Università di Pisa

Dipartimento di Informatica

Technical Report : TR-00-05

# Domain and Event Structure Semantics for Petri Nets with Read and Inhibitors arcs

*P. Baldan, N. Busi, A. Corradini, G.M. Pinna*

May 19, 2000

ADDR: Corso Italia 40,56125 Pisa,Italy. TEL: +39-50-887111. FAX: +39-50-887226

# Domain and Event Structure Semantics for Petri Nets with Read and Inhibitor Arcs

P. Baldan[1], N. Busi[2], A. Corradini[1] and G. M. Pinna[3]

[1] Dipartimento di Informatica - Università di Pisa
[2] Dipartimento di Scienze dell'Informazione - Università di Bologna
[3] Dipartimento di Matematica - Università di Siena

**Abstract.** We propose a functorial concurrent semantics for Petri nets extended with *read* and *inhibitor* arcs, that we call *inhibitor nets*. Along the lines of the seminal work of Winskel on safe nets and extending a previous work in [5] for nets with read arcs, the truly concurrent semantics is given at a categorical level via a chain of functors leading from the category **SW-IN** of semi-weighted inhibitor nets to the category **Dom** of finitary prime algebraic domains (equivalent to the category **PES** of prime event structures). As an intermediate semantic model, we introduce *inhibitor event structures*, an extension of prime event structures able to faithfully capture the dependencies among events which arise in the presence of read and inhibitor arcs.

## Introduction

Several generalizations of Petri nets [27, 29] have been proposed in the literature to overcome the expressiveness limitations arising from the simplicity of the classical model. At a very basic level Petri nets have been extended with two new kinds of arcs, namely *read arcs* (also called test, read, activator or positive contextual arcs) [12, 24, 18, 32] and *inhibitor arcs* (also called negative contextual arcs) [2, 24], which allow a transition to check for the presence, resp. absence of resources (tokens), which are not affected by the firing of the transition. Read arcs are able to faithfully represent the situations where a resource is read but not consumed (read-only accesses). They have been used to model concurrent accesses to shared data (e.g., read operations in a database) [30, 13], to study temporal efficiency in asynchronous systems [32] and to give a truly concurrent semantics to concurrent constraint programs [23, 7]. Inhibitor arcs have been introduced in [2] to solve a synchronization problem not expressible in classical Petri nets. A study of the expressiveness of inhibitor arcs, along with a comparison with other extensions proposed in the literature, namely priorities, exclusive-or transitions and switches, is carried out in [16, 26]. In particular it is worth stressing that inhibitor arcs make the model Turing complete, essentially because they allow to simulate the zero-testing operation of RAM machines which cannot be expressed neither by flow nor by read arcs. Inhibitor arcs have been employed for performance evaluation of distributed systems [3], to provide $\pi$-calculus with a net semantics [9] and to show the existence of an expressiveness gap between two different semantics of a process algebra based on Linda coordination primitives [10].

The purpose of this paper is to give a truly concurrent semantics to Petri nets extended with read and inhibitor arcs, that we will call *inhibitor nets*. We follow the seminal work on ordinary safe nets of [25, 34], where the semantics is given at a categorical level via a chain of coreflections (a particularly nice kind of adjunction), leading from the category **S-N** of safe (marked) P/T nets to the category **Dom** of finitary prime algebraic domains, through the categories **O-N** of occurrence nets and **PES** of prime event structures (PES's), the last step being an equivalence of categories.

$$\begin{tikzcd}
\mathbf{S\text{-}N} \arrow[r, shift right=1.5, "\mathscr{U}"'] \arrow[r, phantom, "\perp"] & \mathbf{O\text{-}N} \arrow[l, hook', shift right=1.5] \arrow[r, shift right=1.5, "\mathscr{E}"'] \arrow[r, phantom, "\perp"] & \mathbf{PES} \arrow[l, shift right=1.5, "\mathscr{N}"'] \arrow[r, shift right=1.5, "\mathscr{L}"'] \arrow[r, phantom, "\sim"] & \mathbf{Dom} \arrow[l, shift right=1.5, "\mathscr{P}"']
\end{tikzcd}$$

As shown in [21, 22] essentially the same construction applies to the wider category of *semi-weighted* nets, i.e. P/T nets in which the initial marking is a set and transitions can generate at most one token in each post-condition (while the generalization to the whole category of P/T nets requires some original technical machinery and allows one to obtain a proper adjunction rather than a coreflection).

The mentioned approach has been extended in [5] to nets with read arcs, referred to as *contextual nets* (see also [33]). The key observation is that prime event structures are not adequate to model in a direct way the dependencies between transition occurrences in a contextual net. The problem is illustrated by the net $N_0$ of Fig. 1 where the same place $s$ is "read" by transition $t_0$ and "consumed" by transition $t_1$ (a read arc is represented by an undirected, horizontal line). The firing of $t_1$ prevents $t_0$ to be executed, so that $t_0$ can never follow $t_1$ in a computation, while the converse is not true, since $t_1$ *can* fire after $t_0$. This situation can be interpreted naturally as an *asymmetric conflict* between the two transitions and cannot be represented faithfully in a PES. To model the behaviour of contextual nets, the paper [5] introduces *asymmetric event structures* (AES's), an extension of prime event structures where the symmetric conflict is replaced by an asymmetric conflict relation. Such a feature is obviously still necessary to be able to model the dependencies arising between events in inhibitor nets, but the nonmonotonic features related to the presence of inhibitor arcs (negative conditions) makes the situation far more complicated.

Consider the net $N_1$ in Fig. 1 where the place $s$, which inhibits transition $t$, is in the post-set of transition $t'$ and in the pre-set of $t_0$ (an inhibitor arc is depicted as a dotted line from $s$ to $t$, ending with an empty circle). The execution of $t'$ inhibits the firing of $t$, which can be enabled again by the firing of $t_0$. Thus $t$ can fire before or after the "sequence" $t'; t_0$, but not in between the two transitions. Roughly speaking there is a sort of atomicity of the sequence $t'; t_0$ w.r.t. $t$. The situation can be more involved since many transitions $t_0, \ldots, t_n$ may have the place $s$ in their pre-set (see the net $N_2$ in Fig. 1). Therefore, after the firing of $t'$, the transition $t$ can be re-enabled by any of the conflicting transitions $t_0, \ldots, t_n$. This leads to a sort of *or*-causality, but only when $t$ fires after $t'$. With a logical terminology we can say that $t$ causally depends on the implication $t' \Rightarrow t_0 \vee t_1 \vee \ldots \vee t_n$.

To face these additional complications in this paper we introduce *inhibitor event structures* (IES's), a generalization of AES's equipped with a ternary relation, called *DE-relation (disabling-enabling relation)* and denoted by $\vdash\!\circ(\cdot,\cdot,\cdot)$, which allows one to model the dependency between transitions in $N_2$ as $\vdash\!\circ(\{t'\}, t, \{t_0, \ldots, t_n\})$. As we will see, the DE-relation is sufficient to represent both causality and asymmetric conflict and thus concretely it is the only relation of an IES.

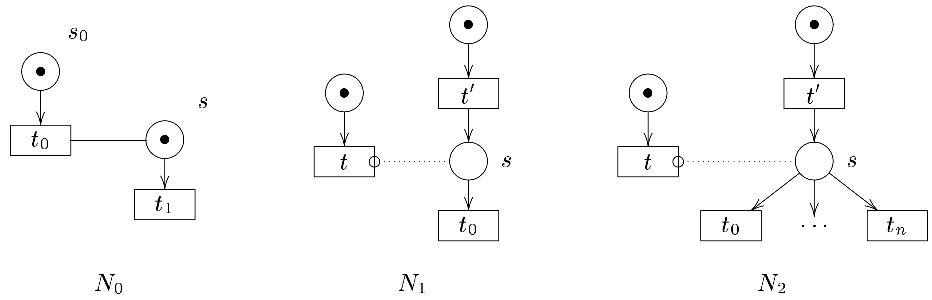



**Fig. 1.** Some basic contextual and inhibitor nets.

As already pointed out in [11], when dealing with inhibitor nets a deterministic computation is not uniquely determined by the events which occur in it. More concretely, in a deterministic process the absence of a token in an inhibitor place, which enables a transition, may arise in two different situations: because the transition producing the token has not been fired yet, or because the transition removing the token has already been fired. For instance, the net $N_1$ of Fig. 1 admits two possible executions involving all its transitions, namely $t; t'; t_0$ and $t'; t_0; t$, which should not be identified from the point of view of causality. To deal with this problem a deterministic process, as defined in [11], includes also a partition of the inhibitor arcs into *before* and *after* arcs. Intuitively, the fact that an inhibitor arc from $s$ to $t$ is classified as "before" means that $t$ must be executed before the place $s$ is filled, while if it is an "after" arc then $t$ must be executed after the token has been removed from $s$.

In a similar way, a configuration of an IES is not uniquely identified as a set of events, but some additional information has to be added which plays a basic role also in the definition of the order on configurations. More concretely, a configuration of an IES is a set of events endowed with a *choice relation* which chooses one among the possible different orders of execution of events constrained by the DE-relation. The configurations of an IES, endowed with a suitable computational order, form a prime algebraic domain, and Winskel's equivalence between **PES** and **Dom** generalizes to a coreflection between the category **IES** of inhibitor event structures and **Dom**.

Inhibitor event structures are also related to *event automata* [28], a class of automata where states are sets of events and the transition relation specifies which events can occur in a certain state. An event automaton is obtained from an IES by considering the partial order of configurations and then forgetting about the history of events, namely identifying different configurations involving the same set of events.

As in the case of ordinary and contextual nets, the connection between nets and event structures is established via an unfolding construction which maps each net into an occurrence net. The complex structure of inhibitor nets computations makes it hard to find an appropriate notion of occurrence inhibitor net. We identify two distinct, both reasonable, notions of occurrence inhibitor net and correspondingly we provide two different unfolding constructions which associate to each semi-weighted inhibitor net an occurrence inhibitor net. In both cases the unfolding gives rise to a functor which is right adjoint to the inclusion. Finally, the unfolding can be naturally abstracted to an IES, having the transitions of the net as events. The main difference with respect to the case of ordinary and contextual nets is the absence of a functor performing the backward step from IES's to occurrence inhibitor nets. We will discuss how the problem is related to the complex kind of causality expressible in IES's, observing that it cannot be overcome by using standard techniques.

The rest of the paper is organized as follows. Section 1 motivates and defines inhibitor event structures. Section 2 studies the relation between IES's, domains and event automata. Section 3 presents the category of inhibitor nets and focuses on the subcategory of semi-weighted inhibitor nets which we shall work with. Section 4 introduces the categories of occurrence inhibitor nets and the corresponding unfolding constructions. Section 5 shows how the unfoldings can be abstracted to an IES semantics. Finally Section 6 draws some conclusions and directions of future research. The interested reader can find in [4] most of the proofs of the results presented in this paper and a detailed treatment of some related issues.

## 1 Inhibitor event structures

This section introduces the class of event structures that we consider adequate for modelling the complex phenomena which arise in the dynamics of inhibitor nets. Let us fix some notational conventions. The powerset of a set $X$ is denoted by $\mathbf{2}^X$, while $\mathbf{2}^X_{fin}$ denotes the set of finite subsets of $X$ and $\mathbf{2}^X_1$ the set of subsets of $X$ of cardinality at most one (singletons or the empty set). In the sequel generic subsets of events will be denoted by upper case letters $A, B, \ldots$, and singletons or empty subsets by $a, b, \ldots$

**Definition 1 (pre-inhibitor event structure).** *A* pre-inhibitor event structure *(pre-IES) is a pair $I = \langle E, \vdash\!\circ \rangle$, where $E$ is a set of* events *and $\vdash\!\circ \; \subseteq \mathbf{2}^E_1 \times E \times \mathbf{2}^E$ is a ternary relation called* disabling-enabling relation (DE-relation *for short*).

