Lending Petri Nets

Massimo Bartoletti, Tiziana Cimoli, G. Michele Pinna*

Dipartimento di Matematica e Informatica, Università degli Studi di Cagliari, Italy

Abstract

We study Lending Petri nets, an extension of Petri nets where places may carry a negative number of tokens. This allows for modeling contracts where a participant may promise to give some of her resources under the guarantee that some other resources will eventually be obtained in exchange. We then propose an interpretation of the Horn fragment of Propositional Contract Logic in Lending Petri nets. In particular, we show that provability in the logic corresponds to reachability of certain markings in nets, and that proof traces correspond to "honored" firing sequences in nets.

Keywords: Petri nets, Contracts, Intuitionistic Logic

1. Introduction

Service-oriented computing (SOC) and cloud computing technologies foster the implementation of complex software systems through the composition of basic building blocks, called *services*. Ensuring reliable coordination of such components is fundamental to avoid critical, possibly irreparable problems, ranging from economic losses in case of commercial activities, to risks for human life in case of safety-critical applications.

Ideally, in the SOC paradigm an application is constructed by dynamically discovering and composing services published by different organizations. Services have to *cooperate* to achieve the overall goals, while at the same time, they have to *compete* to achieve the specific goals of their stakeholders. These goals may be conflicting, e.g., in case of mutually distrusted organizations. Thus, services must play a double role: while cooperating together, they have to protect themselves against other services misbehavior (either unintentional or malicious).

The lack of precise guarantees about the reliability and security of services is a main deterrent for industries wishing to move their applications and business to the cloud [1]. Quoting from [1], "absent radical improvements in security technology, we expect that users will use *contracts* and courts, rather than clever security engineering, to guard against provider malfeasance".

Indeed, contracts are already a key ingredient in the design of SOC applications. For instance, in the approaches based on multi-party session types [2, 3], *global types* are used to specify the overall behavior (i.e. the choreography) of a distributed application; a global type is then projected into *local types*, which specify the behavior expected from each service involved in the whole application. The local types can be interpreted as the service contracts: if the actual implementation of each service respects its contract, then the overall application is guaranteed to enjoy some correctness properties, e.g., deadlock freedom and session fidelity. Another approach is the bottom-up one: each service advertises its contract (a local view), and a contract broker combines those services whose contracts admit an *agreement*. This can be done, for instance, by using session types as contracts, and by taking as agreement the possibility to synthesise from them a choreography — i.e. a global view — whose projections are the contracts themselves [4].

☆Work partially supported by Aut. Region of Sardinia under grants L.R.7/2007 CRP-17285 (TRICS), P.I.A. 2010 Project "Social Glue", by MIUR PRIN 2010-11 project "Security Horizons", and by EU COST Action IC1201 (BETTY).

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

Contracts may be seen as a way to formally specify and regulate the exchange of resources among the participants involved in an interaction. Typically, these resources are exchanged in a circular way: a participant provides the others with some resource, in order to obtain some others resources in return. For instance, assume that a participant A wants to obtain 1TB disk space from a cloud storage provider B, which in turn asks a payment of \$100. We could model the contract of A as "pay \$100", and that of B as "receive \$100, and then provide A with 1TB disk space". Intuitively, these two contracts admit an agreement: indeed, if both A and B are honest, then each one will perform its due action, so leading to a correct execution of the contracts. However, it would be unsafe for A to advertise a contract which just states "pay \$100", because this would admit an agreement also with the contract of a malicious provider which accepts the payment from A, and then gives nothing in return. To cope with this issue, A would like to advertise a stronger contract, e.g., "receive 1TB disk space from B, and then pay \$100". However, this contract would not admit an agreement with the one of the provider: since no one is willing to do the first move, we reach a deadlock situation, where the two resources are not exchanged.

Scenarios like the one outlined above are typical in interorganizational processes, where services are mutually distrusting, and may pursue their providers goals to the detriment of the other ones. The role of contracts in these competitive scenarios is twofold: on the one hand, they must allow participants to find agreements when there is a matching between the requested and offered resources; on the other hand, they must somehow protect their participants from interactions with malicious counterparts.

Petri nets [5] offer a natural way to formalize contracts: a resource can be seen as the presence of a token in a given place, and transferring a resource can be seen as firing a transition which moves the token to another place. For instance, the contract "receive \$100, and then provide A with 1TB disk space" can be formalised as a Petri net with two places (called, say, \$100 and 1TB), and a transition taking one token from place \$100 and putting one token in place 1TB. However, when composing this net with the one modelling the dual contract "receive 1TB disk space, and then pay \$100 to B", we obtain a net which can fire no transitions, as intuitively predicted above. To overcome this deadlock situation would need to weaken the conditions under which a resource is transferred, e.g., the transition of A's contract could be fired in the absence of a token in place 1TB, *under the guarantee* that B's transition will be eventually fired. A possible way to state this requirement is to record, after firing A's transition, that the resource \$100 has been given "on credit"; the contracts will admit an agreement only if credits will be honored, whatever the future choices of the participants.

Contribution. In this paper we propose a model of contracts based on Petri nets, along the lines of [6]. Differently from standard Petri nets, our *Lending Petri nets* (in short, LPNs) allow places to give tokens "on credit": technically, when a place gives a token on credit, its marking may become negative (whereas markings are always non-negative in standard Petri nets). To represent contracts, we enrich LPNs with some additional information: the participants associated to each transition, and their objectives. Taking inspiration from [7], we then interpret contracts as games, where each participant has a strategy to choose which transitions to fire in order to reach her objectives.

A set of contracts admits an agreement whenever, in their composition, each participant has a winning strategy, which allows her to reach the objectives, or make some other participant *liable* of a contract violation. LPNs can model contracts which, at the same time, admit an agreement and protect their participants. In the above scenario, participant A could formalise her contract as an LPN with a transition which takes one token on credit from place 1TB, and produces one token in place \$100. When this LPN is composed with the one of B which moves a token from \$100 to 1TB, there is a correct exchange of resources, and so the contracts admit an agreement. Instead, when the LPN of A is composed with the one of B which just takes the token from \$100 (and gives nothing in return), there is no agreement, because the credit \$100 is not honored.

Lending Petri nets preserve one of the main results of [6], i.e., compositional verification (Theorem 3.15). More precisely, if we have an agreement among a set of contracts, then we can indepently refine each of them (e.g., into a more detailed implementation), and be guaranteed that the composition of the refined contracts still enjoys agreement.

The other main contribution of this paper is a correspondence between Lending Petri nets and a logical

$N, N', \ldots$	Lending Petri nets	$\mathsf{A}, \mathsf{B}, \ldots \in \mathcal{A}$	Participants
$p, q, s, \ldots \in S$	Places	$\mathcal{C}, \mathcal{C}', \ldots$	Contract nets
$t, t', \ldots \in T$	Transitions	$\pi : S \cup T \rightharpoonup \mathcal{A}$	Ownership function
$F \subseteq (S \times T) \cup (T \times S)$	Weight function	$\gamma : \mathcal{A} \rightharpoonup \mathbf{2}^{S \to \mathbb{Z}}$	Objectives function
$L \subseteq (S \times T)$	Lending function	$\Sigma, \Sigma', \ldots$	Strategies
$\ell : S \cup T \rightharpoonup \mathcal{L} \ni a, b, \ldots$	Labeling	$\mathcal{C} \Downarrow_{\mathsf{A}}$	Agreement
$m : S \to \mathbb{Z}$	Marking	$\mathcal{C} \downarrow_{\mathsf{A}}$	Weak termination

Table 1: Summary of notation.

model of contracts, namely Propositional Contract Logic (PCL [8]). This is an extension of intuitionistic propositional logic (IPC), featuring a new binary connective $\twoheadrightarrow$, called *contractual implication*. The intuition is that, while the formula $(a \to b) \wedge (b \to a)$ in IPC is a "vicious circle", from which one can deduce neither a nor b, the formula $(a \twoheadrightarrow b) \wedge (b \twoheadrightarrow a)$ is a "virtuous circle", which entails both a and b. The relation with LPNs is clear: $a \to b$ is like a transition in a standard Petri net, which consumes one token from a and produces a token in b; instead, $a \twoheadrightarrow b$ is like a transition in an LPN, which puts a token in b also when the one in a is missing: in such case, a later transition is required to eventually honor the credit. We exploit this insight to provide a sound and complete model of the Horn fragment of PCL. More precisely, in Definition 4.3 we associate each Horn PCL theory Δ with an LPN $\mathcal{P}(\Delta)$, and in Theorem 4.10 we show that an atom is provable in Δ if and only if a certain marking is reachable in $\mathcal{P}(\Delta)$. In Theorem 4.28 we push further the correspondence between PCL and LPNs, by showing that proof traces [9] of a Horn PCL theory Δ are exactly the honored firing sequences in $\mathcal{P}(\Delta)$.

Structure of the paper. The rest of this paper is organized as follows. We introduce Lending Petri nets in Section 2. In Section 3 we use them as a model for contracts, and we set up a game-theoretic framework for contract agreement. In Section 4 we show that Lending Petri nets are a model of Horn PCL theories, and that honored firing sequences in LPNs correspond to proof traces in PCL. Finally, in Section 5 we discuss some related work, and we draw some conclusions. Table 1 summarises the syntactic categories and some notation used throughout the paper.

2. Lending Petri Nets

To introduce the kind of nets we will use in this paper, we start by recapping the usual notion of Petri nets [5]. A Petri net is a tuple (S, T, F, m_0), where S is a set of *places*, T is a set of *transitions* (with the constraint that $S \cap T = \emptyset$), $F \colon (S \times T) \cup (T \times S) \to \mathbb{N}$ is a *weight function*, and $m_0 : S \to \mathbb{N}$ is a function from places to natural numbers, called *marking*, which models the initial state of the net. Intuitively, $F(s, t) = n$ means that if the transition t can be fired, then n tokens must be 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 Petri nets by allowing transitions to fire even in the absence of the required number of tokens. However, this is done in a controlled manner: each time a transition is fired, only a fixed number of tokens can be taken on credit, and credits must be eventually honored. Technically, this is obtained by extending Petri nets with a *lending function* $L : S \times T \to \mathbb{N}$, which specifies how many tokens a transition may borrow from a place. The intuition is that if $F(s, t) = n$ and $L(s, t) = l$, then firing the transition t costs $n + l$ tokens, of which at least n must be already available in place s, while the other l can be taken on credit.

Additionally, we equip Lending Petri nets with a labeling ℓ of places and transitions, where labels are drawn from a set $\mathcal{L}$. These labels will be used later on in Section 2.2 to define composition of nets, similarly to the role played by input/output interfaces in open nets ([6] and [10, 11]).

Definition 2.1 (Lending Petri net). *A* lending Petri net *(LPN) is a tuple* $N = (S, T, F, L, \ell, m_0)$ *where:*

- (S, T, F, m_0) *is a Petri net,*

- $L\colon S \times T \to \mathbb{N}$ *is the* lending function*,*
- $\ell : S \cup T \rightharpoonup \mathcal{L}$ *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 proposed model is obviously a conservative extension of the standard one: indeed, a 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 last requirement asks that transitions cannot happen spontaneously. In literature, when Petri nets are used to model specific systems (and to reason on them, as e.g. in [12]), spontaneous transitions may be allowed. However, when the focus of the study is on *causal dependencies*, spontaneous transitions may cause problems, as dependencies may arise without a *justification* [13]. Since in this paper we are interested in causal dependencies (in particular, in circular ones), we rule out spontaneous transitions.

The drawing conventions we adopt are mostly standard: places are depicted as circles, transitions as squares, and arcs connecting transition to places are decorated with their weights. Lending arcs are drawn like standard ones, thus in case of 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 (only written when nonzero). We do not draw any arc between a place and a transition if both standard and lending arcs have zero weights. Further, we stipulate that subscripts on the net name carry over the names of the net components.

2.1. Semantics of LPNs

The behaviour of LPNs is defined by extending that of standard Petri nets. We define the *pre-set* ${}^\bullet x$ and the *post-set* $x^\bullet$ of a transition/place x as usual. We extend these standard notions to the lending function, by introducing the *lending pre-set* ${}^\circledcirc t$ of a transition t and the *lending post-set* $s^\circledcirc$ of a place s.

$$
\begin{array}{lll}
{}^\bullet x &=& \{y \in T \cup S \mid F(y,x) > 0\} \qquad x^\bullet = \{y \in T \cup S \mid F(x,y) > 0\} \\
{}^\circledcirc t &=& \{s \in S \mid L(s,t) > 0\} \qquad s^\circledcirc = \{t \in T \mid L(s,t) > 0\}
\end{array}
$$

All these definitions are lifted to sets of transitions/places in the obvious way.

The 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. If the marking of the place p is $n > 0$ we write n occurrences of p. If the marking of the place p is negative and equal to $-n$ we write n occurrences of $\overline{p}$, denoting with $\overline{p}$ a token lent by place p and not yet given back. Finally, we denote with $\emptyset$ the marking where each place contains no token. For instance, we display the marking $m = \{p_1 \mapsto 2, p_2 \mapsto -1\}$ as $p_1, p_1, \overline{p_2}$.

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 is accordingly updated. To do that, each place s in ${}^\bullet t \cup {}^\circledcirc t$ 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.

Definition 2.2 (**Step**). *Let $N = (S, T, F, L, \ell, m_0)$ 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[1] *t from m to m' (in symbols, $m \xrightarrow{t} m'$) whenever t is enabled at m, and, for all $s \in S$:*

$$m'(s) \;=\; m(s) - \big(F(s,t) + L(s,t)\big) + F(t,s)$$

[1]The word *step* is usually reserved to the execution of a subset of transitions, but here we prefer to stress the computational interpretation.

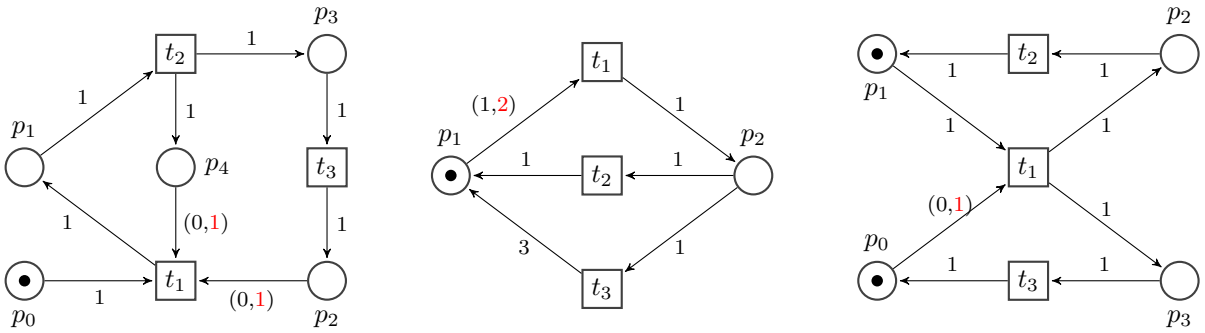


Figure 1: Three Lending Petri nets.

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.

Note that not all reachable markings represent good states of a system: indeed, 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. We call *honored markings* those markings which model states where all credits have been honored.

Definition 2.3 (**Honored marking**). *A marking m of N is* honored *iff $m(s) \geq 0$ for all places s of N.*

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

Example 2.4. *Consider the LPN N_1 in Figure 1. The initial marking is represented by 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 at the initial marking. We have exactly one maximal firing sequence:*

$$p_0 \xrightarrow{t_1} p_1, \overline{p_2}, \overline{p_4} \xrightarrow{t_2} \overline{p_2}, p_3 \xrightarrow{t_3} \emptyset$$

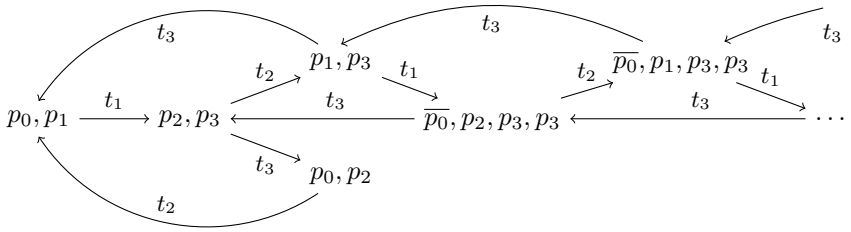
Note that the marking reached after firing all the three transitions is honored.

Consider now the LPN N_2 in Figure 1. The firing sequences of N_2 are described by the following LTS, with initial state p_1:

$$p_1 \xrightarrow{t_1} \overline{p_1}, \overline{p_1}, p_2 \xrightarrow{t_2} \overline{p_1}, \qquad \overline{p_1}, \overline{p_1}, p_2 \xrightarrow{t_3} p_1$$

The transition t_1 is enabled at p_1, and to fire it borrows 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 non-honored marking 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.

In the net N_3 of Figure 1, the transition t_1 is enabled at the initial marking p_0, p_1, and though it can lend a token from the place p_0, there is no reason to do so, as the place already contains a token. Firing t_2 and t_3 (in any order) leads again to the initial marking. Some of the firing sequences of N_3 are described by the following LTS (partially drawn):



2.2. Composing LPNs

We now introduce a notion of composition of LPNs. The intuition is rather simple: each time a labeled transition is executed in a component, tokens are produced in the equally-labeled places of the other component. The labeled places are the *interface* of the LPN (interface places), and the labeled transitions of a components are connected to the interface of the other component with an arc, connecting each transition to each place with the same label. The interface places without outgoing transitions play the role of *outputs*, while the others play the role of *inputs*. Notice that an input place may have a non empty pre-set.

