Models of Circular Causality*

Massimo Bartoletti[1], Tiziana Cimoli[1], G. Michele Pinna[1], and Roberto Zunino[2]

[1] Dipartimento di Matematica e Informatica, Università di Cagliari, Cagliari, Italy
[2] Dipartimento di Matematica, Università degli Studi di Trento, Italy

Abstract. Causality is often interpreted as establishing dependencies between events. The standard view is that an event b causally depends on an event a if, whenever b occurs, then a has already occurred. If the occurrences of a and b mutually depend on each other, i.e. a depends on b and vice versa, then (under the standard notion of causality) neither of them can ever occur. This does not faithfully capture systems where, for instance, an agent promises to do event a provided that b will be *eventually* done, and vice versa. In this case, the circularity between the causal dependencies should allow both a and b to occur, in any order. In this paper we review three models for circular causality, one based on logic (declarative), one based on event structures (semantical), and one based on Petri nets (operational). We will cast them in a coherent picture pointing out their relationships.

1 Motivations

Circular dependencies are a natural aspect of many kinds of interactions. For instance, consider two mutually distrusting participants, Alice and Bob, who want to exchange their resources. Alice wants Bob's resource, and *vice versa*, but only one resource at a time can be transferred. A possible interaction is that where Alice makes the first move, by giving her resource to Bob. At this point, Bob can either give his resource to Alice, or he may even choose not to. Since Alice and Bob do not trust each other, each one is expecting that the other gives their resource first. The two participants are stuck in a situation where no one can move. This is a classical issue, discussed by philosophers at least since Hobbes' Leviathan [14].

The above scenario expresses the basic idea behind *circular causality*: Alice wants to do her action after Bob's action has happened, and *vice versa*. This situation can be represented in many ways in many models.

From a logical point of view, we may say that "*Alice gives her resource to Bob*" and "*Bob gives his resource to Alice*" are two atomic propositions a and b, and we can model the behaviour of Alice and Bob with the formulae:

$$b \to a \qquad\qquad a \to b \tag{1}$$

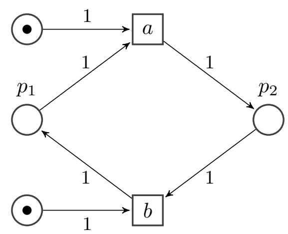
where $\to$ denotes e.g. intuitionistic implication [17].

* Work partially supported by Aut. Reg. Sardinia LR 7/07 CRP-17285 (TRICS), PIA 2010 "Social Glue", by MIUR PRIN 2010-11 "Security Horizons", and by EU COST Action IC1201 (BETTY).

From a semantical point of view, a and b can be considered *events* of an *event structure* [22], which is one of the classical models of concurrent systems. In event structures, causality among events is represented by *enablings* of the form $X \vdash e$, meaning that the event e can only occur after all the events in the set X have already occurred. Then, our Alice-Bob scenario can be modelled by an event structure with the following enablings:

$$\{b\} \vdash a \qquad\qquad \{a\} \vdash b \tag{2}$$

Finally, in a more operational perspective, a and b can be seen as transitions of the following Petri net N [19]:



We can now notice that all the above formalisations of the Alice-Bob scenario share some kind of misfeature: neither the atom a nor the atom b can be deduced in the logical theory in (1); neither the events a nor b can ever happen (i.e. they are not *reachable*) in the event structure in (2); and neither transitions a or b can be fired in the net N.

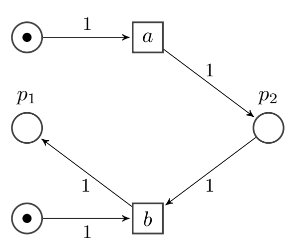
To solve the impasse, one of the participants must do the first move. Alice may decide to give her resource to Bob, without conditions, hoping for his in exchange. This adjustment clearly resolves the circularity issue (since one of the dependencies is removed). From the logical point of view, the modified scenario can be described by the theory:

$$a \qquad\qquad a \to b \tag{3}$$

where both a and b are deducible. From the semantic point of view, we obtain an event structure with the following enablings:

$$\emptyset \vdash a \qquad\qquad \{a\} \vdash b \tag{4}$$

where both a and b are reachable. Finally, the Petri net is adjusted as follows (we name the resulting net N'):



where both transitions a and b can be performed.

However, this new scenario has a flaw: we have lost the information that Alice is willing to trade her resource *only* if she receives what she wants in exchange. Consider for instance that Bob wants to give his resource to Carl, first, and that Carl needs that resource forever. Assume also that this information is not shared with Alice. Since Alice is saying that she will give her resource without conditions, she gives away her resource, but she will receive nothing. If we model this situation in our three settings, in the logic we have a and $a \to c$ (Carl's request), from which we deduce a but not b. In the event structure we have enablings $\emptyset \vdash a$ and $a \vdash c$, and so a is reachable while b is not. In the Petri net, we just replace transition b in N' with a new transition c, and we have that the transition a is fired while b is not. Never has Alice got what she wants, but she always has to do something. We would like to express Alice's constraints in such a way that, if Bob is not promising to give her what she wants, then she is not obliged to do anything.

Our answer to this problem is the introduction of a novel kind of circular dependency of the form "Alice gives her resource to Bob" on the *promise* that "Bob will eventually gives his to Alice". Alice may give away her resource *now*, but only if Bob promises to give her his one. In the logical approach, this is done in [11] by extending Intuitionistic Propositional Logic with a new kind of implication ($\twoheadrightarrow$); event structures are extended in [7] with a new kind of enabling ($\Vdash$); Petri nets are extended with the possibility of lending tokens [4]. Thus, let us consider again our Alice-Bob example. In the logic, we have:

$$b \twoheadrightarrow a \qquad\qquad a \to b \tag{5}$$

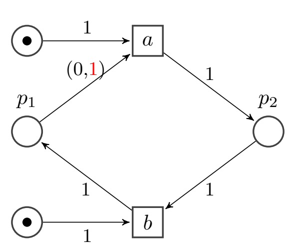
and due to the reduction rules for $\twoheadrightarrow$, now both a and b are deducible.

In event structures, we have:

$$b \Vdash a \qquad\qquad a \vdash b \tag{6}$$

and since $\Vdash$ decouples causality from the order in which events happens, both a and b are reachable.

In Petri nets, we add an arc from p_1 to a labelled in such a way that it means that a token may be lent from p_1:



Since transition a can lend a token, the firing sequence $p_1 \xrightarrow{a} p_2 \xrightarrow{b} p_1$ is now possible.

Consider again the situation where Bob and Carl are trying to deceive Alice. Were we to match Alice's condition with Bob's c, we would end with a satisfactory situation: in the logic, $b \twoheadrightarrow a$ and c do *not* imply a; in the event structure with enablings $b \Vdash a$ and c, event a is not reachable; in Petri nets, an "honored" marking is not reachable.

This paper is organized as follows: Sections 2 to 4 gently introduce the three models considered in this paper: the logic PCL, *lending* Petri nets, and event structures with circular causality. For each of these models, we will give some examples to stress the flavor of these approaches. Section 5 contains some of the results connecting these three models. Finally, in Section 6 we draw some conclusions.