Informally, if $\vdash\!\circ(\{e'\}, e, A)$ then the event $e'$ inhibits the event $e$, which can be enabled again by one of the events in $A$. The first argument of the relation can be also the empty set $\emptyset$, $\vdash\!\circ(\emptyset, e, A)$ meaning that the event $e$ is inhibited in the initial state of the system. Moreover the third argument (the set of events $A$) can be empty, $\vdash\!\circ(\{e'\}, e, \emptyset)$ meaning that there are no events that can re-enable $e'$ after it has been disabled by $e$.

The DE-relation is sufficient to represent both causality and asymmetric conflict and thus, concretely, it is the only relation of a (pre-)IES. In fact, if $\vdash\!\circ(\emptyset, e, \{e'\})$ then the event $e$ can be executed only after $e'$ has been fired. This is exactly what happens in a PES when $e'$ causes $e$, or in symbols when $e' < e$. Notice that more generally, if $\vdash\!\circ(\emptyset, e, A)$ then we can imagine $A$ as a set of disjunctive causes for $e$, since at least one of the events in $A$ will appear in every history of the event $e$; intuitively we can think that $e$ causally depends on $\bigvee A$. This generalization of causality, restricted to the case in which the set $A$ is pairwise conflictual (namely all distinct events in $A$ are in conflict), will be represented in symbols as $A < e$. Notice that under the assumption that $A$ is pairwise conflictual, when $A < e$ *exactly one* event in $A$ appears in each history of $e$. Therefore, in particular, for any event $e' \in A$, if $e$ and $e'$ are executed in the same computation then surely $e'$ must precede $e$. Similar notions of or-causality have been studied in general event structures [34], flow event structures [6] and in bundle event structures [19, 20].

Furthermore, if $\vdash\!\circ(\{e'\}, e, \emptyset)$ then $e$ can never follow $e'$ in a computation since there are no events which can re-enable $e$ after the execution of $e'$. Instead the converse order of execution is admitted, namely $e$ *can* fire before $e'$. This situation is naturally interpreted as an *asymmetric conflict* between the two events and it is written $e \nearrow e'$. Asymmetric conflict can be also seen as a *weak* form of *causal dependency*, in the sense that if $e \nearrow e'$ then $e$ precedes $e'$ in all computations containing both events. This explains why a rule below imposes asymmetric conflict to include (also generalized) causality, by asking that $A < e$ implies $e' \nearrow e$ for all $e' \in A$.

Finally, cycles of asymmetric conflict are used to define a notion of conflict on sets of events. If $e_0 \nearrow e_1 \ldots e_n \nearrow e_0$ then all such events cannot appear together in the same computation, since each one should precede the others. This fact is formalized via a conflict relation on sets of events $\#\{e_0, e_1 \ldots, e_n\}$. In particular, binary (symmetric) conflict is represented by asymmetric conflict in both directions.

We are now ready to present the definition of the dependency relations in a pre-IES, making precise their informal interpretation discussed above.

**Definition 2 (dependency relations).** *Let $I = \langle E, \vdash\!\circ \rangle$ be a pre-*IES*. The relations of* (generalized) causality $<\subseteq \mathbf{2}^E \times E$, asymmetric conflict $\nearrow\subseteq E \times E$ *and* conflict $\# \subseteq \mathbf{2}^E_{fin}$ *are defined by the following set of rules:*

$$\frac{\vdash\!\circ(\emptyset, e, A) \quad \#_p A}{A < e} \ (<1) \qquad \frac{A < e \quad \forall e' \in A.\ A_{e'} < e' \quad \#_p(\cup\{A_{e'} \mid e' \in A\})}{(\cup\{A_{e'} \mid e' \in A\}) < e} \ (<2)$$

$$\frac{\vdash\!\circ(\{e'\}, e, \emptyset)}{e \nearrow e'} \ (\nearrow 1) \qquad \frac{e \in A < e'}{e \nearrow e'} \ (\nearrow 2) \qquad \frac{\#\{e, e'\}}{e \nearrow e'} \ (\nearrow 3)$$

$$\frac{e_0 \nearrow \ldots \nearrow e_n \nearrow e_0}{\#\{e_0, \ldots, e_n\}} \ (\#1) \qquad \frac{A' < e \quad \forall e' \in A'.\ \#(A \cup \{e'\})}{\#(A \cup \{e\})} \ (\#2)$$

*where $\#_p A$ means that the events in $A$ are pairwise conflictual, namely $\#\{e, e'\}$ for all $e, e' \in A$ with $e \neq e'$. We will use the infix notation for the binary conflicts, writing $e\#e'$ instead of $\#\{e, e'\}$. Moreover we will write $e < e'$ to indicate $\{e\} < e'$.*

The basic rules $(<1)$, $(\nearrow 1)$ and $(\#1)$, as well as $(\nearrow 2)$ and $(\nearrow 3)$ are justified by the discussion above. Rule $(<2)$ generalizes the transitivity of the causality relation. If $A < e$ and for every event $e' \in A$ we can find a set of events $A_{e'}$ such that $A_{e'} < e'$, then the union of all such sets, namely $\cup\{A_{e'} \mid e' \in A\}$, can be seen as (generalized) cause of $e$, provided that it is pairwise conflictual. Observe that in particular, if $\{e'\} < e$ and $\{e''\} < e'$ then $\{e''\} < e$. Rule $(\#2)$ expresses a kind of hereditarity of the conflict with respect to causality. Suppose $A' < e$ and that any event $e' \in A'$ is in conflict with $A$, namely $\#(A \cup \{e'\})$ for any $e' \in A'$. Since by definition of $<$ the execution of $e$ must be preceded by an event in $A'$ we can conclude that also $e$ is in conflict with $A$, i.e.,

$\#(A \cup \{e\})$. In particular by taking $A' = \{e'\}$ and $A = \{e''\}$ we obtain that if $\{e'\} < e$ and $\#\{e', e''\}$ then $\#\{e, e''\}$.

Note that we can have $\vdash\!\circ\,(\emptyset, e, \emptyset)$, meaning that event $e$ can never be executed. In this case, by rule $(< 1)$, we deduce $\emptyset < e$ and thus, by rule $(\#2)$, we have $\#\{e\}$, i.e., the event $e$ is in conflict with itself.

An inhibitor event structure is defined as a pre-IES where events related by the DE-relation satisfy a few further requirements suggested by the intended meaning of such relation. Furthermore the causality and asymmetric conflict relations must be induced "directly" by the DE-relation.

**Definition 3 (inhibitor event structure).** *An* inhibitor event structure (IES) *is a pre-*IES $I = \langle E, \vdash\!\circ \rangle$ *satisfying, for all* $e \in E$, $a \in \mathbf{2}^E_1$ *and* $A \subseteq E$,

1. *if* $\vdash\!\circ\,(a, e, A)$ *then* $\#_p A$ *and* $\forall e' \in a.\ \forall e'' \in A.\ e' < e''$*;*
2. *if* $A < e$ *then* $\vdash\!\circ\,(\emptyset, e, A)$*;*
3. *if* $e \nearrow e'$ *then* $\vdash\!\circ\,(\{e'\}, e, \emptyset)$*.*

Given a pre-IES $I$ satisfying only (1) it is always possible to "saturate" the relation $\vdash\!\circ$ in order to obtain an IES where the relations of causality and (asymmetric) conflict are exactly the same as in $I$.

**Proposition 1.** *Let* $I = \langle E, \vdash\!\circ \rangle$ *be a pre-*IES *satisfying condition (1) of Definition 3. Then* $\overline{I} = \langle E, \vdash\!\circ' \rangle$*, where* $\vdash\!\circ' = \vdash\!\circ \cup \{(\emptyset, e, A) \mid A < e\} \cup \{(\{e\}, e', \emptyset) \mid e \nearrow e'\}$ *is a* IES*. Moreover the relations of causality, asymmetric conflict and conflict in* $\overline{I}$ *are the same as in* $I$*.*

We next define the category of IES's by introducing a notion of IES-morphism which generalizes both PES and AES-morphisms.

**Definition 4 (category IES).** *Let* $I_0 = \langle E_0, \vdash\!\circ_0 \rangle$ *and* $I_1 = \langle E_1, \vdash\!\circ_1 \rangle$ *be two* IES*'s. An* IES-morphism $f : I_0 \to I_1$ *is a partial function* $f : E_0 \to E_1$ *such that for all* $e_0, e_0' \in E_0$, $A_1 \subseteq E_1$,

1. $(f(e_0) = f(e_0')) \wedge (e_0 \neq e_0') \quad \Rightarrow \quad e_0 \#_0 e_0'$*;*
2. $A_1 < f(e_0) \quad \Rightarrow \quad \exists A_0 \subseteq f^{-1}(A_1).\ A_0 < e_0$*;*
3. $\vdash\!\circ_1(\{f(e_0')\}, f(e_0), A_1) \quad \Rightarrow \quad \exists A_0 \subseteq f^{-1}(A_1).\ \exists a_0 \subseteq \{e_0'\}.\ \vdash\!\circ_0(a_0, e_0, A_0)$.

*where it is intended that* $f(e_0)$ *and* $f(e_0')$ *are defined. We denote by* **IES** *the category of inhibitor event structures and* IES*-morphisms.*

Condition (1) is the usual condition of event structure morphisms which allows one to confuse only conflictual branches of computations. Condition (2) can be seen as a generalization of the requirement of preservation of causes, namely of the property $\lfloor f(e) \rfloor \subseteq f(\lfloor e \rfloor)$, of PES (and AES) morphisms. Finally, condition (3), as it commonly happens for event structures morphisms, just imposes the preservation of computations by asking, whenever some events in the image are constrained in some way, that stronger constraints are present in the pre-image. More precisely suppose that $\vdash\!\circ\,(\{f(e_0')\}, f(e_0), A_1)$. Thus we can have a computation where $f(e_0')$ is executed first and $f(e_0)$ can be executed only after one of the events in $A_1$. Otherwise the computation can start with the execution of $f(e_0)$. According to condition (3), $e_0$ and $e_0'$ are subject in $I_0$ to the same constraint of their images or, when $a_0 = \emptyset$ or $A_0 = \emptyset$, to stronger constraints selecting one of the possible orders of execution. It is worth stressing that, since $A_i < e_i$ can be equivalently expressed as $\vdash\!\circ(\emptyset, e_i, A_i)$, condition (2) is essentially an adaptation of (3), which is needed to cover the case in which the first argument of the DE-relation is the empty set.

The category **PES** of prime event structures can be viewed as a full subcategory of **IES**. This result substantiates the claim that IES's (and constructions on them) are a "conservative" extension of PES's.

**Proposition 2 (prime and inhibitor event structures).** *Let $\mathscr{J}_i : \mathbf{PES} \to \mathbf{IES}$ be the functor defined as follows. To any* PES $P = \langle E, \leq, \# \rangle$ *the functor $\mathscr{J}_i$ associates the* IES $\langle E, \vdash\!\circ \rangle$ *where the DE-relation is defined by $\vdash\!\circ(\emptyset, e, \{e''\})$ if $e'' < e$ and $\vdash\!\circ(\{e'\}, e, \emptyset)$ if $e\#e'$, and for any* PES *morphism $f : P_1 \to P_2$ its image $\mathscr{J}_i(f)$ is $f$ itself. Then the functor $\mathscr{J}_i$ is a full embedding of* **PES** *into* **IES**.

More generally, it is possible to show that the category of asymmetric event structures introduced in [5] fully embeds into **IES** (see [4]). Also (extended) bundle event structures [20] and prime event structures with possible events [28] can be seen as special classes of IES's.

