_1, \text{A-WK}_2, \text{A-WK}_3\}$$

Definition 22. *The net* $\mathcal{N}(s;b) = \langle S, T, F, I, \ell\rangle$ *is the net associated to the prefix* $(s;b)$*, where*

- $S = \{{}^*\langle \mathsf{a}, i\rangle, \langle \mathsf{a}, i\rangle^*, \langle \mathsf{a}, i\rangle^\dagger \mid \mathsf{a} \in \mathcal{S} \wedge 1 \leq i \leq s(\mathsf{a})\} \cup \{{}^*\langle c, 1\rangle, \langle c, 1\rangle^*, \langle c, 1\rangle^\dagger \mid c \in \mathcal{W} \cap s\} \cup \{{}^*\langle b, 0\rangle, \langle b, 0\rangle^*, \langle b, 0\rangle^\dagger\}$;
- $T = \bigcup_{r\in \mathsf{Rules}} T_r(s;b)$,
- ~~$F = \bigcup_{t\in T}({}^\bullet t \times \{t\}) \cup (\{t\} \times t^\bullet)$ the one defined above~~$F = \bigcup_{r\in \mathsf{Rules}} F_r(s;b)$;
- ~~$I = \bigcup_{t\in T}({}^\circ t \times \{t\})$~~$I = \bigcup_{r\in \mathsf{Rules}} I_r(s;b)$; *and*
- ℓ ~~*is as previously described.*~~ $= \bigcup_{r\in \mathsf{Rules}} \ell_r(s;b)$.

Lemma 7. *Let* $(s;b)$ *be a prefix. Then the net* $\mathcal{N}((s;b))$ *is a* cBN.

Proof. By putting together the proofs of Lemma 4, Lemma 5 and Lemma 6. □

We discuss now how to associate the marking $\mathcal{M}((s;b))$ to the prefix $(s;\beta)$ in the net $\mathcal{N}((s;b))$ where possibly $\beta = b[l]$ for some key l and some of the action in s may have a key k or an invalid key $\dagger l$. To ease the notation we write s as $s_1 + s_2 + s_3$ where s_1 are the unexecuted actions, s_2 are the actions with non-invalidated keys, and s_3 actions with invalidated keys ($\dagger$). Therefore $s_1 \in \mathbb{N}^{\mathcal{A}}$, $s_2 \in \mathbb{N}^{\mathcal{A}\times\mathcal{K}}$ which is a set and $s_3 \in \mathbb{N}^{\mathcal{A}\times\dagger\mathcal{K}}$ which is again set. Finally, given $t \in \mathbb{N}^{\mathcal{A}\times\mathcal{K}}$ (or $t \in \mathbb{N}^{\mathcal{A}\times\dagger\mathcal{K}}$), with $\tilde{t}$ we denote the multiset in $\mathbb{N}^{\mathcal{A}}$ defined as $\tilde{t}(\mathsf{a}) = \sum_{k\in\mathcal{K}} t(\mathsf{a}, k)$.

The marking (we indicate only the marked places) is then

$$\begin{aligned}\mathcal{M}((s;\beta)) &= \{{}^*\langle \mathsf{a}, i\rangle \mid 1 \leq i \leq s_1(\mathsf{a}) \wedge \mathsf{a} \in \mathcal{A}\}\\ &= \cup\{\langle \mathsf{a}, i\rangle^*, \langle \mathsf{a}, i\rangle^\dagger \mid s_1(\mathsf{a}) < i \leq s_1(\mathsf{a}) + \tilde{s_2}(\mathsf{a}) \wedge \mathsf{a} \in \mathcal{A}\}\\ &= \cup\{\langle \mathsf{a}, i\rangle^* \mid s_1(\mathsf{a}) + \tilde{s_2}(\mathsf{a}) < i \leq s_1(\mathsf{a}) + \tilde{s_2}(\mathsf{a}) + \tilde{s_3}(\mathsf{a})\}\\ &= \cup\{{}^*\langle b, 0\rangle \mid \beta = b\} \cup \{\langle b, 0\rangle^*, \langle b, 0\rangle^\dagger \mid \beta = (b, l)\}\end{aligned}$$

Observe that the transitions $\langle \mathsf{a}, i\rangle$ corresponding to an a with an invalidated key has no token in the key place $\langle \mathsf{a}, i\rangle^\dagger$. Hereafter, we will consider nets up-to permutation of natural numbers.

Example 22. *Consider the prefix* $(\mathsf{a}, c; b)$ *introduced in example 17. The net representing all the forward (see Figure 17), backward (see Figure 18) and concert (see Figure 20) transitions is depicted in Figure 21.*

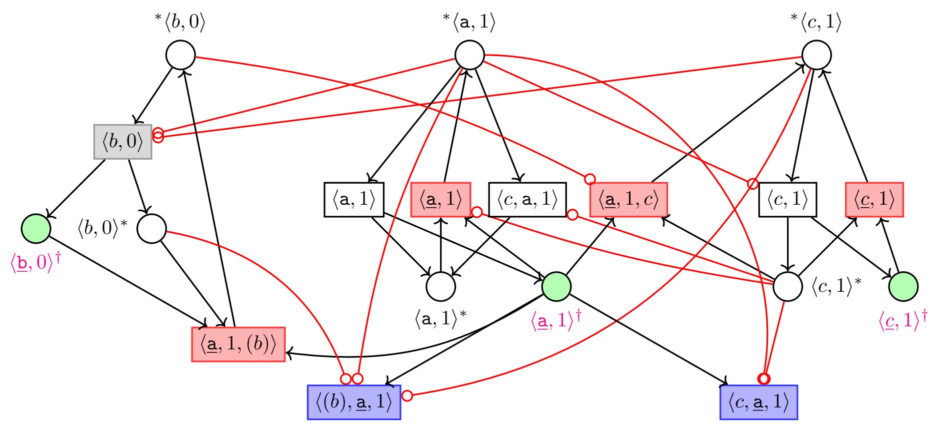

Figure 21: The overall net of $(\mathsf{a}, c; b)$

$\mathcal{M}((\mathsf{a}, c; b))$ *is such that* ${}^*\langle \mathsf{a}, 1\rangle$, ${}^*\langle c, 1\rangle$ *and* ${}^*\langle b, 0\rangle$ *are marked and all the other places do not have any token.* $\mathcal{M}((\mathsf{a}[1], c; b))$ *is, instead of, such that* $\langle \mathsf{a}, 1\rangle^*$, $\langle \mathsf{a}, 1\rangle^\dagger$, ${}^*\langle c, 1\rangle$ *and* ${}^*\langle b, 0\rangle$ *are marked, and the other places are unmarked, and* $\mathcal{M}((\mathsf{a}[\dagger 1], c; b))$ *has the places* $\langle \mathsf{a}, 1\rangle^*$, ${}^*\langle c, 1\rangle$ *and* ${}^*\langle b, 0\rangle$ *marked.*

The encoding of a prefixed process $(s; b).\mathsf{S}$ is obtained as the disjoint union of the nets encoding the prefix $(s; b)$ and the continuation S with the addition of the inhibitor arcs that prevent the execution of forward actions in S and reverse actions of $(s; b)$ as appropriate.

Definition 23. *Let* $N = \langle S, T, F, I, \ell\rangle$ *be a* cBN *such that* $|\mathcal{Y}| = 1$. *Then* $\min_{\mathcal{S}}(N) = \{t \in T \mid t \in T_{\mathcal{C}(N,1)} \setminus \mathsf{weak}(\mathcal{C}(N, 1)) \ \wedge\ \ell(t) \in \mathcal{A}\}$ *is the set of minimal strong transitions of* N *and* $\min_{\mathcal{W}}(N) = \mathsf{weak}(\mathcal{C}(N, 1))$.

The idea is that $\min_{\mathcal{S}}(N)$ are the first transitions that have to be executed in a cBN with just one component, which accounts to say that the cBN is a well-stratified one~~, beside being key-complete,~~ and $\min_{\mathcal{W}}(N)$ contains the unique weak transition of $\mathcal{C}(N, 1)$.

Proposition 5. *Let $N = \mathcal{N}(s; b)$ be the* cBN *associated to the prefix $(s; b)$. Then $\min_{\mathcal{S}}(N) = T_{\text{STR}_s} \cup T_{\text{STR}_w}$ is the set of minimal strong transitions of N, and $\min_{\mathcal{W}}(N) = T_{\text{WK}}$.*

Proof. N is a cBN and $\mathcal{Y}$ contains just one index. The transitions in $\min_{\mathcal{S}}(N)$ are precisely those with a label in $\mathcal{A}$ and that are not a weak transition, and $\min_{\mathcal{W}}(N)$ is the unique weak transition. □

With the aid of the notions of minimal transitions we can state what is the encoding of the `CCB` process $(s; b).\mathtt{S}$.

Definition 24. *Let $(s; b).\mathtt{S}$ be a `CCB` process, $\mathcal{N}(\mathtt{S}) = \langle S_{\mathtt{S}}, T_{\mathtt{S}}, F_{\mathtt{S}}, I_{\mathtt{S}}, \ell_{\mathtt{S}}\rangle$ and $\mathcal{N}(s; b) = \langle S_{(s;b)}, T_{(s;b)}, F_{(s;b)}, I_{(s;b)}, \ell_{(s;b)}\rangle$ be the nets associated with $\mathtt{S}$ and $\mathcal{N}(s; b)$. Then, $\mathcal{N}((s; b).\mathtt{S})$ is the net defined as*

- $S_{(s;b).P} = S_{\mathtt{S}} \uplus S_{(s;b)}$*;*
- $T_{(s;b).P} = T_{\mathtt{S}} \uplus T_{(s;b)}$*;*
- $F_{(s;b).P} = F_{\mathtt{S}} \uplus F_{(s;b)}$*;*
- $I_{(s;b).P} = I_{\mathtt{S}} \uplus I_{(s;b)} \;\cup\; \mathtt{Fw} \cup \mathtt{Bw}$*, where* $\mathtt{Fw} = (\{{}^*\langle \mathtt{a}, i\rangle \in S_{(s;b)}\} \cup \{\langle b, 0\rangle^*\}) \times \min_{\mathcal{S}}(\mathcal{N}(\mathtt{S}))$ *and* $\mathtt{Bw} = \min_{\mathcal{S}}(\mathcal{N}(\mathtt{S}))^{\bullet} \times T_{(s;b)}$*; and*
- $\ell_{(s;b).\mathtt{S}} = \ell_{\mathtt{S}} \uplus \ell_{(s;b)}$.

