Synthesising ENI-Systems with Interval Order Semantics

Maciej Koutny[1,*], Marta Pietkiewicz-Koutny[1]

[1] *School of Computing, Newcastle University*
Urban Sciences Building, 1 Science Square, Newcastle Helix, Newcastle upon Tyne, NE4 5TG, United Kingdom

Abstract

Elementary net systems with inhibitor arcs are a class of fundamental Petri net models with very simple markings which are sets of places. Their standard semantics is based on sequences of executed transitions or, alternatively, as labelled total orders. In this paper, we introduce semantics based on interval (partial) orders which allows one to describe behaviours where transitions have non-atomic duration. For such a semantical model, we consider the net synthesis problem, and show that the standard notion of a region of transition system (providing input to the synthesis procedure) can still be applied after suitable modifications.

1. Introduction

Petri nets are a general model of concurrent systems which emerged in the 1960's as a counterpart to the state machines that were used so successfully to model sequential systems. A particular advantage of Petri nets is that the model allows one to both specify concurrent system designs, and the behaviours of such systems. It is generally acknowledged that concepts related to fundamental notions of concurrency theory, such as causality and independence, can be well explained using the framework provided by Petri nets (see, e.g., [1, 2, 3, 4]). A fundamental class of Petri nets in that respect are Elementary Net Systems (EN-systems) [5]. In this paper, we consider EN-systems extended with inhibitor arcs which allow testing for emptiness of places, which results in Elementary Net Systems with Inhibitor Arcs (ENI-systems) [6, 7, 8, 9].

In general, the execution semantics of Petri nets (i.e., the representation of individual runs or observations) is captured by total orders of executed transitions (or, equivalently, by firing sequences), or stratified orders of executed transitions where simultaneity is transitive (or, equivalently, by step sequences). Having said that, it was argued by Wiener in 1914 [10] (and later, more formally, in [11]) that any execution that can be observed by a single observer must be an interval order, and so the most precise (qualitative) observational semantics is based on interval orders, where simultaneity is often non-transitive.

In this paper, extending the ideas presented in [12], we first show how one can generate interval order executions of ENI-systems in a direct way, without the need to modify the original system

International Workshop on Petri Nets and Software Engineering, June, 2024, Geneva, Switzerland
*Corresponding author.
maciej.koutny@ncl.ac.uk (M. Koutny); marta.koutny@ncl.ac.uk (M. Pietkiewicz-Koutny)
0000-0003-4563-1378 (M. Koutny)

specification (e.g., by splitting transitions into explicit beginnings and endings) as it was done, for example, in [13, 14, 15]. We also define Interval Reachability Graphs (IR-graphs) which are finite generators of potentially infinite sets of interval orders defined by ENI-systems. IR-graphs are a subclass of Interval Transition Systems (IT-systems) which differ from the standard transition systems since instead of having their arcs labelled by executed transitions, they have states labelled by sets of transitions (interpreted as transitions currently being executed). Then, assuming the interval order semantics of ENI-systems, we consider the problem of synthesising ENI-systems from given IT-systems.

We approach the new synthesis problem using the standard synthesis approach based on the theory of regions [16, 17, 18]. If one considers sequential behaviours of nets, a transition system is realised by a net *iff* it is isomorphic to the sequential reachability graph (or case graph) of this net. Ehrenfeucht and Rozenberg investigated the realisation of transition systems by elementary nets and produced an axiomatic characterisation of all realisable transition systems in terms of their regions [16, 17]. As in the existing literature about Petri net synthesis, we demonstrate that all ENI-system realisable IT-systems are characterised by suitably adapted State Separation and Forward Closure axioms.

The paper is organised as follows. In the next section, we recall basic facts about labelled partial orders and ENI-systems, including their standard interleaving semantics. Section 3 introduces our first major contribution, viz. the interval order semantics of ENI-systems which does not rely on transition splitting, and the resulting IR-graph representation. The latter is a subclass of IT-systems generating interval orders discussed in Section 4. Section 5 comprises our second main contribution, viz. a full characterisation of those IT-systems which can be synthesised to ENI-systems, and a procedure to do so. Section 6 contains concluding remarks and references to relevant literature.

2. Preliminaries

2.1. Partial orders

Labelled partial orders with domain elements representing executed actions (events) are commonly used in concurrency theory to formalise different notions of dynamic semantics.

A (*strict labelled*) *partial order* is a triple $po = \langle X, \prec, \ell \rangle$ such that X (or X_{po}) is a set, $\prec$ (or $\prec_{po}$) is a binary relation over X which is irreflexive and transitive, and ℓ (or ℓ_{po}) is a labelling for the elements of X. The *maximal* elements of po are $\max_{po} = \{x \in X \mid \neg\exists y \in X : \ x \prec y\}$. For all $x \neq y \in X$, $x \frown y$ if $x \not\prec y \not\prec x$, i.e., $\frown$ relates *unordered* elements. In this paper all partial orders are assumed to be *finite*.