## 2 Inhibitor event structures, domains and event automata

The paper [34] shows that the categories **PES** of prime event structures and **Dom** of finitary prime algebraic domains are equivalent, via the functors $\mathscr{P} : \mathbf{Dom} \to \mathbf{PES}$ and $\mathscr{L} : \mathbf{PES} \to \mathbf{Dom}$. This section establishes a connection between IES's and finitary prime algebraic domains, by showing that the mentioned result generalizes to the existence of a categorical coreflection between **IES** and **Dom**. Then we study the problem of removing the non-executable events from an IES, by characterizing the full subcategory $\mathbf{IES^e}$ consisting of the IES's where all events are executable, as a coreflective subcategory of **IES**. Finally, we investigate the relation between IES's and event automata by showing that the construction taking an IES into the corresponding domain can be easily modified in order to produce an event automaton.

### 2.1 The domain of configurations

The domain associated to an IES is obtained by considering the family of its configurations with a suitable order. Since here computations involving the same events may be different from the point of view of causality, a configuration is not uniquely identified as a set of events, but some additional information has to be added which plays a basic role also in the definition of the order on configurations. More concretely, a configuration of an IES is a set of events endowed with a *choice relation* which chooses among the possible different orders of execution of events constrained by the DE-relation.

Consider a set of events $C$ of an inhibitor event structure $I$, and suppose $e', e, e'' \in C$ and $\vdash\!\circ(\{e'\}, e, A)$ for some $A$, with $e'' \in A$. We already noticed that in this case there are two possible orders of execution of the three events (either $e; e'; e''$ or $e'; e''; e$), which cannot be identified from the point of view of causality. A choice relation for $C$ must choose one of them by specifying that $e$ precedes $e'$ or that $e''$ precedes $e$. To ease the definition of the notion of choice relation, we first introduce, for a given set of events $C$, the set *choices*$(C)$, a relation on $C$ which "collects" *all* the possible precedences between events induced by the DE-relation. A choice relation for $C$ is then defined as suitable subset of *choices*$(C)$.

**Definition 5 (choice).** *Let $I = \langle E, \vdash\!\circ \rangle$ be an* IES *and let $C \subseteq E$. We denote by choices$(C)$ the set*

$$\{(e, e') \mid \exists A.\ \vdash\!\circ_C(\{e'\}, e, A)\} \cup \{(e'', e) \mid \exists A.\ \vdash\!\circ_C(a, e, A) \wedge\ e'' \in A\} \subseteq C \times C,$$

*where the restriction of $\vdash\!\circ(,,)$ to $C$ is defined by $\vdash\!\circ_C(a, e, A)$ if and only if $\vdash\!\circ(a, e, A')$ for some $A'$, with $e \in C$, $a \subseteq C$ and $A = A' \cap C$.*
*A* choice *for $C$ is a relation $\hookrightarrow_C \subseteq$ choices$(C)$ such that*

1. *if $\vdash\!\circ_C(a, e, A)$ then $\exists e' \in a.\ e \hookrightarrow_C e'$ or $\exists e'' \in A.\ e'' \hookrightarrow_C e$;*
2. *$\hookrightarrow_C$ is* acyclic*;*
3. *$\forall e \in C.\ \{e' \in C \mid e' \hookrightarrow^*_C e\}$ is* finite.

Condition (1) intuitively requires that whenever the DE-relation permits two possible orders of execution, the relation $\hookrightarrow_C$ chooses one of them. The fact that $\hookrightarrow_C \subseteq$ *choices*$(C)$ ensures that $\hookrightarrow_C$ does not impose more precedences than necessary. Conditions (2) and (3) guarantee that the precedences specified by $\hookrightarrow_C$ do not give rise to cyclic situations and that each event must be preceded only by finitely many others. Notice that the acyclicity of $\hookrightarrow_C$ ensures that exactly one of the two possible choices in condition (1), namely either $\exists e' \in a.\ e \hookrightarrow_C e'$ or $\exists e'' \in A.\ e'' \hookrightarrow_C e$ is taken. It is worth observing that conditions (2) and (3) can be equivalently rephrased by saying that $\hookrightarrow^*_C$ is a finitary partial order.

Configurations of PES's and AES's are required to be conflict free and downward closed with respect to causality. The following proposition shows that the property of admitting a choice implies a generalization of causal closedness and conflict freeness. Furthermore any choice must agree with the asymmetric conflict (since both relations impose an order of execution on events).

**Proposition 3.** *Let $I = \langle E, \vdash\!\circ \rangle$ be an* IES *and let $C \subseteq E$ be a subset of events such that there exists a choice $\hookrightarrow_C$ for $C$. Then*

1. *for any $e \in C$, if $A < e$ then $A \cap C \neq \emptyset$;*
2. *$\nearrow_C \subseteq \hookrightarrow_C$;*
3. *for any $A \subseteq C$ it is not the case that $\# A$.*

A configuration of an IES is now introduced as a set of events endowed with a choice relation. Proposition 3 above shows how this definition generalizes the notion of PES and AES configuration.

**Definition 6 (configuration).** *Let $I = \langle E, \vdash\!\circ \rangle$ be an* IES*. A* configuration *of $I$ is a pair $\langle C, \hookrightarrow_C \rangle$, where $C \subseteq E$ is a set of events and $\hookrightarrow_C \subseteq C \times C$ is a* choice *for $C$.*

In the sequel, with abuse of notation, we will often denote a configuration and the underlying set of events with the same symbol $C$, referring to the corresponding choice relation as $\hookrightarrow_C$.

The computational order on configurations is a generalization of that introduced in [5] for AES's.

**Definition 7 (extension).** *Let $I = \langle E, \mapstochar\!\!\circ \rangle$ be an* IES *and let $C$ and $C'$ be configurations of $I$. We say that $C'$* extends *$C$ and we write $C \sqsubseteq C'$, if*

1. *$C \subseteq C'$;*
2. *$\forall e \in C.\ \forall e' \in C'.\ e' \hookrightarrow_{C'} e \quad \Rightarrow \quad e' \in C$;*
3. *$\hookrightarrow_C \subseteq \hookrightarrow_{C'}$.*

*The poset of all configurations of $I$, ordered by extension, is denoted by $Conf(I)$.*

The basic idea is that a configuration $C$ can be extended only by adding events which are not supposed to happen before other events already in $C$, as expressed by condition (2). Moreover the extension relation takes into account the choice relations of the two configurations. Intuitively, condition (3) serves to ensure, together with (2), that the past history of events in $C$ remains the same in $C'$.

The intuition on the extension relation can be enforced by observing that it can be equivalently characterized as $C \sqsubseteq C'$ iff $C \subseteq C'$ and the inclusion $i : \langle C, \hookrightarrow_C^* \rangle \to \langle C', \hookrightarrow_{C'}^* \rangle$ is a rigid embedding, i.e., it is monotone and for $e \in C$, $e' \in C'$, $e' \hookrightarrow_{C'}^* e$ implies $e' \in C$. Furthermore, it can be shown that if $C \sqsubseteq C'$ then $\hookrightarrow_C = \hookrightarrow_{C'} \cap (C \times C)$, and thus the inclusion of $C$ into $C'$ is also *order monic*, namely for each $e, e' \in C$, if $e \hookrightarrow_{C'}^* e'$ then $e \hookrightarrow_C^* e'$. Roughly speaking this means that $C \sqsubseteq C'$ whenever $C$ coincides with a "truncation" of $C'$.

The history of an event in a configuration $C$ is formally defined as a suitable subconfiguration of $C$.

**Definition 8 (history).** *Let $I$ be an* IES *and let $C \in Conf(I)$ be a configuration. For any $e \in C$ we define the* history *of $e$ in $C$ as the configuration $\langle C[\![e]\!], \hookrightarrow_{C[\![e]\!]} \rangle$, where $C[\![e]\!] = \{e' \in C \mid e' \hookrightarrow_C^* e\}$ and $\hookrightarrow_{C[\![e]\!]} = \hookrightarrow_C \cap (C[\![e]\!] \times C[\![e]\!])$.*

Recall that, by definition, the reflexive and transitive closure of a choice is a finitary partial order, and thus each history $C[\![e]\!]$ is a *finite* configuration. Furthermore, it is easy to see that $C[\![e]\!] \sqsubseteq C$.

Given a pairwise compatible set of configurations $X \subseteq Conf(I)$ of an IES $I$,[1] has it is possible to show that its greatest lower bound is given by $\bigsqcap X = \langle \bigcap_{C \in X} C, \bigcap_{C \in X} \hookrightarrow_C \rangle$, while its least upper bound is $\bigsqcup X = \langle \bigcup_{C \in X} C, \bigcup_{C \in X} \hookrightarrow_C \rangle$. Furthermore, for any $C_1$ and $C_2$ in $X$, if they contain a common event $e$, then the history of $e$ in the two configurations is the same, namely $C_1[\![e]\!] = C_2[\![e]\!]$. By exploiting such properties, we can prove that the poset of configurations of an IES has the desired algebraic structure.

**Theorem 1 (configurations form a domain).** *Let $I$ be an* IES. *Then $\langle Conf(I), \sqsubseteq \rangle$ is a (finitary prime algebraic) domain. The complete primes of $Conf(I)$ are the possible histories of events in $I$, i.e.*

$$Pr(Conf(I)) = \{C[\![e]\!] \mid C \in Conf(I), e \in C\}.$$

[1] Recall that a subset $X$ of a partial order $\langle D, \sqsubseteq \rangle$ is called pairwise compatible if each pair of elements $x, y \in X$, the set $\{x, y\}$ has an upper bound in $D$.

### 2.2 A coreflection between IES and Dom

To prove that the construction which associates the domain of configurations to an IES lifts to a functor from **IES** to **Dom** a basic result is the fact that IES-morphisms preserve configurations. Observe that since configurations are not simply sets of events it is not completely obvious, a priori, what should be the image of a configuration through a morphism. Let $f : I_0 \to I_1$ be an IES-morphism and let $C_0$ be a configuration of $I_0$. According to the intuition underlying IES (and general event structures) morphisms, we expect that any possible execution of the events in $C_0$ can be simulated in $f(C_0)$. But the converse implication is not required to hold, namely the level of concurrency in $f(C_0)$ may be higher. For instance we can map two causally related events $e_0 \leq e_1$ to a pair of concurrent events. Hence we cannot pretend that the whole image of the choice relation of $C_0$ is a choice for $f(C_0)$, but just that there is a choice for $f(C_0)$ included in such image. By the properties of choices, there is only one choice on $f(C_0)$ included in the image of $\hookrightarrow_{C_0}$, which is obtained as the intersection of the image of $\hookrightarrow_{C_0}$ with $choices(f(C_0))$.

Given a function $f : X \to Y$ and a relation $r \subseteq X \times X$, we will denote by $f(r)$ the relation in $Y$ defined as $f(r) = \{(y, y') \mid \exists (x, x') \in r.\ f(x) = y \ \wedge \ f(x') = y'\}$.

**Lemma 1.** *Let $f : I_0 \to I_1$ be an* IES*-morphism and let $\langle C_0, \hookrightarrow_0 \rangle \in Conf(I_0)$. Then the pair $\langle C_1, \hookrightarrow_1 \rangle$ with $C_1 = f(C_0)$ and $\hookrightarrow_1 = f(\hookrightarrow_0) \cap choices(f(C_0))$, namely the unique choice relation on $C_1$ included in $f(\hookrightarrow_{C_0})$, is a configuration in $I_1$. Moreover the function $f^* : Conf(I_0) \to Conf(I_1)$ which associates to each configuration $C_0$ the configuration $C_1$ defined as above, is a domain morphism.*

The previous lemma implies that the construction taking an IES into its domain of configurations can be viewed as a functor.