The causal constraints imposed by the prefix operator are captured by the inhibitor arcs in `Fw` and `Bw`. The set `Fw` gives the constraints for the forward execution. Intuitively, a forward transition $\langle \mathtt{c}, i\rangle$ in $\mathcal{N}(\mathtt{S})$ that is initially enabled (i.e., $\langle \mathtt{c}, i\rangle \in \min_{\mathcal{S}}(\mathcal{N}(\mathtt{S}))$) is inhibited because either (i) ~~some~~an action $\mathtt{a}$ in the strong part of the prefix s has not be fired (i.e., a place ${}^*\langle \mathtt{a}, i\rangle$ is marked) or (ii) the weak prefix b has been already fired (i.e., $\langle b, 0\rangle^*$ is marked). The constraints for reversibility are stated by `Bw`, and basically says that any transition t of the prefix is inhibited if any action in the continuation has been executed, i.e., a place in the post set of some of the minimal transitions in $\mathcal{N}(P)$ is marked.

Example 23. *In Figure 22 the forward transitions of the term $(\mathtt{a}, \mathtt{c}; b).(\mathtt{d}; e)$ are depicted, highlighting how the two components, the one corresponding to $(\mathtt{a}, \mathtt{c}; b)$ and the other to $(\mathtt{d}; e)$ are connected with the inhibitor arcs. It should be stressed that all the other transitions of $(\mathtt{a}, \mathtt{c}; b)$ are inhibited when one of $(\mathtt{d}; e)$ is executed.*

Lemma 8. *The net $\mathcal{N}(\mathtt{S})$ is an* cBN*.*

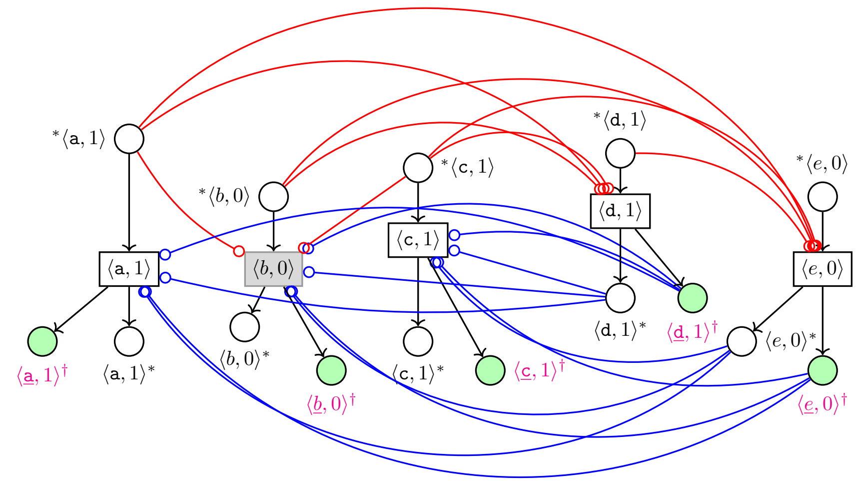

Figure 22: The net $\mathcal{N}((\mathtt{a}, \mathtt{c}; b).(\mathtt{d}; e))$ without the reversing, the move, the promotions and the concerting transitions.

Proof. The proof is by induction on the structure of $\mathtt{S}$. We prove that the net is a cBN and it has just one component. If $\mathtt{S}$ is $\mathtt{0}$ then the thesis follows ~~as $\mathcal{N}((s; b).\mathtt{0})$ is $\mathcal{N}((s; b))$ and $\mathcal{Y}$ associated to it is a singleton.~~ Consider $\mathtt{S}$ as $(s; b).\mathtt{S}'$ and by inductive hypothesis that $\mathcal{N}(\mathtt{S}')$ is a cBN and it is such that $\mathcal{Y}$ has just one element $\mathcal{N}(\mathtt{S}')$ is a well-stratified pBN ~~which is also key-complete.~~. Then $\mathcal{N}((s; b).\mathtt{S}')$ is by construction a well-stratified pBN, and then it is a cBN as required. □

The marking associated to $(s; \beta).\mathtt{S}$, where again $\beta = b[l]$ for some key l and some of the action in s may have a key k or an invalid key $\dagger l$, is then the union of $\mathcal{M}((s; \beta))$ and $\mathcal{M}(\mathtt{S})$ which is well defined as the places of the net are disjoint.

Example 24. *Consider the net partially shown in Figure 22 where just the forward transitions of* $(\mathtt{a}, \mathtt{c}; b).(\mathtt{d}; e)$ *are shown, and consider* $(\mathtt{a}[1], \mathtt{c}[2]; b).(\mathtt{d}[3]; e)$. *In this case the marked places are* ${}^*\langle b, 0\rangle$, ${}^*\langle e, 0\rangle$, $\langle \mathtt{a}, 1\rangle^*$, $\langle \mathtt{c}, 1\rangle^*$, $\langle \mathtt{d}, 1\rangle^*$, $\langle \mathtt{a}, 1\rangle^\dagger$, $\langle \mathtt{c}, 1\rangle^\dagger$ *and* $\langle \mathtt{d}, 1\rangle^\dagger$.

It should be underlined that $\mathit{std}(\mathtt{S})$ is defined by considering all the places of the kind ${}^*\langle x, i\rangle$ marked in $\mathcal{M}(\mathtt{S})$, where x is the name of an action~~.~~ and this gives a straightforward implementation of the predicate $\mathit{std}(\mathtt{S})$. ~~We come now to the how to encode the rules for composition of CCB terms.~~It remains to discuss how the rules for composition of CCB terms are encoded. We write

$\gamma(t_1, t_2)$ in lieu of $\gamma(\ell(t_1), \ell(t_2))$. Given two disjoint nets N_1 and N_2 defined as $N_i = \langle S_i, T_i, F_i, I_i, \ell_i\rangle$, we write $(N_1 \| N_2)$ for the set of transitions for forward synchronisations, aka bonds, is defined as

$$T_{(N_1\|N_2)} = \{\langle t_1 \| t_2\rangle \mid t_1 \in T_1 \wedge t_2 \in T_2 \wedge \gamma(t_1, t_2)\downarrow\}$$

The function $\mathsf{bk}(\_)$ takes a place, an action and a possible synchronisation (bond) and gives a new key place stating that the synchronisation (bond) has been executed and the place ~~involved~~ concerns that action, and otherwise it gives the place it received in input:

$$\mathsf{bk}(p, \mathsf{a}, \alpha) = \begin{cases} \alpha\langle \mathsf{a}, i\rangle^\dagger & \textit{if } p = \langle \underline{\mathsf{a}}, i\rangle^\dagger \wedge i > 0 \\ p & \textit{otherwise} \end{cases}$$

This mapping transforms a key $\langle \underline{\mathsf{a}}, i\rangle^\dagger$ (representing an unbound key) by another one $\alpha\langle \underline{\mathsf{a}}, i\rangle^\dagger$ representing a fresh key for the action a generated by the synchonisation α. We also use $\mathsf{bk}(\_)$ for its obvious extension to set of places.

Therefore, taking the bond $\langle t_1 \| t_2\rangle \in T_{(N_1\|N_2)}$ we connect it to the places in N_1 and N_2 as follows:

$$\begin{aligned} {}^\bullet\langle t_1\|t_2\rangle &= {}^\bullet t_1 \cup {}^\bullet t_2 \\ \langle t_1\|t_2\rangle^\bullet &= \mathsf{bk}({t_1}^\bullet, \ell(t_1), \langle t_1\|t_2\rangle) \cup \mathsf{bk}({t_2}^\bullet, \ell(t_2), \langle t_1\|t_2\rangle) \\ {}^\circ\langle t_1\|t_2\rangle &= {}^\circ t_1 \cup {}^\circ t_2 \end{aligned}$$

Every transition $t_1\|t_2$ represents the synchronisation of two forward actions, one from each net. It basically describes the jointly execution of the two transitions. For this reason, the pre, post and inhibitor set roughly corresponds to the union of the respective set, with the amendment of generating a new pair of keys, i.e., if $\langle \mathsf{a}, j\rangle^\dagger \in {t_i}^\bullet$, with $i \in \{1, 2\}$ that key is represented by $(t_1\|t_2)\langle \mathsf{a}, j\rangle^\dagger$. In this way we will distinguish among the concurrent execution of t_1 and t_2 from their synchronised execution $t_1\|t_2$ by associating a different keys of one with respect to the other.