The partial order is *total* whenever, for all $x \neq y \in X$, $x \prec y$ or $y \prec x$. Moreover, it is *interval* whenever, for all $x, y, z, w \in X$, if $x \prec z$ and $y \prec w$ then $x \prec w$ or $y \prec z$. The adjective 'interval' derives from the following result (c.f. [19]):

> *A partial order $\langle X, \prec, \ell \rangle$ is interval iff there are two integer-valued mappings on X, β and ε, such that, for all $x, y \in X$, $\beta(x) < \varepsilon(x)$ and $x \prec y \iff \varepsilon(x) < \beta(y)$.*

The mappings β and ε above are usually interpreted as 'beginnings of' and 'endings of' events represented by the elements of X.

As an example, consider four transactions in the distributed environment (e.g., local computations involving multiple communications by message passing): a, b, c, and d. Moreover, suppose that a precedes b and c precedes d. Suppose further that a does not precede d and c does not precede b. Then, it is possible for two messages α (from d to a) and β (from b to c) to be sent and delivered. Hence $\alpha_{snd} \prec \alpha_{rcv} \prec \beta_{snd} \prec \beta_{rcv} \prec \alpha_{snd}$, which is impossible. Hence a precedes d or c precede b, and so the precedence relationship between the four transactions is an interval order.

The relevance of interval orders follows from an observation, credited to Wiener [10], that any execution of a physical system that can be observed by a single observer must be an interval order. It implies that the most precise observational semantics should be defined in terms of interval orders (cf. [11]). In the area of concurrency theory, the use of interval orders can be traced back to [20, 21, 22, 23], and processes of concurrent systems with interval order semantics were studied in [24, 25]. Interval orders were used to investigate communication protocols in [26], using the approach of [27]. Interval semantics (ST-semantics) was investigated for Petri nets with read arcs [28] and was discussed in the context of distributability of concurrent systems in [29].

2.2. Elementary net systems with inhibitor arcs and their standard semantics

Definition 1 (ENI-system)**.** *An* elementary net system with inhibitor arcs *(or* ENI-*system) is a tuple* $eni = \langle P,T,F,I,m_0\rangle$*, where* P *and* T *are disjoint finite sets of* nodes*, called respectively* places *and* transitions*,* $F \subseteq (T\times P)\cup(P\times T)$ *is the* flow relation*,* $I \subseteq P\times T$ *is the set of* inhibitor arcs*, and* $m_0 \subseteq P$ *is the* initial marking *(in general, any subset of places is a* marking*). We denote:*

- *For a node* x*,* ${}^\bullet x = \{y \mid \langle y,x\rangle \in F\}$ *and* $x^\bullet = \{y \mid \langle x,y\rangle \in F\}$.
- *For a set of nodes* X*,* ${}^\bullet X = \bigcup\{{}^\bullet x \mid x \in X\}$ *and* $X^\bullet = \bigcup\{x^\bullet \mid x\in X\}$.
- *For every transition* t*,* ${}^\circ t = \{p \in P \mid \langle p,t\rangle \in I\}$.
- $und(eni) = \langle P,T,F,\varnothing,m_0\rangle$.

We then require that the following hold, for all transitions t *and places* p*:*

1. ${}^\bullet t \neq \varnothing \neq t^\bullet$ *and* ${}^\bullet t \cap (t^\bullet \cup {}^\circ t) = \varnothing$.
2. *There is a place* q *such that* ${}^\bullet p = q^\bullet$*,* $p^\bullet = {}^\bullet q$*, and* $p \in m_0 \iff q \notin m_0$.
3. *If* q *is a place such that* ${}^\bullet p = {}^\bullet q$ *and* $p^\bullet = q^\bullet$ *then* $p \in m_0 \iff q \in m_0$.
4. *There is no other place* q *such that* ${}^\bullet p = {}^\bullet q$*,* $p^\bullet = q^\bullet$*, and* ${}^\circ p = {}^\circ q$. ◇

Note that Definition 1(2,3,4) is included in order to simplify Definition 2. Also, $und(eni)$ can be regarded as an elementary net system (or EN-system, as defined in [12]) *underlying eni*, and some of the subsequent definitions introduced for *eni* are conservative extensions of those provided for $und(eni)$ in [12].

In diagrams, places are represented by circles, transitions by rectangles, the flow relation by directed arcs, the inhibitor arcs by edges with small circles, and a marking by small black dots drawn inside places belonging to the marking.

Example 1. *Figure 1 shows an* ENI*-system such that, e.g.,* ${}^\bullet b = \{p_1,p_4\}$*,* $c^\bullet = \{p_4,p_6\}$*, and* ${}^\circ d = \{p_4\}$. ◇

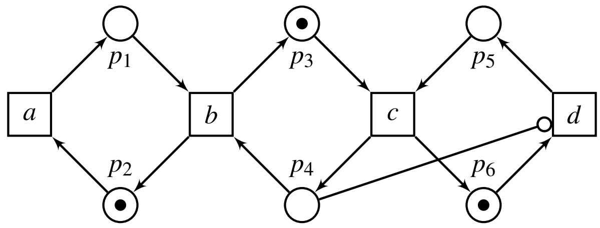

Figure 1: ENI-system.