Definition 2.5 (**Input and output places**). *For an LPN N, we define the set of* output places *$out(N)$, and the set of* input places *$in(N)$, respectively as follows:*

$$out(N) \;=\; \{s \in S \;\mid\; \ell(s) \neq \bot \text{ and } s^\bullet \cup s^{\circledcirc} = \emptyset\} \qquad in(N) \;=\; \{s \in S \;\mid\; \ell(s) \neq \bot \text{ and } s^\bullet \cup s^{\circledcirc} \neq \emptyset\}$$

Composition of LPNs is subject to some conditions, which altogether take the name of *correct labeling*, and are collected in Definition 2.6. The transitions of each component 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 (requirement (a) of Definition 2.6). Accordingly, all the labeled places in the post-set of a transition should carry the same label (requirement (b) of Definition 2.6). Finally, interface places are not initially marked (requirement (c)).

Definition 2.6 (**Correctly labeled LPN**). *An LPN (S, T, F, L, ℓ, m_0) 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) $\forall t \in {}^\bullet s.\ |\{\ell(s') \;\mid\; s' \in t^\bullet \;\wedge\; \ell(s') \neq \bot\}| = 1$,

(c) $m_0(s) = 0$.

Two LPNs are composed by adding a flow arc connecting transitions to the appropriate interface places. 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, as the output place can be safely removed, and the transitions putting tokens in the output place are directly connected to the input one. This models an asynchronous communication channel between nets, which does not preserve the order of messages (as usual in open nets, see Section 5). We require that arcs connecting a labeled transition to a labeled place have always weight 1.

Definition 2.7 (**Composition of LPNs**). *Let $N = (S, T, F, L, \ell, m_0)$ and $N' = (S', T', F', L', \ell', m'_0)$ be two correctly labeled LPNs. We say that N, N' are* composable *whenever*

- $S \cap S' = \emptyset = T \cap T'$,

$$
\begin{aligned}
\hat{S} &= (S \cup S') \setminus (\mathbb{S} \cup \mathbb{S}'), \ \textit{where} \\
&\quad \mathbb{S} = \{s \in \textit{out}(N) \mid \ell(s) \in \ell'(\textit{in}(N'))\} \text{ and } \mathbb{S}' = \{s \in \textit{out}(N') \mid \ell'(s) \in \ell(\textit{in}(N))\} \\
\hat{F}(s,t) &= \begin{cases} F(s,t) & \text{if } s \in S \text{ and } t \in T \\ F'(s,t) & \text{if } s \in S' \text{ and } t \in T' \\ 0 & \text{otherwise} \end{cases} \\
\hat{F}(t,s) &= \begin{cases} F(t,s) & \text{if } s \in S \text{ and } t \in T \\ F'(t,s) & \text{if } s \in S' \text{ and } t \in T' \\ 1 & \text{if } t \in T \text{ and } s \in S' \text{ and } \ell(t) = \ell'(s) \\ 1 & \text{if } t \in T' \text{ and } s \in S \text{ and } \ell'(t) = \ell(s) \\ 0 & \text{otherwise} \end{cases} \\
\hat{L}(s,t) &= \begin{cases} L(s,t) & \text{if } s \in S \text{ and } t \in T \\ L'(s,t) & \text{if } s \in S' \text{ and } t \in T' \\ 0 & \text{otherwise} \end{cases} \\
\hat{\ell}(x) &= \begin{cases} \ell(x) & \text{if } x \in S \cup T \\ \ell'(x) & \text{otherwise} \end{cases} \\
\hat{m}_0(s) &= \begin{cases} m_0(s) & \text{if } s \in S \\ m'_0(s) & \text{if } s \in S' \end{cases}
\end{aligned}
$$

Figure 2: Composition of two LPNs.

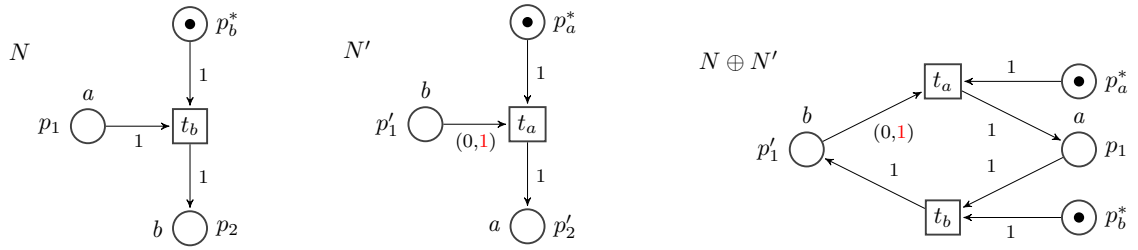


Figure 3: Two LPNs and their pairwise composition.

- $\forall t \in T,\ \forall s \in S.\ (\ell(s) \neq \bot \implies F(t,s) \leq 1)$,
- $\forall t \in T',\ \forall s \in S'.\ (\ell'(s) \neq \bot \implies F'(t,s) \leq 1)$

and in such case their composition $N \oplus N'$ *is the LPN* $\left(\hat{S}, T \cup T', \hat{F}, \hat{L}, \hat{\ell}, \hat{m}_0\right)$ *in Figure 2.*

Observe that composing two nets N and N' such that $\ell(S \cup T) \cap \ell'(S' \cup T') = \emptyset$ results in the disjoint union of the two nets.

Example 2.8. *Consider the LPNs in Figure 3. In* N*, the transition* t_b *can be fired only if a token is present in the input place* p_1 *(labeled* a*). In* N'*, the transition* t_a *is enabled, as it may lend a token from the input place* p'_1 *labeled* b*. The composition of these two nets is* $N \oplus N'$*, where now the execution of the transition* t_a *puts a token in* p_1 *(the resulting marking is* $p_1, p_b^*, \overline{p'_1}$*), and then firing* t_b *leads to the empty marking.*

Lemma 2.9. *For all composable LPNs* N, N'*:*

(a) $\textit{in}(N \oplus N') = \textit{in}(N) \cup \textit{in}(N')$*;*

(b) $\textit{out}(N \oplus N') = (\textit{out}(N) \cup \textit{out}(N')) \setminus (\mathbb{S} \cup \mathbb{S}')$.

The following proposition shows that composition of LPNs is associative and commutative.

Proposition 2.10. *Let N_1, N_2 and N_3 be pairwise composable LPNs. Then:*

(a) $N_1 \oplus N_2 = N_2 \oplus N_1$, *and*

(b) $N_1 \oplus (N_2 \oplus N_3) = (N_1 \oplus N_2) \oplus N_3$.

Proof. Commutativity is straightforward by Definition 2.7. We show that also associativity holds.

First, observe that the LPNs $N = N_1 \oplus (N_2 \oplus N_3)$ and $N' = (N_1 \oplus N_2) \oplus N_3$ have the same transitions. Let S be the set of places of N, let S' be that of N', let S_{12} that of $N_1 \oplus N_2$, and S_{23} that of $N_2 \oplus N_3$. We now prove that $S = S'$. Below, for i, j, k pairwise distinct, we denote with S_{ij} the set of places of the LPN $N_i \oplus N_j$, and with $\mathbb{S}_i^j$, $\mathbb{S}_{ij}^k$ and $\mathbb{S}_i^{jk}$ the sets:

$$\begin{aligned}
\mathbb{S}_i^j &= \{s \in out(N_i) \mid \ell(s) \in \ell(in(N_j))\} \\
\mathbb{S}_{ij}^k &= \{s \in out(N_i \oplus N_j) \mid \ell(s) \in \ell(in(N_k))\} \\
\mathbb{S}_i^{jk} &= \{s \in out(N_i) \mid \ell(s) \in \ell(in(N_j \oplus N_k))\}
\end{aligned}$$

In particular, we have that:

$$S_{12} = (S_1 \cup S_2) \setminus (\mathbb{S}_1^2 \cup \mathbb{S}_2^1) \tag{1}$$

$$S_{23} = (S_2 \cup S_3) \setminus (\mathbb{S}_2^3 \cup \mathbb{S}_3^2) \tag{2}$$

$$\begin{aligned}
\mathbb{S}_1^{23} &= \{s \in out(N_1) \mid \ell(s) \in \ell(in(N_2 \oplus N_3))\} \\
&= \{s \in out(N_1) \mid \ell(s) \in \ell(in(N_2) \cup in(N_3))\} \\
&= \{s \in out(N_1) \mid \ell(s) \in \ell(in(N_2))\} \cup \{s \in out(N_1) \mid \ell(s) \in \ell(in(N_3))\} \\
&= \mathbb{S}_1^2 \cup \mathbb{S}_1^3
\end{aligned} \tag{3}$$

$$\begin{aligned}
\mathbb{S}_3^{12} &= \{s \in out(N_3) \mid \ell(s) \in \ell(in(N_1 \oplus N_2))\} \\
&= \{s \in out(N_3) \mid \ell(s) \in \ell(in(N_1) \cup in(N_2))\} \\
&= \{s \in out(N_3) \mid \ell(s) \in \ell(in(N_1))\} \cup \{s \in out(N_3) \mid \ell(s) \in \ell(in(N_2))\} \\
&= \mathbb{S}_3^1 \cup \mathbb{S}_3^2
\end{aligned} \tag{4}$$

$$\begin{aligned}
\mathbb{S}_{12}^3 &= \{s \in out(N_1 \oplus N_2) \mid \ell(s) \in \ell(in(N_3))\} \\
&= \{s \in (out(N_1) \cup out(N_2)) \setminus (\mathbb{S}_1^2 \cup \mathbb{S}_2^1) \mid \ell(s) \in \ell(in(N_3))\} \\
&= \{s \in out(N_1) \setminus \mathbb{S}_1^2 \mid \ell(s) \in \ell(in(N_3))\} \cup \{s \in out(N_2) \setminus \mathbb{S}_2^1 \mid \ell(s) \in \ell(in(N_3))\} \\
&= (\mathbb{S}_1^3 \setminus \mathbb{S}_1^2) \cup (\mathbb{S}_2^3 \setminus \mathbb{S}_2^1)
\end{aligned} \tag{5}$$

$$\begin{aligned}
\mathbb{S}_{23}^1 &= \{s \in out(N_2 \oplus N_3) \mid \ell(s) \in \ell(in(N_1))\} \\
&= \{s \in (out(N_2) \cup out(N_3)) \setminus (\mathbb{S}_2^3 \cup \mathbb{S}_3^2) \mid \ell(s) \in \ell(in(N_1))\} \\
&= \{s \in out(N_2) \setminus \mathbb{S}_2^3 \mid \ell(s) \in \ell(in(N_1))\} \cup \{s \in out(N_3) \setminus \mathbb{S}_3^2 \mid \ell(s) \in \ell(in(N_1))\} \\
&= (\mathbb{S}_2^1 \setminus \mathbb{S}_2^3) \cup (\mathbb{S}_3^1 \setminus \mathbb{S}_3^2)
\end{aligned} \tag{6}$$

Summing up:

$$\begin{aligned}
S &= (S_1 \cup S_{23}) \setminus (\mathbb{S}_1^{23} \cup \mathbb{S}_{23}^1) \\
&= \big(S_1 \cup ((S_2 \cup S_3) \setminus (\mathbb{S}_2^3 \cup \mathbb{S}_3^2))\big) \setminus \big(\mathbb{S}_1^2 \cup \mathbb{S}_1^3 \cup (\mathbb{S}_2^1 \setminus \mathbb{S}_2^3) \cup (\mathbb{S}_3^1 \setminus \mathbb{S}_3^2)\big) && \text{by (2), (3), (6)} \\
&= (S_1 \cup S_2 \cup S_3) \setminus \big(\mathbb{S}_1^2 \cup \mathbb{S}_1^3 \cup \mathbb{S}_2^1 \cup \mathbb{S}_2^3 \cup \mathbb{S}_3^1 \cup \mathbb{S}_3^2\big) \\
&= (((S_1 \cup S_2) \setminus (\mathbb{S}_1^2 \cup \mathbb{S}_2^1)) \cup S_3) \setminus (\mathbb{S}_{12}^3 \cup \mathbb{S}_3^{12}) \\
&= (S_{12} \cup S_3) \setminus (\mathbb{S}_{12}^3 \cup \mathbb{S}_3^{12}) && \text{by (1), (5), (4)} \\
&= S'
\end{aligned}$$

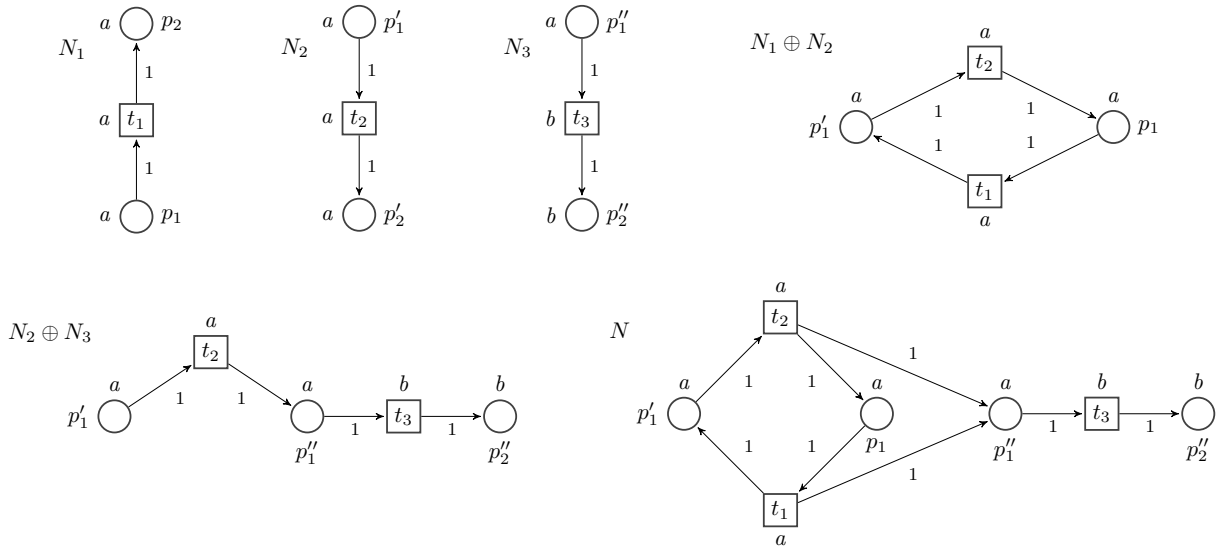


Figure 4: Three LPNs and their compositions $N_1 \oplus N_2$, $N_2 \oplus N_3$, $N = N_1 \oplus (N_2 \oplus N_3) = (N_1 \oplus N_2) \oplus N_3$.

Therefore, the places of N coincide with those of N'. The flow relations are the same in both N and N': indeed, from each labeled transition belonging to N_1 and each equally labeled interface place of N_2 and N_3, an arc with weight 1 is added, and the same for the transitions in N_2 and N_3. The same holds for the lending relation, which is inherited by those of the components, as well as the initial marking and the labeling. In conclusion, we obtain $N = N'$. □

Example 2.11. *Consider the LPNs N_1, N_2 and N_3 in Figure 4. The net $N_1 \oplus N_2$ is obtained by removing places p_2 and p'_2, and adding arcs from t_1 to p'_2, and from t_2 to p_1; the net $N_2 \oplus N_3$ is obtained by removing the place p'_2, and adding an arc from t_2 to p''_1. Finally, N can be either obtained from N_1 and $N_2 \oplus N_3$, removing the place p_2 and adding an arc from transition t_2 to p_1, and one from t_1 to p'_1 and p''_1, or from $N_1 \oplus N_2$ and N_3, by simply adding the arcs from t_1 to p''_1, and from t_2 to p''_1.*

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.

Definition 2.12 (Subnet). *Let $N = (S, T, F, L, \ell, m_0)$ be an LPN, and let $T' \subseteq T$. We define the* subnet $N|_{T'} = (S', T', F', L', \ell', m'_0)$*, where:*

(a) $S' = \{s \in S \mid \exists t \in T'.\ \big(F(t,s) > 0 \text{ or } F(s,t) > 0 \text{ or } L(s,t) > 0\} \cup \{s \in S \mid m_0(s) > 0\}$,

(b) $F' = F\,|_{(S' \times T') \cup (T' \times S')}$

(c) $L' = L\,|_{S' \times T'}$

(d) $\ell' = \ell\,|_{S' \cup T'}$

(e) $m'_0 = m_0\,|_{S'}$.

The composition $\oplus$ of two LPNs N_1 and N_2 does not have the property that, restricting to the transitions of one of the components, we obtain the LPN we started with, i.e., $(N_1 \oplus N_2)|_{T_i} \neq N_i$, for $i \in \{1, 2\}$. In fact, in $N_1 \oplus N_2$ there may be more places bearing a given label with respect to N_i, and as flow arcs are added, these places are not discharged when considering the subnet generated by T_i, with $i \in \{1, 2\}$. However these places are not initially marked, hence it may be that the nets have the same traces.

Definition 2.13 (**Trace equivalence**). *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 2.14. *For two composable LPNs N_1, N_2, we have that $N_i \sim (N_1 \oplus N_2)|_{T_i}$, for $i = 1, 2$.*

Proof. Consider two composable LPNs $N_1 = \big(S_1, T_1, F_1, L_1, \ell_1, m_1^0\big)$ and $N_2 = \big(S_1, T_1, F_1, L_1, \ell_1, m_1^0\big)$, and their composition $N_1 \oplus N_2 = \big(S, T, F, L, \ell, m^0\big)$ where $T = T_1 \cup T_2$, $S = (S_1 \cup S_2) \setminus (\mathbb{S}_1 \cup \mathbb{S}_2)$, and where F, L, ℓ and the initial marking are defined according to Definition 2.7. As N_1 and N_2 are composable, their transitions are disjoint, hence the set of transitions of $(N_1 \oplus N_2)|_{T_i}$ is T_i. According to Definition 2.12, the set $\tilde{S}$ of places of $(N_1 \oplus N_2)|_{T_i}$ comprises exactly those places connected to a transition in T_i, hence $S_i \setminus \mathbb{S}_i \subseteq \tilde{S}$. The weight function F and the lending function L restricted to $S_i \setminus \mathbb{S}_i$ and T_i are precisely F_i and L_i, and the places in the initial marking of N_i are contained in $S_i \setminus \mathbb{S}_i$. We observe that each place in $\tilde{S} \setminus (S_i \setminus \mathbb{S}_i)$ belongs to $out((N_1 \oplus N_2)|_{T_i})$, hence it is never used to enable a transition in the firing sequences of $(N_1 \oplus N_2)|_{T_i}$. Similarly, places in $\mathbb{S}_i \subseteq out(N_i)$ do not play any role in the firing sequences of N_i. Therefore, the firing sequences of $(N_1 \oplus N_2)|_{T_i}$ coincide with those of N. We can conclude that $Tr((N_1 \oplus N_2)|_{T_i}) = Tr(N_i)$. □

3. Contract nets

In this section we use LPNs to model behavioural contracts for concurrent systems. In this setting, the role of LPNs is to specify the obligations of a set of *participants* $\mathsf{A}, \mathsf{B}, \ldots \in \mathcal{A}$, who interact by exchanging resources (represented by tokens). Each transition t is owned by a single participant $\pi(t) \in \mathcal{A}$, which in any state may (or may not) have the obligation to fire such transition. The resources in the places $s \in {}^\bullet t \cup {}^{\circledcirc} t$ may possibly belong to a participant different from $\pi(t)$, and they are acquired by $\pi(t)$ when t is fired. When doing so results in a negative number of tokens in some places, it means that $\pi(t)$ has a debit, for which he is *liable* until it is honored.