**Proposition 4.** *There exists a functor $\mathscr{L}_i$ : **IES** $\to$ **Dom** defined as $\mathscr{L}_i(I) = Conf(I)$ for each* IES *$I$ and $\mathscr{L}_i(f) = f^*$ for each* IES*-morphism $f : I_0 \to I_1$.*

A functor going back from domains to IES's, namely $\mathscr{P}_i$ : **Dom** $\to$ **IES** can be obtained simply as the composition of the functor $\mathscr{P}$ : **Dom** $\to$ **PES**, defined by Winskel, with the full embedding of **PES** into **IES** discussed in Proposition 2. The functor $\mathscr{P}_i$ is left adjoint to $\mathscr{L}_i$ and thus they establish a coreflection between **IES** and **Dom**.

**Theorem 2 (coreflection between IES and Dom).** *The functor $\mathscr{P}_i$ : **Dom** $\to$ **IES** is left adjoint to $\mathscr{L}_i$ : **IES** $\to$ **Dom**. The counit of the adjunction at an* IES *$I$ is the function $\epsilon_I : \mathscr{P}_i \circ \mathscr{L}_i(I) \to I$, mapping each history of an event $e$ into the event $e$ itself, i.e., $\epsilon_I(C[\![e]\!]) = e$, for all $C \in Conf(I)$ and $e \in C$.*

It is worth stressing that the above result, together with Winskel's equivalence between the category **Dom** of domains and the category **PES** of prime event structures, allow to translate an IES $I$ into a PES $\mathscr{P}(\mathscr{L}_i(I))$. The universal characterization of the construction intuitively ensures that the obtained PES is the "best approximation" of $I$ in the category **PES**. By the characterization of the complete prime elements in the domain of configurations (see Theorem 1) we have that the events in $\mathscr{P}(\mathscr{L}_i(I))$ are the possible histories of the events in $I$. The picture below depicts the PES corresponding to a basic IES containing the events $\{e, e', e_0, \ldots, e_n\}$ related by the DE-relation as $\vdash\!\circ(\{e'\}, e, \{e_0, \ldots, e_n\})$. We explicitly represent a history of an event $e$ as a set of events, where $e$ appears in boldface style.

$$
\begin{tikzcd}
 & \{\mathbf{e}\} \arrow[rrr, dotted, no head, "\#" description] \arrow[d] & & & \{\mathbf{e'}\} \arrow[dl] \arrow[dr] & \\
 & \{e,\mathbf{e'}\} \arrow[dl] \arrow[dr] & & \{e',\mathbf{e_0}\} \arrow[d] & \cdots & \{e',\mathbf{e_n}\} \arrow[d] \\
\{e,e',\mathbf{e_0}\} & \cdots & \{e,e',\mathbf{e_n}\} & \{e',e_0,\mathbf{e}\} & \cdots & \{e',e_n,\mathbf{e}\}
\end{tikzcd}
$$

### 2.3 Removing non-executable events

The non-executability of events in an IES is not completely captured by the proof system of Definition 2, in the sense that we cannot derive $\#\{e\}$ for every non-executable event. Here we propose a semantic approach to rule out unused events from an IES, namely we simply remove from a given IES all events which do not appear in any configuration. Nicely, this can be done functorially and the subcategory $\mathbf{IES^e}$ of IES's where all events are executable turns out to be a coreflective subcategory of $\mathbf{IES}$. Moreover, the coreflection between $\mathbf{IES}$ and $\mathbf{Dom}$ restricts to a coreflection between $\mathbf{IES^e}$ and $\mathbf{Dom}$.

**Definition 9.** *We denote by $\mathbf{IES^e}$ the full subcategory of $\mathbf{IES}$ consisting of the* IES*'s $I = \langle E, \vdash\!\circ \rangle$ such that for any $e \in E$ there exists $C \in Conf(I)$ with $e \in C$.*

Any IES is turned into an $\mathbf{IES^e}$ object by forgetting the events which do not appear in any configuration.

**Definition 10.** *We denote by $\Psi : \mathbf{IES} \to \mathbf{IES^e}$ the functor mapping each* IES *$I$ into the $\mathbf{IES^e}$ object $\Psi(I) = \overline{\langle \psi(E), \vdash\!\circ_{\psi(E)} \rangle}$, where $\psi(E)$ is the set of executable events in $I$, namely*

$$\psi(E) = \{e \in E \mid \exists C \in Conf(I).\ e \in C\}.$$

*Moreover if $f : I_0 \to I_1$ is an* IES*-morphism then $\Psi(f) = f_{|\psi(E_0)}$. With $\mathscr{J}_{ies} : \mathbf{IES^e} \to \mathbf{IES}$ we denote the inclusion.*

It is easy to verify that, if $I$ is a $\mathbf{IES^e}$ object and $I'$ an arbitrary IES, then any IES-morphism $f : I \to \Psi(I')$ is also a morphism $f : I \to I'$. This implies that the inclusion of $\mathbf{IES^e}$ into $\mathbf{IES}$ is left adjoint to $\Psi$, i.e., $\Psi \vdash \mathscr{J}_{ies}$, and thus that $\mathbf{IES^e}$ is a coreflective subcategory of $\mathbf{IES}$.

Finally observe that the functor $\mathscr{P}_i : \mathbf{Dom} \to \mathbf{IES}$ maps each domain into the encoding of a PES, which is clearly an object in $\mathbf{IES^e}$. Therefore it is easy to prove that the coreflection between $\mathbf{IES}$ and $\mathbf{Dom}$ restricts to a coreflection between $\mathbf{IES^e}$ and $\mathbf{Dom}$.

**Corollary 1.** *Let $\mathscr{P}_i^{\mathbf{e}} : \mathbf{Dom} \to \mathbf{IES^e}$ and $\mathscr{L}_i^{\mathbf{e}} : \mathbf{IES^e} \to \mathbf{Dom}$ denote the restrictions of the functors $\mathscr{P}_i$ and $\mathscr{L}_i$. Then $\mathscr{P}_i^{\mathbf{e}} \dashv \mathscr{L}_i^{\mathbf{e}}$.*

### 2.4 From inhibitor event structures to event automata

Event automata [28] are a kind of automata arising as a generalization of the domain of configurations of an event structure. The states of an event automaton are sets of events and the transition relation, which specifies how one state can evolve in another, is constrained only by the fact that each step must add exactly one event. Since states are sets of events, without any additional structure, the past history of computations is partly forgot.

**Definition 11.** *An* event automaton *is a triple $EA = (E, St, \mapsto)$, where $E$ is a set of* events*, $St \subseteq \mathbf{2}^E_{fin}$ is a set of* states *and $\mapsto \subseteq St \times St$, called the* transition relation*, satisfies: for all $X, Y \in St$, if $X \mapsto Y$ then $Y = X \cup \{e\}$ with $e \notin X$, for some $e \in E$. An event automaton $EA$ is called* simple *if $\emptyset \in St$ and $\emptyset \mapsto^* s$ for all $s \in St$, i.e., each state is reachable from the empty state.*

Since in this section we will consider only simple event automata, the qualification *simple* will be omitted. Event automata are turned into a category by defining a suitable notion of morphism, which, as for event structures, respects the transition relation.

**Definition 12.** *Let $EA = (E, St, \mapsto)$ and $EA' = (E', St', \mapsto')$ be two event automata. An* event automata morphism *$f : EA \to EA'$ is a partial function $f : E \to E'$ such that for each state $s \in St$ its image $f(s) \in St'$, $f$ is injective on $s$, and if $s \mapsto s \cup \{e\}$ and $f(e)$ is defined then $f(s) \mapsto' f(s) \cup \{f(e)\}$. We denote by* **EvAut** *the category of event automata and event automata morphisms.*

In [28] it is shown how the configurations of various brands of event structures give rise to (special kinds of) event automata. Similarly, here we show that a slight modification of the construction which associates to an IES its domain of configurations defines a functor from **IES** to **EvAut**. Given an IES $I = \langle E, \vdash\!\circ \rangle$, the corresponding event automata $\mathscr{A}(I)$ is obtained by considering the domain of configurations of $I$ and then forgetting the choice relations of configurations. More formally

$$\mathscr{A}(I) = \langle E, St, \mapsto \rangle,$$

where $St = \{C \subseteq E \mid \exists \hookrightarrow_C .\ \langle C, \hookrightarrow_C \rangle \in \mathit{Conf}(I)\}$, and $C_1 \mapsto C_2$ when for $i \in \{1,2\}$ there are choice relations $\hookrightarrow_i$ such that $\langle C_i, \hookrightarrow_i \rangle \in \mathit{Conf}(I)$ and $\langle C_1, \hookrightarrow_1 \rangle \sqsubseteq \langle C_2, \hookrightarrow_2 \rangle$, with $|C_2 - C_1| = 1$, namely $\langle C_1, \hookrightarrow_1 \rangle$ is an immediate predecessor of $\langle C_2, \hookrightarrow_2 \rangle$. It is easy to verify that the above construction gives rise to a functor.

**Proposition 5.** *The operation $\mathscr{A}$ :* **IES** $\to$ **EvAut** *defined on objects as above and which acts as identity on morphisms is a well-defined functor.*

## 3 A category of inhibitor nets

*Inhibitor nets* are an extension of ordinary Petri nets where, by means of read and inhibitor arcs, transitions can check both for the presence and for the absence of tokens in places of the net. To give the formal definition we need some notation for multisets. Let $A$ be a set; a *multiset* of $A$ is a function $M : A \to \mathbb{N}$. The set of multisets of

$A$ is denoted by $\mu A$. The usual operations and relations on multisets, like multiset union $+$ or multiset difference $-$, are used. We write $M \leq M'$ if $M(a) \leq M'(a)$ for all $a \in A$. If $M \in \mu A$, we denote by $[\![M]\!]$ the multiset defined as $[\![M]\!](a) = 1$ if $M(a) > 0$ and $[\![M]\!](a) = 0$ otherwise, obtained by changing all non-zero coefficients of $M$ to 1; sometimes $[\![M]\!]$ will be confused with the corresponding subset $\{a \in A \mid [\![M]\!](a) = 1\}$ of $A$. A *multirelation* $f : A \to B$ is a multiset of $A \times B$. We will limit our attention to finitary multirelations, namely multirelations $f$ such that the set $\{b \in B \mid f(a,b) > 0\}$ is finite. Multirelation $f$ induces in an obvious way a function $\mu f : \mu A \to \mu B$, defined as $\mu f(\sum_{a\in A} n_a \cdot a) = \sum_{b\in B} \sum_{a\in A} (n_a \cdot f(a,b)) \cdot b$ (possibly partial, since infinite coefficients are disallowed). If $f$ satisfies $f(a,b) \leq 1$ for all $a \in A$ and $b \in B$, i.e. $f = [\![f]\!]$, then we sometimes confuse it with the corresponding set-relation and write $f(a,b)$ for $f(a,b) = 1$.

**Definition 13 (inhibitor net).** *A* (marked) inhibitor Petri net (i-net) *is a tuple* $N = \langle S, T, F, C, I, m\rangle$, *where*

- $S$ *is a set of* places*;*
- $T$ *is a set of* transitions*;*
- $F = \langle F_{pre}, F_{post}\rangle$ *is a pair of multirelations from* $T$ *to* $S$*;*
- $C$ *and* $I$ *are relations between* $T$ *and* $S$*, called the* context *and* inhibitor relation*, respectively;*
- $m$ *is a multiset of* $S$*, called the* initial marking.

*If the inhibitor relation* $I$ *is empty then* $N$ *is called a* contextual net (c-net).