Until Section 4.1, we assume that $eni = \langle P, T, F, I, m_0 \rangle$ is a fixed ENI-system.

The basic dynamic behaviour of ENI-systems is defined by sequences of executed transitions.

Definition 2 (firing sequences of ENI-system)**.** *The* firing sequences *of eni, denoted by* SEQ_{eni}*, are generated as follows.*

- *The empty sequence λ is a firing sequence of eni, and it leads to marking* $\mathrm{mar}_\lambda = m_0$.
- *Let σ be a firing sequence of eni leading to marking* mar_σ*, and t be a transition such that* ${}^\bullet t \subseteq \mathrm{mar}_\sigma$ *and* ${}^\circ t \cap \mathrm{mar}_\sigma = \varnothing$ *(t is enabled at* mar_σ*). Then σt is a firing sequence of eni leading to marking* $\mathrm{mar}_{\sigma t} = (\mathrm{mar}_\sigma \setminus {}^\bullet t) \cup t^\bullet$. ◇

Proposition 1. *Let* $\sigma \in \mathsf{SEQ}_{eni}$*,* $t \in T$*, and* $p, q \in P$.

1. $\mathsf{SEQ}_{eni} \subseteq \mathsf{SEQ}_{und(eni)}$.
2. *If* ${}^\bullet p = q^\bullet$ *and* $p^\bullet = {}^\bullet q$*, then* $p \in m_\sigma \iff q \notin m_\sigma$.
3. *If* ${}^\bullet t \subseteq \mathrm{mar}_\sigma$ *and* ${}^\circ t \cap \mathrm{mar}_\sigma = \varnothing$*, then* $t^\bullet \cap \mathrm{mar}_\sigma = \varnothing$.

Proof. Straightforward, from the definitions which are conservative extensions of those introduced for EN-systems, and results in [12]. □

Example 2. *Figure 1 shows an* ENI*-system where, intuitively, three components represented by cyclic sub-nets progress independently, but any action shared by two components can be executed only if both of them do so. Moreover, d can only be executed if p_4 is empty. Given a marking, a transition of an* ENI*-system is enabled if its pre-places are marked, and the post-places and inhibitor places are not marked (this follows from Definitions 1 and 2, and Proposition 1).* ◇

One can also associate labelled total orders of transition occurrences with the executed interleaving sequences of transitions. In what follows, the n-th occurrence of transition t will be denoted by $t^{(n)}$ and called *event*. The labelling function ℓ in total and partial orders in this paper maps events (occurrences of transitions) to their names.

Definition 3 (total orders of ENI-system)**.** *The* total orders *of eni, denoted by* TPO_{eni}*, are generated as follows.*

- $tpo_\varnothing = \langle \varnothing, \varnothing, \varnothing \rangle$ *is a total order of eni, and it leads to marking* $\mathrm{mar}_{tpo_\varnothing} = m_0$.

- *Let* $tpo = \langle X, \prec, \ell \rangle$ *be a total order of eni leading to marking* mar_{tpo}*, and t be a transition such that* ${}^{\bullet}t \subseteq \mathrm{mar}_{tpo}$ *and* ${}^{\circ}t \cap \mathrm{mar}_{tpo} = \varnothing$*. Then,*

$$tpo' = \langle X \cup \{x\}, \prec \cup (X \times \{x\}), \ell \cup \{\langle x, t \rangle\} \rangle$$

is a total order of eni leading to $\mathrm{mar}_{tpo'} = (\mathrm{mar}_{tpo} \setminus {}^{\bullet}t) \cup t^{\bullet}$*, where* $x = t^{(1+|\ell^{-1}(t)|)}$. ◇

Proposition 2.

1. $\mathsf{TPO}_{eni} \subseteq \mathsf{TPO}_{und(eni)}$.
2. $\{\mathrm{mar}_{tpo} \mid tpo \in \mathsf{TPO}_{eni}\} \subseteq \{\mathrm{mar}_{tpo} \mid tpo \in \mathsf{TPO}_{und(eni)}\}$.

Proof. Straightforward, from the definitions which are conservative extensions of those introduced for EN-systems, and results in [12]. □

There is a canonical way of associating a labelled total order with a finite sequence of transitions $\sigma = t_1 \dots t_k$ ($k \geq 0$), namely $\xi(\sigma) = \langle \{x_1, \dots, x_k\}, \prec, \ell \rangle$, where $x_1 \prec \cdots \prec x_k$ and, moreover, each $x_i = t_i^{(k_i)}$ is such that $\ell(x_i) = t_i$, and k_i is the number of occurrences of t_i in $t_1 \dots t_i$.

Proposition 3. ξ *induces a bijection between* SEQ_{eni} *and* TPO_{eni}.

Proof. Straightforward, from the definitions which are conservative extensions of those introduced for EN-systems, and results in [12]. □

In what follows, we will assume that each transition of an ENI-system occurs in at least one firing sequence, i.e., there are no dead transitions.