Besides the obligations, we consider the participant objectives. The function γ associates each participant A involved in a contract with a set of markings, which represents the states where A has a positive payoff. Formally, $\gamma(\mathsf{A})$ is a set of *partial markings*, i.e., functions $\tilde{m} : S \rightharpoonup \mathbb{Z}$. The intuition is that A has a positive payoff in each marking m such that there exists some $\tilde{m} \in \gamma(\mathsf{A})$ such that $m(s) = \tilde{m}(s)$, for all $s \in \mathrm{dom}(\tilde{m})$, and this is denoted with $\gamma(\mathsf{A}) \models m$.

A *contract net* is an LPN together with the mappings π and γ.

Definition 3.1 (**Contract net**). *A* contract net $\mathcal{C}$ *is a triple (N, π, γ), where $N = (S, T, F, L, \ell, m_0)$ is an LPN, $\pi : S \cup T \rightharpoonup \mathcal{A}$, and $\gamma : \mathcal{A} \rightharpoonup \mathbf{2}^{S \rightharpoonup \mathbb{Z}}$. Additionally, we require that:*

(a) $\mathsf{A} \in \pi(T) \iff \gamma(\mathsf{A}) \neq \bot$

(b) $\forall t \in T.\ \pi(t^\bullet) = \{\pi(t)\} \neq \{\bot\}$

(c) $\gamma(\mathsf{A}) \models m \implies \forall s \in \pi^{-1}(\mathsf{A}).\ m(s) \geq 0$

(d) $\tilde{m} \in \gamma(\mathsf{A}) \implies \forall s \in \pi^{-1}(\mathsf{A}).\ s \notin out(N)$

Requirement (a) states that γ is defined for all participants having transitions in the LPN. Requirement (b) implies that each transition of a contract net belongs to a participant, and the places in the post-set of a transition belong to the same participant as well. Requirement (c) asks that the objectives of a participant A only comprise markings where A has no debits. Finally, requirement (c) states that there are no objectives on output places.

3.1. Prudent transitions

The key intuition of contract nets is that a transition is considered an obligation for A if and only if firing it will not make A definitively liable, and will make her still capable of reaching some marking in $\gamma(A)$. Indeed, not respecting these conditions is a failure for A, since she can be blamed for a contract breach, or

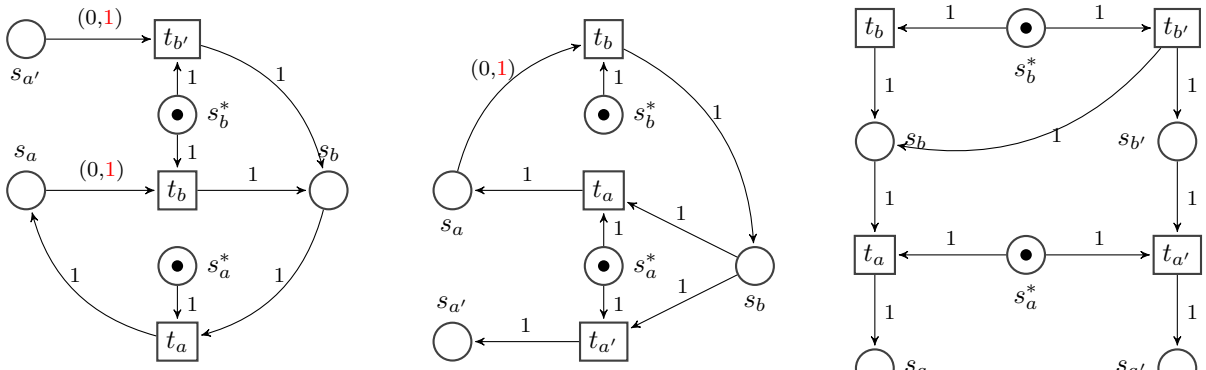


Figure 5: Three LPNs.

she will never be able to reach her objectives. Thus, our aim is to guarantee that *prudent* transitions, i.e. those which represent actual obligations, are identified correctly.

In order to provide a precise notion of prudence, we will interpret the token game of LPNs as a multi-player concurrent game, where participants can play by choosing their moves through individual *strategies*. Intuitively, a participant A wins when, in all the plays conforming to her strategy, she reaches a marking in $\gamma(\mathsf{A})$, or some other participant is (definitively) liable.

Our game-theoretic setting will make it possible to correctly render the fact that, from the point of view of a participant A, her choices are *angelic*, while the choices of the other participants are *demonic*. We will show that our winning property coincides with the *weak termination* property of [6] when we restrict to standard Petri nets and those strategies, which accept all the prudent transitions (Proposition 3.10).

We now formalize our game-theoretic setting. A strategy Σ for A is a function which associates each marking m to a set of enabled transitions of A.

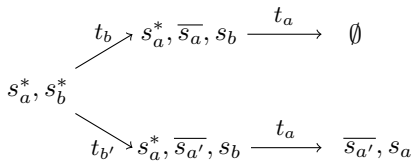
Definition 3.2 (**Strategy**). *We say that* $\Sigma : (S \to \mathbb{Z}) \to \mathbf{2}^T$ *is a* strategy *for* A *if* $t \in \Sigma(m)$ *implies that* $\pi(t) = \mathsf{A}$ *and* t *is enabled at* m. *We say that a firing sequence* $m_0 \xrightarrow{t_1} \cdots \xrightarrow{t_n} m_n$ conforms *to a strategy* Σ *for* A *when, for all* $i \in 1..n$, *if* t_i *is a transition of* A, *then* $t_i \in \Sigma(m_{i-1})$.

The definitions of prudent strategies and of innocent participants are mutually coinductive. A participant A is considered *innocent* at a marking m when she has no prudent transitions in m (otherwise A is *liable*). Given a marking m, a transition t is *prudent* at m whenever there exists a prudent strategy Σ which allows A to fire t at m. A strategy for A is prudent whenever, in all firing sequences where all other participants are innocent, the debits of A are eventually honored. Below we will denote with $\overline{\mathsf{A}}$ the set of all participants excluding A, and with $\overline{\Sigma}$ their overall strategy.

Definition 3.3 (**Prudence and innocence**). *We say that:*

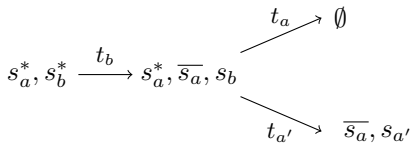
- *A strategy* Σ *for* A *is* prudent *iff, for all firing sequences* $m_0 \to^* m$ *conforming to* Σ, *and where all* $\mathsf{B} \in \overline{\mathsf{A}}$ *are innocent at* m, *we have that* $\forall s \in \pi^{-1}(\mathsf{A}).\ m(s) \geq 0$.
- *A transition* t *is* prudent *at* m *iff there exists a prudent strategy* Σ *such that* $t \in \Sigma(m)$.
- *A participant is* innocent *at* m *iff she has no prudent transitions at* m *(otherwise she is* liable*).*
- *A strategy for* A *is* innocent *if, for all* m, $\Sigma(m) = \emptyset$ *implies that* A *is innocent at* m.

Example 3.4. *Let $\mathcal{C} = (N_4, \pi, \gamma)$, where N_4 is the leftmost LPN depicted in Figure 5, $\pi(t_a) = \mathsf{A}$, and $\pi(t_b) = \pi(t_{b'}) = \mathsf{B}$. The participant objectives are irrelevant in this example (they will be defined in Example 3.6). The maximal firing sequences of $\mathcal{C}$ are described by the following LTS with initial state s_a^*, s_b^*:*



The transition t_b is prudent for B in the initial marking s_a^*, s_b^*, while $t_{b'}$ is not. It is easy to check that the strategy which allows B to choose only t_b at the first step is prudent: indeed, after that step, A either is liable, or she fires the transition t_a, which honours the debit of B. The imprudence of transition $t_{b'}$ follows from the fact that if such transition is fired, then B can no longer reach an honored marking.

Consider now the contract net (N_5, π, γ), where N_5 is depicted in Figure 5, $\pi(t_a) = \pi(t_{a'}) = \mathsf{A}$, and $\pi(t_b) = \mathsf{B}$. Here the transition t_b is prudent for B. To see why, consider the firing sequences of N_5, described by the following LTS with initial state s_a^*, s_b^*:



Indeed, even if A's strategy is to fire t_a, so making the marking at s_a negative, prudency of t_b only takes into account the debits of B (and not those of A).

We now define when a strategy is *winning*. To win, a participant A has to reach some of her objectives in all firing sequences conforming to her strategy, and to an arbitrary strategy for the other participants. Note that not all the possible strategies of the context are considered: actually, those where some $\mathsf{B} \neq \mathsf{A}$ is definitively liable are losing strategies for B, hence they are neglected.

Definition 3.5 (Winning strategy). *We say that a strategy Σ is* winning *for A iff, for all innocent strategies $\overline{\Sigma}$ of $\overline{\mathsf{A}}$, and for all firing sequences $m_0 \to^* m$ conforming to Σ and $\overline{\Sigma}$, there exists a firing sequence $m \to^* m'$ conforming to Σ and $\overline{\Sigma}$ such that $\gamma(\mathsf{A}) \models m'$.*

Example 3.6. *In the contract net $\mathcal{C}$ considered in Example 3.4, assume that $\pi(s_a) = \pi(s'_a) = \mathsf{A}$, and that the goals of A and B are the following:*

$$\gamma(\mathsf{A}) = \{\tilde{m} \mid \tilde{m}(s_a^*) = 0 \wedge \tilde{m}(s_a), \tilde{m}_{\mathsf{A}}(s'_a) \geq 0\} \qquad \gamma(\mathsf{B}) = \{\tilde{m} \mid \tilde{m}(s_a^*) = 0 \wedge \tilde{m}(s_b) \geq 0\}$$

which model the objective for A to consume the resource generated by t_b or by $t_{b'}$, and for B to have the resource generated by t_a. Participant B has a winning strategy in $\mathcal{C}$: indeed, B can either choose to fire t_b or t'_b at the first step, and then either make A culpable, or make it fire t_a. Instead, A has no winning strategies, because if B chooses to fire t'_b, then the marking at s'_a will remain negative. Hence, intuitively A does not agree on the contract $\mathcal{C}$. Note that in the LPN of $\mathcal{C}$ the only choice is that between t_b and $t_{b'}$, which is angelic from the point of view of B, and demonic from that of A.

Lemma 3.7. *If Σ is a winning strategy, then it is prudent.*

Proof. Let Σ be a winning strategy for A, let $\overline{\Sigma}$ be the context strategy, and let $m_0 \to^* m$ be a firing sequence conforming to Σ and $\overline{\Sigma}$. Since Σ is winning for A, then there exists a firing sequence $m \to^* m'$ conforming to Σ and $\overline{\Sigma}$, where $\gamma(\mathsf{A}) \models m'$. By item (c) of Definition 3.1, we conclude that m' is honored for A, hence Σ is a prudent strategy. □

We now define when a contract net admits an agreement among all the involved participants.

Definition 3.8 (**Agreement**). *We say that* A agrees on $\mathcal{C}$ *(in symbols,* $\mathcal{C} \Downarrow_{\mathsf{A}}$*) whenever* A *has a winning strategy in* $\mathcal{C}$*. We say that* $\mathcal{C}$ *admits an agreement (in symbols,* $\mathcal{C} \Downarrow$*) whenever* $\mathcal{C} \Downarrow_{\mathsf{A}}$ *for each participant* A*.*

We now relate our notion of agreement with *weak termination*, the property used in [6] to characterize good behaviour of open Petri nets. Weak termination captures the intuition that, notwithstanding the marking reached by the system, it is always possible for A to reach a marking in her objectives.

Definition 3.9 (**Weak termination**). *We say that* $\mathcal{C}$ *weakly terminates for* A *(in symbols,* $\mathcal{C} \downarrow_{\mathsf{A}}$*) iff:*

$$\forall m : \big(m_0 \to^* m \implies \exists m'.\, m \to^* m' \,\wedge\, \gamma(\mathsf{A}) \models m'\big)$$

Note that in the firing sequence $m_0 \to^* m$ both the choices of A and of the context are considered demonic, while in the firing sequence $m \to^* m'$ all the choices are considered angelic. This is different from our definition of agreement, because there the choices of A are always angelic, while those of the context are always demonic. Thus, the notions of agreement and of weak termination are not comparable, in general (that is, agreement does not imply weak termination, nor *vice versa*). However, we can formally relate them in the special case of nets without lending arcs (as those considered in [6]).

Proposition 3.10. *For all participants* A*, let the* maximal prudent strategy Σ^p_{A} *and the* maximal enabled strategy Σ^e_{A} *be defined, respectively, as follows:*

$$\Sigma^p_{\mathsf{A}}(m) \;=\; \{t \in \pi^{-1}(\mathsf{A}) \;\mid\; t \text{ is prudent at } m\} \qquad \Sigma^e_{\mathsf{A}}(m) \;=\; \{t \in \pi^{-1}(\mathsf{A}) \;\mid\; t \text{ is enabled at } m\}$$

Let $\mathcal{C}$ *be a contract net. For all markings* m *of* N*, we have:*

(a) $\Sigma^p_{\mathsf{A}}(m) \subseteq \Sigma^e_{\mathsf{A}}(m)$

(b) If N *has no lending arcs, then* $\Sigma^p_{\mathsf{A}}(m) \supseteq \Sigma^e_{\mathsf{A}}(m)$

(c) If Σ^e_{A} *is winning for* A*, then* $\mathcal{C} \downarrow_{\mathsf{A}}$*.*

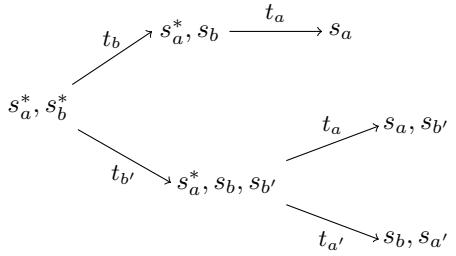
Proof. For item (a), let t be prudent for A at m. By Definition 3.3, there exists some prudent strategy Σ of A such that $t \in \Sigma(m)$. By Definition 3.2, since Σ is a strategy, then t must be enabled at m.

For item (b), let t be enabled at m, and let $m_0 \to^* m$ be a firing sequence conforming to Σ^e_{A}. Since N has no lending arcs, it must be $m(s) \geq 0$ for all $s \in \pi^{-1}(\mathsf{A})$. Hence, t is prudent at m, and so $t \in \Sigma^p_{\mathsf{A}}(m)$.

For item (c), assume that $m_0 \to^* m$, for some marking m. Clearly, the firing sequence $m_0 \to^* m$ conforms to the maximal enabled strategy Σ^e_{A}. Let $\overline{\Sigma}$ be an innocent strategy of $\overline{\mathsf{A}}$ conforming to $m_0 \to^* m$. Since Σ^e_{A} is winning for A, then by Definition 3.5 it follows that there exists a firing sequence $m \to^* m'$ which conforms to Σ^e_{A} and $\overline{\Sigma}$, and such that $\gamma(\mathsf{A}) \models m'$. By Definition 3.9, we conclude that $\mathcal{C} \downarrow_{\mathsf{A}}$. □

Notice that Proposition 3.10 above implies that, in the absence of lending arcs, if the maximal prudent strategy is winning, then we also obtain weak termination. However, agreement (with a non-prudent strategy) does not imply weak termination: Indeed, in the following example we show a contract net which admits an agreement, although the maximal prudent strategy is not winning (and weak termination does not hold as well).

Example 3.11. *Consider the contract net* $\mathcal{C} = (N_6, \pi, \gamma)$*, where* N_6 *is depicted in Figure 5,* $\pi(t_a) = \pi(t_{a'}) = \mathsf{A}$*,* $\pi(t_b) = \pi(t_{b'}) = \mathsf{B}$*, the objective of* B *is to have a token in* s_a*, and that of* A *is to have no tokens in* s^*_a*. The firing sequences of* N_6 *are described by the following LTS, with initial state* s^*_a, s^*_b*:*



We have that both A *and* B *agree on* $\mathcal{C}$*: indeed, the strategy which allows* B *to fire* t_b *(but not* $t_{b'}$*) is winning for* B*, and the maximal prudent strategy is winning for* A*. Instead, the maximal prudent strategy is not winning for* B*, because if* $t_{b'}$ *is fired, then* A *can choose to fire* $t_{a'}$*, which prevents* B *from reaching his objective. Note that weak termination does hold for* A*, but not for* B*.*

3.2. Composition and refinement

Contract nets can be composed along the way outlined in Definition 2.7, with some further requirements about participants and labels.