We assume, without loss of generality, that $S \cap T = \emptyset$. Moreover, we require that for each transition $t \in T$, there exists a place $s \in S$ such that $F_{pre}(t,s) > 0$. In the following when considering an i-net $N$, we will assume that $N = \langle S, T, F, C, I, m\rangle$. Moreover superscripts and subscripts on the net names carry over the names of the net components. For instance $N_i = \langle S_i, T_i, F_i, C_i, I_i, m_i\rangle$.

Let $N$ be an i-net. As usual, the functions from $\mu T$ to $\mu S$ induced by the multirelations $F_{pre}$ and $F_{post}$ are denoted by ${}^{\bullet}(\ )$ and $(\ )^{\bullet}$, respectively. If $A \in \mu T$ is a multiset of transitions, ${}^{\bullet}A$ is called its *pre-set*, while $A^{\bullet}$ is called its *post-set*. Moreover, by $\underline{A}$ we denote the *context* of $A$, defined as $\underline{A} = C([\![A]\!])$, and by ${}^{\circledcirc}A = I([\![A]\!])$ the *inhibitor set* of $A$. The same notation is used to denote the functions from $S$ to $\mathbf{2}^T$ defined as, for $s \in S$, ${}^{\bullet}s = \{t \in T \mid F_{post}(t,s) > 0\}$, $s^{\bullet} = \{t \in T \mid F_{pre}(t,s) > 0\}$, $\underline{s} = \{t \in T \mid C(t,s)\}$ and ${}^{\circledcirc}s = \{t \in T \mid I(t,s)\}$.

A finite multiset of transitions $A$ is enabled at a marking $M$, if $M$ contains the pre-set of $A$ and an additional multiset of tokens which covers the context of $A$. Furthermore no token must be present nor produced by the transitions in the places of the inhibitor set of $A$.

**Definition 14 (token game).** *Let* $N$ *be an i-net and let* $M$ *be a* marking *of* $N$*, i.e., a multiset* $M \in \mu S$*. A finite multiset* $A \in \mu T$ *is* enabled *at* $M$ *if* ${}^{\bullet}A + \underline{A} \leq M$ *and* $[\![M + A^{\bullet}]\!] \cap {}^{\circledcirc}A = \emptyset$*. The* transition relation *between markings is defined as*

$$M\,[A\rangle\,M' \qquad \textit{iff} \qquad A \text{ is enabled by } M \text{ and } M' = M - {}^{\bullet}A + A^{\bullet}.$$

Step and firing sequences, as well as reachable markings are defined in the usual way.

**Definition 15 (i-net morphism).** *Let $N_0$ and $N_1$ be i-nets. An* i-net morphism *$h : N_0 \to N_1$ is a pair $h = \langle h_T, h_S \rangle$, where $h_T : T_0 \to T_1$ is a partial function and $h_S : S_0 \to S_1$ is a multirelation such that (1) $\mu h_S(m_0) = m_1$ and (2) for each $t \in T$,*

*(a) $\mu h_S({}^\bullet t) = {}^\bullet h_T(t)$* *(c) $\mu h_S(\underline{t}) = \underline{h_T(t)}$*

*(b) $\mu h_S(t^\bullet) = h_T(t)^\bullet$* *(d) $[\![h_S]\!]^{-1}({}^{\circledcirc}h_T(t)) \subseteq {}^{\circledcirc}t$.*

*where $[\![h_S]\!]$ is the set relation underlying the multirelation $h_S$. We denote by* **IN** *the category having i-nets as objects and i-net morphisms as arrows, and by* **CN** *its full subcategory of c-nets.*

Conditions (1), (2.a) - (2.b) are the defining conditions of Winskel's morphisms on ordinary nets, while (2.c) is the obvious condition which takes into account contexts.[2] Condition (2.d) regarding the inhibitor arcs can be understood if we think of morphisms as simulations. Like preconditions and contexts must be preserved to ensure that the morphism maps computations of $N_0$ into computations of $N_1$, similarly, inhibitor conditions, which are negative conditions, must be reflected. In fact observe that condition (2.d) on inhibiting places can be rewritten as

$$s_1 \in [\![\mu h_S(s_0)]\!] \;\wedge\; I_1(h_T(t_0), s_1) \quad \Rightarrow \quad I_0(t_0, s_0),$$

which shows more explicitly that inhibitor arcs are reflected. In particular, if $h_S$ is a *total function* then

$$I_1(h_T(t_0), h_S(s_0)) \quad \Rightarrow \quad I_0(t_0, s_0).$$

Although, to the best of our knowledge, no other categories of inhibitor nets have been introduced in the literature, our i-net morphisms can be seen as a generalization of the process mappings of [8, 11].

By the next proposition i-net morphisms preserve the token game, and thus marking reachability.

**Proposition 6 (morphisms preserve the token game).** *Let $N_0$ and $N_1$ be i-nets, and let $h = \langle h_T, h_S \rangle : N_0 \to N_1$ be an i-net morphism. Then for each $M, M' \in \mu S$ and $A \in \mu T$*

$$M\,[A\rangle\, M' \quad \Rightarrow \quad \mu h_S(M)\,[\mu h_T(A)\rangle\, \mu h_S(M').$$

*Therefore i-net morphisms preserve reachable markings, i.e. if $M_0$ is a reachable marking in $N_0$ then $\mu h_S(M_0)$ is reachable in $N_1$.*

As in [34, 22, 5] we will restrict our attention to a subclass of nets where each token produced in a computation has a uniquely determined history. The next definition introduces the corresponding subcategory of **IN**.

[2] The category of contextual nets considered in [5] slightly differs from **CN** for the fact that contexts can be general multisets rather than sets and the inhibitor relation is absent rather than empty. Our choices here are only aimed at simplifying the presentation. It is easy to see that **CN** fully embeds in the category of [5], and that all the results and constructions in that paper can be developed also for **CN**.

**Definition 16 (semi-weighted and safe i-nets).** *A* semi-weighted *i-net is an i-net $N$ such that the initial marking $m$ is a set and $F_{post}$ is a relation (i.e., $t^\bullet$ is a set for all $t \in T$). We denote by* **SW-IN** *the full subcategory of* **IN** *having semi-weighted i-nets as objects; the corresponding subcategory of c-nets is denoted by* **SW-CN**.

*A semi-weighted i-net is called* safe *if also $F_{pre}$ is a relation and each reachable marking is a set.*

## 4 Occurrence i-nets and the unfolding constructions

Generally speaking, an occurrence net provides a static representation of some computations of a net, in which the events (firing of transitions) and the relationships between events are made explicit. In [5] the notion of (nondeterministic) occurrence net has been generalized to the case of nets with read arcs. Here, the presence of the inhibitor arcs and the complex kind of dependencies they induce on transitions makes it hard to find an appropriate notion of occurrence i-net.

As a first step, consider an i-net where backward conflicts are not admitted, namely where each place is in the post-set of at most one transition. The condition of finiteness and acyclicity of the causes, which ensures that each transition of an occurrence net is firable in some computation, has no obvious generalizations in this setting. Indeed even the notion of set of causes of a transition becomes unclear. In fact, as already observed, a transition $t$ to be fired may require the absence of tokens in a place $s$, and therefore it can be seen as a (weak) cause of the transition $t'$ in the pre-set of $s$ if it fires before $s$ is filled, or as a consequence of any of the transitions $t_1, \ldots, t_n$ in the post-set of $s$, otherwise (see the net $N_2$ in Fig. 1).

In this section we present two different, in our opinion both reasonable, notions of occurrence i-net and, correspondingly, we develop two unfolding constructions.

In the first construction, given an i-net $N$, we consider the underlying contextual net $N_c$, obtained from $N$ by forgetting the inhibitor arcs. Then, disregarding the inhibitor arcs, we apply to $N_c$ the unfolding construction for contextual nets defined in [5], which produces an occurrence contextual net $\mathcal{U}_a(N_c)$. Finally, if a place $s$ and a transition $t$ were originally connected by an inhibitor arc in the net $N$, then we insert an inhibitor arc between each copy of $s$ and each copy of $t$ in $\mathcal{U}_a(N_c)$, thus obtaining the unfolding $\mathcal{U}_i(N)$ of the net $N$. Then the characterization of the unfolding as a universal construction can be lifted from contextual to inhibitor nets. Furthermore, in this way the unfolding of an inhibitor net is decidable, in the sense that the problem of establishing if a possible transition occurrence actually appears in the unfolding is decidable. This fact, besides being nice from a theoretical point of view, may be helpful if one wants to use the unfolding in practice to prove properties of the modelled system. The price to pay is that, differently from what happens for ordinary and contextual nets, some of the transitions in the unfolding may not be firable, since they are generated without taking care of inhibitor places. Therefore not all the transitions of the unfolding correspond to a concrete firing of a transition of the original net, but only those which are executable.

In the second approach, the dependency relations (of causality and asymmetric conflict) for a net are defined only with respect to a fixed assignment for the net (playing a role similar to choices) which specifies for any inhibitor arc $(t, s)$ if the inhibited transition $t$ is executed before or after the place $s$ is filled and in the second case which one of the transitions in the post-set $s^\bullet$ of the inhibitor place consumes the token. Then the firability of a transition $t$ amounts to the existence of an assignment which is acyclic on the transitions which must be executed before $t$. Relying on this idea we can define a notion of occurrence net where each transition is really executable. The corresponding unfolding construction produces a net where the mentioned problem of the existence of non-firable transitions disappears. However, in this way, as a consequence of the Turing completeness of inhibitor nets (see, e.g., [1]) the produced unfolding is not decidable.

### 4.1 Lifting the unfolding from contextual to inhibitor nets

In the first approach, the unfolding construction disregards the inhibitor arcs. Consequently the notion of occurrence i-net is defined without taking into account the dependencies between transitions induced by such kind of arcs.

Given a safe i-net $N$ let us define the *read causality relation* as the least transitive relation $<_r$ on $S \cup T$ such that $s <_r t$ if $s \in {}^\bullet t$, $t <_r s$ if $s \in t^\bullet$, and $t <_r t'$ if $t^\bullet \cap \underline{t'} \neq \emptyset$, the only novelty with respect to ordinary nets being the last clause stating that a transition causally depends on transitions generating tokens in its context. The *read asymmetric conflict* $\nearrow_r$ is defined by taking $t \nearrow_r t'$ if $t'$ consumes a token in the context of $t$, namely $\underline{t} \cap {}^\bullet t' \neq \emptyset$, in such a way that the firing of $t'$ inhibits $t$. Moreover $t \nearrow_r t'$ if $(t \neq t' \ \wedge \ {}^\bullet t \cap {}^\bullet t' \neq \emptyset)$ to capture the usual symmetric conflict, and finally, according to the weak causality interpretation of the asymmetric conflict, $t \nearrow_r t'$ whenever $t <_r t'$.

**Definition 17 (occurrence i-nets).** *An* occurrence i-net $N$ *is a safe i-net* $N$ *where causality* $\leq_r$ *is a finitary partial order, asymmetric conflict* $\nearrow_r$ *is acyclic on the causes of each transition, for all* $s \in S$ $|{}^\bullet s| \leq 1$ *and* $m = \{s \in S \mid {}^\bullet s = \emptyset\}$.

*The full subcategory of* **SW-IN** *having occurrence i-nets as objects is denoted by* **O-IN**.