3. Interval order semantics of ENI-systems

The standard execution semantics of ENI-systems implicitly assumes that events are executed instantaneously (atomically), or that their duration is negligible. In the semantical model adopted in this paper, firing of transitions is *transaction-like*. By this we mean that when event x based on transition (action) t starts its execution, it removes tokens from all the places in ${}^{\bullet}t$, and when x ends its execution, then the tokens present in the places in $t^{\bullet}$ become available for other transitions. It is also important to note that certain relationships between net transitions will impose additional constraints on the relationships between their occurrences (events). In particular, if event z based on transition v adds a token to an inhibitor place of t, then z cannot directly precede event x (based on t), and when v removes token from an inhibitor place of t, then z must finish before x starts.

Following the approach first introduced in [12], we now define an abstract interval order semantics for *eni*. Similarly as in Definitions 2 and 3, we will use an inductive approach to define interval order semantics of ENI-systems. This leads to the following question:

> *Given an interval order execution ipo, resulting from extending the initial empty interval order by events, what could we say about the interval order obtained after starting another transition? In other words, what could we say about ipo′ derived from ipo after starting a single event x?*

Our answer is based on the following key observations:

(i) All non-maximal events in *ipo* should precede x.

(ii) The maximal events in *ipo* belong to three categories: *Cntd* comprises events which must be continued, *Fin* events which must be finished, and the remaining events may be continued or finished. More precisely,

- (a) $z \in \mathit{Cntd}$ if $\ell(z)^\bullet \cap {}^\circ t \neq \varnothing$, and then $z \frown_{ipo'} x$.
- (b) $z \in \mathit{Fin}$ if $\ell(z)^\bullet \cap {}^\bullet t \neq \varnothing$ or ${}^\bullet\ell(z) \cap {}^\circ t \neq \varnothing$, and then $z \prec_{ipo'} x$.
- (c) $z \in \max_{ipo} \setminus (\mathit{Cntd} \cup \mathit{Fin})$ otherwise, and then $z \prec_{ipo'} x$ or $z \frown_{ipo'} x$.

(iii) $\mathit{Cntd} \cap \mathit{Fin} = \varnothing$ as we cannot have both $z \prec_{ipo'} x$ and $z \frown_{ipo'} x$.

Intuitively, the maximal events in *ipo* can be considered 'pending' before starting x, and can either be finished 'just before' x started, or continued to be finished after the execution of x has started.

In case (a) above, $z \in \mathit{Cntd}$ has to continue as its finishing would put a token in a place which acts as an inhibitor for t.

In case (b) above, we have two different reasons for finishing $z \in \mathit{Fin}$. First, if $\ell(z)^\bullet \cap {}^\bullet t \neq \varnothing$, then x needs a token produced by z to start its execution (this is also required when generating interval order semantics of EN-systems in [12]). Second, if ${}^\bullet\ell(z) \cap {}^\circ t \neq \varnothing$ then there is a place p inhibiting t which is also a pre-place for $\ell(z)$. If we allowed $z \frown_{ipo'} x$, then x and z would become overlapping events and so the order of their beginnings would be undetermined in terms of the interval order semantics; in particular, x could start before z violating the inhibitor arc constraint.

In case (c) above, z can be either terminated or continued as it has no impact on the executability of x. This is, in fact, the source of non-determinism which is not present in the standard interleaving execution semantics of *eni*.

As explained in Section 6, the above treatment of interval order executions is consistent with the semantics of inhibitor arcs proposed in, e.g., [15], where transitions are split and each inhibitor arc is replaced by a pair of inhibitor arcs.

3.1. Interval orders generated by ENI-systems

Definition 4 (interval orders of ENI-system)**.** *The* interval orders *of eni, denoted by* IPO_{eni}*, are generated as follows.*

- $ipo_\varnothing = \langle \varnothing, \varnothing, \varnothing \rangle$ *is an interval order of eni, and it leads to marking* $\mathrm{mar}_{ipo_\varnothing} = m_0$.
- *Let* $ipo = \langle X, \prec, \ell \rangle$ *be an interval order of eni leading to marking* mar_{ipo}*, and* t *be a transition such that*

$$^\bullet t \subseteq \mathrm{mar}_{ipo} \setminus \ell(\mathit{Cntd})^\bullet \quad \text{and} \quad {}^\circ t \cap (\mathrm{mar}_{ipo} \setminus \ell(\mathit{Cntd})^\bullet) = \mathit{Cntd} \cap \mathit{Fin} = \varnothing\ ,$$

where

$$\begin{array}{lcl} \mathit{Cntd} & = & \{z \in \max_{ipo} \mid \ell(z)^\bullet \cap {}^\circ t \neq \varnothing\} \\ \mathit{Fin} & = & \{z \in \max_{ipo} \mid \ell(z)^\bullet \cap {}^\bullet t \neq \varnothing \ \vee\ {}^\bullet\ell(z) \cap {}^\circ t \neq \varnothing\}\ . \end{array}$$