Example 25. *Consider* $\mathtt{S}_i = (\mathsf{a}_i, c_i; b_i)$ *for* $i = 1, 2$ *and assume a synchronisation algebra* γ*, defined such that* $\gamma(\mathsf{a}_1, \mathsf{a}_2) = \mathsf{a}$*,* $\gamma(b_1, b_2) = b$ *and* $\gamma(c_1, c_2) = c$*. Let* $N_i = \mathcal{N}(\mathtt{S}_i)$*, which are isomorphic to* $\mathcal{N}((\mathsf{a}, c; b))$ *in the example example 17. Then,*

$$T_{(N_1\|N_2)} = \{\langle\langle \mathsf{a}_1, 1\rangle\|\langle \mathsf{a}_2, 1\rangle\rangle, \langle\langle c_1, 1\rangle\|\langle c_2, 1\rangle\rangle, \langle\langle c_1, 1\rangle\|\langle c_2, \mathsf{a}_2, 1\rangle\rangle, \langle\langle c_1, \mathsf{a}_1, 1\rangle\|\langle c_2, 1\rangle\rangle\}$$

we use to keys for this transitions, one to be use to promote on the left and on *other on the right.*

$$\alpha = \langle\langle \mathtt{a}_1, 1\rangle \| \langle \mathtt{a}_2, 1\rangle\rangle : \quad \begin{aligned} {}^\bullet\alpha &= \{{}^*\langle \mathtt{a}_1, 1\rangle, {}^*\langle \mathtt{a}_2, 1\rangle\} \\ \alpha^\bullet &= \{\langle \mathtt{a}_1, 1\rangle^*, \langle \mathtt{a}_2, 1\rangle^*, \alpha\langle \mathtt{a}_1, 1\rangle^\dagger, \alpha\langle \mathtt{a}_2, 1\rangle^\dagger\} \\ {}^\circ\alpha &= \emptyset \end{aligned}$$

$$\alpha = \langle\langle c_1, 1\rangle \| \langle c_2, 1\rangle\rangle : \quad \begin{aligned} {}^\bullet\alpha &= \{{}^*\langle c_1, 1\rangle, {}^*\langle c_2, 1\rangle\} \\ \alpha^\bullet &= \{\langle c_1, 1\rangle^*, \langle c_2, 1\rangle^*, \alpha\langle c_1, 1\rangle^\dagger, \alpha\langle c_2, 1\rangle^\dagger\} \\ {}^\circ\alpha &= \{{}^*\langle \mathtt{a}_1, 1\rangle, {}^*\langle \mathtt{a}_2, 1\rangle\} \end{aligned}$$

$$\alpha = \langle\langle c_1, 1\rangle \| \langle c_2, \mathtt{a}_2, 1\rangle\rangle : \quad \begin{aligned} {}^\bullet\alpha &= \{{}^*\langle c_1, 1\rangle, {}^*\langle \mathtt{a}_2, 1\rangle\} \\ \alpha^\bullet &= \{\langle c_1, 1\rangle^*, \alpha\langle c_1, 1\rangle^\dagger, \langle \mathtt{a}_2, 1\rangle^*\} \\ {}^\circ\alpha &= \{{}^*\langle \mathtt{a}_1, 1\rangle, \langle c_2, 1\rangle^*\} \end{aligned}$$

The last one, i.e., $\langle c_1, \mathtt{a}_1, 1 \| c_2, 1\rangle$ *is defined analogously. The labeling of these transitions is*

$$\ell\langle t \| t'\rangle = \gamma(t, t')$$

Figure 23 illustrates the case for $\langle\langle \mathtt{a}_1, 1\rangle \| \langle \mathtt{a}_2, 1\rangle\rangle$*. It should be noted that* $\langle\langle \mathtt{a}_1, 1\rangle \| \langle \mathtt{a}_2, 1\rangle\rangle$ *produces in* $\alpha\langle \mathtt{a}_1, 1\rangle^\dagger$ *and* $\alpha\langle \mathtt{a}_2, 1\rangle^\dagger$ *instead of in* $\langle \underline{\mathtt{a}}_1, 1\rangle^\dagger$ *and* $\langle \underline{\mathtt{a}}_2, 1\rangle^\dagger$*, as done by the transitions* $\langle \mathtt{a}_1, 1\rangle$ *and* $\langle \mathtt{a}_1, 2\rangle$*. In this way, none of the reversing rules (fig. 20) can be applied to undone* $\mathtt{a}_1$ *and* $\mathtt{a}_2$ *independently; they need to be reversed in a coordinated way, as will be described later.*

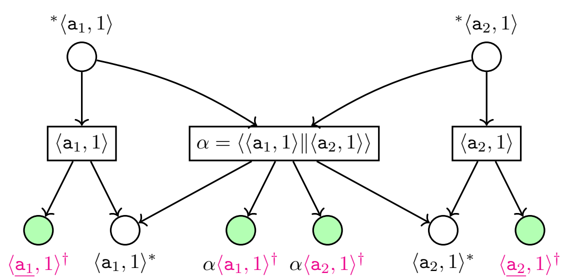

Figure 23: Forward synchronisations of $\mathtt{a}$ in $(\mathtt{a}, c; b) \| (\mathtt{a}, c; b)$

The $\mathsf{bk}(t, \mathtt{a}, \alpha)$ used previously can be considered as a kind of *template* for the concerting rules (among those there are also the reversing ones), hence we define $\mathsf{bt}(t, \mathtt{a}, \alpha)$ as the transition such that ${}^\bullet\mathsf{bt}(t, \mathtt{a}, \alpha) = \mathsf{bk}({}^\bullet t, \mathtt{a}, \alpha), \mathsf{bt}(t, \mathtt{a}, \alpha)^\bullet = \mathsf{bk}(t^\bullet, \mathtt{a}, \alpha)$ and ${}^\circ\mathsf{bt}(t, \mathtt{a}, \alpha) = \mathsf{bk}({}^\circ t, \mathtt{a}, \alpha)$. This notation will help to define the templates for the concerted action. To understand which templates should be considered, we need to understand which are the possible participants to a concerted action, which is the result of the concerted rule. Let t be a transition such that $\langle \underline{\mathtt{a}}, j\rangle^\dagger \in {}^\bullet t$, all its versions that can be considered in concerting with respect to a set of synchronisation $\mathbb{S}$ are the following:

$$\mathsf{conc}(t, \mathbb{S}) = \mathbb{T}_1 \cup \mathbb{T}_2$$

where

$$\mathbb{T}_1 = \{\mathsf{bt}(t, \mathtt{a}, \alpha) \mid \langle \underline{\mathtt{a}}, j\rangle^{\dagger} \in {}^{\bullet}t \wedge \alpha \in \mathbb{S}\}$$

$$\mathbb{T}_2 = \{\mathsf{bt}(\mathsf{bt}(t, \mathtt{a}, \alpha), c, \beta) \mid \{\langle \underline{\mathtt{a}}, j\rangle^{\dagger}, \langle c, 1\rangle^{\dagger}\} \subseteq {}^{\bullet}t \wedge \alpha, \beta \in \mathbb{S}\}$$

$\mathsf{conc}(\cdot,\cdot)$ scales to subset of transitions: $\mathsf{conc}(T, \mathbb{S}) = \bigcup_{t\in T} \mathsf{conc}(t, \mathbb{S})$. The idea is that any transition in T gives place to another version of it in which the keys corresponds to some of the possible synchronisations in $\mathbb{S}$ (the $\mathbb{T}_1$ part) and the second one ($\mathbb{T}_2$) is for transitions $\langle \underline{\mathtt{a}}, i, c\rangle$ ~~arising from the rule STR$_{m}$;~~, which may have also a bounded key that will be invalidated. The set $\mathsf{conc}(T, \mathbb{S})$ contains transitions that cannot be used, but they are sorted out when actually combining them in parallel (concerting them). For the time being, the set of candidate bonded synchronisations is:

$$T_1 \oplus T_2 = \{\langle t_1 \oplus t_2\rangle \mid t_1 \in \mathsf{conc}(T_1, T_1\|T_2) \wedge t_2 \in \mathsf{conc}(T_2, T_1\|T_2)\}$$

We will take pairs of transitions that require the same key. We say two transitions t and t' *agree on the bounded keys for* $\mathtt{a}$ and $\mathtt{b}$, written $t\{\mathtt{a} = \mathtt{b}\}t'$, if and only if $\exists \alpha, i, j.\alpha\langle \underline{\mathtt{a}}, i\rangle^{\dagger} \in {}^{\bullet}t \wedge \alpha\langle \underline{\mathtt{b}}, j\rangle^{\dagger} \in {}^{\bullet}t'$, and the α implies that they come from the same synchronization; We also say that t does not use a bond on $\mathtt{a}$, written $t\{\neq \mathtt{a}\}$, if does not use a bond key for it, i.e., $t\{\neq \mathtt{a}\} \iff \neg\exists \alpha, i.\alpha\langle \underline{\mathtt{a}}, i\rangle^{\dagger} \in {}^{\bullet}t$.

We will use the labels to filter out the allowed combinations, and we obtain the following sets of concerting transitions, which contains also the standard reversing for bonds:

$$\begin{aligned}
T_A &= \{\langle t_1 \oplus t_2\rangle \mid \ell(t_1) = \underline{\mathtt{a}} \wedge \ell(t_2) = \underline{\mathtt{b}} \wedge t\{\mathtt{a} = \mathtt{b}\}t' \wedge \gamma(\mathtt{a}, \mathtt{b}) \downarrow\}\\
T_B &= \{\langle t_1 \oplus t_2\rangle \mid \ell(t_1) = \underline{\mathtt{a}} \wedge \ell(t_2) = \mathtt{b} \wedge t_1\{\neq \mathtt{a}\} \wedge t_2\{\neq \mathtt{b}\}\}\\
T_C &= \{\langle t_1 \oplus t_2\rangle \mid \ell(t_1) = \underline{\mathtt{a}}, (b) \wedge \ell(t_2) = c \wedge t_1\{\neq \mathtt{a}\}\}\\
T_D &= \{\langle t_1 \oplus t_2\rangle \mid \ell(t_1) = \underline{\mathtt{a}}, (b) \wedge \ell(t_2) = \underline{\mathtt{d}} \wedge t_1\{\mathtt{a} = \mathtt{d}\}t_2\}\\
T_E &= \{\langle t_1 \oplus t_2\rangle \mid \ell(t_1) = \underline{\mathtt{a}}, \underline{\mathtt{d}}, (b) \wedge \ell(t_2) \in \{c, c, \langle c\rangle\} \wedge \gamma(\mathtt{a}, \mathtt{d}) \downarrow, \gamma(b, c) \downarrow\}\\
T_F &= \{\langle t_1 \oplus t_2\rangle \mid \ell(t_1) = \underline{\mathtt{a}}, (b), \beta \wedge \beta \in \{c, c, \langle c\rangle\} \wedge \ell(t_2) = \underline{\mathtt{d}} \wedge \gamma(\mathtt{a}, \mathtt{d}) \downarrow, \gamma(b, c) \downarrow\}\\
T_G &= \{\langle t_1 \oplus t_2\rangle \mid \ell(t_1) = \underline{\mathtt{a}}, (b) \wedge \ell(t_2) \in \underline{\mathtt{d}}, \beta \wedge \beta \in \{c, c, \langle c\rangle\} \wedge \gamma(\mathtt{a}, \mathtt{d}) \downarrow, \gamma(b, c) \downarrow\}
\end{aligned}$$

The above sets are related with the possible synchronisations involving reverse actions. As a matter of fact, T_A stands for the transitions corresponding to rule ($\underline{\text{COM}}$), i.e., the synchronisation of reverse actions. All the remaining ones, correspond to rule (CONCERT PAR), and are associated with the definition of the operator $\oplus$ on CCB labels. The sets T_B, T_C and T_D corresponds respectively to the first three cases in the definition of $\oplus$, while the final three corresponds to the last case in the definition of $\oplus$ (we have split the cases here for technical convenience).