The relations $\leq_r$ and $\nearrow_r$ associated to an i-net $N$ are exactly the relations of causality and asymmetric conflict of the underlying c-net, as defined in [5], and thus occurrence i-nets are occurrence c-nets in the sense of [5], enriched with inhibitor arcs. To formalize this fact it is useful to introduce a functor mapping each i-net to the underlying c-net and the inclusion of the subcategory of c-nets into the category of i-nets (see the figure below).

$$\begin{tikzcd} \mathbf{O\text{-}CN} \arrow[r, hook, shift left] \arrow[r, phantom, "\perp"] & \mathbf{SW\text{-}CN} \arrow[l, shift left, "\mathscr{U}_a"] \arrow[r, hook, shift left, "\mathscr{I}_{ci}"] & \mathbf{SW\text{-}IN} \arrow[l, shift left, "\mathscr{R}_{ic}"] & \mathbf{O\text{-}IN} \arrow[l, hook', "\mathscr{I}_O"'] \end{tikzcd}$$

**Definition 18.** *We denote by* $\mathscr{R}_{ic} : \mathbf{SW\text{-}IN} \to \mathbf{SW\text{-}CN}$ *the functor which maps each i-net into the underlying c-net with an empty inhibitor relation, namely defined by* $\mathscr{R}_{ic}(\langle S, T, F, C, I, m\rangle) = \langle S, T, F, C, \emptyset, m\rangle$, *and by* $\mathscr{I}_{ci} : \mathbf{SW\text{-}CN} \to \mathbf{SW\text{-}IN}$ *the inclusion.*

As mentioned above, it is easy to see that an occurrence i-net is a safe i-net $N$ such that $\mathscr{R}_{ic}(N)$ is an occurrence c-net in the sense of [5].

We remark that, since the above definition does not take into account the inhibitor arcs of the net, we are not guaranteed that each transition in an occurrence i-net is firable. For instance, in the i-net $N_3$ below the only transition $t$ can never fire, but, by looking at the underlying c-net $\mathscr{R}_{ic}(N_3)$ depicted aside, it is immediate to see that $N_3$ is an occurrence i-net.


$N_3$ $\qquad\qquad$ $\mathscr{R}_{ic}(N_3)$

Let **O-CN** denote the category of occurrence c-nets, namely the full subcategory of **O-IN** having only c-nets as objects. The paper [5] defines an unfolding functor $\mathscr{U}_a$ : **SW-CN** $\to$ **O-CN**, mapping each semi-weighted c-net to an occurrence c-net. The functor $\mathscr{U}_a$ is shown there to be right adjoint to the inclusion functor of **O-CN** into **SW-CN**. By suitably using the functors $\mathscr{R}_{ic}$ and $\mathscr{I}_{ci}$ we can lift both the construction and the result to inhibitor nets.

**Definition 19 (unfolding).** *Let $N$ be a semi-weighted i-net. Consider the occurrence c-net $\mathscr{U}_a(\mathscr{R}_{ic}(N)) = \langle S', T', F', C', \emptyset, m'\rangle$ and the folding morphism $f_N : \mathscr{U}_a(\mathscr{R}_{ic}(N)) \to \mathscr{R}_{ic}(N)$. Define an inhibiting relation on the net $\mathscr{U}_a(\mathscr{R}_{ic}(N))$ by taking for $s' \in S'$ and $t' \in T'$*

$$I'(s', t') \quad \textit{iff} \quad I(f_N(s'), f_N(t')).$$

*Then the unfolding $\mathscr{U}_i(N)$ of the net $N$ is the occurrence i-net $\langle S', T', F', C', I', m'\rangle$ and the* folding morphism *is given by $f_N$ seen as a function from $\mathscr{U}_i(N)$ into $N$.*

The fact that $\mathscr{U}_i(N)$ is an occurrence i-net immediately follows from its construction. Furthermore, since the place component of $f_N$ is a total function, according to condition (2.d) in the definition of i-net morphism, the unfolding $\mathscr{U}_i(N)$ can be characterized as the *least* i-net which extends $\mathscr{U}_a(N)$ with the addition of inhibitor arcs in a way that $f_N : \mathscr{U}_i(N) \to N$ is a well defined i-net morphism.

The unfolding construction is functorial, namely we can define a functor $\mathscr{U}_i$ : **SW-IN** $\to$ **O-IN**, which acts on arrows as $\mathscr{U}_a \circ \mathscr{R}_{ic}$. In other words, given $h : N_0 \to N_1$, the arrow $\mathscr{U}_i(h) : \mathscr{U}_i(N_0) \to \mathscr{U}_i(N_1)$ is obtained by interpreting $h$ as a morphism between the c-nets underlying $N_0$ and $N_1$, taking its image via $\mathscr{U}_a$, and then considering the map $\mathscr{U}_a(h)$ as an arrow from $\mathscr{U}_i(N_0)$ to $\mathscr{U}_i(N_1)$. Such a functor establishes a coreflection between semi-weighted i-nets and occurrence i-nets.

**Theorem 3.** *The unfolding functor $\mathscr{U}_i$ :* **SW-IN** $\to$ **O-IN** *is right adjoint to the obvious inclusion functor $\mathscr{I}_O$ :* **O-IN** $\to$ **SW-IN** *and thus establishes a coreflection between* **SW-IN** *and* **O-IN**.

*The component at an object $N$ in* **SW-IN** *of the counit of the adjunction, $f : \mathscr{I}_O \circ \mathscr{U}_i \stackrel{.}{\to} 1$, is the folding morphism $f_N : \mathscr{U}_i(N) \to N$.*

### 4.2 Executable occurrence i-nets

The second approach is inspired by the notion of deterministic process of an i-net in [8]. As mentioned in the introduction, the inhibitor arcs of the net underlying a process are partitioned into two subsets: the *before* inhibitor arcs and *after* inhibitor arcs. Then the dependencies induced by such a partition are required to be acyclic in order to guarantee the firability of all the transitions of the net in a single computation. Following this idea, to ensure that each transition of a nondeterministic occurrence net is firable in some computation, we require, for each transition $t$, the existence of a so-called assignment which partitions the inhibitor arcs into before and after arcs, without introducing cyclic dependencies on the transitions which must be executed before $t$.

**Definition 20 (assignment).** *Let $N$ be a safe i-net. An* assignment *for $N$ is a function $\rho : I \to T$ such that, for all $(t,s) \in I$, $\rho(t,s) \in {}^\bullet s \cup s^\bullet$.*

Intuitively, an assignment $\rho$ specifies for each inhibitor arc $(t,s)$, if the transition $t$ fires *before* or *after* the place $s$ receives a token. If $\rho(t,s) \in {}^\bullet s$ then $(t,s)$ is a before arc, while if $\rho(t,s) \in s^\bullet$ then $(t,s)$ is an after arc. In the last case, since we are considering possibly nondeterministic nets and thus the place $s$ may be in the pre-set of several transitions, the assignment specifies also which of the transitions in $s^\bullet$ consumes the token.

Given a safe net $N$ once an assignment $\rho$ for $N$ is fixed, new dependencies arise between the transitions of the net, formalized by means of the relations $\prec_i^\rho$ and $\nearrow_i^\rho$. We define $t \prec_i^\rho t'$ iff $\exists s \in {}^{\circledcirc}t' \cap {}^\bullet t.\ \rho(t',s) = t$ and $t \nearrow_i^\rho t'$ iff $\exists s \in {}^{\circledcirc}t \cap t'^\bullet.\ \rho(t,s) = t'$. Observe that, as suggested by the adopted symbols, the additional dependencies can be seen as a kind of causality and asymmetric conflict, respectively. In fact if $t \prec_i^\rho t'$, then $t'$ can happen only after $t$ has removed the token from $s$, and thus $t$ acts as a kind of cause for $t'$. If $t \nearrow_i^\rho t'$ then if both $t$ and $t'$ happen in the same computation then necessarily $t$ occurs before $t'$, since $t'$ generates a token in a place $s$ which inhibits $t$, while according to the interpretation of $\rho$, $t$ must occur before the place $s$ is filled.

Under a fixed assignment $\rho$, we can introduce a kind of generalized causality and asymmetric conflict by joining the relations $\leq_r$ and $\nearrow_r$ defined in the previous subsection with the additional dependencies induced by the inhibitor arcs. We define $<_\rho = (<_r \cup \prec_i^\rho)^+$ and $\lhd_\rho = <_\rho \cup \nearrow_r \cup \nearrow_i^\rho$, i.e., $\lhd_\rho$ records both kinds of dependency. Furthermore, for $x \in S \cup T$ we denote by $\lfloor x \rfloor_\rho$ the set $\{t \in T \mid t \leq_\rho x\}$, and similarly, for $X \subseteq S \cup T$, we define $\lfloor X \rfloor_\rho = \bigcup\{\lfloor x \rfloor_\rho \mid x \in X\}$.

Now we are ready to introduce executable occurrence i-nets, which refine occurrence i-nets by constraining all the transitions of the net to be firable.

**Definition 21 (executable occurrence i-net).** *An* executable occurrence i-net *is a safe net $N$ such that (i) for all $t \in T$ there exists an assignment $\rho$ such that $(\lhd_\rho)_{\lfloor t \rfloor_\rho}$ is acyclic and $\lfloor t \rfloor_\rho$ is finite, (ii) for all $s \in S$ $|{}^\bullet s| \leq 1$, and (iii) $m = \{s \in S \mid {}^\bullet s = \emptyset\}$.*

It is not difficult to see that each executable occurrence i-net is an occurrence i-net. We denote by $\mathbf{O\text{-}IN^e}$ the full subcategory of **O-IN** having executable occurrence i-nets as objects.

As the reader probably noticed, the notions of assignment and choice are strictly related. Formally, as we will see in the next section, each occurrence i-net $N$ can be mapped to an IES and, for any subset $X \subseteq T$, an assignment $\rho$ for $N$ such that $X = \lfloor X \rfloor_\rho$ and $\lhd_\rho$ is acyclic and finitary on $X$ uniquely determines a choice turning $X$ in a configuration of the IES corresponding to $N$.

We remark that it is not possible to require the existence of a single assignment $\rho$ such that $\lhd_\rho$ is acyclic on $\lfloor t \rfloor_\rho$ for each transition $t$ of the net. For instance, such an assignment does not exist for the net in Fig. 2, although each of its transitions can fire in some computation (and for each transition there exists an assignment $\rho$ for which $\lhd_\rho$ is acyclic on its causes).

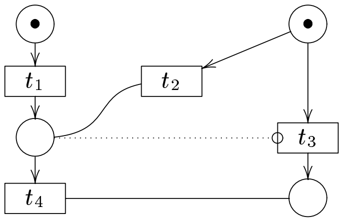



**Fig. 2.** A net for which there exists no assignment $\rho$ making relation $\lhd_\rho$ acyclic on the whole net.

Now, a notion of concurrent set of places of an executable occurrence i-net can be naturally defined.

**Definition 22 (concurrency).** *A set of places $M \subseteq S$ is called* concurrent*, written $conc(M)$, if there is an assignment $\rho$ such that*

*i. for all $s, s' \in M$ $\neg(s <_\rho s')$,*
*ii. $\lfloor M \rfloor_\rho$ is finite and*
*iii. $\lhd_\rho$ acyclic on $\lfloor M \rfloor_\rho$.*

It is possible to show that, as for ordinary and contextual nets, a set of places $M$ is concurrent if and only if there is a reachable marking in which all the places of $M$ contain a token.

**Proposition 7.** *Let $N$ be an executable occurrence i-net and let $M \subseteq S$. Then $conc(M)$ iff there exists a reachable marking $M'$ such that $M \subseteq M'$.*

The above result allows one to prove a basic property of executable occurrence i-nets, namely the fact that each transition of such a net can fire in some computation (and thus each place contains a token at some reachable marking).

**Proposition 8.** *Let $N$ be an executable occurrence i-net. Then for each transition $t \in T$ there exists a reachable marking such that $t$ is enabled at $M$.*

We can now introduce a different unfolding construction, that, when applied to a semi-weighted i-net $N$, produces an executable occurrence i-net.