Then

$$ipo' = \langle X \cup \{x\}, \prec \cup\, ((X \setminus \mathit{Exec}) \times \{x\}), \ell \cup \{\langle x, t \rangle\} \rangle$$

is an interval order of eni, and it leads to marking $\mathrm{mar}_{ipo'} = (\mathrm{mar}_{ipo} \setminus {}^{\bullet}t) \cup t^{\bullet}$*, where* $Cntd \subseteq Exec \subseteq \max_{ipo} \setminus Fin$ *and* $x = t^{(1+|\ell^{-1}(t)|)}$.
We also denote $ipo \rightarrow_{eni} ipo'$ *and* $ipo \xrightarrow{t:\ell(Exec)}_{eni} ipo'$. ◇

Intuitively, *Cntd* are the maximal events of *ipo* which we must 'keep' executing simultaneously with *x*, *Fin* are the maximal events of *ipo* which must be 'finished' before the start of *x*, and *Exec* are all those maximal events of *ipo* which we keep 'executing' simultaneously with *x* (and so $Cntd \subseteq Exec$ and $Fin \cap Exec = \varnothing$). The annotation of the arc, $t{:}\ell(Exec)$, means that the move from *ipo* to *ipo*′ has been achieved by executing transition *t*, while the transitions in $\ell(Exec)$ that started earlier are still active.

The nondeterministic execution of *t* results from having 2^k, where $k = |\max_{ipo} \setminus (Cntd \cup Fin)|$, possibilities for choosing *Exec*.

Definition 4 extends conservatively a corresponding definition of interval order semantics introduced for EN-systems in [12]. More precisely, inhibitor arcs only restrict the enabling of transition *t*, the next to be executed, while the rest of construction remains the same. We can therefore carry over a number of suitably adapted results established in [12].

Proposition 4. *Assume the notation as in Definition 4. Then:*

1. $t \notin \ell(\max_{ipo})$.
2. $t^{\bullet} \cap \mathrm{mar}_{ipo} = \varnothing$.
3. $\max_{ipo'} \setminus \max_{ipo} = \{x\}$.
4. $Cntd \subseteq \max_{ipo'} \setminus \{x\} \subseteq \max_{ipo} \setminus Fin$.
5. $\ell_{ipo'}(\max_{ipo'} \setminus \max_{ipo}) = \{t\}$.
6. $\ell_{ipo'}(\max_{ipo'} \setminus \{x\}) = \ell(Exec) \subseteq \ell(\max_{ipo} \setminus Fin)$.
7. *If* $x \neq y \in X$ *are such that* $\ell(x) = \ell(y)$*, then* $x \prec y$ *or* $y \prec x$.
8. *If* $ipo \xrightarrow{t:V}_{eni} ipo'$ *and* $ipo \xrightarrow{t:V}_{eni} ipo''$*, then* $ipo' = ipo''$.

Proposition 5.

1. IPO_{eni} *is a set of labelled interval orders.*
2. $\mathsf{TPO}_{eni} \subseteq \mathsf{IPO}_{eni} \subseteq \mathsf{IPO}_{und(eni)}$.
3. $\{\mathrm{mar}_{ipo} \mid ipo \in \mathsf{IPO}_{eni}\} \subseteq \{\mathrm{mar}_{\sigma} \mid \sigma \in \mathsf{SEQ}_{und(eni)}\}$.

Leading to the same marking is not enough to ensure that two generated interval orders have the same extensions. The next definition adds another requirement.

Definition 5 (extension equivalent interval orders of ENI-system)**.** *Two interval orders of eni, ipo and ipo*′*, are* extension equivalent *if* $\mathrm{mar}_{ipo} = \mathrm{mar}_{ipo'}$ *and* $\ell_{ipo}(\max_{ipo}) = \ell_{ipo'}(\max_{ipo'})$.
We denote this by $ipo \sim_{eni} ipo'$. ◇

The above relation is an equivalence relation. Moreover, the following result will be needed to define states of ENI-systems.

Proposition 6. *If* $ipo \sim_{eni} ipo'$ *and* $ipo \xrightarrow{t:V}_{eni} ipo_o$*, then there is* ipo'_o *such that* $ipo' \xrightarrow{t:V}_{eni} ipo'_o$ *and* $ipo_o \sim_{eni} ipo'_o$.