Then, the set of reverse and concerted transitions is therefore

$$T_{\underline{(N_1\|N_2)}} = T_A \cup T_B \cup T_C \cup T_D \cup T_F \cup T_G \cup T_E$$

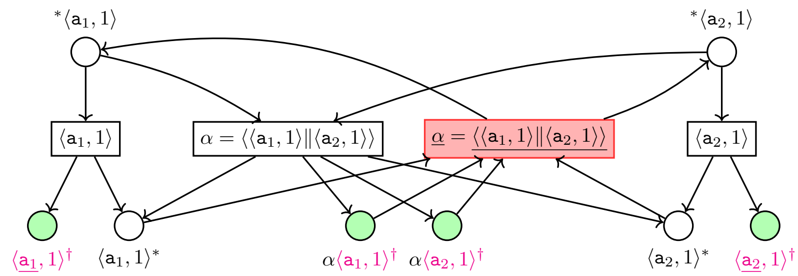

Figure 24: Reverse of the synchronisations of a in $(\mathsf{a}, c; b) \| (\mathsf{a}, c; b)$

The set of transitions T_A contains the expected reverse of a bond transition, whereas the others cases contains the transitions needed for concerting.

In all cases, the pre sets and inhibitor arcs are straightforward, i.e.,

$$^\bullet\langle t_1 \oplus t_2\rangle = {}^\bullet t_1 \cup {}^\bullet t_2 \qquad {}^\circ\langle t_1 \oplus t_2\rangle = {}^\circ t_1 \cup {}^\circ t_2$$

For the post sets, the only provision is the cases in which there exists c s.t. $\langle \underline{c}, 1\rangle^\dagger \in t_1{}^\bullet \cup t_2{}^\bullet$. Consider $(\mathsf{a}[k]; c) \| (\mathsf{a}[k]; d) \| (\mathsf{e}[j], c; d)$ where each action synchronises with itself. A concert rule allows to reach $(\mathsf{a}[\dagger l]; c) \| (\mathsf{a}; d) \| (\mathsf{e}[j], c[l]; d)$. Here the question is which key we should associate to the prefix $c[l]$. Since its not free, it cannot be reversed, hence we cannot mark $\langle c, 1\rangle^\dagger$. For this reason, we consider a further key $\varkappa\langle c, 1\rangle^\dagger$ for every weak action in the strong part of the prefix. We can now establish the post sets of the transitions in

$$\langle t_1 \oplus t_2\rangle^\bullet = \begin{cases} \mathsf{bk}(t_1{}^\bullet \cup t_2{}^\bullet, c, \varkappa) & \langle t_1 \oplus t_2\rangle \notin T_B \ \wedge \exists c.\langle c, 1\rangle^\dagger \ \in t_1{}^\bullet \cup t_2{}^\bullet \\ t_1{}^\bullet \cup t_2{}^\bullet & \end{cases}$$

Example 26. *We illustrates the case discussed above by showing what happens when considering $\langle\langle (b_1), \underline{\mathsf{a}_1}, 1\rangle \oplus \langle c_2, \underline{\mathsf{a}_2}, 1\rangle\rangle$ assuming that $\gamma(b_1, c_2)$ and $\gamma(\mathsf{a}_1, \mathsf{a}_2)$ are defined. This will generate the key place $\varkappa\langle c_2, 1\rangle^\dagger$.*

As before, abusing from notation, we have the new transitions defined as

$$\varkappa t = \begin{cases} \mathsf{bk}(t, c, \varkappa) & \textit{if} \, \exists c.\langle c, 1\rangle^\dagger \in {}^\bullet t \\ t & \textit{otherwise} \end{cases}$$

This is lifted to sets of transitions as follows

$$\varkappa T = \{\varkappa t \mid t \in (T \setminus \{\langle \underline{c}, 1\rangle \mid c \in \mathcal{W}\}) \wedge \exists c.\langle c, 1\rangle^\dagger \in {}^\bullet t\}$$

Example 27. *The Figure 24 shows a part of the net associated to $(\mathsf{a}, c; b) \| (\mathsf{a}, c; b)$ with a forward synchronization and the transition implementing the reverse of*

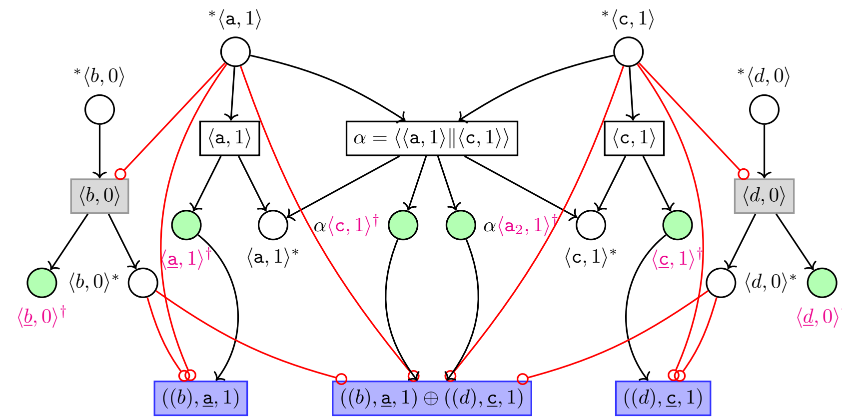

Figure 25: Concerting in $(\mathtt{a}; b)\|(\mathtt{c}; d)$ with $\gamma(\mathtt{a}, \mathtt{c}) \downarrow$ and $\gamma(b, d) \downarrow$

this transition (which belongs to T_A in $T_{\underline{(N_1\|N_2)}}$). In the Figure 25 it is shown how the transitions $((b), \underline{\mathtt{a}}, 1)$ and $((d), \underline{\mathtt{c}}, 1)$ are concerted *together (which belongs to T_G in $T_{\underline{(N_1\|N_2)}}$) where $N_1 = \mathcal{N}((\mathtt{a}; b))$ and $N_2 = \mathcal{N}((\mathtt{c}; d))$. The net $\mathcal{N}((\mathtt{a}; b)\|(\mathtt{c}; d))$ is partially depicted.*

We are now ready to show how to encode the parallel composition of two `CCB` terms.

Definition 25. *Let $\mathtt{P_1}$ and $\mathtt{P_2}$ be two CCB terms and $\mathcal{N}(\mathtt{P_i}) = N_i$ with $N_i = \langle S_i, T_i, F_i, I_i, \ell_i\rangle$ for $i = 1, 2$, the associated* cBN*s, which we assume to be disjoint (i.e., $S_1 \cap S_2 = \emptyset$ and $T_1 \cap T_2 = \emptyset$). Then $\mathcal{N}(\mathtt{P_1}\|\mathtt{P_2})$ is the net $\langle S, T, F, I, \ell\rangle$ where*

- $S = S_1 \ \cup\ S_2 \ \cup\ S_{(N_1\|N_2)} \ \cup\ S_{\underline{(N_1\|N_2)}} \cup S_{\varkappa(T_1\cup T_2)}$*; ;*
- $T = T_1 \ \cup\ T_2 \ \cup\ T_{(N_1\|N_2)} \ \cup\ T_{\underline{(N_1\|N_2)}} \cup T_{\varkappa(T_1\cup T_2)}$*;*
- $F = F_1 \ \cup\ F_2 \ \cup\ F_{(N_1,N_2)} \ \cup\ F_{\underline{(N_1\|N_2)}} \cup F_{\varkappa(T_1\cup T_2)}$*;*
- $I = I_1 \ \cup\ I_2 \ \cup\ I_{(N_1,N_2)} \ \cup\ I_{\underline{(N_1\|N_2)}} \cup I_{\varkappa(T_1\cup T_2)}$*; and*
- $\ell = \ell_1 \ \cup\ \ell_2 \ \cup\ \ell_{(N_1\|N_2)} \ \cup\ \ell_{\underline{(N_1\|N_2)}} \cup \ell_{\varkappa(T_1\cup T_2)}$

where S_{nk}, F_{nk}, I_{nk} and ℓ_{nk}, with $nk \in \{(N_1\|N_2), \underline{(N_1\|N_2)}, \varkappa(T_1 \cup T_2)\}$ are defined as illustrated above.

We have again an cBN, as shown by the following result.

Lemma 9. *Let $\mathtt{P_1}$ and $\mathtt{P_2}$ be two CCB terms and $\mathcal{N}(\mathtt{P_i}) = \langle S_i, T_i, F_i, I_i, \ell_i\rangle$ for $i = 1, 2$, the associated disjoint* cBN*s. Then $\mathcal{N}(\mathtt{P_1}\|\mathtt{P_2})$ is a* cBN.