2 A logical approach to circular causality

Propositional Contract Logic (PCL [11]) extends intuitionistic propositional logic (IPC) with the connective $\twoheadrightarrow$, called *contractual implication.*

Definition 1 (PCL syntax). *The formulae $A, B, \ldots$ of PCL are defined as follows, where we assume that $a, b, \ldots$ range over a given set of atoms.*

$$A, B \ ::= \ \bot \mid \top \mid a \mid \neg A \mid A \vee B \mid A \wedge B \mid A \rightarrow B \mid A \twoheadrightarrow B$$

The natural deduction system for PCL [3] extends that for IPC with the rules $(\twoheadrightarrow\text{I1})$, $(\twoheadrightarrow\text{I2})$, and $(\twoheadrightarrow\text{E})$ in Figure 1 (wherein, in all the rules, Δ is a set of PCL formulae). Provable formulae are contractually implied, according to rule $(\twoheadrightarrow\text{I1})$. Rule $(\twoheadrightarrow\text{I2})$ provides $\twoheadrightarrow$ with the same weakening properties of intuitionistic implication $\rightarrow$. The paradigmatic rule is $(\twoheadrightarrow\text{E})$, which allows for the elimination of contractual implication $\twoheadrightarrow$. Compared to the rule $(\rightarrow\text{E})$ for elimination of $\rightarrow$ in IPC, the only difference is that in the context used to deduce the antecedent A, rule $(\twoheadrightarrow\text{E})$ also allows for using as hypothesis the consequence B.

$$\frac{}{\Delta, A \vdash A}\ (\text{Id}) \qquad \frac{\Delta \vdash A \qquad \Delta \vdash B}{\Delta \vdash A \wedge B}\ (\wedge\text{I}) \qquad \frac{\Delta \vdash A \wedge B}{\Delta \vdash A}\ (\wedge\text{E1}) \qquad \frac{\Delta \vdash A \wedge B}{\Delta \vdash B}\ (\wedge\text{E2})$$

$$\frac{\Delta \vdash A}{\Delta \vdash A \vee B}\ (\vee\text{I1}) \qquad \frac{\Delta \vdash B}{\Delta \vdash A \vee B}\ (\vee\text{I2}) \qquad \frac{\Delta \vdash A \vee B \qquad \Delta, A \vdash r \qquad \Delta, B \vdash r}{\Delta \vdash r}\ (\vee\text{E})$$

$$\frac{\Delta, A \vdash B}{\Delta \vdash A \rightarrow B}\ (\rightarrow\text{I}) \qquad \frac{\Delta \vdash A \rightarrow B \qquad \Delta \vdash A}{\Delta \vdash B}\ (\rightarrow\text{E})$$

$$\frac{\Delta \vdash B}{\Delta \vdash A \twoheadrightarrow B}\ (\twoheadrightarrow\text{I1}) \qquad \frac{\Delta \vdash A \twoheadrightarrow B \qquad \Delta, B \vdash A}{\Delta \vdash B}\ (\twoheadrightarrow\text{E})$$

$$\frac{\Delta \vdash A \twoheadrightarrow B \qquad \begin{array}{l}\Delta, A' \vdash A \\ \Delta, B \vdash A' \twoheadrightarrow B'\end{array}}{\Delta \vdash A' \twoheadrightarrow B'}\ (\twoheadrightarrow\text{I2})$$

Fig. 1. Natural deduction system for PCL (rules for $\neg$ and $\bot$ omitted).

Example 1. Let $\Delta = A \to B, B \twoheadrightarrow A$. A proof of $\Delta \vdash A$ in natural deduction is:

$$\dfrac{\Delta \vdash B \twoheadrightarrow A \qquad \dfrac{\Delta, A \vdash A \to B \qquad \overline{\Delta, A \vdash A}^{\,(\text{ID})}}{\Delta, A \vdash B}\,(\to\text{E})}{\Delta \vdash A}\,(\twoheadrightarrow\text{E})$$

As in the previous example, we can show that the following is a theorem of PCL:

$$(A \twoheadrightarrow B) \ \wedge\ (B \twoheadrightarrow A) \ \to\ A \ \wedge\ B \qquad \text{(THEOREM)}$$

whereas the following is *not* a theorem (neither of PCL nor of IPC):

$$(A \to B) \ \wedge\ (B \to A) \ \to\ A \ \wedge\ B \qquad \text{(NOT A THEOREM)}$$

The above theorem highlights the different nature of contractual and intuitionistic implication: the former allows for a form of *circular* reasoning, while the latter does not. Some other characterizing theorems of PCL are outlined below:

$$\vdash (A \twoheadrightarrow B) \wedge (B \twoheadrightarrow A) \ \to\ A \wedge B \tag{7}$$

$$\vdash (A_1 \twoheadrightarrow A_2) \wedge \cdots \wedge (A_{n-1} \twoheadrightarrow A_n) \wedge (A_n \twoheadrightarrow A_1) \ \to\ A_1 \wedge \cdots \wedge A_n \tag{8}$$

$$\vdash \textstyle\bigwedge_{i \in 1..n} \big((A_1 \wedge \cdots \wedge A_{i-1} \wedge A_{i+1} \wedge \cdots \wedge A_n) \twoheadrightarrow A_i \big) \ \to\ A_1 \wedge \cdots \wedge A_n \tag{9}$$

$$\vdash (A \twoheadrightarrow B) \to (A \to B) \tag{10}$$

$$\nvdash (A \to B) \to (A \twoheadrightarrow B) \tag{11}$$

$$\vdash (A' \to A) \ \wedge\ (A \twoheadrightarrow B) \ \wedge\ (B \to B') \ \to\ (A' \twoheadrightarrow B') \tag{12}$$

Theorem (7) models a binary *handshaking*; (8) is a generalization to the multi-party case, where the $(i+1)$-th party, in order to do A_{i+1}, relies on an action A_i made by the i-th party; (9) is a sort of "greedy" handshaking, because now a party does A_i only provided that *all* the other parties do their actions, i.e. $A_1, \ldots, A_{i-1}, A_{i+1}, \ldots, A_n$. Theorem (10) states that contractual implication is *stronger* than intuitionistic implication, while (11) says that the converse does not hold. The consequence in a contractual implication can be arbitrarily weakened, while the precondition can be arbitrarily strengthened (12).

The main results about PCL, among which consistency (under the full set of rules, which also deal with negation and $\bot$) and decidability, are established in [11]. The proof of decidability follows the lines of the one for IPC given by Kleene in [15]. The result relies on a formulation of the PCL sequent calculus with implicit structural rules (to limit the proof search space of a given sequent, as in Kleene's G3 calculus) and the subformula property, obtained as consequence of the cut-elimination theorem.

While PCL is clearly a conservative extension of IPC, there cannot be sound and complete homomorphic encodings of PCL into IPC: that is, $\twoheadrightarrow$ cannot be regarded as syntactic sugar for some IPC context.

Definition 2 (Homomorphic encoding). *A homomorphic encoding m is a function from PCL formulae to IPC formulae such that: m is the identity on* *prime formulas,* $\top$*, and* $\bot$*; it acts homomorphically on* $\wedge, \vee, \rightarrow, \neg$*; it satisfies* $m(A \twoheadrightarrow B) = \mathcal{C}[m(A), m(B)]$ *for some fixed IPC context* $\mathcal{C}(\bullet, \bullet)$*.*

Of course, each homomorphic encoding is uniquely determined by the context $\mathcal{C}$. Several *complete* encodings, (i.e. satisfying $\vdash p \implies \vdash_{IPC} m_i(p)$) exist: for instance, $m_0(A \twoheadrightarrow B) = m_0(B)$ and $m_1(A \twoheadrightarrow B) = (m_1(B) \rightarrow m_1(A)) \rightarrow m_1(B)$ are both complete encodings. However, there can be no *sound* encodings. Indeed, a sound encoding would allow us to derive Peirce's axiom in PCL, violating the fact that PCL conservatively extends IPC [10].

Theorem 1. *If* m *is a homomorphic encoding of* PCL *into IPC, then* m *is not sound, i.e. there exists a* PCL *formula* A *such that* $\vdash_{IPC} m(A)$ *and* $\nvdash A$*.*

3 An operational approach to circular causality

In Petri nets [19] dependencies among transitions are encoded by stipulating that tokens produced by a transition are consumed by others, and it may well be that two or more transitions may share places in such a way that tokens produced by one are used by the others and vice versa. Thus, on an abstract level, the issue of circularity is already present in the general Petri nets setting, and circularity is not considered a relevant issue. This is not longer true when considering nets where transitions represent events and the requirement that a transition is executed just once is enforced, *e.g.* occurrence nets (adopting the terminology of van Glabbeek and Plotkin in [21]). In this case to establish a circular dependency a transition should use a token produced by another transition which in turn expects (one of) the token produced by the former. According to the ordinary interpretation of firing in Petri nets these transitions cannot fire.

To overcome this problem *debit* arcs have been introduced in [20]. In a net with debit arcs transitions may be executed even if some tokens are not available. The motivations behind this approach rely on language theoretic considerations (Petri nets are indeed a kind of automata able to recognise a class of languages [23]): nets with debit arcs are inspired by the so called *blind-one way multicounter machines* described by Greibach in [13].