Definition 3.12 (**Contract net composition**)**.** *Let* N_1 *and* N_2 *be composable LPNs such that* $\ell_1(T_1) \cap \ell_2(T_2) = \emptyset$*, and let* $\mathcal{C}_1 = (N_1, \pi_1, \gamma_1)$ *and* $\mathcal{C}_2 = (N_2, \pi_2, \gamma_2)$ *be contract nets such that* $\pi_1(T_1) \cap \pi_2(T_2) = \emptyset$*. We define* $\mathcal{C}_1 \oplus \mathcal{C}_2 = (N_1 \oplus N_2, \pi, \gamma)$*, where*

$$\pi(x) = \begin{cases} \pi_i(x) & \text{if } \pi_i(x) \neq \bot \\ \mathsf{A} & \text{if } (\pi_1 \cup \pi_2)(x) = \bot,\ {}^\bullet x \neq \emptyset \text{ and } \forall t \in {}^\bullet x : \pi(t) = \mathsf{A} \\ \bot & \text{otherwise} \end{cases}$$

$$\gamma(\mathsf{A}) = \begin{cases} \bigcup_{m \in \gamma_1(\mathsf{A})} \delta_1(m) & \text{if } \gamma_1(\mathsf{A}) \neq \bot \\ \bigcup_{m \in \gamma_2(\mathsf{A})} \delta_2(m) & \text{if } \gamma_2(\mathsf{A}) \neq \bot \\ \bot & \text{otherwise} \end{cases} \qquad \delta_i(m) = \left\{ \tilde{m} \;\middle|\; \begin{array}{ll} \tilde{m}(s) = m(s) & \text{if } s \in S_i \\ \tilde{m}(s) \geq 0 & \text{if } s \in S_j \cap (\pi_i^{-1}(\mathsf{A}) \cap T_i)^\bullet, j \neq i \\ \tilde{m}(s) = \bot & \text{otherwise} \end{array} \right\}$$

We say that $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *are* composable *whenever* $\mathcal{C}_1 \oplus \mathcal{C}_2$ *respect the constraints in Definition 3.1.*

Two contract nets are composable whenever their transitions have different labels, $\pi_1(T_1) \cap \pi_2(T_2) = \emptyset$, and the resulting structure is a contract net. The set of participants of the composition is obtained as expected: each transition inherits its participant, and each input place without an assigned participant may get a new one, provided that all the transitions putting tokens in it are associated with the same participant. The objective mappings are inherited from each component. Observe that we require that an action can be performed only by one of the components, but the other may *use* the tokens produced by the execution of such action. Associativity and commutativity of composition between contract nets follow from the corresponding properties of the composition of the underlying LPNs.

Example 3.13. *Consider the contract nets* $\mathcal{C} = (N, \pi, \gamma)$ *and* $\mathcal{C}' = (N', \pi', \gamma')$*, where* N *and* N' *are the LPNs in Figure 3,* $\pi(t_b) = \mathsf{B} = \pi(p_2)$*, and* $\pi'(t_a) = \mathsf{A} = \pi'(p_2')$*. The objective function of* $\mathcal{C}$ *is defined as follows:* $\gamma(\mathsf{A}) = \bot$*, and* $\gamma(\mathsf{B})$ *contains the partial markings* $\tilde{m}$ *such that* $\tilde{m}(p_b^*) = 0$*,* $\tilde{m}(p_2) \geq 0$*, and* $\tilde{m}$ *is undefined in* p_1*. The objective function of* $\mathcal{C}'$ *is defined as follows:* $\gamma'(\mathsf{B}) = \bot$*, and* $\gamma(\mathsf{A})$ *contains the partial markings* $\tilde{m}$ *such that* $\tilde{m}(p_a^*) = 0$*,* $\tilde{m}(p_2') \geq 0$*, and* $\tilde{m}$ *is undefined in* p_1'*.*

The contract nets $\mathcal{C}$ *and* $\mathcal{C}'$ *are composable, and their composition is* $\mathcal{C} \oplus \mathcal{C}' = (N \oplus N', \hat{\pi}, \hat{\gamma})$*, where* $\hat{\pi}$ *is such that* $\hat{\pi}(p_1') = \mathsf{B}$ *and* $\hat{\pi}(p_1) = \mathsf{A}$*,* $\hat{\gamma}(\mathsf{A})$ *contains the partial markings* $\tilde{m}$ *such that* $\tilde{m}(p_a^*) = 0$*,* $\tilde{m}(p_b^*) = \bot = \tilde{m}(p_1)$ *and* $\tilde{m}(p_1') \geq 0$*, and* $\hat{\gamma}(\mathsf{B})$ *contains the partial markings* $\tilde{m}$ *such that* $\tilde{m}(p_b^*) = 0$*,* $\tilde{m}(p_a^*) = \bot = \tilde{m}(p_1')$ *and* $\tilde{m}(p_1) \geq 0$*.*

Assume now that π *is defined also for* p_1*, and* $\pi(p_1) = \mathsf{A}$*. The two contract nets are still composable, and their composition is exactly as in the previous case. Instead, if* $\pi(p_1) = \mathsf{C}$*, then the contract nets are no longer composable, as the result of the operation is not a contract net (as* $\pi(p_1) \neq \pi(t_b)$*, so violating constraint* (b) *of Definition 3.1).*

We now introduce a notion of refinement between two contract nets (similar to the notion of *accordance* in [6]), and then we show that it allows for compositional verification (Theorem 3.15).

Definition 3.14 (**Refinement**). *A contract net* $\mathcal{C}'$ refines $\mathcal{C}$ *(in symbols,* $\mathcal{C}' \sqsubseteq \mathcal{C}$*) iff*

$$\forall \mathsf{A}.\ \left(\mathcal{C} \Downarrow_{\mathsf{A}} \implies \mathcal{C}' \Downarrow_{\mathsf{A}}\right)$$

If a contract net $\mathcal{C}$ is obtained by a composition of contract nets, i.e., $\mathcal{C} = \bigoplus_i \mathcal{C}_i$, we can ask what happens if there is some $\mathcal{C}'_i$ which refines $\mathcal{C}_i$, for each i. The following theorem gives the desired answer, that is a compositional criterion to check if an agreement of a SOC application is possible. One starts from an global specification (e.g. a choreography), projects it into a set of local views, and then refines each of them into a service implementation. These services can be verified independently (for refinement), and it is guaranteed that their composition still enjoys the desired property.

Theorem 3.15. *Let* $\mathcal{C} = \bigoplus_{i \in 1..n} \mathcal{C}_i$ *be such that* $\mathcal{C} \Downarrow$*, and let* $\mathcal{C}' = \bigoplus_{i \in 1..n} \mathcal{C}'_i$ *be such that* $\mathcal{C}'_i \sqsubseteq \mathcal{C}_i$*, for all* $i \in 1..n$*. Then,* $\mathcal{C}' \Downarrow$*.*

Proof. We prove the statement for $n = 2$; the generalization is obvious. As $\mathcal{C} = \mathcal{C}_1 \oplus \mathcal{C}_2$ admits an agreement and $\mathcal{C}'_1$ refines $\mathcal{C}_1$ we know that $\mathcal{C}'' = \mathcal{C}'_1 \oplus \mathcal{C}_2$ admits an agreement as well by virtue of Proposition 2.14 (in fact, traces equivalence implies that strategies are preserved). Now consider that $\mathcal{C}'_2$ refines $\mathcal{C}_2$, and that $\mathcal{C}'' = \mathcal{C}'_1 \oplus \mathcal{C}_2$ admits an agreement, hence also $\mathcal{C}' = \mathcal{C}'_1 \oplus \mathcal{C}'_2$ admits an agreement as well, as required. □

4. LPNs as a model of Propositional Contract Logic

In this section we establish a correspondence between a logical model for contracts, namely Propositional Contract Logic (PCL [8]), and Lending Petri nets. PCL extends intuitionistic propositional logic with a connective $\twoheadrightarrow$ (called *contractual implication*) in order to allow for circular assume-guarantee reasoning. For instance, $(b \circ a) \wedge (a \bullet b) \to a \wedge b$ is a theorem in PCL provided that $\circ = \twoheadrightarrow$ or $\bullet = \twoheadrightarrow$ (or both). The insight of $\twoheadrightarrow$ in PCL is similar to that lending arcs in LPNs: to prove a formula ψ from a clause $\varphi \twoheadrightarrow \psi$, one needs to prove φ, but to do that one can somehow take ψ "on credit".

In Theorem 4.10 we will show that a particular class of LPNs, that is *occurrence* LPNs, can be used to give a model of the Horn fragment of PCL. This result is particularly relevant also because it gives us a clear insight about how a linear variant of PCL (not studied yet) would have to work on its Horn fragment. To further strengthen the correspondence between PCL and LPNs, we will show in Theorem 4.28 that another crucial notion in PCL, namely that of *proof traces*, has a clear counterpart in the realm of LPNs.

4.1. Propositional Contract Logic

PCL formulae, ranged over greek letters $\varphi, \varphi', \ldots$, are defined as follows, where we assume that the prime formulae $a, b, \ldots$ coincide with the atoms in $\mathcal{L}$:

$$\varphi \ ::= \ \bot \mid \top \mid a \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \to \varphi \mid \varphi \twoheadrightarrow \varphi$$

The natural deduction system for PCL [9] extends that for IPC with the last three rules in Figure 6 (wherein, in all the rules, Δ is a set of PCL formulae, and where Δ, φ is a shorthand for $\Delta \cup \{\varphi\}$). Provable formulae are contractually implied, according to rule ($\twoheadrightarrow$I1). Rule ($\twoheadrightarrow$I2) provides $\twoheadrightarrow$ with the same weakening properties of $\to$. The paradigmatic rule is ($\twoheadrightarrow$E), which allows for the elimination of $\twoheadrightarrow$. Compared to the rule ($\to$E) for elimination of $\to$ in IPC, the only difference is that in the context used to deduce the antecedent φ, rule ($\twoheadrightarrow$E) also allows for using as hypothesis the consequence ψ.

A simple example of a natural deduction proof in PCL follows.

Example 4.1. *Let* $\Delta = a \to b, b \twoheadrightarrow a$*. A proof of* $\Delta \vdash a$ *in natural deduction is the following:*

$$\dfrac{\Delta \vdash b \twoheadrightarrow a \qquad \dfrac{\Delta, a \vdash a \to b \quad \Delta, a \vdash a}{\Delta, a \vdash b}\ (\to\mathrm{E})}{\Delta \vdash a}\ (\twoheadrightarrow\mathrm{E})$$

from which we can obtain a proof of $\Delta \vdash b$*, by using rule* ($\to$E)*.*

$$\frac{}{\Delta, \varphi \vdash \varphi}\ (\text{ID}) \qquad \frac{\Delta \vdash \varphi \qquad \Delta \vdash \psi}{\Delta \vdash \varphi \wedge \psi}\ (\wedge\text{I}) \qquad \frac{\Delta \vdash \varphi \wedge \psi}{\Delta \vdash \varphi}\ (\wedge\text{E1}) \qquad \frac{\Delta \vdash \varphi \wedge \psi}{\Delta \vdash \psi}\ (\wedge\text{E2})$$

$$\frac{\Delta \vdash \varphi}{\Delta \vdash \varphi \vee \psi}\ (\vee\text{I1}) \qquad \frac{\Delta \vdash \psi}{\Delta \vdash \varphi \vee \psi}\ (\vee\text{I2}) \qquad \frac{\Delta \vdash \varphi \vee \psi \qquad \Delta, \varphi \vdash \rho \qquad \Delta, \psi \vdash \rho}{\Delta \vdash \rho}\ (\vee\text{E})$$

$$\frac{\Delta, \varphi \vdash \psi}{\Delta \vdash \varphi \to \psi}\ (\to\text{I}) \qquad \frac{\Delta \vdash \varphi \to \psi \qquad \Delta \vdash \varphi}{\Delta \vdash \psi}\ (\to\text{E})$$

$$\frac{\Delta \vdash \psi}{\Delta \vdash \varphi \twoheadrightarrow \psi}\ (\twoheadrightarrow\text{I1}) \qquad \frac{\Delta \vdash \varphi \twoheadrightarrow \psi \qquad \Delta, \varphi' \vdash \varphi \qquad \Delta, \psi \vdash \varphi' \twoheadrightarrow \psi'}{\Delta \vdash \varphi' \twoheadrightarrow \psi'}\ (\twoheadrightarrow\text{I2}) \qquad \frac{\Delta \vdash \varphi \twoheadrightarrow \psi \qquad \Delta, \psi \vdash \varphi}{\Delta \vdash \psi}\ (\twoheadrightarrow\text{E})$$

Figure 6: Natural deduction system for PCL.

The decidability of the provability relation $\vdash$ of PCL has been proved in [8], by exploiting the cut elimination property enjoyed by the sequent calculus of PCL, which has been shown equivalent to the natural deduction system in [9].

In this paper we shall focus on the *Horn fragment* of PCL, which comprises atoms, conjunctions, and non-nested implications (both intuitionistic and contractual). This fragment is particularly insightful, because it has strong relations with LPNs, as we will show later. Hereafter, we let X, Y range over conjunctions of atoms $\bigwedge\{a_1, \ldots, a_n\}$ with $n \geq 0$, and we let $\top$ denote $\bigwedge \emptyset$. We denote with $\mathcal{L}(\Delta)$ the set of atoms occurring in Δ. When clear from the context, we shall use X, Y to denote interchangeably conjunctions or *sets* of atoms. We use $\sigma, \eta, \ldots$ to range over sequences of atoms, and we denote with $\overline{\sigma}$ the set of atoms in σ. Furthermore, we denote with σ_i the prefix of σ containing exactly i atoms.

Definition 4.2 (**Horn PCL theory**). *A* Horn PCL theory *is a finite set of clauses of the form $X \to a$ or $X \twoheadrightarrow a$. We identify the atomic formula a with the clause $\top \to a$.*

4.2. Encoding PCL *into LPNs*

In this section we exploit LPNs to give a model to the Horn fragment of PCL. 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. 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 (i.e., transitions labeled with atoms in X) have been fired.

Definition 4.3 (**Mapping PCL to LPNs**). *For a Horn* PCL *theory Δ, we define $\mathcal{P}(\Delta)$ as the LPN (S, T, F, L, ℓ, m_0) in Figure 7.*

We briefly comment below the construction in Figure 7. For each clause $X \circ a$ in Δ (with $\circ \in \{\to, \twoheadrightarrow\}$), 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 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

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

Figure 7: Mapping from Horn PCL theories to Lending Petri Nets.

$F(t,s)$	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						

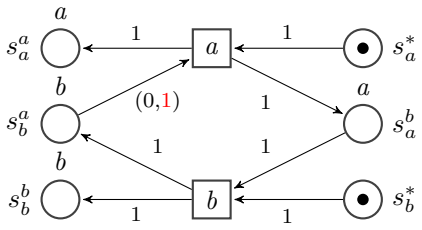


Figure 8: LPN obtained from the PCL theory Δ of Example 4.4.

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 4.4. *Let $\Delta = a \rightarrow b,\ b \twoheadrightarrow a$. According to Definition 4.3, $\mathcal{P}(\Delta)$ has the following places and transitions:*

$$T = \{t_a, t_b\},\ \textit{where } t_a = (b, a, \twoheadrightarrow), t_b = (a, b, \rightarrow)$$
$$S = \{s_a^a, s_a^b, s_a^*, s_b^a, s_b^b, s_b^*\},\ \textit{where } s_a^a = (a, t_a), s_a^b = (a, t_b), s_a^* = (a, *), s_b^a = (b, t_a), s_b^b = (b, t_b), s_b^* = (b, *)$$

The arcs and the labels of $\mathcal{P}(\Delta)$ are depicted in Figure 8. Observe that the LPN $\mathcal{P}(\Delta)$ has exactly one maximal firing sequence, i.e.:

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

Example 4.5. *Let $\Delta = a \rightarrow b,\ b \twoheadrightarrow a,\ b \wedge c \twoheadrightarrow a$. The translation gives the LPN in Figure 9. The transitions are*

$$T = \{t_a, t_{a'}, t_b\},\ \textit{where } t_a = (b \wedge c, a, \twoheadrightarrow), t_{a'} = (b, a, \twoheadrightarrow), t_b = (a, b, \rightarrow)$$

and the places are

$$S = \{s_a^a, s_a^b, s_a^{a'}, s_a^*, s_b^a, s_b^b, s_b^{a'}, s_b^*, s_c^a, s_c^b, s_c^{a'}, s_c^*\},\ \textit{where } s_a^a = (a, t_a), s_a^b = (a, t_b), s_a^{a'} = (a, t_{a'}), s_a^* = (a, *),$$
$$s_b^a = (b, t_a), s_b^b = (b, t_b), s_b^{a'} = (b, t_{a'}), s_b^* = (b, *), s_c^a = (c, t_a), s_c^b = (c, t_b), s_c^{a'} = (c, t_{a'}), s_c^* = (c, *)$$

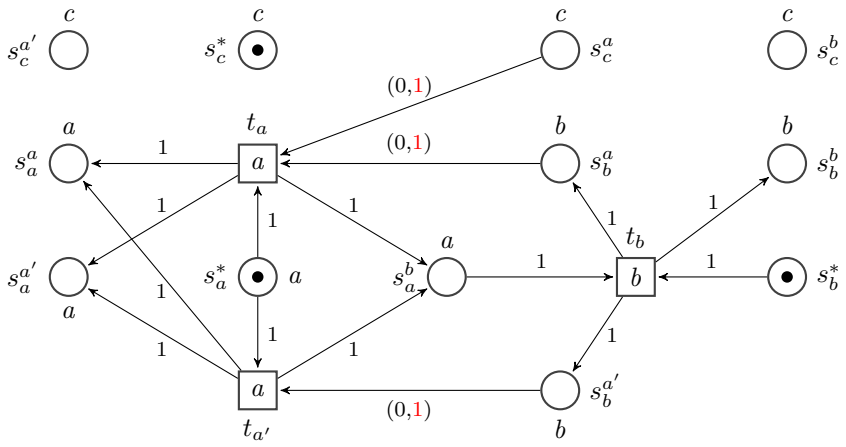


Figure 9: LPN obtained from the PCL theory Δ of Example 4.5.