Proof. As $\mathcal{N}(\mathtt{P_i}) = \langle S_i, T_i, F_i, I_i, \ell_i \rangle$ are CBNs, there exists for each a partition with $\mathcal{Y}_i$ for $i = 1, 2$. Now, the only requirement to check is that if we are considering a bond between the two, there is one of the reversing bond. But this is clear by constructions (these are the transition in T_A above). The thesis follows. □

The marking associated with $\mathtt{P}_1 \| \mathtt{P}_2$ cannot be straightfowardly inferred from the markings $\mathcal{M}(\mathtt{P_i})$ because of the loose of information about keys. Consider the process $(\mathtt{a}[1]; b)\|(\mathtt{a}[1]; c)$ and note that "1" is a shared key. If processes $(\mathtt{a}[1]; b)$ and $(\mathtt{a}[1]; c)$ are taken independently, such shared key would be considered as two different private ones; hence the two actions $\mathtt{a}$ would appeare as executed independently. For this reason, we ubstitute shared keys in a parallel composition $\mathtt{P}_1 \| \mathtt{P}_2$ by invalidated keys (recall that invalidated keys translates into unmarked key places). For each shared key, we record the transitions names. Then, for these pairs of names, we mark the key places of the synchronization. Given $\mathtt{P}_1 \| \mathtt{P}_2$, $\phi(\mathtt{P}_1 \| \mathtt{P}_2)$ gives the set of shared keys together with the names of the transitions in each of the nets $\mathcal{N}(\mathtt{P_i})$ they are associated to, and in $\mathtt{P}_1 \| \mathtt{P}_2$ it substitutes the shared key with an invalid one. More formally for each $k \in \mathit{key}(\mathtt{P}_1) \cap \mathit{key}(\mathtt{P}_2)$ the mapping returns the triple (k, t_1, t_2) where t_i are the transitions in $\mathcal{N}(\mathtt{P_i})$ which corresponds to the actions in $\mathtt{P}_1$ and $\mathtt{P}_2$ with that key, and at the same time invalidate the key in $\mathtt{P}_1 \| \mathtt{P}_2$. The triple (k, t_1, t_2) is used to mark the places $\alpha\langle \mathtt{a}, i\rangle^\dagger$ and $\alpha\langle \mathtt{b}, j\rangle^\dagger$, where $\mathtt{a}$ and $\mathtt{b}$ are the actions with the same key and α is $\langle t_1 \| t_2\rangle$, which are the key places added by the various bonds. Thus $\phi(\mathtt{P}_1 \| \mathtt{P}_2) = (\Omega, \mathtt{P}'_1 \| \mathtt{P}'_2)$ where Ω is the set of triples and $\widehat{\mathtt{P}_1} \| \widehat{\mathtt{P}_2}$ is the CCB term where the various shared keys have been invalidated. Hence when defining $\mathcal{M}(\mathtt{P}_1 \| \mathtt{P}_2)$ we have $\mathcal{M}(\widehat{\mathtt{P}_1}) \cup \mathcal{M}(\widehat{\mathtt{P}_2}) \cup \{\alpha(t_i)^\dagger \mid \exists (k, t_1, t_2) \in \Omega \ \wedge \ \alpha = t_1 \oplus t_2\}$ where $(t_i)^\dagger$ is the key place associated to the transition t_i.

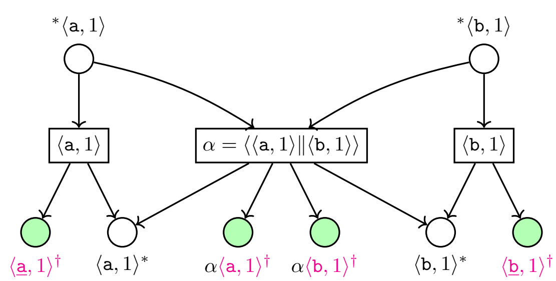

Figure 26: Part of the net $\mathcal{N}((\mathtt{a}; c)\|(\mathtt{b}; d))$

Example 28. *Consider* $(\mathtt{a}[1]; c)\|(\mathtt{b}[1]; d)$ *and assume that* γ *on* $(\mathtt{a}, \mathtt{b})$ *is defined. Then* $\phi((\mathtt{a}[1]; c)\|(\mathtt{b}[1]; d))$ *gives* $\Omega = \{(1, \langle \mathtt{a}, 1\rangle, \langle \mathtt{b}, 1\rangle\}$ *and* $(\mathtt{a}[\dagger 1]; c)\|(\mathtt{b}[\dagger 1]; d)$ *and* $\mathcal{M}((\mathtt{a}[1]; c)\|(\mathtt{b}[1]; d))$ *is* $\mathcal{M}((\mathtt{a}[\dagger 1]; c)) \cup \mathcal{M}((\mathtt{b}[\dagger 1]; d)) \cup \{\alpha\langle \mathtt{a}, 1\rangle^\dagger, \alpha\langle \mathtt{b}, 1\rangle^\dagger\}$.

The last operation we have to encode is the restriction.

Definition 26. *Let* $\texttt{P}$ *be a* CCB *term and let* $\mathcal{N}(\texttt{P}) = \langle S_{\texttt{P}}, T_{\texttt{P}}, F_{\texttt{P}}, I_{\texttt{P}}, \ell_{\texttt{P}} \rangle$ *be the associated* cBN. *Then* $\mathcal{N}(\texttt{P} \setminus L)$ *is the net* $\langle S, T, F, I, \ell \rangle$ *where* $S = S_{\texttt{P}} \uplus \{s_x \mid x \in L\}$, $T = T_{\texttt{P}}$, $F = F_{\texttt{P}}$, $\ell = \ell_{\texttt{P}}$ *and* $I = I_{\texttt{P}} \uplus \{(s_x, t) \mid x \in L \ \wedge\ \ell_{\texttt{P}}(t) = x\}$.

Obviously we have that the resulting construction is a cBN.

Lemma 10. *Let* $\texttt{P} \setminus L$ *be a* CCB *term. Then* $\mathcal{N}(\texttt{P} \setminus L)$ *is a* cBN.

The marking associated to the term $\texttt{P} \setminus L$ in the net $\mathcal{N}(\texttt{P} \setminus L)$ is $\mathcal{M}(\texttt{P})$.

We can now put together what we have seen so far and we have the following result, which summarize that our encoding gives a covalent bond net.

Theorem 1. *Let* $\texttt{P}$ *be a* CCB *term, then* $\mathcal{N}(\texttt{P})$ *is a* cBN.

We prove now that our encoding is full and faithful, meaning that each move in the transition system associated to the CCB terms is matched by a fireable transition in the encoding of the term, and if one fires a transition in the net which encodes a CCB term, then the corresponding move can be performed also on the term.

We write $\mu \equiv \mathsf{L}$ if L is obtained by removing the keys in μ. Moreover, we write $\mathcal{N}(\texttt{P}) \xrightarrow{\mathsf{L}} \mathcal{N}(\texttt{Q})$ if there exists a transition t with label L, *i.e.*, $\ell(t) = \mathsf{L}$ that is enabled at $(\mathcal{N}(\texttt{P}), \mathcal{M}(\texttt{P}))$ and $(\mathcal{N}(\texttt{Q}), \mathcal{M}(\texttt{Q}))$ is the result of firing t in $(\mathcal{N}(\texttt{P}), \mathcal{M}(\texttt{P}))$.

Theorem 2. *If* $\texttt{P}$ *is well-formed then*

1. *if* $\texttt{P} \xrightarrow{\mu} \texttt{Q}$ *then* $\mathcal{N}(\texttt{P}) \xrightarrow{\mathsf{L}} \mathcal{N}(\texttt{Q})$ *and* $\mu \equiv \mathsf{L}$; *and*

2. *if* $\mathcal{N}(\texttt{P}) \xrightarrow{\mathsf{L}} (N, \mathsf{m})$ *then there exists* $\texttt{Q}$ *such that* $(N,\ \mathsf{m}) = \mathcal{N}(\texttt{Q})$ *and* $\texttt{P} \xrightarrow{\mu} \texttt{Q}$ *and* $\mu \equiv \mathsf{L}$.

Proof. (1) It follows by induction on the structure of the derivation $\texttt{P} \xrightarrow{\mu} \texttt{Q}$. We show a few interesting cases.

(CASE STR$_s$) $\texttt{P} = (\mathsf{a}, s; b).\texttt{S}$, and $\texttt{P}' = (\mathsf{a}[k], s; b).\texttt{S}$, and $\mu = \mathsf{a}[k]$ and $\mathsf{a} \in \mathcal{S}$, and *std*($\texttt{S}$) and *fresh*$(k, s)$. Then, $(N, \mathsf{m}) = (\mathcal{N}(\mathsf{a}, s; b), \mathcal{M}(\mathsf{a}, s; b))$. By definition of the encoding there exists j such that $\langle \mathsf{a}, j \rangle \in T_{\mathrm{STR}_s}((\mathsf{a}, s; b))$ and $\ell\langle \mathsf{a}, j \rangle = \mathsf{a}$ and ${}^*\langle \mathsf{a}, j \rangle \in \mathsf{m}$. Since $\texttt{S}$ is standard, $\min_{\mathsf{S}}(N_{\texttt{S}})^{\bullet} = \emptyset$, hence ${}^{\circ}\langle \mathsf{a}, j \rangle = \emptyset$. Consequently, $(N, \mathsf{m}) \xrightarrow{\mathsf{a}} (N, \mathsf{m} \setminus \{{}^*\langle \mathsf{a}, j \rangle\} \cup \{\langle \mathsf{a}, j \rangle^*, \langle \mathsf{a}, j \rangle^{\dagger}\}) = (\mathcal{N}((\mathsf{a}[k], s; b).\texttt{S}), \mathcal{M}((\mathsf{a}[k], s; b).\texttt{S})$.

(CASE STR$_m$) $\texttt{P} = (c, \mathsf{a}, s; b).\texttt{S}$, and $\texttt{P}' = (c, \mathsf{a}[\dagger l], s; b).\texttt{S}$, and $\mu = c[k]$, and *std*($\texttt{S}$) and *fresh*(k, t). Then, $(N, \mathsf{m}) = (\mathcal{N}((c, \mathsf{a}, s; b)), \mathcal{M}((c, \mathsf{a}, s; b)))$. By definition of encoding $\langle c, \mathsf{a}, i \rangle \in T_{\mathrm{STR}_m}(c, \mathsf{a}, s; b)$ and $\ell\langle c, \mathsf{a}, i \rangle = c$ and ${}^*\langle \mathsf{a}, i \rangle \in \mathsf{m}$. Moreover, $\mathsf{m}(\langle\langle c, 1\rangle^*\rangle) = 0$ (because c appears without key in the prefix). Hence, $\langle c, \mathsf{a}, i \rangle$ is enabled. Then, $(N, \mathsf{m}) \xrightarrow{\mathsf{a}} (N, (\mathsf{m} \setminus \{{}^*\langle \mathsf{a}, i \rangle)\} \cup \{\langle \mathsf{a}, i \rangle^*)\} = (\mathcal{N}((c, \mathsf{a}[\dagger l], s; b).\texttt{S}), \mathcal{M}((c, \mathsf{a}[\dagger l], s; b).\texttt{S}))$ (note that the transition does not produce a token in the place $\langle \mathsf{a}, i \rangle^{\dagger}$, hence the key associated with $\langle \mathsf{a}, i \rangle\rangle$ is invalidated).