Proof. It follows directly from Definition 4. □

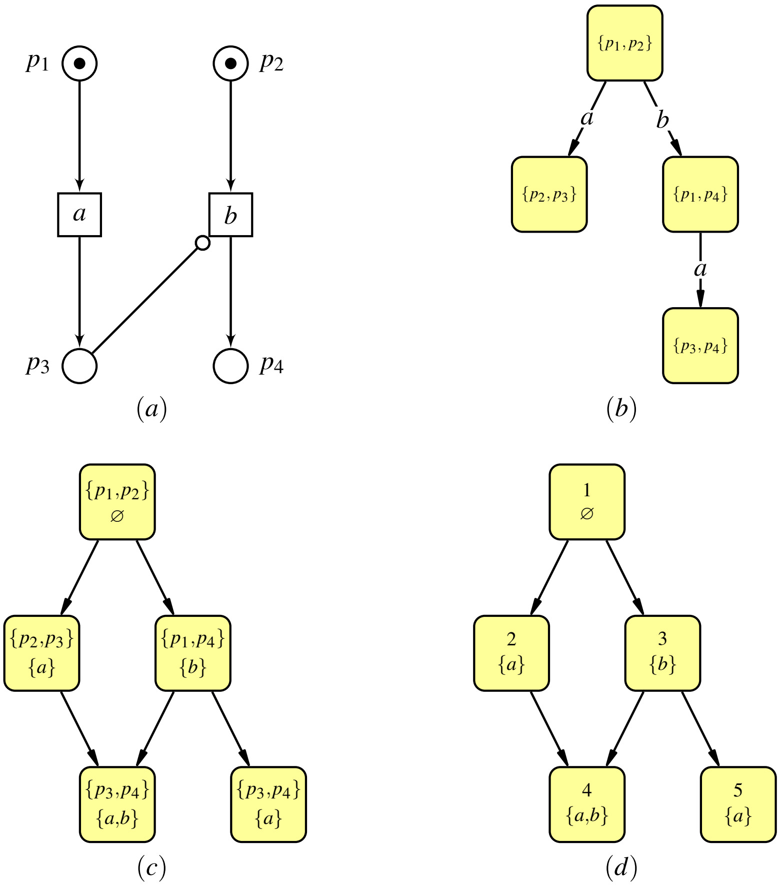

Figure 2: (a) ENI-system; (b) its interleaving reachability graph; (c) its IR-graph; and (d) an isomorphic IT-system.

3.2. Reachable states and interval reachability graphs

In the standard semantics of ENI-systems, one usually associates the notion of a 'reachable system state' with that of the marking reached after executing a firing sequence. This, in turn, leads to the notion of the reachability graph of an ENI-system. Such graphs can be seen, in particular, as generators of all the firing sequences that can be executed.

As observed in [12], markings alone are insufficient to identify states of EN-systems under the interval order semantics. A solution to this problem proposed there, and one which we adopt in the case of ENI-systems, is to associate a state of *eni* with all those interval orders which lead to the same marking, and have the same set of labels of maximal events. The reason is that all the 'continuations' for such interval orders are the same.

We then define the reachability graph of an ENI-system.

Definition 6 (interval reachability graph of ENI-system)**.** *The* interval reachability graph (or IR-graph) *of eni is* $irg_{eni} = \langle Q, \to, q_0, \iota \rangle$*, where:*

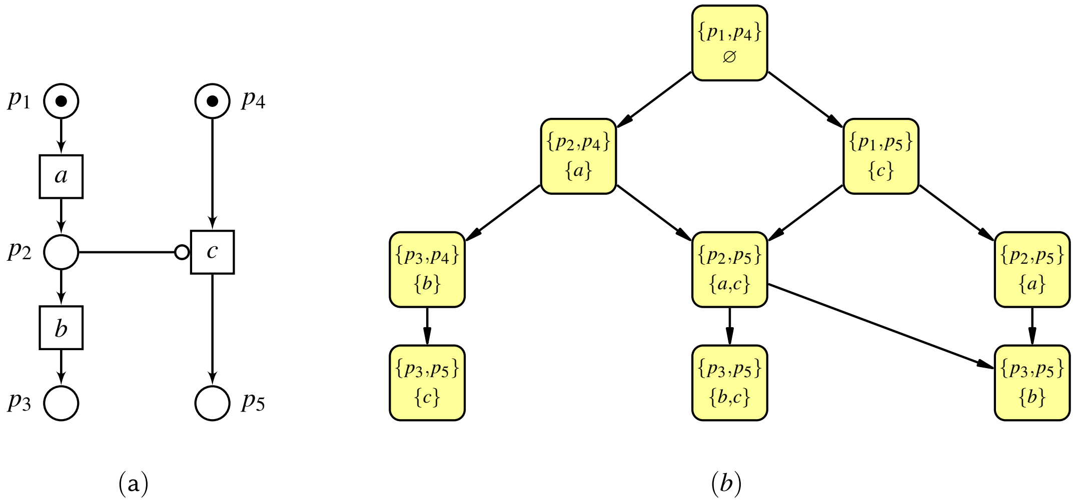

Figure 3: (*a*) ENI-system (some of the redundant places that are required by Definition 1 are omitted); and (*b*) its IR-graph.

1. $Q = \{state_{eni}(ipo) \mid ipo \in \mathsf{IPO}_{eni}\}$, *where* $state_{eni}(ipo) = \langle \mathrm{mar}_{ipo}, \ell_{ipo}(\max_{ipo})\rangle$ *is the state corresponding to* $ipo \in \mathsf{IPO}_{eni}$.
2. $\rightarrow = \{\langle state_{eni}(ipo), state_{eni}(ipo')\rangle \mid ipo \rightarrow_{eni} ipo'\}$ *are the arcs.*
3. $q_0 = state_{eni}(ipo_{\varnothing})$ *is the initial state.*
4. $\iota : Q \rightarrow 2^T$ *is the labelling such that we have* $\iota(state_{eni}(ipo)) = \ell_{ipo}(\max_{ipo})$, *for every* $ipo \in \mathsf{IPO}_{eni}$. ◇