The transition t_a is enabled at the initial marking s_a^*, s_b^*, s_c^*, and firing it would result in the marking $s_a^a, s_a^b, s_a^{a'}, s_b^*, s_c^*, \overline{s_b^a}, \overline{s_c^a}$. This marking cannot be honored, because there is no transition labeled with c. The corresponding maximal firing sequence is:

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

Instead, firing the other transition for a, namely $t_{a'}$, would result in the marking $s_a^a, s_a^b, s_a^{a'}, s_b^*, s_c^*, \overline{s_b^{a'}}$. There, the transition labeled b can be fired. The maximal firing sequence is then

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

where the final marking is honored.

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 [14]. We enumerate in Lemma 4.6 below some basic properties of the LPNs associated to Horn PCL theories.

Lemma 4.6. *Let $\mathcal{P}(\Delta) = (S, T, F, L, \ell, m_0)$, for some Horn* PCL *theory Δ. We have that:*

(a) $\mathcal{P}(\Delta)$ is correctly labeled,

(b) $\forall s \in S.\ (m_0(s) = 1 \implies {}^\bullet s = \emptyset \ \wedge\ \ell(s) = \bot)$,

(c) $\forall t \in T.\ \forall s \in t^\bullet.\ \ell(t) = \ell(s)$

(d) $\forall t, t' \in T.\ \ell(t) = \ell(t') \implies \exists_1 s \in {}^\bullet t \cap {}^\bullet t'.\ m_0(s) = 1$,

(e) $m_0 \to^* m \implies \forall s \in S.\ m(s) \in \{-1, 0, 1\}$,

(f) $m_0 \to^* \xrightarrow{t} \to^* \xrightarrow{t'} \implies \ell(t) \neq \ell(t')$

Proof. Items (a)–(d) follow from an easy inspection of Definition 4.3. Items (e) and (f) are immediate consequences of (d). □

In the LPN associated to a Horn PCL theory Δ, the initially marked places have an empty pre-set (item (b)), and all the places in the post-set of a transition are equally labeled (item (c)). Item (d) ensures that transitions with the same label are mutually exclusive, since they share a control place with exactly one token. Item (e) implies that each place may contain at most one token, either on credit (positive) or on debit (negative). Finally, item (f) states that, in each trace of the LPN, the same label cannot occur twice.

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 $\mathcal{P}(\Delta_1) \cdots \mathcal{P}(\Delta_n)$.

Proposition 4.7. *For all Δ_1, Δ_2, we have that $\mathcal{P}(\Delta_1, \Delta_2) \sim \mathcal{P}(\Delta_1) \oplus \mathcal{P}(\Delta_2)$.*

Proof. By Definition 4.3, the transitions of $\mathcal{P}(\Delta_1) \oplus \mathcal{P}(\Delta_2)$ coincide with those of $\mathcal{P}(\Delta_1, \Delta_2)$. The net $\mathcal{P}(\Delta_1, \Delta_2)$ may have more places than $\mathcal{P}(\Delta_1) \oplus \mathcal{P}(\Delta_2)$. This may happen for two reasons:

- each atom in $a \in \mathcal{L}(\Delta_i)$ which is not in $\mathcal{L}(\Delta_j)$ generates the places (a, t), with $t \in T_j$, $i \neq j$;
- some of the output places of $\mathcal{P}(\Delta_i)$ are removed in the composition. These places are precisely the places (a, t) with $t \in T_i$ such that $\ell_i((a, t)) \in \ell_j(T_j)$, with $i \neq j$.

However, these places do not play any role in the token game: indeed, by an easy inspection of Definition 4.3, these are output places in $\mathcal{P}(\Delta_1, \Delta_2)$, as there is no transition that can consume or lend tokens from these places. By Definition 4.3, the other places are connected to transitions (with flow or lending arcs) exactly in the same way in both nets. Hence the thesis. □

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 4.8 (Configuration). *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:*

- $\overline{m} = \{a \in \mathcal{L} \mid m((a, *)) = 0\}$
- $\Omega(m) = \{\ell(s) \mid m(s) < 0\}$.

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

Proof. Assume that $\overline{m} = \overline{m'}$ and $\Omega(m) = \Omega(m')$, and by contradiction suppose that there exists a place s in $\mathcal{P}(\Delta)$ such that $m(s) \neq m'(s)$. We have two cases, according to the form of s. If s has the form $(a, *)$, the thesis follows directly from the fact that $\overline{m} = \overline{m'}$. Otherwise, if the place s is of the form (a, t), with $\ell(t) = b$, then by item (e) of Lemma 4.6 it must be $m(s) \in \{-1, 0, 1\}$. So, we have the following three cases:

- $m(s) = 1$. This means that a transition labelled a has been executed, but the transition t has not, thus $b \notin \overline{m}$. Now, if $m'(s) = 0$ then the transition t has been executed, hence $b \in \overline{m'}$, but then $\overline{m} \neq \overline{m'}$, which contradicts our hypotheses. If $m'(s) = -1$ then the transition t is connected with s through a lending arc, and then $a \in \Omega(m')$. Since $a \notin \Omega(m)$, this would imply that $\Omega(m') \neq \Omega(m)$, which again leads to a contradiction.
- $m(s) = 0$. If $m'(s) = 1$ we are in the same case as above. If $m'(s) = -1$ then $a \in \Omega(m') = \Omega(m)$. Thus there is a place $s' = (a, t')$ such that $m(s') = -1$ and $\ell(t') = b$, otherwise $\Omega(m') \neq \Omega(m)$. But by construction this place is exactly s, contradicting the assumption that $m(s) = 0$.

- $m(s) = -1$. Already covered by the previous cases. □

In Theorem 4.10 below we state one of the main results of this section, namely that our construction maps the provability relation of PCL into the reachability of certain configurations in the associated LPN.

Theorem 4.10. *Let Δ be a Horn* PCL *theory, and let X be a conjunction of atoms. Then:*

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

Since the proof of the above statement is quite long, and it requires some additional notions and results, we devote to it the whole following subsection.

4.3. Proof of Theorem 4.10

We exploit the notion of *event structure with circular causality* (CES) introduced in [15] and further studied in [16, 9]. In a nutshell, a CES is a tuple $(E, \vdash, \Vdash, \#)$ where E is a set of *events*, $\# \subseteq E \times E$ is an irreflexive relation, called *conflict relation*, and $\vdash, \Vdash \subseteq \mathbf{2}^E_{fin} \times E$ are two relations, called respectively *causality* and *circular causality relation*. A set of events X is *conflict free* (in symbols, $CF(X)$) when $\forall x, y \in X.\ \neg(x \# y)$, and it is required that $X \circ e$ (for $\circ \in \{\vdash, \Vdash\}$) implies $CF(X)$. Further, the relations $\vdash$ and $\Vdash$ are *saturated*, i.e. if $X \circ a$, $X \subseteq Y$ and $CF(Y)$, then $Y \circ a$, with $\circ \in \{\vdash, \Vdash\}$.

Each conflict free sequence of events $\sigma = \langle e_0 \cdots e_i \cdots \rangle$ without duplicates uniquely identifies a computation in the CES $\mathcal{E}$, of the form:

$$(\emptyset, \emptyset) \xrightarrow{e_0} (\overline{\sigma_1}, \Gamma(\sigma_1)) \cdots \xrightarrow{e_i} (\overline{\sigma_{i+1}}, \Gamma(\sigma_{i+1})) \cdots$$

The first element of each pair is the set of events occurred so far, while the second element is the least set of events done "on credit", i.e. performed in the absence of a causal justification. Hereafter, we denote with $\overline{\sigma}$ the set of events in the sequence σ, and with σ_i the prefix of σ containing exactly i events. For all sequences $\eta = \langle e_0\ e_1 \cdots \rangle$, in [9] we define:

$$\Gamma(\eta) \ = \ \{e_i \in \overline{\eta} \ \mid \ \overline{\eta_i} \nvdash e_i \ \wedge\ \overline{\eta} \nVdash e_i\} \tag{7}$$

The set $\mathcal{R}_\mathcal{E}$ of *reachable events*, i.e. those which occur in some honored computation of $\mathcal{E}$, is then defined as:

$$\mathcal{R}_\mathcal{E} \ = \ \{e \in E \ \mid \ \exists \sigma.\ e \in \overline{\sigma} \text{ and } \Gamma(\sigma) = \emptyset\} \tag{8}$$

A computation η is *honored* iff $\Gamma(\eta) = \emptyset$, while we say that η is *honorable* iff it is the prefix of an honored one. Note that the empty sequence ε is honorable (and honored). In conflict free CES, $\sigma\, e$ is honorable whenever σ is honorable and either e is $\vdash$-enabled by the past events $\overline{\sigma}$, or it is $\Vdash$-enabled by $\mathcal{R}_\mathcal{E}$.

Example 4.11. *Let $\mathcal{E}$ be the conflict-free* CES *with (minimal) enablings $\{a\} \vdash b$ and $\{b\} \Vdash a$. Then, $\mathcal{E}$ has the following maximal computations:*

$$(\emptyset, \emptyset) \xrightarrow{a} (\{a\}, \{a\}) \xrightarrow{b} (\{a, b\}, \emptyset) \qquad\qquad (\emptyset, \emptyset) \xrightarrow{b} (\{b\}, \{b\}) \xrightarrow{a} (\{a, b\}, \{b\})$$

from which it follows that $\eta = \langle a\, b \rangle$ is honored, and so $\mathcal{R}_\mathcal{E} = \{a, b\}$.

A preliminary observation about computations in CES is that they allow for steps without a causal justification: for instance, the second computation of the CES of Example 4.11 is not honorable, because none of the available enablings can justify the firing of event b in the first step. However, for the sake of reachability we are interested in honorable computations only, hence hereafter we can restrict our attention to those computations where a step $(\overline{\sigma}, \Gamma(\sigma)) \xrightarrow{e}$ is either justified by an enabling $X \vdash e$ with $X \subseteq \overline{\sigma}$, or at least by some circular enabling $Y \Vdash e$.

A Horn PCL theory Δ can be associated to a conflict-free CES $\mathcal{E}(\Delta)$ as follows: the set of events E consists of the atoms of PCL, a clause $X \to a$ is associated to the saturation of $X \vdash a$, and $X \twoheadrightarrow a$ to

the saturation of $X \Vdash a$; the conflict relation $\#$ is empty. For instance, the theory $\Delta = a \to b,\ b \twoheadrightarrow a$ is associated to the CES $\mathcal{E}(\Delta)$ in Example 4.11.

Given a Horn PCL theory Δ, Theorem 6.4 in [16] states that $\Delta \vdash \mathcal{R}_{\mathcal{E}(\Delta)}$, i.e. that the atoms provable in Δ coincide with the events reachable in $\mathcal{E}(\Delta)$ [2]. For instance, from $\Delta = a \to b,\ b \twoheadrightarrow a$ we can deduce both a and b, (Example 4.1), which is coherent with the fact that $\mathcal{R}_{\mathcal{E}(\Delta)} = \{a, b\}$ (Example 4.11).

Our plan, to prove a correspondence between LPNs and PCL, is to exploit this result, by first establishing a correspondence between LPNs and CES. Note that the latter is not completely straightforward, because firing sequences in LPNs (non-deterministically) decide which $\twoheadrightarrow$-transition to fire in an *eager* fashion, while CES computations are *lazy* in the way they use circular enablings: indeed, to remove an event e from the set of credits $\Gamma(\sigma)$, any one of the enablings $X \Vdash e$ in $\mathcal{E}$ such that $X \subseteq \overline{\sigma}$ can be used. Hence, relate LPNs with CES we start by providing CES with a notion of *eager* computations, which preserves reachability, while making explicit which enabling is used at each computation step.

Definition 4.12. *An* eager computation *of a* CES $\mathcal{E}$ *is a sequence τ of minimal enablings of $\mathcal{E}$. We define the function ev as $ev(X \circ e) = e$ for $\circ \in \{\vdash, \Vdash\}$, and we extend it to sequences/sets of enablings as expected. Eager computations are subject to a* well-formedness *condition, which we define inductively as follows:*

- *the empty computation ε is well-formed;*
- *$\tau \cdot (X \vdash e)$ is well-formed iff τ is well-formed, $CF(ev(\overline{\tau}) \cup \{e\})$, and $X \subseteq ev(\overline{\tau}) \not\ni e$;*
- *$\tau \cdot (X \Vdash e)$ is well-formed iff τ is well-formed, $CF(ev(\overline{\tau}) \cup \{e\})$, and $e \notin ev(\overline{\tau})$.*

Hereafter, we shall only consider well-formed eager computations.

The notion of credits in (7) is lifted to eager computations as follows. Events justified by an enabling $X \vdash e$ are *not* credits, as we are considering well-formed computations where such enabling can only be used after X has been done. Events justified by $X \Vdash e$, instead, are justified in τ if and only if the set of all events fired in τ includes X. Then, a credit of τ is an event in τ with no justifications.

Definition 4.13. *We define the* credits *of an eager computation τ as follows:*

$$\Gamma(\tau) \;=\; \{e \;\mid\; (X \Vdash e) \in \overline{\tau} \;\wedge\; X \not\subseteq ev(\overline{\tau})\}$$

The notion of reachability for eager computations is similar to that for lazy ones:

$$\mathcal{R}_{\mathcal{E}}^{eager} \;=\; \bigcup \{ev(\overline{\tau}) \;\mid\; \tau \text{ well-formed and } \Gamma(\tau) = \emptyset\}$$

Example 4.14. *The* CES $\mathcal{E}$ *in Example 4.11 has exactly one maximal eager computation:*

$$\tau = (\{b\} \Vdash a)\ (\{a\} \vdash b)$$

and from Definition 4.13 we have that $\Gamma(\tau_0) = \emptyset$, $\Gamma(\tau_1) = \{a\}$, and $\Gamma(\tau_2) = \Gamma(\tau) = \emptyset$.

We now introduce a notion of *coherence* between a lazy and an eager computation, which relates σ and τ whenever they fire the same sequence of events, and have exactly the same credits on all prefixes.

Definition 4.15. *A lazy computation $\sigma = \langle e_0 \cdots e_n \rangle$ is* coherent *with an eager computation τ whenever:*

$$\forall i \in 1..n+1.\quad \overline{\sigma_i} = \overline{ev(\tau_i)} \;\wedge\; \Gamma(\sigma_i) = \Gamma(\tau_i)$$

Example 4.16. *Consider the conflict free* CES *with enablings $\{a\} \vdash b$, $\{b\} \Vdash a$, and $\{b, c\} \Vdash a$. We have that $\tau = (\{b\} \Vdash a)\ (\{a\} \vdash b)$ is coherent with $\sigma = \langle a\, b \rangle$, whereas $\tau' = (\{b, c\} \Vdash a)\ (\{a\} \vdash b)$ is not, because $\Gamma(\tau') = \{b\}$ while $\Gamma(\sigma) = \emptyset$.*

[2] The mapping in [16] is a bijection from finite conflict-free CES to Horn PCL theories; here we use the inverse mapping.

From an eager computation τ it is straightforward to construct a lazy computation σ coherent with τ: it suffices to define $\sigma = ev(\tau)$, i.e. the sequence of events fired by τ. The inverse construction is a bit more involved. In Definition 4.17 we show how to associate an eager computation to a lazy one; Lemma 4.19 will show that this construction produces coherent eager computations.

Definition 4.17. *Let $\sigma = \langle e_0 \cdots e_n \rangle$ be a lazy computation of a* CES*. We construct an eager computation $\tau = \langle (X_0 \circ_0 e_0) \cdots (X_n \circ_n e_n) \rangle$ as follows:*

(a) if $\overline{\sigma_i} \vdash e_i$, then $X_i = X$ and $\circ_i = \vdash$, for some minimal enabling $X \vdash e_i$ of $\mathcal{E}$ such that $X \subseteq \overline{\sigma_i}$.

(b) otherwise, if $e_i \notin \Gamma(\sigma)$, it means that the credit e_i has been honored in σ, i.e. there exists some $j > i$ such that $e_i \in \Gamma(\sigma_j)$ and $e_i \notin \Gamma(\sigma_{j+1})$. We define $X_i = X$ and $\circ_i = \Vdash$, for some minimal enabling $X \Vdash e_i$ such that $X \subseteq \overline{\sigma_{j+1}}$.

(c) otherwise, it must be $e_i \in \Gamma(\sigma)$, i.e. the credit e_i has not been honored in σ. Then, we define $X_i = X$ and $\circ_i = \Vdash$ for some minimal enabling $X \Vdash e$.

Example 4.18. *Consider the* CES *with enablings $\{b, c\} \Vdash a$, $\{b\} \Vdash a$, $\{a\} \vdash b$, $\{a, b\} \vdash c$, and let $\sigma = \langle a\, b\, c \rangle$. The construction in Definition 4.17 associates to σ two lazy computations:*

$$\tau = (\{b\} \Vdash a)\ (\{a\} \vdash b)\ (\{a, b\} \vdash c) \qquad\qquad \tau' = (\{b, c\} \Vdash a)\ (\{a\} \vdash b)\ (\{a, b\} \vdash c)$$

Observe that both τ and τ' are coherent with σ.

Lemma 4.19 below shows that any τ obtained from σ according to the construction in Definition 4.17 is coherent with σ. Together with the inverse construction, we can then conclude that an event belongs to some honored lazy computation if and only if it belongs to some honored eager computation, hence $\mathcal{R}_{\mathcal{E}} = \mathcal{R}_{\mathcal{E}}^{eager}$.