We are more interested in characterising in an operational way the capability of a place to *lend* tokens allowing in this way the execution of a transition otherwise *blocked*. To this aim we present *Lending Petri nets*, defined in [4] and further studied in [5], which are basically debit nets with some additional constraints. A Petri net is a tuple $\langle S, T, F, m_0 \rangle$, where S is a set of *places*, T is a set of *transitions* (such that that $S \cap T = \emptyset$), $F \colon (S \times T) \cup (T \times S) \rightarrow \mathbb{N}$ is a *weight function*, and $m_0 : S \rightarrow \mathbb{N}$ is a function from places to natural numbers, called *marking*, which models the initial state of the net. $F(s, t) = n$ means that the transition t can be fired whenever n tokens are available at place s, while $F(t, s') = m$ means that firing the transition t will result in m tokens added to place s'. Lending Petri nets extend standard nets by allowing transitions to fire even in the absence of the required number of tokens. However, this is done in a controlled manner: only a fixed number of tokens can be obtained "on credit", and credits must be eventually honored. We introduce a *lending function* $L : S \times T \to \mathbb{N}$, which specifies how many tokens a transition may borrow from a place. Thus if $F(s,t) = n$ and $L(s,t) = l$, then firing the transition t costs $n+l$ tokens, of which only l can be taken on credit. We equip Lending Petri nets with a labelling ℓ of places and transitions, where labels are drawn from a set $\mathcal{L}$. These labels have a role similar to the one played by input/output interfaces in the open nets as defined in [1], and play a major role when defining operations on these nets.

Definition 3. *A* lending *Petri net (LPN) is a tuple* $N = \langle S, T, F, L, \ell, m_0 \rangle$ *where: (a)* $\langle S, T, F, m_0 \rangle$ *is a Petri net, (b)* $L\colon S \times T \to \mathbb{N}$ *is the* lending function*, and (c)* $\ell : S \cup T \rightharpoonup \mathcal{L}$ *is a partial* labeling *of places and transitions, Further, we require that for each* $t \in T$*, there exists some* $s \in S$ *such that* $F(s,t) + L(s,t) > 0$*.*

The last requirement says simply that no transition can happen *spontaneously* but must consume or *lend* some tokens.

This model is clearly a conservative extension of the classical one: indeed, an ordinary Petri net is an LPN where the lending function is constant and equal to 0, which means that no token can be borrowed from any place. The drawing conventions we adopt are mostly standard, the unique difference is for arcs connecting places to transitions we have a pair of natural numbers, the first representing the weight of the *standard* arcs (possibly 0) and the second the weight of the lending ones (in red, only written when nonzero). We omit the arc between a place and a transition if standard and lending arcs have null weights.

We define the *pre-set* and the *post-set* of a transition/place as usual: ${}^\bullet x = \{y \in T \cup S \mid F(y,x) > 0\}$ and $x^\bullet = \{y \in T \cup S \mid F(x,y) > 0\}$, respectively. These are lifted to sets of transitions/places in the obvious way. The current state of a net is described by a *marking*, which in the case of LPNs is no longer constrained to be a function from places to natural numbers, but it is a function $m : S \to \mathbb{Z}$ from places to *integers* (with the exception of the initial marking m_0 that must be non-negative). We shall adopt the following drawing convention for markings. First, we associate each place p with a *co-place* $\overline{p}$, which represents a negative token in p. Then, a marking m is represented as a multiset of places and co-places, contanining $m(p)$ occurrences of p if $m(p)$ is positive, and $-m(p)$ occurrences of $\overline{p}$ is $m(p)$ is negative. For instance, we represent the marking $m = \{p_1 \mapsto 2, p_2 \mapsto -1\}$ as $p_1, p_1, \overline{p_2}$. We denote with $\emptyset$ the empty multiset.

The behavior of a net is described by a labeled relation between markings, where labels are transitions in T. Intuitively, a transition t can be fired at a certain marking whenever each place in the pre-set of t contains enough tokens: more precisely, each place $s \in {}^\bullet t$ must contain at least $F(s,t)$ tokens. If a transition t is enabled at a marking m then it can be *fired*, leading to a new marking where the number of tokens in the places of the net is accordingly updated. To do that, each place s in the pre-set of t gives away $F(s,t) + L(s,t)$ tokens (of which, only $F(s,t)$ need to be already available at s, while the others can be taken on credit), and it receives $F(t,s)$ tokens.

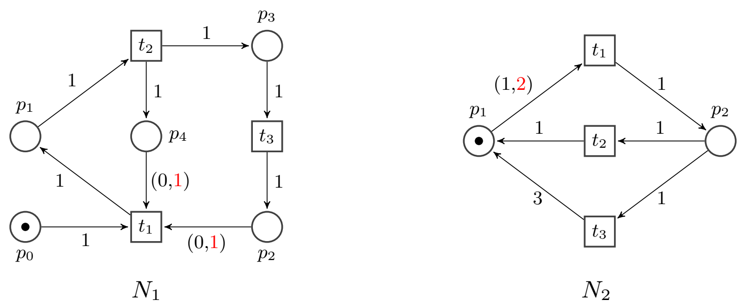


Fig. 2. Two Lending Petri nets.

Definition 4. *Let* $N = \langle S, T, F, L, \ell, m_0 \rangle$ *be an LPN. We say that* $t \in T$ *is* enabled at m *iff* $m(s) \geq F(s,t)$ *for all* $s \in {}^\bullet t$. *We have a* step t *from* m *to* m' *(in symbols,* $m \xrightarrow{t} m'$*) whenever* t *is enabled at* m*, and, for all* $s \in S$*:* $m'(s) = m(s) - \big(F(s,t) + L(s,t)\big) + F(t,s)$

A consequence of this notion is that the number of tokens in a place can become negative, if the weight of the lending arc is not zero.

A *firing sequence* is a finite sequence of steps. The *trace* of a firing sequence is the string of labels associated to its transitions, *i.e.* the trace of $m_0 \xrightarrow{t_1} m_1 \cdots m_{n-1} \xrightarrow{t_n} m_n$ is the string $\ell(t_1) \cdots \ell(t_n)$, which is the empty string ε when $n = 0$, and it is undefined when $\ell(t_i)$ is undefined for some i. The set of all traces of a net N is denoted with $Tr(N)$. As usual, we denote with $\rightarrow^*$ the reflexive and transitive closure of $\rightarrow$. Hereafter, we denote with $Mk(N)$ the set of *reachable markings* of a net N, *i.e.* those markings m for which there exists a firing sequence starting at m_0 and leading to m.

Not all reachable markings represent good states of a system: a marking where some places have a negative number of tokens models a state where some resources have been taken on credit, but the credit has not been honored yet. *Honored markings* are those markings which model states where all credits have been honored. Thus in honoured markings the possible circular dependencies among transitions have been *solved*.

Definition 5. *A marking* m *of* N *is* honored *iff* $m(s) \geq 0$ *for all places* s *of* N*.*

If the net has no lending arcs, all the reachable markings are honored. An honored firing sequence is a firing sequence where the final marking is honored.

Example 2. Consider the LPN N_1 in Figure 2. The initial marking is the multiset p_0. The transition t_1 is enabled at p_0 as it may borrow tokens from places p_2 and p_4. The other two transitions (t_2 and t_3) are not enabled. We have exactly one maximal firing sequence: $p_0 \xrightarrow{t_1} p_1, \overline{p_2}, \overline{p_4} \xrightarrow{t_2} \overline{p_2}, p_3 \xrightarrow{t_3} \emptyset$. Note that the marking reached after firing all the three transitions is a non-negative one, hence it is honored.

Consider now the LPN N_2 in Figure 2. The transition t_1 is enabled, as it may borrow two tokens from place p_1. Firing t_1 leads to the marking $\overline{p_1}, \overline{p_1}, p_2$. Then, if the transition t_2 is fired, one token is given back to place p_1, and we reach a deadlock, *i.e.* a not honored state where no transitions are enabled. Instead, if the transition t_3 is fired then we return to the initial state, with one token at place p_1.

LPNs are intended to represent systems, hence a notion of composition should be introduced. The idea is that labelled places are the *interface* of the LPN. Those without outgoing transitions play the role of *outputs*, whereas those incoming transitions play the role of *inputs*. If a net N has an input place, and N' has an output place with the same label, then in their composition $N \oplus N'$ these places will be plugged together. This models an asynchronous communication channel between nets, which does not preserve the order of messages (as usual in nets, see e.g. [1]). A transition with a certain label of a component is supposed to produce tokens in all the interface places of the other component, that have the same label.