(Case wk) $\mathtt{P} = (t; b).\mathtt{S}$, and $\mathtt{P}' = (t; b[k]).\mathtt{S}$, and $\mu = (b)[k]$ and $std(\mathtt{S})$, and $fresh(k, t)$. By definition of encoding $\langle b, 0\rangle \in T_{\mathrm{WK}}(t; b)$ and $\ell\langle b, 0\rangle = b$ and ${}^*\langle b, 0\rangle \in \mathsf{m}$. Moreover, for all $\mathsf{a} \in \mathcal{A}$ and $1 \leq j \leq t(\mathsf{a})$ we have ${}^*\langle \mathsf{a}, i\rangle \notin \mathsf{m}$ (because $t \in \mathbb{N}^{\mathcal{A}\times\mathcal{K}}$). Hence, $\langle b, 0\rangle$ is enabled and the proof is completed as in the previous case.

(Case com) $\mathtt{P} = \mathtt{P} \parallel \mathtt{Q}$, $\mathtt{P}' = \mathtt{P}' \parallel \mathtt{Q}'$, $\mu = \gamma(\mathsf{a}, \mathsf{c})[k]$ and $\mathtt{P} \xrightarrow{\mathsf{a}[k]} \mathtt{P}'$ and $\mathtt{Q} \xrightarrow{\mathsf{c}[k]} \mathtt{Q}'$. By inductive hypothesis, $\mathcal{N}(\mathtt{P}) \xrightarrow{\mathsf{a}} \mathcal{N}(\mathtt{P}')$ and $\mathcal{N}(\mathtt{Q}) \xrightarrow{\mathsf{c}} \mathcal{N}(\mathtt{Q}')$. Hence, there are two transitions t_1 and t_2 s.t. ${}^\bullet t_1 \subseteq \mathcal{M}(\mathtt{P})$ and $\ell(t_1) = \mathsf{a}$ and ${}^\bullet t_2 \subseteq \mathcal{M}(\mathtt{Q})$ and $\ell(t_2) = \mathsf{c}$. Hence, $t_1 \| t_2 \in T_{\mathtt{P}\|\mathtt{Q}}$. By inspecting the rules that have label in $\mathcal{A}$ that are in the domain on γ, we conclude that they are transitions of the shape of T_{STR_s}, T_{STR_w}, ore $T_{\mathrm{STR}_m}(\mathsf{a}, c; b)$. Since the presets in all cases are of the form ${}^*\langle \mathsf{a}, i\rangle$, it holds that that ${}^\bullet t_i \subseteq \mathcal{M}(\mathtt{P}\|\mathtt{Q})$ implies ${}^\bullet(t_1\|t_2) \subseteq \mathcal{M}(\mathtt{P}\|\mathtt{Q})$. By definition of the encoding, it is also the case that ${}^\circ t_1 \cap \mathcal{M}(\mathtt{P}) = \emptyset$ and ${}^\circ t_2 \cap \mathcal{M}(\mathtt{Q}) = \emptyset$, which imply ${}^\circ t_1\|t_2 \cap \mathcal{M}(\mathtt{P}\|\mathtt{Q}) = \emptyset$. Hence $t_1\|t_2$ is enabled at $\mathcal{M}(\mathtt{P}\|\mathtt{Q})$. Consequently, $\mathcal{N}(\mathtt{P}\|\mathtt{Q}) \xrightarrow{\gamma(\mathsf{a},\mathsf{c})} \mathcal{N}(\mathtt{P}'\|\mathtt{Q}')$.

(Case $\underline{\text{com}}$) $\mathtt{P} = \mathtt{P} \parallel \mathtt{Q}$, $\mathtt{P}' = \mathtt{P}' \parallel \mathtt{Q}'$, $\mu = \underline{\gamma(\mathsf{a}, \mathsf{c})}[k]$ and $\mathtt{P} \xrightarrow{\underline{\mathsf{a}}[k]} \mathtt{P}'$ and $\mathtt{Q} \xrightarrow{\underline{\mathsf{c}}[k]} \mathtt{Q}'$. By inductive hypothesis, $\mathcal{N}(\mathtt{P}) \xrightarrow{\underline{\mathsf{a}}} \mathcal{N}(\mathtt{P}')$ and $\mathcal{N}(\mathtt{Q}) \xrightarrow{\underline{\mathsf{c}}} \mathcal{N}(\mathtt{Q}')$. Hence, there are two transitions t_1 and t_2 s.t. ${}^\bullet t_1 \subseteq \mathcal{M}(\mathtt{P})$ and $\ell(t_1) = \underline{\mathsf{a}}$ and ${}^\bullet t_2 \subseteq \mathcal{M}(\mathtt{Q})$ and $\ell(t_2) = \underline{\mathsf{c}}$. By inspecting the rules that have label in $\underline{\mathcal{A}}$ s.t. the forward transition is the domain on γ, we conclude that they are transitions of the shape of $T_{\underline{\mathrm{STR}}} \cup T_{\underline{\mathrm{STR}}_p} \cup T_{\underline{\mathrm{STR}}_m}$. We analyse the case in which t_1 comes from which $T_{\underline{\mathrm{STR}}_m}$ (the other follows by analogous arguments). Hence, $t_1 = \langle \underline{\mathsf{a}}, i, d\rangle$, ${}^\bullet\langle \underline{\mathsf{a}}, i, d\rangle = \{\langle \underline{\mathsf{a}}, i\rangle^\dagger, \langle d, 1\rangle^*, \langle d, 1\rangle^\dagger\}$, $\langle \underline{\mathsf{a}}, i, d\rangle^\bullet = \{{}^*\langle d, 1\rangle\}$, ${}^\circ\langle \underline{\mathsf{a}}, i, d\rangle = \{\langle b, 0\rangle^*\}$. First note that $\langle \underline{\mathsf{a}}, i\rangle^\dagger \notin \mathcal{M}(\mathtt{P}\|\mathtt{Q})$ (because the key between a and c is bound). By definition of the encoding, there should some synchronisation $\alpha \in \mathcal{N}(\mathtt{P}\|\mathtt{Q})$ such that $\alpha\langle \underline{\mathsf{a}}, i\rangle^\dagger \in \mathcal{M}(\mathtt{P}\|\mathtt{Q})$. In case $\langle d, 1\rangle^\dagger \in \mathcal{M}(\mathtt{P}\|\mathtt{Q})$, we can conclude that $\mathsf{bt}(t, \mathsf{a}, \alpha)$ is enabled at $\mathcal{M}(\mathtt{P}\|\mathtt{Q})$. If $\langle d, 1\rangle^\dagger \notin \mathcal{M}(\mathtt{P}\|\mathtt{Q})$, then, there exists a synchronisation β such that $\mathsf{bt}(\mathsf{bt}(t, \mathsf{a}, \alpha), c, \beta)$ is enabled at $\mathcal{M}(\mathtt{P}\|\mathtt{Q})$. Call such transition t_1'. By reasoning analogously for t_2, we can conclude that there is a transition t_2' enabled at $\mathcal{M}(\mathtt{P}\|\mathtt{Q})$. Consequently, there is a transition $\langle t_1' \oplus t_2'\rangle$ (coming from T_A) that is enabled at $\mathcal{M}(\mathtt{P}\|\mathtt{Q})$. Hence, we can conclude that $\mathcal{N}(\mathtt{P}\|\mathtt{Q}) \xrightarrow{\underline{\gamma(\mathsf{a},\mathsf{c})}} \mathcal{N}(\mathtt{P}'\|\mathtt{Q}')$.

(2) follows by induction on the structure of $\mathtt{P}$. ~~If $\mathtt{P}$ is a prefix then the case follows by analysis on the shape of the fired transition. For sequential composition, the proof follows by inductive hypothesis and by noting that the conditions in definition 25 ensure that a transition under a prefix can proceed only when all forward transitions in the strong prefix have been fired and the weak prefix is still enabled. For parallel composition, the case follows by analysis on the shape of the fired transition by reasoning analogously to cases (com) and $\underline{\text{com}}$ above. If $\mathtt{P}$ is a restriction, the case follows straightforwardly.~~ Let t be the fired

transition and note that $\ell(t) = \mathtt{L}$. Hence,

$$(\mathcal{N}(\mathtt{P}), \mathcal{M}(\mathtt{P})) \xrightarrow{\mathtt{L}} (\mathcal{N}(\mathtt{P}), \mathcal{M}(\mathtt{P}) \setminus {}^{\bullet}t \ \cup t^{\bullet})$$

(CASE $\mathtt{P} = \alpha.\mathtt{S}$). Then, $\mathcal{N}(\mathtt{P}) = \mathcal{N}(\alpha.\mathtt{S})$ and $\mathcal{M}(\mathtt{P}) = \mathcal{M}(\alpha.\mathtt{S}) = \mathcal{M}(\alpha) \uplus \mathcal{M}(\mathtt{S})$. By definition 24, there are two cases:

- $t \in T_\alpha$: $(\mathcal{N}(\alpha.\mathtt{S}), \mathcal{M}(\alpha.\mathtt{S})) \xrightarrow{\mathtt{L}} (\mathcal{N}(\alpha.\mathtt{S}), \mathcal{M}(\alpha) \setminus {}^{\bullet}t \ \cup t^{\bullet} \uplus \mathcal{M}(\mathtt{S}))$.