**Definition 23 ((executable) unfolding).** *Let $N$ be a semi-weighted i-net. The* (executable) unfolding $\mathscr{U}_i^{\mathbf{e}}(N) = \langle S', T', F', C', I', m' \rangle$ *of the net $N$ and the* folding morphism $f_N = \langle f_T, f_S \rangle : \mathscr{U}_i^{\mathbf{e}}(N) \to N$ *are the unique executable occurrence i-net and i-net morphism satisfying the following equations:*

$$
\begin{array}{l}
m' = \{\langle \emptyset, s\rangle \mid s \in m\} \\
S' = m' \cup \{\langle t', s\rangle \mid t' \in T' \ \wedge \ s \in f_T(t')^{\bullet}\} \\
T' = \{t' \mid t' = \langle M_p, M_c, t\rangle \ \wedge \ t \in T \ \wedge \ M_p \cup M_c \subseteq S' \ \wedge \ M_p \cap M_c = \emptyset\} \\
\qquad\qquad \wedge \ conc(M_p \cup M_c) \wedge \ \mu f_S(M_p) = {}^{\bullet}t \ \wedge \ \mu f_S(M_c) = \underline{t} \\
\qquad\qquad \wedge \ \exists \rho.\ (\lfloor t' \rfloor_\rho \textit{ finite} \ \wedge \ \lhd_\rho \textit{ acyclic on } \lfloor t' \rfloor_\rho)\}
\end{array}
$$

$$
\begin{array}{lll}
F'_{pre}(t', s') & \text{iff} & t' = \langle M_p, M_c, t\rangle \ \wedge \ s' \in M_p \quad (t \in T) \\
F'_{post}(t', s') & \text{iff} & s' = \langle t', s\rangle \quad (s \in S) \\
C'(t', s') & \text{iff} & t' = \langle M_p, M_c, t\rangle \ \wedge \ s' \in M_c \quad (t \in T) \\
I'(t', s') & \text{iff} & f_S(s', s) \ \wedge \ I(f_T(t'), s)
\end{array}
$$

$$
\begin{array}{lll}
f_T(t') = t & \text{iff} & t' = \langle M_p, M_c, t\rangle \\
f_S(s', s) & \text{iff} & s' = \langle x, s\rangle \quad (x \in T' \cup \{\emptyset\})
\end{array}
$$

As usual, places and transitions in the unfolding represent respectively tokens and firing of transitions in the original net. Each item of the unfolding is a copy of an item in the original net, enriched with the corresponding "history". The folding morphism $f$ maps each item of the unfolding to the corresponding item in the original net.

As one would expect, the two proposed unfolding constructions are tightly related, in the sense that $\mathscr{U}_i^{\mathbf{e}}(N)$ can be obtained from $\mathscr{U}_i(N)$ simply by removing the non executable transitions. This fact can be formalized by first defining a "pruning" functor $\Pi : \mathbf{O\text{-}IN} \to \mathbf{O\text{-}IN}^{\mathbf{e}}$ which removes the non executable transitions from a general occurrence i-net thus producing an executable occurrence i-net. Next one can show that, given an executable occurrence i-net $N$ and any occurrence i-net $N'$, a morphism $f : N \to N'$ is also a morphism from $N$ to $\Pi(N')$, and thus that the pruning functor $\Pi : \mathbf{O\text{-}IN} \to \mathbf{O\text{-}IN}^{\mathbf{e}}$ is right adjoint to the inclusion functor $\mathscr{J}^{\mathbf{e}} : \mathbf{O\text{-}IN}^{\mathbf{e}} \to \mathbf{O\text{-}IN}$, and $\mathbf{O\text{-}IN}^{\mathbf{e}}$ is a coreflective subcategory of $\mathbf{O\text{-}IN}$. At this point, one can formally state the relationship between $\mathscr{U}_i^{\mathbf{e}}(N)$ and $\mathscr{U}_i(N)$, which provides also an indirect proof of the universality of the new unfolding construction.

**Proposition 9.** *For any semi-weighted i-net $N$, $\mathscr{U}_i^{\mathbf{e}}(N) = \Pi(\mathscr{U}_i(N))$. Therefore $\mathscr{U}_i^{\mathbf{e}}$ is right adjoint to the inclusion functor and they establish a coreflection between* $\mathbf{SW\text{-}IN}$ *and* $\mathbf{O\text{-}IN}^{\mathbf{e}}$.

## 5 Inhibitor event structure semantics for i-nets

To provide an event structure and a domain semantics for i-nets we investigate the relationship between occurrence i-nets and inhibitor event structures. The kind of dependencies arising among transitions in an occurrence i-net can be naturally represented by the DE-relation, and therefore the IES corresponding to an occurrence i-net is obtained by forgetting the places and taking the transitions of the net as events. Furthermore morphisms between occurrence i-nets naturally restrict to morphisms between the corresponding IES's, and therefore the semantics is given via a functor $\mathscr{E}_i : \mathbf{O\text{-}IN} \to \mathbf{IES}$.

The construction, when applied to an executable occurrence i-net, restricts to a functor $\mathscr{E}_i^{\mathbf{e}} : \mathbf{O\text{-}IN^e} \to \mathbf{IES^e}$.

The converse step, from IES's to occurrence i-nets, instead, is shown to be very problematic. An object level construction can be easily performed, associating to each IES a corresponding i-net. However such a construction does not give rise to a functor. We briefly discuss the origin of this problem, showing that it is intimately connected to or-causality.

### 5.1 From occurrence i-nets to IES's

Let us show first how an IES can be extracted from an occurrence i-net.

**Definition 24.** *Let $N$ be an occurrence i-net. The pre-*IES *associated to $N$ is defined as $I_N^p = \langle T, \vdash\!\circ_N^p \rangle$, with $\vdash\!\circ_N \subseteq \mathbf{2}_1^T \times T \times \mathbf{2}^T$, given by: for $t, t' \in T$, $t \neq t'$ and $s \in S$*

1. *if $t^\bullet \cap ({}^\bullet t' \cup \underline{t'}) \neq \emptyset$ then $\vdash\!\circ_N^p(\emptyset, t', \{t\})$*
2. *if $({}^\bullet t \cup \underline{t}) \cap {}^\bullet t' \neq \emptyset$ then $\vdash\!\circ_N^p(\{t'\}, t, \emptyset)$;*
3. *if $s \in {}^{\circledcirc}t$ then $\vdash\!\circ_N^p({}^\bullet s, t, s^\bullet)$.*

*The* IES *associated to $N$, denoted by $I_N = \langle T, \vdash\!\circ_N \rangle$, is obtained by saturating $I_N^p$, i.e., $I_N = \overline{I_N^p}$.*

The first two clauses of the definition encode, by using the DE-relation, the causal dependencies and the asymmetric conflicts induced by the flow and read arcs (we could have written if $t <_r t'$ then $\vdash\!\circ_N^p(\emptyset, t', \{t\})$ and if $t \nearrow_r t'$ then $\vdash\!\circ_N^p(\{t'\}, t, \emptyset)$). The last clause fully exploits the expressiveness of the DE-relation to represent the dependencies induced by inhibitor places. Notice that $I_N^p$ is a pre-IES satisfying also condition (1) of the definition of IES. Thus, by Proposition 1, it can be "saturated" to obtain the IES $I_N$.

It can be shown that the transition component of an i-net morphism is an IES-morphism between the corresponding IES's. Hence we get the existence of a functor which maps each i-net to the corresponding IES defined as in Definition 24 and each i-net morphism to its transition component.

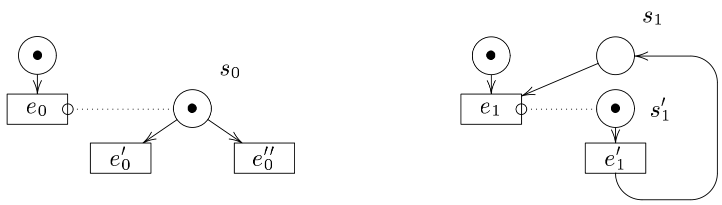
**Definition 25.** *We denote by $\mathscr{E}_i : \mathbf{O\text{-}IN} \to \mathbf{IES}$ the functor defined as $\mathscr{E}_i(N) = I_N$ for each occurrence i-net $N$ and $\mathscr{E}_i(h : N_0 \to N_1) = h_T$ for each morphism $h : N_0 \to N_1$.*

By exploiting the relation between choices and assignments mentioned before, one can verify that if $N$ is an executable occurrence i-net then $\mathscr{E}_i(N)$ is an $\mathbf{IES^e}$ object. Therefore the functor $\mathscr{E}_i$ restricts to a functor $\mathscr{E}_i^{\mathbf{e}} : \mathbf{O\text{-}IN^e} \to \mathbf{IES^e}$.

The coreflection between **IES** ($\mathbf{IES^e}$) and **Dom** can be finally used to obtain a domain semantics, and, by exploiting Winskel's equivalence, a prime event structure semantics for semi-weighted i-nets. As explained in Section 2, the PES semantics is obtained from the IES semantics by introducing an event for each possible different history of events in the IES.

### 5.2 From IES's to i-nets: an open problem

In [34] Winskel maps each prime event structure into a canonical occurrence net, via a free construction which generates for each set of events related in a certain way by the dependency relations a unique place that induces that kind of relation on the events. Winskel's construction can be generalized to i-nets and IES's, in such a way that when applied to any $\mathbf{IES^e}$ object $I$ the construction produces an occurrence i-net $\mathcal{N}_i(I)$, and $\mathcal{E}_i(\mathcal{N}_i(I)) \simeq I$ (see [4]). However a problem exists, which appears hard to get rid of and which makes impossible to turn the construction into a functor. Consider two IES's $I_0$ and $I_1$, obtained by saturating the pre-IES's $\langle\{e_0, e_0', e_0''\}, \{(\emptyset, e_0, \{e_0', e_0''\})\}\rangle$, and $\langle\{e_1, e_1'\}, \{(\emptyset, e_1, \{e_1'\})\}\rangle$. The problem can be understood by looking at (part of) the nets $\mathcal{N}_i(I_0)$ and $\mathcal{N}_i(I_1)$ produced by the construction and represented in the figure below.




part of $\mathcal{N}_i(I_0)$ part of $\mathcal{N}_i(I_1)$

The presence of the place $s_0$ in $\mathcal{N}_i(I_0)$ is justified by the fact that $\{e_0', e_0''\} < e_0$, while both $s_1$ and $s_1'$ in $\mathcal{N}_i(I_1)$ are generated to induce the dependency $e_1' < e_1$. It is easy to see that $f : I_0 \to I_1$, defined by $f(e_0) = e_1$ and $f(e_0') = f(e_0'') = e_1'$, is a well defined IES-morphism, but it cannot be extended to the places of the nets above in order to obtain an i-net morphism, the problem being that no place in ${}^\bullet e_0$ can be mapped to $s_1$. Intuitively, this is related to the fact that in an occurrence i-net a situation of or-causality can be induced only by a place like $s_0$, which inhibits a transition and is consumed by other transitions. However, to have a net morphism from $\mathcal{N}_i(I_0)$ to $\mathcal{N}_i(I_1)$ we would need a place $s_0'$ in the post-set of $e_0'$ and $e_0''$ and in the preset of $e_0$, inducing or-causality by means of a backward conflict, which is forbidden in occurrence i-nets. Instead, at the level of IES's, when one forgets the state, the two dependency situations become indistinguishable.