The composition of LPNs we introduce is subject to some conditions, which altogether take the name of *correct labeling*, and are collected in Definition 6. The transitions of each components are labeled with actions, and the tokens produced by these transitions may carry this information. When these tokens are produced in labeled places, we require that this information is preserved (this is the requirement (a) of Definition 6). Accordingly to the same intuition, all the labeled places in the post-set of a transition should carry the same label (requirement (b) of Definition 6). Finally, input/output places are not initially marked (requirement (c)). This is because we want have input/output places as the *communication* medium among the components.

Definition 6. *A LPN* $\langle S, T, F, L, \ell, m_0 \rangle$ *is* correctly labeled *iff for all* $s \in S$ *such that* $\ell(s) \neq \bot$*: (a)* $\forall t, t' \in {}^\bullet s.\ \ell(t) = \ell(s) = \ell(t')$ *(b)* $\forall t \in {}^\bullet s.\ |\{\ell(s') \mid s' \in t^\bullet \ \wedge\ \ell(s') \neq \bot\}| = 1$*, and (c)* $m_0(s) = 0$*.*

The underlying idea of LPN composition is rather simple: input and output places with the same label are merged together and the flow relation is defined accordingly. Formally, the output places s in N with a label occurring in N' are removed, and the ingoing transitions of s are connected to the input places in N' with label $\ell(s)$. Furthermore, if a component has a transition t with the same label of a place s of the other component, then a flow arc is created from the transition to the place. We require that arcs connecting a labeled transition to a labeled place have always weight 1. All the other ingredients of the composed net are inherited from the components.

Definition 7. *Let* $N = \langle S, T, F, L, \ell, m_0 \rangle$ *and* $N' = \langle S', T', F', L', \ell', m'_0 \rangle$ *be two correctly labeled LPNs. We say that* N, N' *are* composable *whenever (a)* $S \cap S' = \emptyset = T \cap T'$*, and (b)* $\forall t \in T,\ \forall s \in S.\ \ell(s) \neq \bot \implies F(t, s) \leq 1$ *and* $\forall t \in T',\ \forall s \in S'.\ \ell'(s) \neq \bot \implies F'(t, s) \leq 1$ *and in such case their composition* $N \oplus N'$ *is the LPN* $\langle \hat{S}, T \cup T', \hat{F}, \hat{L}, \hat{\ell}, \hat{m}_0 \rangle$ *in Figure 3.*

$$\hat{S} = (S \setminus \mathbb{S}) \cup (S' \setminus \mathbb{S}')$$
$$\textit{where } \mathbb{S} = \{s \in S \mid \ell(s) \in \ell'(S') \text{ and } s^\bullet = \emptyset\}$$
$$\textit{and } \mathbb{S}' = \{s' \in S' \mid \ell'(s') \in \ell(S) \text{ and } s'^\bullet = \emptyset\}$$

$$\hat{F}(s,t) = \begin{cases} F(s,t) & \text{if } s \in S \text{ and } t \in T \\ F'(s,t) & \text{if } s \in S' \text{ and } t \in T' \end{cases}$$

$$\hat{F}(t,s) = \begin{cases} F(t,s) & \text{if } s \in S \text{ and } t \in T \\ F'(t,s) & \text{if } s \in S' \text{ and } t \in T' \\ F(t,s') & \text{if } s \in S' \text{ and } s' \in \mathbb{S} \text{ and } t \in T \text{ and } \ell(t) = \ell'(s) \\ F'(t,s') & \text{if } s \in S' \text{ and } s' \in \mathbb{S}' \text{ and } t \in T' \text{ and } \ell'(t) = \ell(s) \\ 1 & \text{if } t \in T \text{ and } s \in S' \text{ and } \ell(t) = \ell'(s) \\ 1 & \text{if } t \in T' \text{ and } s \in S \text{ and } \ell'(t) = \ell(s) \end{cases}$$

$$\hat{L}(s,t) = \begin{cases} L(s,t) & \text{if } s \in S \text{ and } t \in T \\ L'(s,t) & \text{if } s \in S' \text{ and } t \in T' \end{cases}$$

$$\hat{\ell}(x) = \begin{cases} \ell(x) & \text{if } x \in S \cup T \\ \ell'(x) & \text{otherwise} \end{cases}$$

$$\hat{m}_0(\hat{s}) = \begin{cases} 1 & \text{if } s \in S \text{ and } m_0(s) = 1, \text{ or } s \in S' \text{ and } m'_0(s) = 1 \\ 0 & \text{otherwise} \end{cases}$$

Fig. 3. Composition of two LPNs.

Observe that composing two nets N and N' such that $\ell(S) \cap \ell'(S') = \emptyset$ results in the disjoint union of the two nets. Further, if the common label $a \in \ell(S) \cap \ell'(S')$ is associated in N to a place s with empty post-set and in N' to a place s' with empty pre-set (or *vice versa*) and the labelings are injective, we obtain precisely the composition between open nets defined in [1].

Example 3. Consider the nets in Figure 4. In the LPN N the transition t_a can be executed only if a token is present in the interface place p_1 labeled a, which has no ingoing to any transition. In the LPN N' the transition t_b is enabled as it may lend a token from the interface place p'_1 labeled b. The result of composition of these two nets is the LPN $N \oplus N'$, where now the execution of the transition t_a puts a token in the interface place p_1 (the resulting marking is $p_1, p^*_b, \overline{p'_1}$) and at this marking firing t_b leads to the empty marking.

Proposition 1. *Let N_i, $i \in \{1, 2, 3\}$ be pairwise composable LPNs. Then $N_1 \oplus N_2 = N_2 \oplus N_1$, and $N_1 \oplus (N_2 \oplus N_3) = (N_1 \oplus N_2) \oplus N_3$.*

The composition $\oplus$ does not have the property that, in general, restricting to the transitions of one of the components, we obtain the LPN we started with.

A *subnet* is a net obtained by restricting places and transitions of a net, and correspondingly the flow function, the lending function and the initial marking.

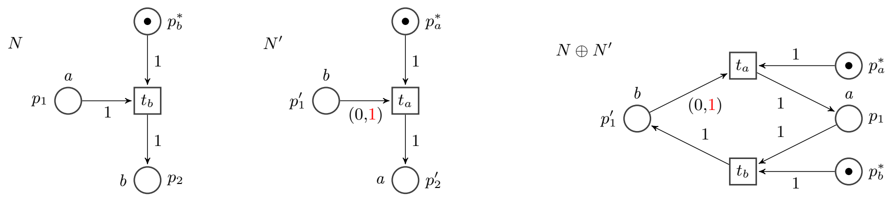


Fig. 4. Two LPNs and their pairwise composition.

Definition 8. *Let $N = \langle S, T, F, L, \ell, m_0 \rangle$ be an LPN, and let $T' \subseteq T$. We define the* subnet $N|_{T'} = \langle S', T', F', L', \ell, m_0' \rangle$, *where: (a) $S' = \{s \in S \mid F(t,s) > 0$ or $F(s,t) > 0$ for $t \in T'\} \cup \{s \in S \mid m_0(s) > 0\}$, (b) $F' = F\,|_{(S' \times T') \cup (T' \times S')}$, (c) $L' = L\,|_{S' \times T'}$, (d) $\ell' = \ell\,|_{S' \cup T'}$, and (e) $m_0' = m_0\,|_{S'}$.*

Definition 9. *Let N and N' be two LPNs. We say that N is* trace equivalent *to N' (in symbols, $N \sim N'$) whenever $Tr(N) = Tr(N')$.*

Proposition 2. *For two composable LPNs N_1, N_2, we have that $N_i \sim (N_1 \oplus N_2)|_{T_i}$, for $i = 1, 2$.*

4 A semantic approach to circular causality

We review now a semantic approach to circularity based on the notion of *event structure with circular causality*, which have been introduced in [6] and further studied in [12] and [7]. Since [18,22], event structures (ES) are one of the classical model for concurrency, and they are at least equipped with a relation (written $\vdash$ in [22]) modelling *causality*, and another one modeling non-determinism (usually rendered in terms of conflicts or consistency). Extensions to ES often use other relations to model other kind of dependencies, *e.g.* or-causality [2]. ES can provide a basic semantic model for concurrent systems, by interpreting the enabling $\{a\} \vdash b$ as: "event b can be done *after* a has been done". We use a relation to model circular causality. Given a set of events E and an irreflexive and symmetric relation representing conflicts (denoted with #), we say that a set $X \subseteq E$ is *conflict-free* ($CF(X)$ in symbols) whenever $\forall e, e' \in X. \neg(e \# e')$. We denote with Con the set $\{X \subseteq_{fin} E \mid CF(X)\}$.