Lemma 4.19. *Let τ be constructed from σ as in Definition 4.17. Then, τ is coherent with σ.*

Proof. The requirement $\forall i \in 1..n+1.\ \overline{\sigma_i} = \overline{ev(\tau_i)}$ is trivially satisfied, because labels are added by the construction of τ in exactly the same order. To prove the requirement $\forall i \in 1..n+1.\ \Gamma(\sigma_i) = \Gamma(\tau_i)$, we proceed by induction on the length of σ. The base case is trivial, because $\Gamma(\sigma_0) = \emptyset = \Gamma(\tau_0)$. For the inductive case, assume that $\sigma = \langle e_0 \cdots e_n \rangle$. By the induction hypothesis, we have that $\Gamma(\tau_i) = \Gamma(\sigma_i)$, for all $i \in 0..n$. Let $\tau = \tau_n \cdot (X_n \circ_n e_n)$. To prove that $\Gamma(\tau) = \Gamma(\sigma)$ we distinguish between two cases, according to the kind of enabling $\circ_n$ used in the last step of τ.

- $\circ_n = \vdash$. Then by (7) we have $e_n \notin \Gamma(\tau)$. Hence:

$$\begin{aligned}
\Gamma(\tau) &= \{e \mid (Y \Vdash e) \in \overline{\tau} \text{ and } Y \not\subseteq ev(\overline{\tau})\} && \text{by Definition 4.13}\\
&= \Gamma(\tau_n) \setminus \{e \mid (Y \Vdash e) \in \overline{\tau_n} \text{ and } Y \subseteq ev(\overline{\tau_n}) \cup \{e_n\}\} && \\
&= \Gamma(\sigma_n) \setminus \{e \mid (Y \Vdash e) \in \overline{\tau_n} \text{ and } Y \subseteq ev(\overline{\tau_n}) \cup \{e_n\}\} && \text{by the ind. hyp.}\\
&= \Gamma(\sigma_n) \setminus \{e \in \overline{\sigma_n} \mid \overline{\sigma_n} \cup \{e_n\} \Vdash e\} && (\star)\\
&= \Gamma(\sigma) && \text{as } X_n \vdash e_n
\end{aligned}$$

  The equality $(\star)$ is justified as follows. Let:

$$\begin{aligned}
A &= \{e \mid (Y \Vdash e) \in \overline{\tau_n} \text{ and } Y \subseteq ev(\overline{\tau_n}) \cup \{e_n\}\}\\
B &= \{e \mid \overline{\sigma_n} \cup \{e_n\} \Vdash e\}
\end{aligned}$$

  The inclusion $A \subseteq B$ holds trivially, since if there exists $(Y \Vdash e) \in \overline{\tau_n}$ with $Y \subseteq ev(\overline{\tau_n}) \cup \{e_n\}$, then $Y \subseteq \overline{\sigma_n} \cup \{e_n\}$, and so by saturation $\overline{\sigma_i} \cup \{e_i\} \Vdash e_j$. Thus, $\Gamma(\sigma_n) \setminus A \supseteq \Gamma(\sigma_n) \setminus B$. The inclusion $B \subseteq A$ does not hold, but it is enough to show that $\Gamma(\sigma_n) \setminus A \subseteq \Gamma(\sigma_n) \setminus B$. To do that, we pick some e such that $e \notin \Gamma(\sigma_n)$ or $e \in B$, and we show that $e \notin \Gamma(\sigma_n)$ or $e \in A$. We have two cases. If $e \notin \Gamma(\sigma_n)$, then the thesis follows trivially. Otherwise, let $e \in B$. There are two subcases. If τ_n contains $Y \Vdash e$ for some Y, then e also belongs to A, from which the thesis follows. Otherwise, assume that τ_n does not contain any such $Y \Vdash e$. Since $e \in ev(\overline{\tau_n})$, then τ_n must contain $Y \vdash e$, for some Y. Since τ_n is well-formed, it must be $Y \subseteq ev(\overline{\tau_n})$. But then, since $ev(\overline{\tau_n}) = \overline{\sigma_n}$, the thesis $e \notin \Gamma(\sigma_n)$ follows.

- $\circ_n = \Vdash\vdash$. We distinguish between two further subcases. If $X_n \subseteq ev(\overline{\tau})$, then $e_n \notin \Gamma(\tau)$, and the proof proceeds similarly to the case $\circ_n = \vdash$ above. Otherwise, we have $e_n \in \Gamma(\tau)$, and:

$$\begin{aligned}
\Gamma(\tau) &= \{e \mid (Y \Vdash\vdash e) \in \overline{\tau} \text{ and } Y \not\subseteq ev(\overline{\tau})\} && \text{by Definition 4.13}\\
&= \big(\Gamma(\tau_n) \setminus \{e \mid (Y \Vdash\vdash e) \in \overline{\tau_n} \text{ and } Y \subseteq ev(\overline{\tau})\}\big) \cup \{e_n\} && \text{as } e_n \in \Gamma(\tau)\\
&= \big(\Gamma(\sigma_n) \setminus \{e \mid (Y \Vdash\vdash e) \in \tau_n \text{ and } Y \subseteq ev(\overline{\tau_n}) \cup \{e_n\}\}\big) \cup \{e_n\} && \text{by the ind. hyp.}\\
&= \big(\Gamma(\sigma_n) \setminus \{e \mid \overline{\sigma_n} \cup \{e_n\} \Vdash\vdash e\}\big) \cup \{e_n\} && (\star)\\
&= \big(\Gamma(\sigma_n) \setminus \{e \mid \overline{\sigma_n} \cup \{e_n\} \Vdash\vdash e\}\big) \cup \{e_n\} && \text{as } \sigma = \sigma_n\, e_n\\
&= \Gamma(\sigma)
\end{aligned}$$

  where the equality $(\star)$ is justified as before, and the last equality is justified as follows. Let $A = \big(\Gamma(\sigma_n) \setminus \{e \mid \overline{\sigma} \Vdash\vdash e\}\big) \cup \{e_n\}$.

  To show the inclusion $\Gamma(\sigma) \subseteq A$, let $a \in \Gamma(\sigma)$. Then, $a = e_i$, for some $i \in 0..n$ such that $\overline{\sigma_i} \nVdash e_i$ and $\overline{\sigma} \Vdash\!\!\!\!\!\!/\; e_i$. We distinguish between two cases:

  - $i < n$. Then, $e \in \Gamma(\sigma_n)$, and since $\overline{\sigma} \Vdash\!\!\!\!\!\!/\; e_i$ then $e_i \notin \{e \mid \overline{\sigma} \Vdash\vdash e\}$. Therefore, $e_i \in A$.
  - $i = n$. Trivial, because $e_n \in A$ by definition of A.

  To show the inclusion $A \subseteq \Gamma(\sigma)$, let $a \in A$. We distinguish between two cases:

  - $a \in \Gamma(\sigma_n)$ and $\overline{\sigma} \Vdash\!\!\!\!\!\!/\; a$. Since $a \in \Gamma(\sigma_n)$, then $\overline{\sigma_n} \nVdash a$. Therefore, $a \in \Gamma(\sigma)$.
  - $a = e_n$. We have three further subcases, according to which one of the cases of the construction in Definition 4.17 has been used to append $X_n \circ_n e_n$ to τ_n.
    (a) this case defines $\circ_n = \vdash$. Since we are assuming $\circ_n = \Vdash\vdash$, this case does not apply.
    (b) this case defines $X_n = X$ for a minimal enabling $X \Vdash\vdash e_n$ such that $X \subseteq \overline{\sigma_j}$, for some $j \leq n$. Since we are under the hypothesis that $X_n \not\subseteq ev(\overline{\tau}) = \overline{\sigma}$, also this case does not apply.
    (c) the last case requires that $e_n \in \Gamma(\sigma)$, which is just our thesis. □

We now relate eager computations in CES with firing sequences in LPNs. To do that, we will record the events which have been used in the premise of an enabling $X \Vdash\vdash e$ of an eager computation τ, but which have not been justified in τ. We call these events the *debits* of τ, and below we shall see that these correspond to the places with a negative marking in the firing sequence associated with τ.

Definition 4.20. *We define the* debits *of an eager computation τ inductively as follows:*

$$\Omega(\varepsilon) = \emptyset \qquad\qquad \Omega(\tau \cdot (X \circ a)) \;=\; (\Omega(\tau) \cup X) \setminus (ev(\overline{\tau}) \cup \{a\})$$

It is easy to check that if τ is coherent with σ and σ is honored, then $\Omega(\tau) = \Gamma(\sigma) = \emptyset$. This follows by the fact that $\Omega(\tau)$ can be given the following equivalent non-inductive specification:

$$\Omega(\tau) \;=\; \bigcup \{X \mid (X \Vdash\vdash e) \in \overline{\tau}\} \setminus ev(\overline{\tau}) \tag{9}$$

Lemma 4.21. *For all eager computations τ, we have that:*

(a) $\Gamma(\tau) \cap \Omega(\tau) = \emptyset$, *and*

(b) $\Gamma(\tau) = \emptyset \iff \Omega(\tau) = \emptyset$.

Proof. Item (a) follows from the fact that $\Gamma(\tau) \subseteq ev(\overline{\tau})$, and that $\Omega(\tau) \cap ev(\overline{\tau}) = \emptyset$. For item (b) we prove the two contrapositives. First, assume that $e \in \Gamma(\tau)$. Then, there exists $X \Vdash\vdash e$ in τ such that $X \not\subseteq ev(\overline{\tau})$, and so by Definition 4.20 it follows that $e \in \Omega(\tau)$. The other direction is symmetric. □

We now show that each eager computation τ of $\mathcal{E}(\Delta)$ can be associated to a firing sequence of $\mathcal{P}(\Delta)$ which preserves the debits, pointwise on the prefixes of τ.

Lemma 4.22. *Let $\tau = \langle (X_1 \circ_1 e_1) \cdots (X_n \circ_n e_n) \rangle$ be an eager computation of $\mathcal{E}(\Delta)$. Then, there exists a firing sequence $m_0 \xrightarrow{t_1} \cdots \xrightarrow{t_n} m_n$ of $\mathcal{P}(\Delta)$ such that:*

$$\forall i \in 1..n : \; t_i = (X_i, e_i, \circ_i) \tag{10}$$

$$\forall i \in 0..n : \big(ev(\overline{\tau_i}) = \overline{m_i} \;\wedge\; \Omega(\tau_i) = \Omega(m_i) \big) \tag{11}$$

Proof. By induction on the length of τ. The base case $\tau = \varepsilon$ is straightforward, because $\overline{m_0} = \bar{\varepsilon} = \emptyset$, and $\Omega(m_0) = \Omega(\varepsilon) = \emptyset$. For the inductive case, let $\tau = \tau'(X_n \circ_n e_n)$. By the induction hypothesis it follows that there exists a firing sequence $m_0 \xrightarrow{t_1} \cdots \xrightarrow{t_{n-1}} m_{n-1}$ in $\mathcal{P}(\Delta)$ satisfying (10). We proceed by cases on the form of the rightmost enabling in τ:

- $X_n \vdash e_n$. Since τ is well-formed, then $X_n \subseteq ev(\overline{\tau'})$. By the construction of $\mathcal{E}(\Delta)$, it must be $X_n \to e_n \in \Delta$. By Definition 4.3, the LPN $\mathcal{P}(\Delta)$ has a transition $t_n = (X_n, e_n, \to)$ with places $s_a = (a, t_n)$ for all $a \in X_n$. Since $e_n \notin \overline{m_{n-1}} = ev(\overline{\tau'}) \supseteq X_n$, then $m_{n-1}(s_a) = 1$ for all $a \in X_n$. Further, since $e_n \notin \overline{m_{n-1}}$, then $m((e_n, *)) = 1$. Thus, the transition t_n is enabled, and firing it leads to a marking m_n such that $\overline{m_n} = \overline{m_{n-1}} \cup \{e_n\}$. Then, $\overline{m_n} = ev(\overline{\tau'}) \cup \{e_n\} = ev(\overline{\tau})$. To prove that $\Omega(m_n) = \Omega(\tau)$, note that:

$$\begin{aligned}
\Omega(\tau) &= (\Omega(\tau') \cup X_n) \setminus (ev(\overline{\tau'}) \cup \{e_n\}) && \text{by Definition 4.20} \\
&= (\Omega(m_{n-1}) \cup X_n) \setminus (\overline{m_{n-1}} \cup \{e_n\}) && \text{by the ind. hyp. (11)} \\
&= (\Omega(m_{n-1}) \cup X_n) \setminus \overline{m_n} && \text{since } \overline{m_n} = \overline{m_{n-1}} \cup \{e_n\} \\
&= \Omega(m_{n-1}) \setminus \overline{m_n} && \text{since } X_n \subseteq \overline{m_{n-1}} \\
&= \Omega(m_{n-1}) \setminus \{e_n\} && \text{since } \Omega(m_{n-1}) \cap \overline{m_{n-1}} = \emptyset \\
&= \Omega(m_n)
\end{aligned}$$

  where the last equation is justified because $e_n \in \overline{m_n}$ implies that $m_n(s) \geq 0$ for all s with $\ell(s) = e_n$.

- $X_n \Vdash e_n$. By the construction of $\mathcal{E}(\Delta)$, it must be $X_n \twoheadrightarrow e_n \in \Delta$. By Definition 4.3, the LPN $\mathcal{P}(\Delta)$ has a transition $t_n = (X_n, e_n, \twoheadrightarrow)$ with places $s_a = (a, t_n)$ for all $a \in X_n$, with $F(s_a, t_n) = 0$ and $L(s_a, t_n) = 1$. Since $e_n \notin \overline{m_{n-1}}$, then $m((e_n, *)) = 1$. Thus, the transition t_n is enabled, and firing it leads to a marking m_n such that $\overline{m_n} = \overline{m_{n-1}} \cup \{e_n\}$. Then, $\overline{m_n} = ev(\overline{\tau'}) \cup \{e_n\} = ev(\overline{\tau})$. To prove that $\Omega(m_n) = \Omega(\tau)$, note that:

$$\begin{aligned}
\Omega(\tau) &= (\Omega(\tau') \cup X_n) \setminus (ev(\overline{\tau'}) \cup \{e_n\}) && \text{by Definition 4.20} \\
&= (\Omega(m_{n-1}) \cup X_n) \setminus (\overline{m_{n-1}} \cup \{e_n\}) && \text{by the ind. hyp. (11)} \\
&= (\Omega(m_{n-1}) \cup X_n) \setminus \overline{m_n} && \text{since } \overline{m_n} = \overline{m_{n-1}} \cup \{e_n\} \\
&= \Omega(m_n)
\end{aligned}$$

  where the last equality is justified as follows. The last transition in the firing sequence, i.e. $t_n = (X, e_n, \twoheadrightarrow)$, produces a token in all places labeled e_n (which then become non-negative, so $e_n \notin \Omega(m_n)$), and removes a token from each place $s_a = (a, t_n)$, with $a \in X_n$. These places become negative if and only if the transitions in their pre-sets have not been fired, i.e. $m_n(s_a) < 0$ iff $a \notin \overline{m_n}$. Therefore, for each $a \in X_n$ we have that $a \in \Omega(m_n)$ iff $a \notin \overline{m_n}$, from which the thesis follows. □

We now show the inverse of Lemma 4.22, i.e. that each firing sequence of $\mathcal{P}(\Delta)$ can be associated to an eager computation of $\mathcal{E}(\Delta)$ which preserved the debits. Notice that here we do not make any assumptions about the honorability of markings.

Lemma 4.23. *Let $m_0 \xrightarrow{t_1} \cdots \xrightarrow{t_n} m_n$ be a firing sequence of $\mathcal{P}(\Delta)$. Then, there exists an eager computation $\tau = \langle (X_1 \circ_1 e_1) \cdots (X_n \circ_n e_n) \rangle$ of $\mathcal{E}(\Delta)$ such that:*

$$\forall i \in 1..n : \ell(t_i) = e_i \tag{12}$$

$$\forall i \in 0..n : \big(ev(\overline{\tau_i}) = \overline{m_i} \ \wedge \ \Omega(\tau_i) = \Omega(m_i)\big) \tag{13}$$

Proof. By induction on the length of the firing sequence. The base case is trivial, because with $\tau = \varepsilon$ we have $\overline{m_0} = \emptyset = ev(\overline{\varepsilon})$ and $\Omega(m_0) = \emptyset = \Omega(\varepsilon)$.

For the inductive case, assume that $m_{i-1} \xrightarrow{t_i} m_i$. By the induction hypothesis, there exists a well-formed eager computation:

$$\tau_{i-1} = \langle (X_1 \circ_1 e_1) \cdots (X_{i-1} \circ_{i-1} e_{i-1}) \rangle$$

of $\mathcal{E}(\Delta)$ such that (12) and (13) hold. Let $t_i = (X_i, e_i, \circ_i)$. We show that $\tau_{i-1} \cdot (X_i \circ_i e_i))$ is a well-formed eager computation of $\mathcal{E}(\Delta)$. We distinguish between two cases, according to $\circ_i$:

- $\circ_i = \rightarrow$. Since $m_{i-1} \xrightarrow{t_i} m_i$, the transition t_i consumes the token in $(e_i, *)$ and tokens in the places $s_a = (a, t_i)$ with $a \in X_i$. The latter have been put by the firing of transitions t_j with $j < i$ and $\ell(t_j) = a$. Since $X_i \rightarrow e_i \in \Delta$ and $X_i \subseteq \overline{\tau_{i-1}}$, we have that $\tau_{i-1} \cdot (X_i \vdash e_i)$, is a well-formed computation of $\mathcal{E}(\Delta)$, and $ev(\overline{\tau_i}) = ev(\overline{\tau_{i-1}}) \cup \{e_i\}$. Since $ev(\overline{\tau_{i-1}}) = \overline{m_{i-1}}$, it follows that $ev(\overline{\tau_i}) = \overline{m_{i-1}} \cup \{e_i\} = \overline{m_i}$, as the token at place $(e_i, *)$ has been consumed.