  We first note that $\min_{\mathtt{S}}(\mathcal{N}(\mathtt{S}))^{\bullet} \cap \mathcal{M}(\alpha.\mathtt{S}) = \emptyset$, otherwise t would not be enabled because of the inhibitor arcs in `Bw` from definition 24. Therefore $\mathtt{S}$ is standard. We proceed by case analysis on the shape of t. We illustrate the case in which $t \in T_{\text{STR}_s}(\alpha)$ (the remaining cases follow analogously). There must exist $\mathtt{a}$, s, b and j such that $\alpha = (\mathtt{a}, s; b)$, $\langle \mathtt{a}, j \rangle \in T_{\text{STR}_s}(\alpha)$ and $\ell\langle \mathtt{a}, j \rangle = \mathtt{a} = \mathtt{L}$ and ${}^{*}\langle \mathtt{a}, j \rangle \in \mathcal{M}(\alpha)$. Consequently,

  $$(\mathcal{N}(\alpha.\mathtt{S}), \mathcal{M}(\alpha) \uplus \mathcal{M}(\mathtt{S})) \xrightarrow{\mathtt{a}} (\mathcal{N}(\alpha.\mathtt{S}), \mathcal{M}(\alpha) \setminus \{{}^{*}\langle \mathtt{a}, j \rangle\} \cup \{\langle \mathtt{a}, j \rangle^{*}, \langle \mathtt{a}, j \rangle^{\dagger}\} \uplus \mathcal{M}(\mathtt{S}))$$

  Since $\mathtt{S}$ is standard, we have that $\alpha.\mathtt{S} = (\mathtt{a}, s; b).\mathtt{S} \xrightarrow{\mathtt{a}[k]} (\mathtt{a}[k], s; b).\mathtt{S}$ by rule (STR_s). The proof is completed by taking $\mathtt{Q} = (\mathtt{a}[k], s; b).\mathtt{S}$ and $\mu = \mathtt{a}[k]$ and noting that $\mathcal{M}(\mathtt{Q}) = \mathcal{M}(\alpha) \setminus \{{}^{*}\langle \mathtt{a}, j \rangle\} \cup \{\langle \mathtt{a}, j \rangle^{*}, \langle \mathtt{a}, j \rangle^{\dagger}\} \uplus \mathcal{M}(\mathtt{S})$.

- $t \in T_{\mathtt{S}}$. We have $\mathcal{M}(\alpha.\mathtt{S}) \cap \{{}^{*}\langle \mathtt{a}, i \rangle \in S_{(s;b)}\} \cup \{\langle b, 0 \rangle^{*}\}) = \emptyset$, as otherwise t would not be enabled because of the inhibitor arcs in `Fw`. Hence, $\alpha = (t; b)$. The proof is completed by applying inductive hypothesis and (CONT).

(CASE $\mathtt{P} = \mathtt{P}_1 \mid \mathtt{P}_2$). If $t \in T_{\mathtt{P}_i}$ then the proof follows by inductive hypothesis and rule PAR. If $t \in T_{(\mathtt{P}_1 \| \mathtt{P}_2)}$, then there exist $t_1 \in T_{\mathtt{P}_1}$ and $t_2 \in T_{\mathtt{P}_2}$ and $\gamma(t_1, t_2)\downarrow$. The proof is completed by applying inductive hypothesis on both t_1 and t_2 and rule COM. If $t \in T_{\underline{(\mathtt{P}_1 \| \mathtt{P}_2)}} \cup T_{\nsim(\mathtt{P}_1 \| \mathtt{P}_2)}$, then the proof follows by applying inductive hypothesis and rule (<u>COM</u>).

(CASE $\mathtt{P} = \mathtt{S} \setminus L$) It follows by inductive hypothesis and rule (RES). □

To recover the original `CCB` one has simply to restrict the actions/transitions to those with the allowed labels. ~~Being our proposal on `CCB` a conservative one, all the interesting biochemical reactions which are described as `CCB` processes can be described using our set of rules and with our net encoding. Therefore analysis techniques arising from the Petri Nets world can be fruitfully used in this setting. Another byproduct of our approach is that we can use behavioural equivalences originated in the Petri Nets theory.~~

7. Conclusions

We have investigated how a calculus for covalent bonding can be rendered into Petri nets in a compositional and conservative way. ~~In details, we have~~We started from the *Calculus of Covalent Bonding* (CCB) [31, 8]. In this calculus created bonds (interactions between actions) can be broken (aka reversed) concerting them with other actions or bonds. Indeed, the *concerting* capability is the distinguishing feature of the calculus. ~~and~~We have first ~~we have rendered its semantics more compositional.~~faced some critical issues the act of concerting has. ~~The~~One difficulty stems from the fact that a concerted action involves the formation of a weak bond along with the undoing of a strong one. ~~Also~~Another one concerns weak bonds that can be promoted (e.g., passed) to strong ones. ~~Hence~~To overcome these criticalities we have remodelled the CCB semantics to make it ~~more~~ compositional, ~~trying to be~~remaining consistent with the original semantics~~.~~as it is described in [8], and we have therefore proposed rules mimicking, in a compositional fashion, the concert rules and the pre-congruence ones. We stress that, being our proposal on CCB a conservative one, all the interesting biochemical reactions which are described as CCB processes can be described using our set of rules.

We have then introduced a class of Petri nets, called *covalent* bond nets (cBNs), where dependencies among actions are modelled via inhibitor arcs, similarly to what is done in [17]. Following [18] we have then presented a compositional encoding of the (compositional) CCB to bond nets. Needless to say the main difficulties arise when composing in parallel two bond nets derived from two CCB processes. This is due to the fact that one has to account for all the possible synchronisations, concerted actions and promotions of bonds. Concerted actions are the most difficult ones to rend on a net, as they have to mimic several derivations of the calculus in just one firing. One peculiarity of our encoding is that keys are rendered as extra places, marking the fact that a certain action have been done. We stress that in our net encoding all the auxiliary predicates like *fresh* or *std* are hardwired. Keys are represented, in the net encoding, as key places, which implies that it is unnecessary to check whether the key is available or not. Well-stratifiedness engineers the predicate *std*.

Our main results are stated by theorem 1 and theorem 2. Theorem 1 serves as sanity check of our encoding, and tells us that a net derived from the encoding of a CCB terms is indeed a cBN. Theorem 2 states an operational correspondence between CCB term and its net counterpart.

The net encoding of a CCB term have advantages stemming from the Petri Nets world, for instance analysis techniques arising from the Petri Nets world can be fruitfully used, or the usage of behavioural equivalences originated in the Petri Nets theory. It has a drawback that all these encodings have, namely that the need of representing all the possible interactions using transitions makes the net rather big. We however stress that this is also a problem when considering the computations of a CCB term.

Several possible lines of future work can be envisaged. The first one could be to see what whether cBNs can be rendered into reversible Prime Event structures [42] following the construction of [17]. Also we could resort to colours in our bond nets to avoid some extra machinery in the parallel composition. This would allow us to integrate it with the CPN tool [43], and to have an integrated tool for modelling bonds.

References

[1] G. Berry, G. Boudol, The chemical abstract machine, in: F. E. Allen (Ed.), Conference Record of the Seventeenth Annual ACM Symposium on Principles of Programming Languages, San Francisco, California, USA, January 1990, ACM Press, 1990, pp. 81–94. doi:10.1145/96709.96717.
URL https://doi.org/10.1145/96709.96717

[2] V. Danos, C. Laneve, Formal molecular biology, Theoretical Computer Science 325 (1) (2004) 69–110. doi:10.1016/j.tcs.2004.03.065.
URL https://doi.org/10.1016/j.tcs.2004.03.065

[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. doi:10.1016/j.tcs.2009.02.037.
URL https://doi.org/10.1016/j.tcs.2009.02.037

[4] C. Laneve, F. Tarissan, A simple calculus for proteins and cells, Electronic Notes in Theoretical Computer Science 171 (2) (2007) 139–154. doi:10.1016/j.entcs.2007.05.013.
URL https://doi.org/10.1016/j.entcs.2007.05.013

[5] C. Laneve, A. Vitale, Expressivity in the kappa family, in: Proceedings of the 24th Conference on the Mathematical Foundations of Programming Semantics, MFPS 2008, Philadelphia, PA, USA, May 22-25, 2008, Vol. 218 of Electronic Notes in Theoretical Computer Science, Elsevier, 2008, pp. 97–109. doi:10.1016/j.entcs.2008.10.007.
URL https://doi.org/10.1016/j.entcs.2008.10.007

[6] B. Aman, G. Ciobanu, Computational power of protein networks, in: Membrane Computing - 17th International Conference, CMC 2016, Revised Selected Papers, Vol. 10105 of Lecture Notes in Computer Science, Springer, 2017, pp. 103–118. doi:10.1007/978-3-319-54072-6_7.
URL https://doi.org/10.1007/978-3-319-54072-6_7

[7] B. Aman, G. Ciobanu, Bonding calculus, Natural Computing 17 (4) (2018) 823–832. doi:10.1007/s11047-018-9709-7.
URL https://doi.org/10.1007/s11047-018-9709-7

[8] S. Kuhn, I. Ulidowski, Local reversibility in a calculus of covalent bonding, Science of Computer Programming 151 (2018) 18–47. doi:10.1016/j.scico.2017.09.008.
URL https://doi.org/10.1016/j.scico.2017.09.008

[9] T. Wright, I. Stark, Modelling patterns of gene regulation in the bond-calculus, in: Proceedings of SASB 2018, the Ninth International Workshop

on Static Analysis and Systems Biology, Freiburg, Germany - August 28th, 2018, Vol. 350 of Electronic Notes in Theoretical Computer Science, Elsevier, 2020, pp. 117–138. doi:10.1016/j.entcs.2020.06.007.
URL https://doi.org/10.1016/j.entcs.2020.06.007

[10] S. Kuhn, B. Aman, G. Ciobanu, A. Philippou, K. Psara, I. Ulidowski, Reversibility in chemical reactions, in: Reversible Computation: Extending Horizons of Computing - Selected Results of the COST Action IC1405, Vol. 12070 of Lecture Notes in Computer Science, Springer, 2020, pp. 151–176. doi:10.1007/978-3-030-47361-7_7.
URL https://doi.org/10.1007/978-3-030-47361-7_7

[11] I. C. C. Phillips, I. Ulidowski, Reversing algebraic process calculi, Journal of Logic and Algebraic Methods in Programming 73 (1-2) (2007) 70–96. doi:10.1016/j.jlap.2006.11.002.
URL https://doi.org/10.1016/j.jlap.2006.11.002