Definition 10. *An event structure with circular causality (* CES *) is a quadruple $\mathcal{E} = (E, \#, \vdash, \Vdash)$ where: (a) E is a set of events, (b) $\# \subseteq E \times E$ is an irreflexive and symmetric relation, called* conflict *relation, (c) $\vdash \subseteq Con \times E$ is the* enabling *relation, and (d) $\Vdash \subseteq Con \times E$ is the* circular enabling *relation, The relations $\vdash$ and $\Vdash$ are* saturated, *i.e. for all $X, Y \in Con$ and for $\circ \in \{\vdash, \Vdash\}$: $X \circ e \;\wedge\; X \subseteq Y \implies Y \circ e$. We say that $\mathcal{E}$ is* finite *when E is finite; we say that $\mathcal{E}$ is* conflict-free *when the conflict relation is empty.*

For a sequence $\sigma = \langle e_0\, e_1 \ldots \rangle$ (possibly infinite), we write $\overline{\sigma}$ for the set of events in σ. We write σ_i for the subsequence $\langle e_0 \ldots e_{i-1} \rangle$. If $\sigma = \langle e_0 \ldots e_n \rangle$ is finite, we write $\sigma\, e$ for the sequence $\langle e_0 \dots e_n\, e\rangle$. The empty sequence is denoted by ε. We adopt the following conventions: $\vdash e$ stands for $\emptyset \vdash e$ and we write $a \vdash b$ for $\{a\} \vdash b$. For a finite, conflict-free set X, we write $X \vdash Y$ for $\forall e \in Y.\ X \vdash e$. For an infinite, conflict-free X, we write $X \vdash Y$ as a shorthand for $\exists X_0 \subseteq_{\mathit{fin}} X.\ X_0 \vdash Y$. All the abbreviations above also apply to $\Vdash$.

A configuration C is a "snapshot" of the behaviour of the system. In [22], a set of events C is a configuration if and only if for each event $e \in C$ it is possible to find a *trace for e in C*, *i.e.* a finite sequence of events containing e, which is closed under the enabling relation:

$$\forall e \in C.\ \exists \sigma = \langle e_0 \dots e_n\rangle.\ e \in \overline{\sigma} \subseteq C\ \wedge\ \forall i \leq n.\ \{e_0, \dots, e_{i-1}\} \vdash e_i$$

We refine the notion in [22] to deal with circular causality. Intuitively, for all events e_i in the sequence $\langle e_0 \dots e_n\rangle$, e_i can either be $\vdash$-enabled by its predecessors, or $\Vdash$-enabled by the *whole* sequence, i.e.:

$$\forall e \in C.\ \exists \sigma = \langle e_0 \dots e_n\rangle.\ e \in \overline{\sigma} \subseteq C\ \wedge\ \forall i \leq n.\ \big(\{e_0, \dots, e_{i-1}\} \vdash e_i\ \vee\ \overline{\sigma} \Vdash e_i\big)$$

Clearly, the configurations of a CES without $\Vdash$-enablings are also configurations in the sense of [22], hence CES are a conservative extension of Winskel's general ES. Differently from ES, if C is a finite configuration of a CES , and σe is a trace for all the events in C, not necessarily σ is a trace for $C \setminus \{e\}$ (see *e.g.*, $\mathcal{E}_2$ in Figure 5).

To allow for reasoning about sets of events which are not configurations, we introduce the auxiliary notion of X*-configuration* in Definition 11 below. In an X*-configuration* C, the set C can contain an event e even in the absence of a justification through a standard/circular enabling — provided that e belongs to the set X. This allows, given an X-configuration, to add/remove any event and obtain a Y-configuration, possibly with $Y \neq X$. We shall say that the events in X have been taken "on credit", to remark the fact that they may have been performed in the absence of a causal justification. With this new concept in mind, we can say that standards configurations are just $\emptyset$-configurations: they represent sets of events where all the credits have been "honoured".

Definition 11. *Let $\mathcal{E} = (E, \#, \vdash, \Vdash)$ be a* CES *, and let $X \subseteq E$. A conflict-free sequence $\sigma = \langle e_0 \dots e_n\rangle \in E^*$ without repetitions is an X*-trace *of $\mathcal{E}$ iff:*

$$\forall i \leq n.\ (e_i \in X\ \ \vee\ \ \overline{\sigma_i} \vdash e_i\ \ \vee\ \ \overline{\sigma} \Vdash e_i) \tag{13}$$

For all $C, X \subseteq E$ we say that C is an X-configuration *of $\mathcal{E}$ iff $CF(C)$ and:*

$$\forall e \in C.\ \ \exists \sigma\ X\textit{-trace}.\ \ e \in \overline{\sigma} \subseteq C \tag{14}$$

The set of all X-traces of $\mathcal{E}$ is denoted by $\mathcal{T}_{\mathcal{E}}(X)$, abbreviated as $\mathcal{T}_{\mathcal{E}}$ when $X = \emptyset$. The set of all X-configurations of $\mathcal{E}$ is denoted by $\mathcal{F}_{\mathcal{E}}(X)$, or just $\mathcal{F}_{\mathcal{E}}$ when $X = \emptyset$.

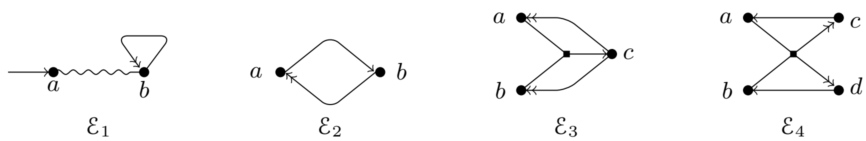


Fig. 5. Four CES . We adopt the following graphical notation for depicting CES : they are denoted as directed hypergraphs, where nodes stand for events. An hyperedge from a set of nodes X to node e denotes an enabling $X \circ e$, where $\circ = \vdash$ if the edge has a single arrow, and $\circ = \Vdash$ if the edge has a double arrow. A conflict $a\#b$ is represented by a waved line between a and b.

Example 4. Consider the CES in Figure 5. $\mathcal{E}_1$ has enablings $\vdash a$, $b \Vdash b$, and conflict $a\#b$. By Definition 11, $\emptyset, \{a\}, \{b\} \in \mathcal{F}_{\mathcal{E}_1}$, but $\{a,b\} \notin \mathcal{F}_{\mathcal{E}_1}$. $\mathcal{E}_2$ has enablings $a \vdash b$ and $b \Vdash a$. Here $\emptyset, \{a,b\} \in \mathcal{F}_{\mathcal{E}_2}$, while neither $\{a\}$ nor $\{b\}$ belong to $\mathcal{F}_{\mathcal{E}_2}$. Also, $\mathcal{F}_{\mathcal{E}_2}(\{b\}) = \{\emptyset, \{b\}, \{a,b\}\}$, and $\mathcal{F}_{\mathcal{E}_2}(\{a\}) = \{\emptyset, \{a\}, \{a,b\}\}$. $\mathcal{E}_3$ has enablings $\{a,b\} \vdash c$, $c \Vdash a$, and $c \Vdash b$. The only non-empty configuration of $\mathcal{E}_3$ is $\{a,b,c\}$. $\mathcal{E}_4$ has enablings $\{a,b\} \Vdash c$, $\{a,b\} \Vdash d$, $c \vdash a$, and $d \vdash b$. We have that $\{a,b,c,d\} \in \mathcal{F}_{\mathcal{E}_4}$. Note that, were one (or both) of the $\Vdash$ turned into a $\vdash$, then the only configuration would have been $\emptyset$.