  We now show that $\Omega(\tau_i) = \Omega(m_i)$.

$$\begin{array}{lll}
\Omega(m_i) & = \{a \mid \exists s \in S.\ a = \ell(s) \text{ and } m_i(s) < 0\} & \text{by Definition 4.8} \\
& = \Omega(m_{i-1}) \setminus \{\ell(t_i)\} & \text{as } \forall s_a \in {}^\bullet t_i.\, m_i(s_a) > 0 \\
& = \big(\Omega(m_{i-1}) \ \cup \ X_i\big) \setminus \big(\overline{m_{i-1}} \cup \ \{e_i\}\big) & \text{as } \Omega(m_{i-1}) \cap \overline{m_{i-1}} = \emptyset \\
& & \text{and } X_i \subseteq \overline{m_{i-1}} \\
& = \big(\Omega(\tau_{i-1}) \ \cup \ X_i\big) \setminus \big(ev(\overline{\tau_{i-1}}) \cup \ \{e_i\}\big) & \text{by the ind. hyp. (13)} \\
& = \Omega(\tau_i) & \text{by Definition 4.20}
\end{array}$$

- $\circ_i = \twoheadrightarrow$. Since $m_{i-1} \xrightarrow{t_i} m_i$, the transition t_i consumes the token in $(e_i, *)$ and tokens in the places $s_a = (a, t_i)$ with $a \in X_i$, lending tokens from those places s_a with $m_{i-1}(s_a) = 0$. All the arcs connecting the places s_a with the transition t_i are lending, hence the transition is enabled at m_{i-1}, and firing t_i produces the marking m_i. Let τ_i as $\tau_{i-1} \cdot (X_i \Vdash e_i)$. The proof that $\overline{m_i} = ev(\overline{\tau_i})$ is similar to the previous case. We show that $\Omega(\tau_i) = \Omega(m_i)$ as follows.

$$\begin{array}{lll}
\Omega(m_i) & = \{a \mid \exists s \in S.\ a = \ell(s) \text{ and } m_i(s) < 0\} & \text{by Definition 4.8} \\
& = \big(\{a \mid \exists s \in S.\ a = \ell(s) \text{ and } m_{i-1}(s) < 0\} \setminus \{e_i\}\big) \cup Y & \\
& \quad \text{where } Y = \{a \mid s_a \in {}^\bullet t_i \text{ and } m_i(s_a) < 0\} & \text{as } m_{i-1} \xrightarrow{t_i} m_i \\
& = \big(\Omega(m_{i-1}) \setminus \{e_i\}\big) \ \cup \ Y & \text{by Definition 4.8} \\
& = \big(\Omega(m_{i-1}) \cup Y\big) \setminus \{e_i\} & \text{as } e_i \notin Y \\
& = \big(\Omega(m_{i-1}) \cup Y\big) \setminus \big(\overline{m_{i-1}} \cup \{e_i\}\big) & \text{as } \overline{m_{i-1}} \cap (\Omega(m_{i-1}) \cup Y) = \emptyset \\
& = \big(\Omega(m_{i-1}) \cup X_i\big) \setminus \big(\overline{m_{i-1}} \cup \ \{e_i\}\big) & \text{as } Y \subseteq X_i \subseteq \Omega(m_{i-1}) \cup Y \\
& = \big(\Omega(\tau_{i-1}) \ \cup \ X_i\big) \setminus \big(ev(\overline{\tau_{i-1}}) \cup \ \{e_i\}\big) & \text{by ind. hyp (13)} \\
& = \Omega(\tau_i) & \text{by Definition 4.20}
\end{array}$$

□

Summing up, we are now able to prove the statement of Theorem 4.10.

$$\frac{}{\varepsilon \in [\![\Delta]\!]}\,(\varepsilon) \qquad \frac{X \to a \in \Delta \qquad \sigma \in [\![\Delta]\!] \qquad X \subseteq \overline{\sigma}}{\sigma\, a \in [\![\Delta]\!]}\,(\to) \qquad \frac{X \twoheadrightarrow a \in \Delta \qquad \sigma \in [\![\Delta, a]\!] \qquad X \subseteq \overline{\sigma}}{\sigma \mid a \subseteq [\![\Delta]\!]}\,(\twoheadrightarrow)$$

Figure 10: Proof traces of Horn PCL.

Proof of Theorem 4.10. For the $\Rightarrow$ direction, assume that $\Delta \vdash X$. We have that:

$$\begin{aligned}
\Delta \vdash X &\iff X \subseteq \mathcal{R}_{\mathcal{E}(\Delta)} && \text{by Theorem 6.4 in [16]}\\
&\iff \exists \sigma \text{ honored lazy computation of } \mathcal{E}(\Delta).\ X \subseteq \overline{\sigma} && \text{by (8)}\\
&\iff \exists \tau \text{ eager computation of } \mathcal{E}(\Delta).\ \Omega(\tau) = \emptyset \text{ and } X \subseteq ev(\overline{\tau})\\
&\implies \exists m \in Mk(\mathcal{P}(\Delta)).\ X \subseteq \overline{m} \text{ and } \Omega(m) = \emptyset && \text{by Lemma 4.22}
\end{aligned}$$

For the $\Leftarrow$ direction, assume that there exists $m \in Mk(\mathcal{P}(\Delta))$ such that $X \subseteq \overline{m}$ and $\Omega(m) = \emptyset$. We have:

$$\begin{aligned}
&\exists m \in Mk(\mathcal{P}(\Delta)).\ X \subseteq \overline{m} \text{ and } \Omega(m) = \emptyset\\
&\implies \exists \tau \text{ eager computation of } \mathcal{E}(\Delta).\ \Omega(\tau) = \emptyset \text{ and } X \subseteq ev(\overline{\tau}) && \text{by Lemma 4.23}\\
&\iff \exists \sigma \text{ honored lazy computation of } \mathcal{E}(\Delta).\ X \subseteq \overline{\sigma}\\
&\iff X \subseteq \mathcal{R}_{\mathcal{E}(\Delta)} && \text{by (8)}\\
&\iff \Delta \vdash X && \text{by Theorem 6.4 in [16]} \qquad \square
\end{aligned}$$

4.4. Proof traces

Each Horn PCL theory Δ induces a set of *proof traces* [9], namely those sequences of atoms which are somehow "compatible" with the sequents of the form $\Delta \vdash Y$. To convey some intuition, consider a theory Δ containing the clause $X \to a$. Then, the elimination rule for $\to$ allows for the following proof:

$$\frac{\Delta \vdash X \to a \qquad \Delta \vdash X}{\Delta \vdash a}\,(\to\text{E})$$

The rule says that, to construct from Δ a proof of a, one first needs a proof of all the atoms in X. If we denote with $[\![\Delta]\!]$ the collection of all proof traces of Δ, and if $\sigma \in [\![\Delta]\!]$ contains all the atoms in X, then to be coherent with rule $(\to\text{E})$ we must also include σa in $[\![\Delta]\!]$.

Consider now the elimination rule for $\twoheadrightarrow$:

$$\frac{\Delta \vdash X \twoheadrightarrow a \qquad \Delta, a \vdash X}{\Delta \vdash a}\,(\twoheadrightarrow\text{E})$$

Here, the intuition is that X needs not necessarily be proved before a: it suffices to prove X by taking a as hypothesis. Assuming that σ is a proof trace of Δ, a (i.e. Δ plus the hypothesis a), the proof traces of Δ must then include all the interleavings between σ and a.

In this section we establish a correspondence between proof traces and honored firing sequences in LPNs. More precisely, Theorem 4.28 below states that each proof trace $\langle e_0 \cdots e_n \rangle$ in $[\![\Delta]\!]$ can be associated to a honored firing sequence in $\mathcal{P}(\Delta)$ with transitions labeled $e_0 \cdots e_n$, and *vice versa*.

We now briefly recap from [9] the notion of proof traces.

Definition 4.24 (**Proof traces** [9])**.** *For a Horn* PCL *theory* Δ*, we define the set of* proof traces $[\![\Delta]\!]$ *by the rules in Figure 10, where for* $\sigma, \eta \in E^*$ *we denote with* $\sigma\eta$ *the concatenation of* σ *and* η*, and with* $\sigma \mid \eta$ *the interleavings of* σ *and* η*. We assume that both concatenation and interleaving remove duplicates from the right, e.g.* $aba \mid ca = ab \mid ca = \{abc, acb, cab\}$*.*

Note that the $(\twoheadrightarrow)$ rule carries a set inclusion in its consequence $\sigma \mid a \subseteq [\![\Delta]\!]$. This is just a convenient shorthand for adding a side condition $\eta \in (\sigma \mid a)$ and changing the conclusion to $\eta \in [\![\Delta]\!]$.

Example 4.25. *Let $\Delta = a \to b,\ b \twoheadrightarrow a$, as in Example 4.4. We can deduce that $ab \in [\![\Delta]\!]$ through the following derivation:*

$$\dfrac{b \twoheadrightarrow a \in \Delta \qquad \dfrac{a \to b \in \Delta, a \qquad \dfrac{\top \to a \in \Delta, a \qquad \overline{\varepsilon \in [\![\Delta, a]\!]}^{(\varepsilon)} \qquad \overline{\top} \subseteq \overline{\varepsilon}}{a \in [\![\Delta, a]\!]}^{(\to)} \qquad \overline{a} \subseteq \overline{a}}{ab \in [\![\Delta, a]\!]}^{(\to)} \qquad \overline{b} \subseteq \overline{ab}}{ab \in ab \mid a \subseteq [\![\Delta]\!]}^{(\twoheadrightarrow)}$$

Notice that $ba \notin [\![\Delta]\!]$: indeed, to derive any non-empty α from Δ one needs to use both $a \to b$ and $b \twoheadrightarrow a$, hence all non-empty proof traces must contain both a and b; since b does not occur at the right of a contractual implication, it cannot be interleaved; thus, ba is not derivable. Therefore, $[\![\Delta]\!] = \{\varepsilon, ab\}$.

Not incidentally, the LPN $\mathcal{P}(\Delta)$ in Example 4.4 has exactly two honored firing sequences, that is ε and $t_a\, t_b$ (whose trace is ab). Theorem 4.28 below will formalize this correspondence.

To prove the main result of this section, Theorem 4.28, we proceed similarly to the proof of Theorem 4.10, by using CES as a bridge between PCL and LPNs. To do that, we provide below a couple of results that will be needed later on in the proof of Theorem 4.28. The first of these results, Lemma 4.26, states that if we consider a proof trace σ as a lazy computation of the CES $\mathcal{E}(\Delta)$, then the credits of σ are empty.

Lemma 4.26. *If $\sigma \in [\![\Delta]\!]$, then $\Gamma(\sigma) = \emptyset$ in $\mathcal{E}(\Delta)$.*

Proof. We prove the following stronger statement: if $\sigma \in [\![\Delta, X]\!]$, then $\Gamma(\sigma) \subseteq X$. To do this, we proceed by induction on the proof of $\sigma \in [\![\Delta, X]\!]$. The base case is when $\sigma = \varepsilon$, for which the statement trivially holds, since $\Gamma(\varepsilon) = \emptyset$. For the inductive case, there are the following two subcases, according to the last rule used in the derivation of $\sigma \in [\![\Delta, X]\!]$:

- $(\to)$. We have that $\sigma = \eta a$, where $\eta \in [\![\Delta, X]\!]$. By the induction hypothesis, $\Gamma(\eta) \subseteq X$. Since $X \to a \in \Delta$, then $X \vdash a \in \mathcal{E}(\Delta)$, and by the premise of rule $(\to)$ we have that $X \subseteq \overline{\eta}$. Therefore, $\Gamma(\sigma) \subseteq \Gamma(\eta) \subseteq X$.
- $(\twoheadrightarrow)$. We have that $\sigma \in \eta \mid a$, where $\eta \in [\![\Delta, X, a]\!]$. By the induction hypothesis, $\Gamma(\eta) \subseteq X \cup \{a\}$. Since $X \twoheadrightarrow a \in \Delta$, then $X \Vdash a \in \mathcal{E}(\Delta)$, and by the premise of rule $(\twoheadrightarrow)$ we have that $X \subseteq \overline{\eta}$. Therefore, $\Gamma(\sigma) \subseteq \Gamma(\eta) \cup \{a\} \subseteq X \cup \{a\}$. Now, $X \subseteq \overline{\eta}$ implies that $a \notin \Gamma(\sigma)$, and so we conclude $\Gamma(\sigma) \subseteq X$. □

Lemma 4.27 below allows for obtaining a proof trace of Δ from an honored lazy computation σ of $\mathcal{E}(\Delta)$.

Lemma 4.27. *For each honored eager computation σ of $\mathcal{E}(\Delta)$, we have that $\sigma \in [\![\Delta]\!]$.*

Proof. We prove the following stronger statement, which also works when σ is not honored. In such general case, we will show that σ is a proof trace of the theory Δ augmented with the credits of σ, i.e.:

$$\sigma \in [\![\Delta, \Gamma(\sigma)]\!]$$

We proceed by induction on the length of σ. The base case $\sigma = \varepsilon$ is trivial, because $\Gamma(\varepsilon) = \emptyset$, and $\sigma = \varepsilon \in [\![\Delta]\!]$ holds by rule (ε). For the inductive case, let $\sigma = \eta a$. Then by the induction hypothesis it follows that $\eta \in [\![\Delta, \Gamma(\eta)]\!]$. We have the following two cases, according to the kind of enabling $\circ$ used to add the event a to η:

- $\circ = \vdash$. Then, there exists a minimal enabling $X \vdash a$ in $\mathcal{E}(\Delta)$ such that $X \subseteq \overline{\eta}$. By rule $(\to)$ we have:

$$\dfrac{X \to a \in \Delta \qquad \eta \in [\![\Delta, \Gamma(\eta)]\!] \qquad X \subseteq \overline{\eta}}{\sigma = \eta a \ \in \ [\![\Delta, \Gamma(\eta)]\!]}$$

  By (7), we have that $\Gamma(\sigma) = \Gamma(\eta) \setminus A$, where

$$A \ = \ \{e \in \overline{\eta} \ \mid \ X \cup \{a\} \Vdash e \text{ and } X \subseteq \overline{\sigma}\} \tag{14}$$

If A is empty, then we already have the thesis. Otherwise, assume that $e \in A$, i.e. $X \cup \{a\} \Vdash e$ for some $X \subseteq \overline{\sigma}$ and $e \in \overline{\eta}$. Since $X \cup \{a\} \Vdash e$ is an enabling in $\mathcal{E}(\Delta)$, then $Y \to e \in \Delta$ for some $Y \subseteq X \cup \{a\}$. By rule $(\twoheadrightarrow)$, we have that:

$$\frac{Y \to e \in \Delta \qquad \sigma \in [\![\Delta, \Gamma(\eta)]\!] \qquad Y \subseteq X \cup \{a\} \subseteq \overline{\sigma}}{\sigma \mid e \subseteq [\![\Delta, \Gamma(\eta) \setminus \{e\}]\!]}$$

and since $e \in \overline{\eta}$, it follows that $\sigma \in (\sigma \mid e)$. The thesis is then obtained by repeating this procedure until A becomes empty.

- $\circ = \Vdash$. By Definition 4.13, we have that $\Gamma(\sigma) = (\Gamma(\eta) \cup \{a\}) \setminus A$, where A is the same as in (14), and the proof proceeds similarly to the previous case. □

We can now prove a correspondence between proof traces of Δ and honored firing sequences in $\mathcal{P}(\Delta)$.

Theorem 4.28. *Let $\sigma = \langle e_0 \cdots e_n \rangle$, and let Δ be a Horn* PCL *theory. Then, $\sigma \in [\![\Delta]\!]$ iff there exists an honored firing sequence $m_0 \xrightarrow{t_0} \cdots \xrightarrow{t_n} m$ in $\mathcal{P}(\Delta)$ such that $\ell(t_i) = e_i$ for all $i \in 0..n$.*

Proof. For the "only if" direction, assume that $\sigma \in [\![\Delta]\!]$. Now, σ is a lazy computation of $\mathcal{E}(\Delta)$, and by Lemma 4.26 we have that $\Gamma(\sigma) = \emptyset$. By Definition 4.17, we obtain an eager computation τ coherent with σ, such that $\Gamma(\tau) = \Gamma(\sigma) = \emptyset$, and so by Lemma 4.21 it follows that $\Omega(\tau) = \emptyset$. By Lemma 4.22, there exists a firing sequence $m_0 \xrightarrow{t_0} \cdots \xrightarrow{t_n} m$ in $\mathcal{P}(\Delta)$ such that:

(a) $\forall i \in 1..n : t_i = (X_i, e_i, \circ_i)$,

(b) $\forall i \in 0..n : \big(ev(\overline{\tau_i}) = \overline{m_i} \ \wedge \ \Omega(\tau_i) = \Omega(m_i)\big)$

From item (a) it follows that $\ell(t_i) = e_i$, and from item (b) it follows that m is honored.

For the "if" direction, assume that $m_0 \xrightarrow{\ell(t_0)} \cdots \xrightarrow{\ell(t_n)} m$ is a firing sequence in $\mathcal{P}(\Delta)$ such that m is honored. By Lemma 4.23, there exists an eager computation $\tau = \langle (X_1 \circ_1 e_1) \cdots (X_n \circ_n e_n) \rangle$ of $\mathcal{E}(\Delta)$ such that:

(a) $\forall i \in 1..n : \ell(t_i) = e_i$,

(b) $\forall i \in 0..n : \big(ev(\overline{\tau_i}) = \overline{m_i} \ \wedge \ \Omega(\sigma_i) = \Omega(m_i)\big)$

Let $\sigma = ev(\tau)$; clearly, σ is an eager computation of $\mathcal{E}(\Delta)$ coherent with τ. Since m is honored, then $\Omega(\tau) = \emptyset$, and so by Lemma 4.21 $\Gamma(\sigma) = \Gamma(\tau) = \Omega(\tau) = \emptyset$, i.e. σ is honored. By Lemma 4.27, we obtain the thesis $\sigma \in [\![\Delta]\!]$. □

5. Related work and conclusions

There are many different proposals of formal models for behavioural contracts, which we may roughly divide into "physical" and "logical" models. Physical contracts take inspiration from formalisms for concurrent systems (e.g., Petri nets [6], event structures [17, 15], and various sorts of process algebras [18, 19, 20, 21, 2]), and they allow to describe the interaction of services in terms of response to events, message exchanges, *etc.* On the other side, logical contracts are typically expressed as formulae of suitable logics, which take inspiration and extend e.g., modal [22, 23], intuitionistic [24, 8], linear [24], deontic [25] logics to model high-level concepts such as promises, obligations, prohibitions, authorizations, *etc.*

Even though logical contracts aim to provide formal models and reasoning tools for real-world Service Level Agreements, existing approaches have not had a great impact on the design of SOC applications. A reason is that there is no evidence on how to relate high-level properties of a contract with properties of the services which have to realize it. In the realm of physical contracts, the gap between contracts and services is narrower. Several papers, e.g., [19, 20, 26, 2, 6], address the issue of relating global properties (e.g., of a choreography) with local properties of the services which implement it (e.g., deadlock freedom,

communication error freedom, session fidelity), in some cases providing automatic tools to project the choreography to a set of services which correctly implement it.