Example 3. *Figure 2 shows both interleaving reachability graph and* IR*-graph of the* ENI*-system in Figure 2(a). In Figure 2(c), we can see that at the state* $\langle\{p_2,p_3\},\{a\}\rangle$ *transition b is enabled because* $a^{(1)} \in Cntd$ *as* $a^\bullet \cap {}^\circ b \neq \varnothing$ *and, therefore, according to Definition 4,* ${}^\bullet b \subseteq \{p_2,p_3\} \setminus \{p_3\}$ *is satisfied and* ${}^\circ b \cap (\{p_2,p_3\} \setminus \{p_3\}) = {}^\circ b \cap \{p_2\} = \varnothing$ *is also satisfied. So, b can join active event* $a^{(1)}$ *and the execution leads to the state* $\langle\{p_3,p_4\},\{a,b\}\rangle$*, where both a and b are active. Observe also that at the state* $\langle\{p_1,p_4\},\{b\}\rangle$*, where a is enabled, we have two possibilities. As* $b^{(1)} \notin Cntd$ *and* $b^{(1)} \notin Fin$*, the new transition a can join currently active b to overlap with it, or can wait for b to finish, and only then to start its execution.* ◇

Example 4. *Note that the maximal events of ipo of a state in an* IR*-graph can belong to Cntd or Fin only with respect to the new transition to be executed. This is clearly visible in the net and* IR*-graph depicted in Figure 3. For example, at the state* $\langle\{p_2,p_4\},\{a\}\rangle$ *of the* IR*-graph in Figure 3(b), two transitions are enabled, viz. b and c. At this state, if we want to execute b,* $a^{(1)} \in Fin$ $(a^\bullet \cap {}^\bullet b \neq \varnothing)$ *and therefore b must wait for a to finish in order to start its execution leading to* $\langle\{p_3,p_4\},\{b\}\rangle$*, where only b is active. However, if we want to execute c at* $\langle\{p_2,p_4\},\{a\}\rangle$*, then* $a^{(1)} \in Cntd$ $(a^\bullet \cap {}^\circ c \neq \varnothing)$ *meaning that according to Definition 4 transition c is indeed enabled at this state, but c can only be executed at* $\langle\{p_2,p_4\},\{a\}\rangle$ *by joining a. Therefore executing c at* $\langle\{p_2,p_4\},\{a\}\rangle$ *leads to* $\langle\{p_2,p_5\},\{a,c\}\rangle$*, where both a and c are active.*

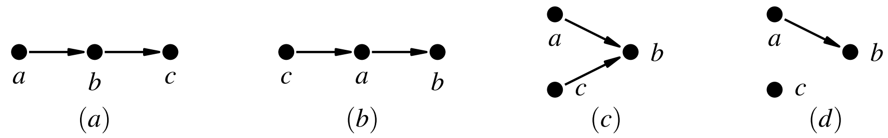

Figure 4: Interval orders generated by different paths in the transition system of Figure 3(b).

Consider then the state $\langle\{p_2, p_5\}, \{a,c\}\rangle$. At this state, b is enabled as $a^{(1)} \in Fin$ for b, and $c^{(1)} \notin Cntd \cup Fin$ for b. Hence we have two possibilities for executing b at this state: it must *wait for a to finish, but it may wait for c to finish (leading to $\langle\{p_3, p_5\}, \{b\}\rangle$) or may overlap with c (leading to $\langle\{p_3, p_5\}, \{b,c\}\rangle$).* ⋄

In the next section, we will show that irg_{eni} is a generator of all the interval orders of *eni*.

4. Transition systems generating interval orders

In general, we are interested in transition systems which are capable of generating interval orders.

Definition 7 (interval transition system)**.** *An* interval transition system over T (or IT-system) *is $its = \langle S, \to, s_0, \iota\rangle$, where S is a finite set of states, $\to \subseteq S \times S$ is the set of arcs, $s_0 \in S$ is the initial state, and $\iota : S \to 2^T$ is the labelling of states. The following hold, for all $s, r, q \in S$:*

1. *All states are reachable from s_0.*
2. *$\iota(s) = \varnothing$ iff $s = s_0$.*
3. *If $s \to r$, then there are $t \in T \setminus \iota(s)$ and $V \subseteq \iota(s)$ such that $\iota(r) = V \cup \{t\}$.*
   We also denote $s \xrightarrow{t:V} r$. Moreover, we denote $s \xrightarrow{t:V}$ (or $s \not\xrightarrow{t:V}$) if there is (resp. there is no) $r \in S$ such that $s \xrightarrow{t:V} r$.
4. *For every $t \in T$, there are $u \in S$ and $V \subseteq T$ such that $u \xrightarrow{t:V}$.*
5. *If $s \xrightarrow{t:V} r$ and $s \xrightarrow{t:V} q$, then $r = q$.* ⋄

Proposition 7. *irg_{eni} is an* IT*-system over T.*

Proof. It follows from Definitions 6 and 4, Propositions 3 and 5 as well as the assumption that each transition of an ENI-system occurs in at least one firing sequence. □

IT-systems are generators of interval orders.