Following [22], we assume the *axiom of finite causes*, that is, we always require an event to be enabled by a *finite* chain of events. For instance, consider the event structure: $\cdots\ e_n \to\ \cdots\ e_3 \to e_2 \to e_1 \to e_0$ For e_0 to happen, an infinite number of events must have happened *before* it. As in [22], we do not consider the set $\{e_i \mid i \geq 0\}$ as a configuration, because a justification of e_0 would require an infinite chain. Similarly, in the CES : $a_0 \twoheadleftarrow a_1 \twoheadleftarrow a_2 \twoheadleftarrow a_3\ \cdots\ \twoheadleftarrow a_n\ \cdots$ where, for a_0 to happen, an infinity of events must happen either *before* or *after* it, the set $\{a_i \mid i \geq 0\}$ is *not* a configuration according to Definition 11, because a justification of a_0 would require an infinite chain.

We relate Winskel's ES with CES in Theorem 3 below. First, we introduce the needed definitions.

Let $\mathcal{F}$ be a family of sets. We say a subset $\mathcal{A}$ of $\mathcal{F}$ is *pairwise compatible* if and only if $\forall e, e' \in \bigcup \mathcal{A}.\ \exists C \in \mathcal{F}.\ e, e' \in C$.

For a set of sets $\mathcal{F}$ we define the following three properties:

Coherence: If $\mathcal{A}$ is a pairwise compatible subset of $\mathcal{F}$, then $\bigcup \mathcal{A} \in \mathcal{F}$.
Finiteness: $\forall C \in \mathcal{F}.\ \forall e \in C.\ \exists C_0 \in \mathcal{F}.\ e \in C_0 \subseteq_{fin} C$
Coincidence-freeness:

$$\forall C \in \mathcal{F}.\ \forall e, e' \in C.\ \big(e \neq e' \implies (\exists C' \in \mathcal{F}.\ C' \subseteq C\ \wedge\ (e \in C' \iff e' \notin C')\big)$$

We say that $\mathcal{F}$ is a *quasi-family of configurations* iff it satisfies coherence and finiteness; if $\mathcal{F}$ also satisfies coincidence-freeness, then we call $\mathcal{F}$ a *family of configurations*. In that case, we say that $\mathcal{F}$ is a family of configurations of E when $\bigcup \mathcal{F} = E$.

A basic result of [22] is that the set of configurations of an ES forms a family of configurations. On the contrary, the set of configurations of a CES does not satisfy *coincidence-freeness*. A counterexample is the CES $\mathcal{E}_2$ in Example 4, where $\{a, b\} \in \mathcal{F}$, but there exists no configuration including only a or b. Indeed, the absence of coincidence-freeness is a peculiar aspect of circularity: if two events are circularly dependent, each configuration that contains one of them must contain them both.

Theorem 2. *For all* CES $\mathcal{E}$*, and for all* $X \subseteq E$*, the set* $\mathcal{F}_{\mathcal{E}}(X)$ *is a quasi-family of configurations.*

Despite faithfully representing the legitimate states of a system where all the credits are honoured, sets of configurations are not a precise semantic model for CES . Indeed, they are not able to discriminate among substantially different CES , e.g. like the following: $\mathcal{E}: \ a \Vdash b,\ b \Vdash a,\ \ \mathcal{E}': \ a \vdash b,\ b \Vdash a$, and $\mathcal{E}'' : a \Vdash b,\ b \vdash a$. It is easy to check that the sets of X-configurations of $\mathcal{E}, \mathcal{E}', \mathcal{E}''$ coincide, for all X. This contrasts with the different intuitive meaning of $\vdash$ and $\Vdash$, which is revealed instead by observing the traces: $\mathcal{T}_{\mathcal{E}} = \{\langle ab \rangle, \langle ba \rangle\}$, $\mathcal{T}_{\mathcal{E}'} = \{\langle ab \rangle\}$, and $\mathcal{T}_{\mathcal{E}''} = \{\langle ba \rangle\}$. To substantiate our feeling that configurations alone are not sufficiently discriminating for CES , in Theorem 3 we show that for all CES $\mathcal{E}$ there exists a CES $\mathcal{E}'$ *without* $\vdash$*-enablings* which has exactly the same configurations of $\mathcal{E}$. Therefore, the meaning of $\vdash$, that is the partial ordering of events, is completely lost by just observing configurations.

Definition 12. *Let* $\mathcal{F}$ *be a quasi-family of configurations of a set* E*. We define the* CES $\hat{\mathcal{E}}(\mathcal{F}) = (E, \#, \emptyset, \Vdash)$ *as follows:*

(a) $e \# e' \iff \forall C \in \mathcal{F}.\ e \notin C \ \vee\ e' \notin C$
(b) $X \Vdash e \iff CF(X) \wedge X \textit{ is finite} \wedge \exists C \in \mathcal{F}.\ e \in C \subseteq X \cup \{e\}$

Theorem 3. *For all quasi-families of configurations* $\mathcal{F}$*, we have* $\mathcal{F}_{\hat{\mathcal{E}}(\mathcal{F})} = \mathcal{F}$*.*

The consequence of this theorem, formalized by the corollary below, is that the $\Vdash$-enabling is the only (circular) causality relation needed, as the standard one can be encoded into this one.

Corollary 1. *For all ES* $\mathcal{E}$*, there exists a* CES $\mathcal{E}'$ *without* $\vdash$*-enablings such that* $\mathcal{F}_{\mathcal{E}} = \mathcal{F}_{\mathcal{E}'}$*.*

The theorem below yields a polynomial-time algorithm for computing the set $\mathcal{R}_{\mathcal{E}}$ of *reachable* events, i.e. those events which belong to some configuration of $\mathcal{E}$. The algorithm exploits Kleene's fixed point theorem, by defining the set $\mathcal{R}_{\mathcal{E}}$ as the greatest fixed point of a monotonic (increasing) function F.

Theorem 4. *For all* $X, Y, Z \subseteq E$*, let:*

$$G_Y(Z) \ = \ Y \cup \{e \ \mid \ Z \vdash e\} \qquad\qquad F(X) \ = \ \mathit{lfp}\, G_{\{e \ \mid \ X \Vdash e\}}$$

Then, for all finite conflict-free CES $\mathcal{E}$*, we have* $\mathcal{R}_{\mathcal{E}} = \mathit{gfp}\, F$

Following the characterization provided by Theorem 4, an algorithm for constructing $\mathcal{R}_{\mathcal{E}}$ can be devised as follows. Let X_0 be the set of all events in $\mathcal{E}$. At step 0, we compute $X_1 = F(X_0)$. This can be done by interpreting the (minimal) $\vdash$-enablings of $\mathcal{E}$ as a set of propositional Horn clauses, and then by applying the forward chaining algorithm with input $\{e \mid X_0 \Vdash e\}$. The forward chaining can be computed in polynomial-time in the number of $\vdash$-enablings. If $X_1 = X_0$, then we have finished, i.e. $X_1 = \mathcal{R}_{\mathcal{E}}$. Otherwise, we compute $X_2 = F(X_1)$ and so on, until reaching a fixed point. In the worst case, this requires $|E|$ steps, hence we have a polynomial-time algorithm for computing $\mathcal{R}_{\mathcal{E}}$.

5 Relating models

We now cast the three formalisms illustrated in the previous sections in a more coherent picture, by pointing out some relations among them. In particular, we show that:

- each conflict-free CES $\mathcal{E}$ can be associated to a Horn PCL theory Δ such that the atoms provable in Δ are exactly the events reachable in $\mathcal{E}$.
- each Horn PCL theory Δ can be associated to a LPN N such that the places marked in some honoured marking of N are exactly the atoms provable in Δ.

Taken together, these results state that finite conflict-free CES have the same expressivity of Horn PCL, and that PCL is no more expressive than LPNs; further, CES and LPNs provide two different models of Horn PCL.

5.1 CES *vs.* PCL

In Definition 13 we show a translation from CES into PCL formulae. In particular, our mapping is a bijection of finite, conflict-free CES into the Horn fragment of PCL , which comprises atoms, conjunctions and non-nested (standard/contractual) implications. When writing $X \vdash e$ we shall mean that X is a minimal set of events such that $(X, e) \in \vdash$ (similarly for $\Vdash$).

The encoding $[\cdot]$ maps an enabling $\vdash$ into an $\rightarrow$-clause, and a circular enabling $\Vdash$ into an $\twoheadrightarrow$-clause.