In this paper, which is an extended and revised version of [27], we propose Lending Petri nets as a model for physical contracts. The notion of LPNs developed here differs with respect to the one in [27]. There, lending capability was confined to places, which were partitioned into *standard* places and *lending* ones, whereas here all the places have the lending capability, provided that there is a lending arc connecting the place to a transition. Thus, in the former definition a *negative* marking was allowed only for standard lending places, whereas now this situation can happen in any place connected to at least one transition with a lending arc. With the new definition, a place s can lend tokens to a transition $t \in s^{\circledcirc}$, while not lending to some other transitions $t' \in s^{\bullet}$. Clearly, a lending place in [27] can be represented in the current model as a place where all outgoing arcs are lending. Hence, the current notion is a conservative extension of the one proposed in [27].

The notion of negative marking, often implemented using *negative* tokens (called also *debit* tokens or *antitokens*), is not a new one in the Petri nets community — although very few papers tackle this notion. Indeed, the interpretation of negative markings does not match the intuition of Petri nets, where tokens are generally intended as resources, and where the marking is a measure of the availability of resources. The intuition of this paper is that negative markings can be exploited to deal with situations where actions are in a *circular* dependency, like the ones arising in contracts. Lending arcs model the intuition that a resource can be given away *on credit*, and a negative marking in a place can be interpreted as the credit made, which must be, sooner or later, *honored*.

Rather than focusing on the quantity of available resources, various approaches relax the requirement that all the places in the pre-set should have enough tokens, and this is modelled by creating debit tokens. In [28], a variant of Petri nets has been used to model an extension of linear logic, called *cancellative linear logic*. In the token game of these nets (called *financial game*), transitions work as in standard Petri nets, but there exists a special move which allows to produce, in any place, a pair token/antitoken. In this way, we can fire any transition (it suffices to produce all the required tokens in its pre-set), but we may end up with a number of antitokens. Each transition in these nets corresponds to a formula of cancellative linear logic, similarly to what we have done by relating PCL with LPNs. A linear implication $a \multimap b$ is realized as a transition consuming from a place a, and producing in a place b, and it can be used in two ways: either one feeds it with the resource a and gets the resource b, or one gets the resource b by introducing at the same time a *debit* $a^{\perp}$, which can be annihilated later on with an occurrence of the resource a. A relevant difference with respect to LPNs is that in [28] pairs tokens/antitokens can always be produced; instead, in our approach, negative markings arise in a more controlled manner, only when lending arcs allow transitions to be fired on credit. In [29] the firing conditions are relaxed in such a way that a step can be executed even though there are not enough tokens in the pre-set of the step. However, with respect to our approach, negative markings are not allowed, which means that tokens taken on credit, i.e. debits, have to be honored in the same step. In this approach tokens can be lent from any place, whereas in our approach tokens can be lent only from specific places and the debits can be rapaid later.

In [30], the idea of places with a negative marking is realized using a new kind of nets, called *debit* Petri nets. The state of a debit Petri net is expressed as a pair (m, d), where m keeps track of the number of tokens in each place, while d counts the antitokens. Tokens and antitokens can be annihilated as in the financial games of [28]. With respect to financial games, debit nets allow for more control, as antitokens may appear only in places with outgoing debit arcs, while in [28] they can appear in any place. Two annihilation strategies are considered in [30]: *instantaneous annihilation*, where tokens and antitokens must cancel out as soon as possible (i.e., either a place contains antitokens, or it contains tokens), and *delayed annihilation*, where tokens and antitokens can coexist in the same place, and be cancelled out at any step. Under the instantaneous annihilation strategy, debit nets are Turing powerful (as debit arcs can encode inhibitor arcs), while under delayed annihilation they do not augment the expressive power of Petri nets. Lending Petri nets use an instantaneous annihilation strategy, and they generalize debit nets by using weight functions (for standard and lending transitions), while debit nets use standard and debit arcs with unit weight. This generalization is convenient to describe contracts in a concise manner (and it has a direct correspondence with Horn PCL clauses, see Section 4), but it does not augment the expressive power of LPNs, which is

equivalent to that of Turing machines.

The notions of nets composition, developed in Section 2.2 and in Section 3.2, is inspired by the one defined in [6] for *open nets*, and extend it. The one defined [6] applies to open nets the idea of net composition presented in [31]. In [6] open nets are nets with an input/output interface, and the places in this interface are either input places (with no incoming arcs) or output places (with no outgoing arcs). The composition of two open nets is then defined by suitably identifying input (output) places of a net with the output (input) places of the other. With respect to [6], the *interface* of an LPN is simply the set of its labeled places and only output places are required to have an empty post-set as well an empty lending post-set, whereas input places may have incoming arcs. We still retain the constraint posed on the initial marking of the places in the interface. In the notion of composition adopted in this paper, if the common label $a \in \ell(S) \cap \ell'(S')$ is associated in N to a place $s \in out(N)$ and in N' to a place $s' \in in(N')$ with empty pre-set (or *vice versa*), and the labelings are injective, we obtain the notion of composition between open nets defined in [6]. The composition results of [6] (in particular, compositional verification stated by Theorem 3.15) hold also in our setting.

Our notion of composition can be related to some other approaches in literature. In [10, 32] composition is achieved in a category oriented way, and the interface is a whole subnet that the components must share up to isomorphisms as specified by the classical push-out construction. As in our approach, places in the interface can be fed either by the component they are part of, or by the other compound net. Instead of, in [11] interface places are a subset of places where tokens can be added or removed without any constraint, and the unique way of disallowing this characteristic is to *close* the net by removing these places from the set of interface place. We do not have a corresponding notion of hiding labeled places, as for us labels are relevant, e.g., when modeling contracts.

The notion of contract nets introduced in Section 3, extends to a *linear* setting the notion of contracts of [33], where event structures (where events model *non-linear* resources) have been used to model participants obligations. A result in [33] establishes that, when obligations are expressed as standard event structures, it is not possible to have contracts which enjoy both agreement and protection, while this can be obtained by using event structures with circular causality [16]. We expect that the former result (mutual exclusion of agreement and protection) can be obtained also in the linear setting, when we consider LPNs without lending arcs; also, agreement and protection could be obtained in contract nets, similarly to the way it is obtained in event structures with circular causality: in Section 4.3 we have already shown some relevant relations between LPNs and CES , which could be exploited to this purpose. In Section 3 we have also related our notion of agreement with the notion of weak termination proposed in [6]; while agreement is somehow more finer-grained than weak termination, (since it discriminates angelic from demonic choices) we have shown that, in the setting of standard Petri nets, the two notions coincide when participants adopt the strategy of firing all and only the enabled transitions.

In Section 4 a suitable subclass of LPNs have been proved to be a model of the Horn fragment of PCL. We have shown that provability in the logic tightly corresponds to reachability of suitables markings (Theorem 4.10), and that proof traces correspond to honored firing sequences (Theorem 4.28). The features of this subclass are the ones stated in Lemma 4.6: in particular, each transition occurs only once in any firing sequence. To prove this result we have resorted to the notion of CES developed in [15, 16, 34]. While to our aims it has been enough to consider conflict free CES, the association among the kind of LPN associated to Horn PCL theories and CES can be generalized. A translation from finite CES (possibly with conflicts) into LPNs could work as follows: the $\vdash$ enablings are translated as transitions without any lending arcs, the $\Vdash$ enablings as transitions with lending arcs, and the conflict among two events e and e' is modelled by an unlabelled place, initially marked, and connected with all the transitions labelled with e or e', and without any incoming arcs.

With respect to the game-theoretic approach pursued in Section 3, Horn PCL theories correspond to occurrence contract nets where resources are used in a non-linear manner. For instance, in the PCL theory $a \to b, a \to c$, the atom a can be used for proving *both* b and c. From the point of view of nets, this is rendered as the fact that there is no need to choose which transition to use to consume the token in a. In the example above, when the token a becomes available, two copies of it are produced: one to be used by the transition which produces b, and the other one to be used by the transition which produces c. The absence

of choices implies that strategies become simpler: to decide if a transition is prudent, it is enough to verify that doing such transition will lead (in some firing sequence) to a configuration where all debits are honored. A similar result has been proved in [9], in the context of event structures with circular causality, and we believe that it can be directly exported to the context of Lending Petri nets, by using the correspondence between firing sequences and computations in CES stated by Lemmas 4.22 and 4.23.

Acknowledgments. We thank Philippe Darondeau, Eric Fabre and Roberto Zunino for useful discussions and suggestions. We thank also the anonymous reviewers that helped us in improving greatly the paper.

References

[1] M. Armbrust, et al., A view of cloud computing, Communication of ACM 53 (4) (2010) 50–58. doi:10.1145/1721654.1721672.

[2] K. Honda, N. Yoshida, M. Carbone, Multiparty asynchronous session types, in: G. C. Necula, P. Wadler (Eds.), Proc. POPL, ACM, 2008, pp. 273–284. doi:10.1145/1328438.1328472.

[3] N. Yoshida, R. Hu, R. Neykova, N. Ng, The Scribble protocol language, in: Proc. TGC, 2013, pp. 22–41. doi:10.1007/978-3-319-05119-2_3.

[4] M. Bartoletti, J. Lange, A. Scalas, R. Zunino, Choreographies in the wild, to appear in Science of Computer Programming, 2015. doi:http://dx.doi.org/10.1016/j.scico.2014.11.015.

[5] W. Reisig, Petri Nets: An Introduction, Vol. 4 of Monographs in Theoretical Computer Science. An EATCS Series, Springer, 1985. doi:10.1007/978-3-642-69968-9.

[6] W. M. P. van der Aalst, N. Lohmann, P. Massuthe, C. Stahl, K. Wolf, Multiparty contracts: Agreeing and implementing interorganizational processes, Computer Journal 53 (1) (2010) 90–106. doi:10.1093/comjnl/bxn064.

[7] M. Bartoletti, T. Cimoli, G. M. Pinna, R. Zunino, Contracts as games on event structures, JLAMP (to appear). doi:10.1016/j.jlamp.2015.05.001.

[8] M. Bartoletti, R. Zunino, A calculus of contracting processes, in: Proc. LICS, IEEE Computer Society, 2010, pp. 332–341. doi:10.1109/LICS.2010.25.

[9] M. Bartoletti, T. Cimoli, P. D. Giamberardino, R. Zunino, Contract agreements via logic, in: M. Carbone, I. Lanese, A. Lluch-Lafuente, A. Sokolova (Eds.), Proc. ICE, Vol. 131 of EPTCS, 2013, pp. 5–19. doi:10.4204/EPTCS.131.2.

[10] P. Baldan, A. Corradini, H. Ehrig, R. Heckel, Compositional semantics for open Petri nets based on deterministic processes, Mathematical Structures in Computer Science 15 (1) (2005) 1–35. doi:10.1017/S0960129504004311.

[11] P. Baldan, F. Bonchi, F. Gadducci, Encoding asynchronous interactions using open Petri nets, in: M. Bravetti, G. Zavattaro (Eds.), Proc. CONCUR, Vol. 5710 of Lecture Notes in Computer Science, Springer, 2009, pp. 99–114. doi:10.1007/978-3-642-04081-8_8.

[12] A. Aghasaryan, E. Fabre, A. Benveniste, R. Boubour, C. Jard, Fault detection and diagnosis in distributed systems: An approach by partially stochastic Petri nets, Discrete Event Dynamic Systems 8 (2) (1998) 203–231. doi:10.1023/A:1008241818642.

[13] R. J. van Glabbeek, The individual and collective token interpretations of Petri nets, in: M. Abadi, L. de Alfaro (Eds.), Proc. CONCUR, Vol. 3653 of Lecture Notes in Computer Science, Springer, 2005, pp. 323–337. doi:10.1007/11539452_26.

[14] R. J. van Glabbeek, G. D. Plotkin, Configuration structures, in: Proc. LICS, IEEE Computer Society, 1995, pp. 199–209. doi:10.1109/LICS.1995.523257.

[15] M. Bartoletti, T. Cimoli, G. M. Pinna, R. Zunino, An event-based model for contracts, in: S. J. Gay, P. Kelly (Eds.), Proc. PLACES, Vol. 109 of EPTCS, 2012, pp. 13–20. doi:10.4204/EPTCS.109.3.

[16] M. Bartoletti, T. Cimoli, G. M. Pinna, R. Zunino, Circular causality in event structures, Fundamenta Informaticae 134 (3-4) (2014) 219–259. doi:10.3233/FI-2014-1101.

[17] T. T. Hildebrandt, R. R. Mukkamala, Declarative event-based workflow as distributed dynamic condition response graphs, in: K. Honda, A. Mycroft (Eds.), Proc. PLACES, Vol. 69 of EPTCS, 2010, pp. 59–73. doi:10.4204/EPTCS.69.3.

[18] L. Bocchi, K. Honda, E. Tuosto, N. Yoshida, A theory of Design-by-Contract for Distributed Multiparty Interactions, in: P. Gastin, F. Laroussinie (Eds.), Proc. CONCUR, Vol. 6269 of Lecture Notes in Computer Science, Springer, 2010, pp. 162–176. doi:10.1007/978-3-642-15375-4_12.

[19] M. Bravetti, I. Lanese, G. Zavattaro, Contract-driven implementation of choreographies, in: C. Kaklamanis, F. Nielson (Eds.), Proc. TGC, Vol. 5474 of Lecture Notes in Computer Science, Springer, 2008, pp. 1–18. doi:10.1007/978-3-642-00945-7_1.

[20] M. Bravetti, G. Zavattaro, Contract based multi-party service composition, in: F. Arbab, M. Sirjani (Eds.), Proc. FSEN, Vol. 4767 of Lecture Notes in Computer Science, Springer, 2007, pp. 207–222. doi:10.1007/978-3-540-75698-9_14.

[21] G. Castagna, N. Gesbert, L. Padovani, A theory of contracts for Web services, ACM Transactions on Programming Languages and Systems 31 (5) (2009) 19:1–19:61. doi:10.1145/1538917.1538920.

[22] M. Abadi, M. Burrows, B. Lampson, G. Plotkin, A calculus for access control in distributed systems, ACM Transactions on Programming Languages and Systems 4 (15) (1993) 706–734. doi:10.1145/155183.155225.

[23] D. Garg, M. Abadi, A modal deconstruction of access control logics, in: R. M. Amadio (Ed.), Proc. FoSSaCS, Vol. 4962 of Lecture Notes in Computer Science, Springer, 2008, pp. 216–230. doi:10.1007/978-3-540-78499-9_16.

[24] M. Abadi, G. D. Plotkin, A logical view of composition, Theoretical Computer Science 114 (1) (1993) 3–30. doi:10.1016/0304-3975(93)90151-I.

[25] C. Prisacariu, G. Schneider, A dynamic deontic logic for complex contracts, Journal of Logic and Algebraic Programming 81 (4) (2012) 458–490. doi:10.1016/j.jlap.2012.03.003.
[26] M. Bravetti, G. Zavattaro, Towards a unifying theory for choreography conformance and contract compliance, in: M. Lumpe, W. Vanderperren (Eds.), Proc. Software Composition, Vol. 4829 of Lecture Notes in Computer Science, Springer, 2007, pp. 34–50. doi:10.1007/978-3-540-77351-1_4.
[27] M. Bartoletti, T. Cimoli, G. M. Pinna, Lending Petri nets and contracts, in: F. Arbab, M. Sirjani (Eds.), Proc. FSEN, Vol. 8161 of Lecture Notes in Computer Science, 2013, pp. 66–82. doi:10.1007/978-3-642-40213-5_5.
[28] N. Martí-Oliet, J. Meseguer, An algebraic axiomatization of linear logic models, in: G. M. Reed, A. W. Roscoe, R. F. Wachter (Eds.), Topology and category theory in computer science, Oxford Univ. Press, 1991, pp. 335–355.
[29] R. Bruni, H. C. Melgratti, U. Montanari, P. Sobocinski, Connector algebras for C/E and P/T nets' interactions, Logical Methods in Computer Science 9 (3). doi:10.2168/LMCS-9(3:16)2013.
[30] P. D. Stotts, P. Godfrey, Place/transition nets with debit arcs, Information Processing Letters 41 (1) (1992) 25–33. doi:10.1016/0020-0190(92)90076-8.
[31] E. Kindler, A compositional partial order semantics for Petri net components, in: P. Azéma, G. Balbo (Eds.), Proc. ICATPN, Vol. 1248 of Lecture Notes in Computer Science, 1997, pp. 235–252. doi:10.1007/3-540-63139-9_39.
[32] P. Baldan, A. Corradini, B. König, A framework for the verification of infinite-state graph transformation systems, Information and Computation 206 (7) (2008) 869–907. doi:10.1016/j.ic.2008.04.002.
[33] M. Bartoletti, T. Cimoli, R. Zunino, A theory of agreements and protection, in: D. A. Basin, J. C. Mitchell (Eds.), Proc. POST, Vol. 7796 of Lecture Notes in Computer Science, Springer, 2013, pp. 186–205. doi:10.1007/978-3-642-36830-1_10.
[34] M. Bartoletti, T. Cimoli, P. D. Giamberardino, R. Zunino, Vicious circles in contracts and in logic, Science of Computer Programming (to appear). doi:10.1016/j.scico.2015.01.005.