[12] V. Danos, J. Krivine, Transactions in RCCS, in: CONCUR 2005 - Concurrency Theory, 16th International Conference, Vol. 3653 of Lecture Notes in Computer Science, Springer, 2005, pp. 398–412. doi:10.1007/11539452_31.
URL https://doi.org/10.1007/11539452_31

[13] I. Cristescu, J. Krivine, D. Varacca, A compositional semantics for the reversible p-calculus, in: 28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS, IEEE Computer Society, 2013, pp. 388–397. doi:10.1109/LICS.2013.45.
URL https://doi.org/10.1109/LICS.2013.45

[14] I. Lanese, C. A. Mezzina, J. Stefani, Reversibility in the higher-order π-calculus, Theor. Comput. Sci. 625 (2016) 25–84. doi:10.1016/j.tcs.2016.02.019.
URL https://doi.org/10.1016/j.tcs.2016.02.019

[15] I. Phillips, I. Ulidowski, S. Yuen, A reversible process calculus and the modelling of the ERK signalling pathway, in: Reversible Computation, 4th International Workshop, RC 2012. Revised Papers, Vol. 7581 of Lecture Notes in Computer Science, Springer, 2012, pp. 218–232. doi:10.1007/978-3-642-36315-3_18.
URL https://doi.org/10.1007/978-3-642-36315-3_18

[16] H. C. Melgratti, C. A. Mezzina, I. Ulidowski, Reversing place transition nets, Log. Methods Comput. Sci. 16 (4).
URL https://lmcs.episciences.org/6843

[17] H. C. Melgratti, C. A. Mezzina, G. M. Pinna, A distributed operational view of reversible prime event structures, in: 36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, IEEE, 2021, pp.

1–13. doi:10.1109/LICS52264.2021.9470623.
URL https://doi.org/10.1109/LICS52264.2021.9470623

[18] H. C. Melgratti, C. A. Mezzina, G. M. Pinna, Towards a truly concurrent semantics for reversible CCS, in: Reversible Computation - 13th International Conference, RC 2021, Vol. 12805 of Lecture Notes in Computer Science, Springer, 2021, pp. 109–125. doi:10.1007/978-3-030-79837-6_7.
URL https://doi.org/10.1007/978-3-030-79837-6_7

[19] A. Philippou, K. Psara, Reversible computation in nets with bonds, Journal of Logical and Algebraic Methods in Programming 124 (2022) 100718. doi:https://doi.org/10.1016/j.jlamp.2021.100718.
URL https://www.sciencedirect.com/science/article/pii/S235222082100081X

[20] G. Ciobanu, G. M. Pinna, Memory associated with membranes systems, Journal on Membrane Computing 3 (2) (2021) 116–132. doi:10.1007/s41965-020-00066-8.
URL https://doi.org/10.1007/s41965-020-00066-8

[21] B. Aman, G. Ciobanu, Controlled reversibility in reaction systems, in: Membrane Computing - 18th International Conference, CMC 2017, Bradford, UK, July 25-28, 2017, Revised Selected Papers, Vol. 10725 of Lecture Notes in Computer Science, Springer, 2018, pp. 40–53. doi:10.1007/978-3-319-73359-3_3.
URL https://doi.org/10.1007/978-3-319-73359-3_3

[22] G. Ciobanu, E. N. Todoran, Correct metric semantics for a biologically-inspired formalism, in: 16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2014, IEEE Computer Society, 2014, pp. 317–324. doi:10.1109/SYNASC.2014.50.
URL https://doi.org/10.1109/SYNASC.2014.50

[23] B. Aman, G. Ciobanu, Mutual exclusion and reversibility in reaction systems, Journal on Membrane Computing 2 (3) (2020) 171–178. doi:10.1007/s41965-020-00043-1.
URL https://doi.org/10.1007/s41965-020-00043-1

[24] B. Aman, G. Ciobanu, Reversible computation in nature inspired rule-based systems, Journal on Membrane Computing 2 (4) (2020) 246–254. doi:10.1007/s41965-020-00053-z.
URL https://doi.org/10.1007/s41965-020-00053-z

[25] B. Aman, G. Ciobanu, R. Glück, R. Kaarsgaard, J. Kari, M. Kutrib, I. Lanese, C. A. Mezzina, L. Mikulski, R. Nagarajan, I. C. C. Phillips, G. M. Pinna, L. Prigioniero, I. Ulidowski, G. Vidal, Foundations of reversible computation, in: Reversible Computation: Extending Horizons of Computing - Selected Results of the COST Action IC1405, Vol. 12070 of Lecture Notes in Computer Science, Springer, 2020, pp. 1–40. doi:

10.1007/978-3-030-47361-7_1.
URL https://doi.org/10.1007/978-3-030-47361-7_1

[26] B. Aman, G. Ciobanu, Reversibility in parallel rewriting systems, Journal of Universal Computer Science 23 (7) (2017) 692–703.
URL http://www.jucs.org/jucs_23_7/reversibility_in_parallel_rewriting

[27] B. Aman, G. Ciobanu, Efficiently solving the bin packing problem through bio-inspired mobility, Acta Informatica 54 (4) (2017) 435–445. doi:10.1007/s00236-016-0264-3.
URL https://doi.org/10.1007/s00236-016-0264-3

[28] B. Aman, P. Battyányi, G. Ciobanu, G. Vaszil, Local time membrane systems and time Petri nets, Theoretical Computer Science 805 (2020) 175–192. doi:10.1016/j.tcs.2018.06.013.
URL https://doi.org/10.1016/j.tcs.2018.06.013

[29] B. Aman, G. Ciobanu, Verification of membrane systems with delays via Petri nets with delays, Theoretical Computer Science 598 (2015) 87–101. doi:10.1016/j.tcs.2015.03.051.
URL https://doi.org/10.1016/j.tcs.2015.03.051

[30] G. Ciobanu, G. M. Pinna, Catalytic and communicating Petri nets are turing complete, Information and Computation 239 (2014) 55–70. doi:10.1016/j.ic.2014.08.008.
URL https://doi.org/10.1016/j.ic.2014.08.008

[31] S. Kuhn, I. Ulidowski, A calculus for local reversibility, in: Reversible Computation - 8th International Conference, RC 2016, Vol. 9720 of Lecture Notes in Computer Science, Springer, 2016, pp. 20–35. doi:10.1007/978-3-319-40578-0_2.

[32] J. A. Bergstra, J. W. Klop, Process algebra for synchronous communication, Information and Computation 60 (1-3) (1984) 109–137. doi:10.1016/S0019-9958(84)80025-X.
URL https://doi.org/10.1016/S0019-9958(84)80025-X

[33] W. J. Fokkink, R. J. van Glabbeek, Precongruence formats with lookahead through modal decomposition, in: 26th EACSL Annual Conference on Computer Science Logic, CSL 2017, August 20-24, 2017, Stockholm, Sweden, Vol. 82 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, pp. 25:1–25:20. doi:10.4230/LIPIcs.CSL.2017.25.
URL https://doi.org/10.4230/LIPIcs.CSL.2017.25

[34] I. Lanese, U. Montanari, Synchronization algebras with mobility for graph transformations, Electronic Notes in Theoretical Computer Science 138 (1) (2005) 43–60. doi:10.1016/j.entcs.2005.05.004.
URL https://doi.org/10.1016/j.entcs.2005.05.004

[35] U. Montanari, F. Rossi, Contextual nets, Acta Informatica 32 (6). doi:10.1007/BF01178907.
URL https://doi.org/10.1007/BF01178907

[36] P. Baldan, N. Busi, A. Corradini, G. M. Pinna, Domain and event structure semantics for Petri nets with read and inhibitor arcs, Theoretical Computer Science 323 (1-3) (2004) 129–189.

[37] M. Nielsen, G. D. Plotkin, G. Winskel, Petri nets, event structures and domains, part I, Theoretical Computer Science 13 (1981) 85–108. doi:10.1016/0304-3975(81)90112-2.
URL https://doi.org/10.1016/0304-3975(81)90112-2

[38] G. M. Pinna, How much is worth to remember? A taxonomy based on Petri Nets unfoldings, in: Applications and Theory of Petri Nets - 32nd International Conference, PETRI NETS 2011, Newcastle, UK, June 20-24, 2011. Proceedings, Vol. 6709 of Lecture Notes in Computer Science, Springer, 2011, pp. 109–128. doi:10.1007/978-3-642-21834-7_7.
URL https://doi.org/10.1007/978-3-642-21834-7_7

[39] G. Casu, G. M. Pinna, Flow unfolding of multi-clock nets, in: Application and Theory of Petri Nets and Concurrency - 35th International Conference, PETRI NETS 2014, Tunis, Tunisia, June 23-27, 2014. Proceedings, Vol. 8489 of Lecture Notes in Computer Science, Springer, 2014, pp. 170–189. doi:10.1007/978-3-319-07734-5_10.
URL https://doi.org/10.1007/978-3-319-07734-5_10

[40] G. Casu, G. M. Pinna, Petri nets and dynamic causality for service-oriented computations, in: Proceedings of the Symposium on Applied Computing, SAC 2017, Marrakech, Morocco, April 3-7, 2017, ACM, 2017, pp. 1326–1333. doi:10.1145/3019612.3019806.
URL https://doi.org/10.1145/3019612.3019806

[41] G. Boudol, Flow event structures and flow nets, in: I. Guessarian (Ed.), Semantics of Systems of Concurrent Processes, Vol. 469 of Lecture Notes in Computer Science, Springer, 1990, pp. 62–95. doi:10.1007/3-540-53479-2_4.
URL https://doi.org/10.1007/3-540-53479-2_4

[42] I. Phillips, I. Ulidowski, Reversibility and asymmetric conflict in event structures, Journal of Logic and Algebraic Methods in Programming 84 (6) (2015) 781–805. doi:10.1016/j.jlamp.2015.07.004.
URL https://doi.org/10.1016/j.jlamp.2015.07.004

[43] K. Jensen, L. M. Kristensen, L. Wells, Coloured Petri nets and CPN tools for modelling and validation of concurrent systems, International Journal on Software Tools for Technology Transfer 9 (3-4) (2007) 213–254. doi:10.1007/s10009-007-0038-x.
URL https://doi.org/10.1007/s10009-007-0038-x