Definition 13. *Let* $\mathcal{E} = \langle E, \#, \vdash, \Vdash \rangle$ *be a conflict-free* CES*. The encoding* $[\mathcal{E}]$ *of* $\mathcal{E}$ *into a Horn PCL theory is defined as follows:*

$$[(X_i \circ e_i)_{i \in I}] = \{[X_i \circ e_i] \mid i \in I\} \qquad \textit{where } [\circ] = \begin{cases} \rightarrow & \textit{if } \circ = \vdash \\ \twoheadrightarrow & \textit{if } \circ = \Vdash \end{cases}$$

$$[X \circ e] = \left(\bigwedge X\right) [\circ]\, e$$

Notice that the encoding above can be inverted, i.e. one can also translate a Horn PCL theory into a conflict-free CES . The following theorem establishes the correctness and completeness of the encoding.

Theorem 5. *Let* $\mathcal{E}$ *be a finite, conflict-free* CES *. An event* e *is reachable in* $\mathcal{E}$ *iff* $[\mathcal{E}] \vdash_{\mathrm{PCL}} e$.

$$
\begin{aligned}
T &= \{(X, a, \rightarrow) \mid X \rightarrow a \in \Delta\} \cup \{(X, a, \twoheadrightarrow) \mid X \twoheadrightarrow a \in \Delta\} \\
S &= \mathcal{L}(\Delta) \times (T \cup \{*\}) \\
F(s,t) &= \begin{cases} 1 & \text{if } \big(s = (a,*) \wedge t = (X, a, -)\big) \bigvee \big(s = (a,t) \wedge t = (\{a\} \cup X, c, \rightarrow)\big) \\ 0 & \text{otherwise} \end{cases} \\
F(t,s) &= \begin{cases} 1 & \text{if } s = (a,t') \wedge t = (X, a, -) \wedge t' \neq * \\ 0 & \text{otherwise} \end{cases} \\
L(s,t) &= \begin{cases} 1 & \text{if } s = (a,t) \wedge t = (\{a\} \cup X, c, \twoheadrightarrow) \\ 0 & \text{otherwise} \end{cases} \\
\ell(x) &= \begin{cases} a & \text{if } x = (a,t) \in S \text{ or } x = (X, a, -) \in T \\ \bot & \text{otherwise} \end{cases} \\
m_0(s) &= \text{if } s = (a,*) \text{ then } 1 \text{ else } 0
\end{aligned}
$$

Fig. 6. Mapping from Horn PCL theories to Lending Petri Nets.

A consequence of Theorem 5 is that we can exploit properties of PCL to derive properties of conflict-free CES. For instance, from the tautology $(a \rightarrow b) \wedge (b \twoheadrightarrow c) \rightarrow (a \twoheadrightarrow c)$ of PCL we deduce that any conflict-free CES with enablings $a \vdash b$ and $b \Vdash c$ can be enriched with the enabling $a \Vdash c$, without affecting the reachable events.

5.2 LPNs *vs.* PCL

The result in Theorem 7 below gives a correspondence between LPNs and Horn PCL theories. Technically, we associate Horn PCL theories with LPNs which preserve the provability relation, in the sense that $\Delta \vdash X$ if and only if the LPN associated to Δ reaches a suitable configuration where all the atoms in X have been fired. The idea of our construction is to translate each Horn clause into a transition of an LPN, labeled with the action in the conclusion of the clause.

Definition 14. *For a Horn* PCL *theory Δ, we define $\mathcal{P}(\Delta)$ as the lending Petri net $\langle S, T, F, L, \ell, m_0 \rangle$ in Figure 6.*

We briefly comment below the construction in Figure 6. For each clause $X \circ a$ in Δ (with $\circ \in \{\rightarrow, \twoheadrightarrow\}$), we introduce a transition of the form $(X, a, \circ)$, and we label it with a (the component X keeps track of the premises of the implication). Places can have two forms: (a, t) for some label a and transition t, or $(a, *)$. Intuitively, a place $(a, *)$ is used to ensure that a transition labeled a can only be fired once, while a place (a, t) (labeled a) is used to collect the tokens produced by transitions labeled a, and to be consumed by transition t. Indeed, the definition of $F(t, s)$ ensures that each transition labeled a puts a token in each

$F(t,s)$	s_a^a	s_a^b	s_a^*	s_b^a	s_b^b	s_b^*
t_a	1	1				
t_b				1	1	

$F(s,t)$	s_a^a	s_a^b	s_a^*	s_b^a	s_b^b	s_b^*
t_a			1			
t_b		1				1

$L(s,t)$	s_a^a	s_a^b	s_a^*	s_b^a	s_b^b	s_b^*
t_a				1		
t_b						

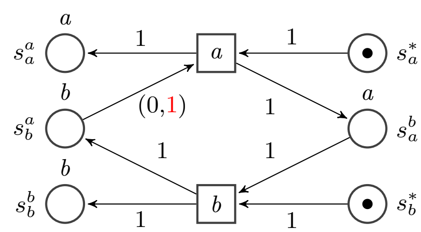


Fig. 7. LPN obtained from the PCL theory Δ of Example 5.

place labeled a, while that of $F(s,t)$ (resp. $L(s,t)$) yields a non-lending (resp. lending) arc from each place (a,t) to t whenever t has a in its premises. Observe that a transition $t = (X, a, \circ)$ puts a token in each place (a, t') with $t' \neq *$, and all the transitions bearing the same labels, say a, are mutually excluding each other, as they share the unique input place $(a, *)$. The initial marking will contain all the places in $\mathcal{L}(\Delta) \times \{*\}$; if a token is consumed from one of these places, then the place will be never marked again. Finally we observe that each transition has a non empty pre-set: for a transition $t = (X, a, \circ)$ we have at least $(a, *)$ in the pre-set, and in particular if $\circ = \twoheadrightarrow$ then the pre-set ${}^\bullet t$ contains exactly $(a, *)$, as ${}^\bullet t$ does not include places connected through lending arcs.

Example 5. Let $\Delta = a \rightarrow b,\ b \twoheadrightarrow a$. According to Definition 14, $\mathcal{P}(\Delta)$ has the following places and transitions:

$$\begin{aligned} T \ &= \{t_a, t_b\},\ \text{where } t_a = (b, a, \twoheadrightarrow), t_b = (a, b, \rightarrow) \\ S \ &= \{s_a^a, s_a^b, s_a^*, s_b^a, s_b^b, s_b^*\},\ \text{where} \\ &\quad s_a^a = (a, t_a), s_a^b = (a, t_b), s_a^* = (a, *), s_b^a = (b, t_a), s_b^b = (b, t_b), s_b^* = (b, *) \end{aligned}$$

The arcs and the labels of $\mathcal{P}(\Delta)$ are depicted in Figure 7. Observe that the LPN $\mathcal{P}(\Delta)$ has exactly one maximal firing sequence, i.e.:

$$s_a^*, s_b^* \xrightarrow{t_a} s_b^*, s_a^a, s_a^b, \overline{s_b^a} \xrightarrow{t_b} s_a^a, s_a^b$$

All the transitions in $\mathcal{P}(\Delta)$ labeled with a consume the token from the place $(a, *)$ in its pre-set, and this place cannot be marked again as it does not belong to the post-set of any transition, hence among them only one can fire. As each transition may be fired at most once, the net associated to a Horn PCL theory is an *occurrence net*, in the sense of van Glabbeek and Plotkin in [21].

A relevant property of $\mathcal{P}$ is that it is an homomorphism with respect to composition of theories. Thus, since both $\oplus$ is associative and commutative, we can construct an LPN from a Horn PCL theory $\Delta_1 \cdots \Delta_n$ componentwise, i.e. by composing the LPNs $\mathcal{P}(\Delta_1) \cdots \mathcal{P}(\Delta_n)$.

Theorem 6. *For all Δ_1, Δ_2, we have that $\mathcal{P}(\Delta_1, \Delta_2) \sim \mathcal{P}(\Delta_1) \oplus \mathcal{P}(\Delta_2)$.*

The reachable markings m of the LPN associated to a Horn PCL theory are completely characterized by a pair $(\overline{m}, \Omega(m))$, called *configuration* of the LPN.

Definition 15. *For a Horn* PCL *theory* Δ*, the* configuration *associated to a marking* $m \in Mk(\mathcal{P}(\Delta))$ *is the pair* $(\overline{m}, \Omega(m))$*, defined as: (i)* $\overline{m} = \{a \in \mathcal{L} \mid m((a,*)) = 0\}$ *(ii)* $\Omega(m) = \{\ell(s) \mid m(s) < 0\}$.