## 6 Conclusions

We have provided a functorial concurrent semantics for Petri nets with read and inhibitor arcs. The proposed constructions, which generalize Winskel's work on safe ordinary nets and the work in [5] on contextual nets, are summarized in the diagram below (where unnamed functors are inclusions).

$$
\begin{tikzcd}
& \mathbf{O\text{-}IN} \arrow[dl, shift right, hook'] \arrow[dl, phantom, "\perp"] \arrow[rr, "\mathscr{E}_i"] \arrow[dd, shift right, "\Pi"'] \arrow[dd, phantom, "\vdash"] & & \mathbf{IES} \arrow[r, shift left, "\mathscr{A}"] \arrow[dd, shift right, "\Psi"'] \arrow[dd, phantom, "\vdash"] \arrow[dr, shift left, "\mathscr{L}_i"] \arrow[dr, phantom, "\perp"] & \mathbf{EvAut} \arrow[l, shift left, "\mathscr{P}_i"] & \\
\mathbf{SW\text{-}IN} \arrow[ur, shift right, "\mathscr{U}_i"'] \arrow[dr, shift right, "\mathscr{U}_i^{\mathbf{e}}"'] & & & & \mathbf{Dom} \arrow[ul, shift left, "\mathscr{P}_i^{\mathbf{e}}"] \arrow[dl, shift left, "\mathscr{P}_i^{\mathbf{e}}"] \arrow[r, shift right, "\mathscr{P}"'] \arrow[r, phantom, "\sim"] & \mathbf{PES} \arrow[l, shift right, "\mathscr{L}"'] \\
& \mathbf{O\text{-}IN^e} \arrow[ul, shift right, hook'] \arrow[ul, phantom, "\perp"] \arrow[uu, shift right, hook'] \arrow[rr, "\mathscr{E}_i^{\mathbf{e}}"] & & \mathbf{IES^e} \arrow[uu, shift right, hook'] \arrow[ur, shift right, "\mathscr{L}_i^{\mathbf{e}}"'] \arrow[ur, phantom, "\perp"] & &
\end{tikzcd}
$$

Unfortunately, the objective of developing a coreflective semantics for inhibitor nets is partially failed, since the construction mapping each occurrence i-net into the corresponding IES is not expressed as a coreflection. The problem of fully extending Winskel's chain of coreflections to i-nets remains an interesting matter of future research. The discussion in Section 5 suggests that a possible solution could be to consider a radically different notion of occurrence i-net (and thus of unfolding construction), allowing for the presence of backward conflicts. In the case of ordinary nets, a corresponding notion has been studied in the literature under the name of flow net [6].

Another issue which remains open regards the possibility of representing event automata by means of IES's. Given an event automaton $EA = (E, St, \mapsto)$, a natural way to construct a corresponding IES would be to define $\mathit{Conflict}(EA) = \{X \subseteq_{fin} E \mid \forall s \in St.\ |s \cap X| \leq 1\}$, $e \leq_{EA} e'$ iff $\forall s \in St.\ e' \in s \Rightarrow e \cap s \neq \emptyset$ and finally introduce $\vdash\!\circ_{EA} \subseteq \mathbf{2}_1^E \times E \times \mathbf{2}^E$ as follows: for $A \in \mathit{Conflict}(EA)$, $a \subseteq E$ such that $\forall e' \in a.\ \forall e'' \in A.\ e' <_{EA} e''$

$$\vdash\!\circ_{EA}(a, e, A) \quad \text{iff} \quad \forall s \in St.\ s \mapsto s \cup \{e\} \Rightarrow (a \subseteq s \Rightarrow A \cap s \neq \emptyset).$$

Although $\langle E, \vdash\!\circ_{EA} \rangle$ is a well-defined IES, not surprisingly the construction does not enjoy good properties and in particular it fails to be functorial. This is essentially due to the generality of the phenomena modelled by event automata, which prevents one to give a faithful representation of event automata by means of IES's. It would be interesting to give an abstract characterization of the subcategory of **EvAut**, whose behaviour can be described by means of IES's.

In the paper we hinted at the relationship between our IES's and other event structure models in the literature. Although a thorough comparison is prevented by lack of space, it can be easily seen that IES's properly generalize prime [34], asymmetric [5], (extended) bundle event structures [20] and prime event structures with possible events [28]. The relation between IES's and flow event structures [6] (with possible flow [28]) is still to be investigated, but although strictly related, likely the two models are not comparable in the sense that there are IES whose sets of configurations cannot be described by a flow event structure and vice versa.

The connection between IES's and event automata suggests also the possibility of comparing our model with other event based models proposed in the literature as generalizations of the family of configurations of event structures, like configuration structures [31] and Chu spaces [15]. In particular it could be interesting to try to give a logical view of IES's, in the style of the presentation of event structures as propositional theories in [31]. To this end, also the logical approach to causality of [14] could provide some interesting hints. Some similarities can be found also with local event structures [17], where, as in the case of IES's, the enabling of events can be non monotone. However a direct comparison appears difficult to carry out since local event structures explicitly represent configurations and concurrent enabling of sets of events, while IES's give an intensional description of such notions by means of the DE-relation. Probably also in this case one could try a comparison at the level of corresponding event automata.

Finally, we recall that a semantics for inhibitor nets, based on a generalization of Mazurkiewicz traces, has been developed in [18]. Such paper assumes a notion of enabling different from ours, allowing for the concurrent firing of steps where a token is generated in the inhibitor set. Consequently concurrent steps may not be serializable and this is the reason why the simultaneity (independence) relation of Mazurkiewicz traces is not sufficiently expressive, and one must consider also a serializability relation which explicitly says if two simultaneous events are serializable and in which order. Understanding if, despite the different notions of enabling, a relationship can be established with our work is left as a matter of future investigation.

## References


1. T. Agerwala. A complete model for representing the coordination of asynchronous processes. Hopkins Computer Research Report 32, John Hopkins University, 1974.
2. T. Agerwala and M. Flynn. Comments on capabilities, limitations and “correctness” of Petri nets. *Computer Architecture News*, 4(2):81–86, 1973.
3. M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. John Wiley, 1995.
4. P. Baldan. *Modelling concurrent computations: from contextual Petri nets to graph grammars*. PhD thesis, Department of Computer Science, University of Pisa, 2000. Available as technical report n. TD-1/00.
5. P. Baldan, A. Corradini, and U. Montanari. An event structure semantics for P/T contextual nets: Asymmetric event structures. In M. Nivat, editor, *Proceedings of FoSSaCS '98*, volume 1378 of *LNCS*, pages 63–80. Springer Verlag, 1998.
6. G. Boudol. Flow Event Structures and Flow Nets. In *Semantics of System of Concurrent Processes*, volume 469 of *LNCS*, pages 62–95. Springer Verlag, 1990.
7. F. Bueno, M. Hermenegildo, U. Montanari, and F. Rossi. Partial order and contextual net semantics for atomic and locally atomic CC programs. *Science of Computer Programming*, 30:51–82, 1998.
8. N. Busi. *Petri Nets with Inhibitor and Read Arcs: Semantics, Analysis and Application to Process Calculi*. PhD thesis, University of Siena, Department of Computer Science, 1998.
9. N. Busi and R. Gorrieri. A Petri Nets Semantics for $\pi$-calculus. In *Proceedings of CONCUR'95*, volume 962 of *LNCS*, pages 145–159. Springer Verlag, 1995.
10. N. Busi, R. Gorrieri, and G. Zavattaro. On the expressiveness of Linda coordination primitives. *Information and Computation*, 2000. To appear.
11. N. Busi and G. M. Pinna. Process semantics for Place/Transition nets with inhibitor and read arcs. *Fundamenta Informaticæ*, 40(2-3):165–197, 1999.
12. S. Christensen and N. D. Hansen. Coloured Petri nets extended with place capacities, test arcs and inhibitor arcs. In M. Ajmone-Marsan, editor, *Applications and Theory of Petri Nets*, volume 691 of *LNCS*, pages 186–205. Springer Verlag, 1993.
13. N. De Francesco, U. Montanari, and G. Ristori. Modeling Concurrent Accesses to Shared Data via Petri Nets. In *Programming Concepts, Methods and Calculi, IFIP Transactions A-56*, pages 403–422. North Holland, 1994.
14. J. Gunawardena. Geometric logic, causality and event structures. In J. C. M. Baeten and J. F. Groote, editors, *Proceedings of CONCUR '91*, volume 527 of *LNCS*, pages 266–280. Springer Verlag, 1991.

15. V. Gupta. *Chu Spaces : A Model for Concurrency.* PhD thesis, Stanford University, Department of Computer Science, August 1994.
16. M. Hack. Petri net languages. Technical Report 159, MIT, Cambridge, Massachusetts, 1976.
17. P. W. Hoogers, H. C. M. Kleijn, and P. S. Thiagarajan. An event structure semantics for general Petri nets. *Theoretical Computer Science*, 153(1–2):129–170, 1996.
18. R. Janicki and M. Koutny. Semantics of inhibitor nets. *Information and Computation*, 123:1–16, 1995.
19. R. Langerak. Bundle Event Structures: A Non-Interleaving Semantics for Lotos. In *$5^{th}$ Intl. Conf. on Formal Description Techniques (FORTE'92)*, pages 331–346. North-Holland, 1992.
20. R. Langerak. *Transformation and Semantics for LOTOS.* PhD thesis, Department of Computer Science, University of Twente, 1992.
21. J. Meseguer, U. Montanari, and V. Sassone. On the semantics of Petri nets. In *Proceedings of CONCUR '92*, volume 630 of *LNCS*, pages 286–301. Springer Verlag, 1992.
22. J. Meseguer, U. Montanari, and V. Sassone. Process versus unfolding semantics for Place/Transition Petri nets. *Theoretical Computer Science*, 153(1-2):171–210, 1996.
23. U. Montanari and F. Rossi. Contextual occurrence nets and concurrent constraint programming. In H.-J. Schneider and H. Ehrig, editors, *Proceedings of the Dagstuhl Seminar 9301 on Graph Transformations in Computer Science*, volume 776 of *LNCS*. Springer Verlag, 1994.
24. U. Montanari and F. Rossi. Contextual nets. *Acta Informatica*, 32(6), 1995.
25. M. Nielsen, G. Plotkin, and G. Winskel. Petri Nets, Event Structures and Domains, Part 1. *Theoretical Computer Science*, 13:85–108, 1981.
26. J.L. Peterson. *Petri Net Theory and the Modelling of Systems.* Prentice-Hall, 1981.
27. C.A. Petri. *Kommunikation mit Automaten.* PhD thesis, Schriften des Institutes für Instrumentelle Matematik, Bonn, 1962.
28. G. M. Pinna and A. Poigné. On the nature of events: another perspective in concurrency. *Theoretical Computer Science*, 138(2):425–454, 1995.
29. W. Reisig. *Petri Nets: An Introduction.* EACTS Monographs on Theoretical Computer Science. Springer Verlag, 1985.
30. G. Ristori. *Modelling Systems with Shared Resources via Petri Nets.* PhD thesis, Department of Computer Science - University of Pisa, 1994.
31. R. J. van Glabbeek and G. D. Plotkin. Configuration structures. In D. Kozen, editor, *Proceedings of* $10^{th}$ *Annual IEEE Symposium on Logic in Computer Science*, pages 199–209. IEEE Computer Society Press, June 1995.
32. W. Vogler. Efficiency of asynchronous systems and read arcs in Petri nets. In *Proceedings of ICALP'97*, volume 1256 of *LNCS*, pages 538–548. Springer Verlag, 1997.
33. W. Vogler, A. Semenov, and A. Yakovlev. Unfolding and finite prefix for nets with read arcs. In *Proceedings of CONCUR'98*, volume 1466 of *LNCS*, pages 501–516. Springer-Verlag, 1998.
34. G. Winskel. Event Structures. In *Petri Nets: Applications and Relationships to Other Models of Concurrency*, volume 255 of *LNCS*, pages 325–392. Springer Verlag, 1987.