The first component is the set of the labels of the transitions that have been executed (the places $(a,*)$ are empty), and the second one is the set of labels of places with a negative marking, which means that the corresponding transitions have not been executed yet (as the LPN is correctly labeled). Clearly, the marking m is honored whenever $\Omega(m)$ is empty.

The following proposition establishes that configurations characterize markings of the LPNs associated to Horn PCL theories.

Proposition 3. *Let* m *and* m' *be markings of* $\mathcal{P}(\Delta)$*, for some Horn* PCL *theory* Δ*. If* $\overline{m} = \overline{m'}$ *and* $\Omega(m) = \Omega(m')$*, then* $m = m'$*.*

In Theorem 7 below we state the core correspondence between lending Petri nets and PCL: our construction maps the provability relation of PCL into the reachability of certain configurations in the associated LPN.

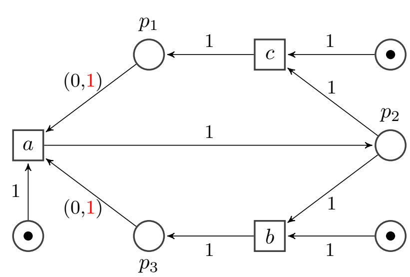
Theorem 7. *For all Horn* PCL *theories* Δ*, and for all conjunctions of atoms* X*:*

$$\Delta \vdash X \iff \exists m \in Mk(\mathcal{P}(\Delta)).\ X \subseteq \overline{m} \ \wedge\ \Omega(m) = \emptyset$$

6 Conclusions

We have presented three formalisms which can model circular causal dependencies, and we have established some relations among them. We conclude by pointing out some differences, as these may open new research directions.

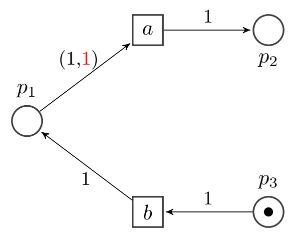
PCL and CES do not have a way to control the usage of resources, whereas LPNs have this feature: once a resource is used, it is not any longer available. For instance, consider the following lending Petri net:



Here the unique resource produced by a can be used either by b or by c, but never by both. Note that in no one of the maximal firing sequencess, i.e. $\emptyset \xrightarrow{a} p_2, \overline{p_1}, \overline{p_3} \xrightarrow{b} \overline{p_1}$ and $\emptyset \xrightarrow{a} p_2, \overline{p_1}, \overline{p_3} \xrightarrow{c} \overline{p_3}$, the reached marking is not honored.

Instead, by modelling the above situation as the PCL theory $a \to b$, $a \to c$ and $(b \wedge c) \twoheadrightarrow a$, we can deduce both $a \wedge b \wedge c$, as the atom a contractually implied by $b \wedge c$ can be "consumed" by *both* implications $a \to b$ and $a \to c$. The logical approach may be possibly accommodated when moving to resource-oriented logics like linear logic (indeed, the idea of connecting Petri nets and linear logic is not new, see [16]); this appears more difficult to obtain when semantic models (like event structures) are considered.

Lending Petri nets, unlike CES and PCL, can express situations where executing a transition depends on the availability of two resources, one of which may possibly be lent. Consider, for instance, the following LPN:



Transition a can be fired after b, because it needs at least one token in place p_1, whiel the other required token can be lent. Hence, we have the following firing sequence: $p_3 \xrightarrow{b} p_1 \xrightarrow{a} p_2, \overline{p_1}$. Were b allowed to fire twice (e.g. with p_3, p_3 as the initial marking), then an honored final marking would be reached.

We finally point out that these models have found a common ground in the framework of *contract-oriented computing* [11,9]. There, participants advertise their contracts to a contract broker. The broker composes contracts which admit some kind of agreement, and then establishes a session among the participants involved in them. In such scenario, the broker guarantees that — even in the presence of malicious participants — no interaction driven by the contract will ever go wrong. At worst, if some participant does not reach her objectives, then some other participant will be culpable of a contract infringement. In this workflow, it is crucial that contract brokers are honest, that is they never establish a session in the absence of an agreement among all the participants. Recall the scenario outlined in Section 1, where Alice and Bob are willing to exchange their resources. In her contract, Alice could promise to give a (unconditionally), declaring that her objective is to obtain b. A malicious contract broker could construct an attack by establishing a session between Alice and Mallory, whose contract just says to take a and give nothing in exchange. Mallory does not violate her contract, because it declares no obligations, and so Alice loses.

Models of circular causality like those presented here can be used by Alice to protect herself against untrusted contract brokers. By advertising the contract $b \twoheadrightarrow a$, Alice is saying that she promises to do a, but only under the guarantee that b will be done. Then, if the broker puts Alice in a session with Mallory (whose contract is *not* guaranteeing b), then Alice will not be culpable if she refuses to do a. Of course, also the contract $b \to a$ would have protected Alice, but this would have limited the interactions to those contexts where some other participants are not protected [8].

References

1. van der Aalst, W.M.P., Lohmann, N., Massuthe, P., Stahl, C., Wolf, K.: Multiparty contracts: Agreeing and implementing interorganizational processes. Comput. J. 53(1) (2010)
2. Baldan, P., Corradini, A., Montanari, U.: Contextual Petri nets, asymmetric event structures, and processes. Inf. Comput. 171(1), 1–49 (2001)
3. Bartoletti, M., Cimoli, T., Giamberardino, P.D., Zunino, R.: Contract agreements via logic. In: Proc. ICE (2013)
4. Bartoletti, M., Cimoli, T., Pinna, G.M.: Lending Petri nets and contracts. In: FSEN 2013. LNCS, vol. 8161, pp. 66–82 (2013)
5. Bartoletti, M., Cimoli, T., Pinna, G.M.: Lending petri nets (submitted), available at `http://tcs.unica.it/publications`
6. Bartoletti, M., Cimoli, T., Pinna, G.M., Zunino, R.: An event-based model for contracts. In: Proc. PLACES (2012)
7. Bartoletti, M., Cimoli, T., Pinna, G.M., Zunino, R.: Circular causality in event structures. Fundamenta Informaticae 134(3-4), 219–259 (2014)
8. Bartoletti, M., Cimoli, T., Zunino, R.: A theory of agreements and protection. In: Proc. POST (2013)
9. Bartoletti, M., Tuosto, E., Zunino, R.: Contract-oriented computing in CO_2. Scientific Annals in Computer Science 22(1), 5–60 (2012)
10. Bartoletti, M., Zunino, R.: A logic for contracts. Tech. Rep. DISI-09-034, DISI - Univ. Trento (2009)
11. Bartoletti, M., Zunino, R.: A calculus of contracting processes. In: LICS (2010)
12. Cimoli, T.: A theory of Agreement and Protection. Ph.D. thesis, University of Cagliari (May 2013)
13. Greibach, S.A.: Remarks on blind and partially blind one-way multicounter machines. Theoretical Computer Science 7, 311–324 (1978)
14. Hobbes, T.: The Leviathan (1651), chapter XIV
15. Kleene, S.: Introduction to metamathematics. North-Holland Publishing Company (1952)
16. Martí-Oliet, N., Meseguer, J.: From Petri nets to linear logic. Mathematical Structures in Computer Science 1(1), 69–101 (1991)
17. Moschovakis, J.: Intuitionistic logic. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy (2008)
18. Nielsen, M., Plotkin, G.D., Winskel, G.: Petri nets, event structures and domains, part i. Theor. Comput. Sci. 13, 85–108 (1981)
19. Reisig, W.: Petri Nets: An Introduction, Monographs in Theoretical Computer Science. An EATCS Series, vol. 4. Springer (1985)
20. Stotts, P.D., Godfrey, P.: Place/transition nets with debit arcs. Inf. Proc. Lett. 41(1) (1992)
21. van Glabbeek, R.J., Plotkin, G.D.: Configuration structures. In: LICS (1995)
22. Winskel, G.: Event structures. In: Advances in Petri Nets. pp. 325–392 (1986)
23. Zielonka, W.: Notes on finite asynchronous automata. Theoretical Informatics and Applications 21(2), 99–135 (1987)