Definition 8 (interval orders of IT-system)**.** *Let $its = \langle S, \to, s_0, \iota\rangle$ be an* IT*-system. Its* interval orders*, denoted by* IPO_{its}*, are the interval orders ipo_π derived from paths π originating at the initial state. They are generated as follows:*

- *$ipo_\pi = ipo_\varnothing$ is the interval order generated by $\pi = s_0$.*
- *Let $\pi = s_0 \ldots s_k$ be a path such that $ipo = ipo_{s_0 \ldots s_{k-1}} = \langle X, \prec, \ell\rangle$ and $s_{k-1} \xrightarrow{t:V} s_k$. Then*

$$ipo_\pi = \langle X \cup \{x\}, \prec \cup\, ((X \setminus Exec) \times \{x\}), \ell \cup \{\langle x, t\rangle\}\rangle$$

  is interval order generated by π, where $Exec = \max_{ipo} \cap\, \ell^{-1}(V)$ and $x = t^{(1+|\ell^{-1}(t)|)}$. ⋄

Example 5. *Figure 4 shows interval orders generated by different paths in the transition system of Figure 3(b). Every state of this transition system (*IR*-graph) is labelled by maximal elements of the interval order (associated with the state) obtained so far by progressing along a particular path. Going along the two 'outer' paths in Figure 3(b), we obtain interval orders depicted in Figure 4(a,b). The 'diamond' at the top of the* IR*-graph indicates that transitions a and c can overlap and that they can do this in more than one way, but in our semantics of* ENI*-systems, we can only express the fact that a started its execution before c, or the other way around. After the diamond, which shows the simultaneity of a and c, there are two possibilities for executing b: (i) after both a and c finished their execution (Figure 4(c)); or (ii) b can start its execution waiting only for a to finish, but joining c which is still active and can overlap with b (Figure 4(d)).* ◇

Proposition 8. $\mathsf{IPO}_{eni} = \mathsf{IPO}_{irg_{eni}}$.

Proof. Follows directly from Definitions 4 and 8 as well as Proposition 7. □

4.1. Isomorphic IT-systems

The standard definition of transition system isomorphism can be adapted for IT-systems as follows.

In this section, $its = \langle S, \to, s_0, \iota \rangle$ and $its_o = \langle S_o, \to_o, s_0^o, \iota_o \rangle$ are fixed IT-systems.

Definition 9 (isomorphism of IT-system)**.** *its and its_o are* isomorphic *if there is a bijection $\psi : S \to S_o$ such that $\psi(s_0) = s_0^o$, $\iota = \iota_o \circ \psi$, and $s \to s_o \iff \psi(s) \to_o \psi(s_o)$, for all $s, s_o \in S$. We denote this by $its \approx its_o$ (or $its \approx_\psi its_o$).* ◇

Directly from Definition 9, we have:

Proposition 9. *$\approx$ is an equivalence relation.*

IT-isomorphism is validated by the following immediate result.

Proposition 10. *$its \approx its_o$ implies* $\mathsf{IPO}_{its} = \mathsf{IPO}_{its_o}$.

The next three results included below are straightforward consequences of the fact that if in an IT-system we replace each arc $s \to r$ by a labelled arc $s \xrightarrow{t:V} r$, where $\{t\} = \iota(r) \setminus \iota(s)$ and $V = \iota(r) \setminus \{t\}$ (and remove the mapping ι), then the result is a deterministic finite state automaton such that each state is reachable from the initial state.

Proposition 11. *If $its \approx its_o$ then there is exactly one ψ such that $its \approx_\psi its_o$.*

Proposition 12. *If $its \approx_\psi its_o$ and $s \in S$, then:*

1. *$s \xrightarrow{t:V} r$ implies that there is exactly one $r_o \in S_o$ such that $\psi(s) \xrightarrow{t:V}_o r_o$; moreover, $\psi(r) = r_o$.*
2. *$\psi(s) \xrightarrow{t:V}_o r_o$ implies that there is exactly one $r \in S$ such that $s \xrightarrow{t:V} r$; moreover, $\psi(r) = r_o$.*

IT-system isomorphism can be established in a rather simple way.

Proposition 13. *Let $\psi : S \to S_o$ be an injective mapping such that $\psi(s_0) = s_0^o$, and the following hold, for all $s \in S$, $t \in T$, and $V \subseteq T$:*

- $s \xrightarrow{t:V} \implies \psi(s) \xrightarrow{t:V}_o$.
- $\psi(s) \xrightarrow{t:V}_o s'_o \implies s \xrightarrow{t:V} s'$, *for some $s' \in S$ such that $s'_o = \psi(s')$.*

Then $its \approx_\psi its_o$.

5. Synthesis

The proposed synthesis procedure will follow the approach used in, e.g., [16, 17, 18, 30, 31, 32, 7, 9], where transition systems are used as behaviour specification from which places of Petri nets are inferred in the form of regions. In our case, transitions systems are IT-systems. They are realisable by ENI-systems with interval order semantics when the IR-graphs of synthesised nets are isomorphic to the initial IT-systems.

Until Definition 11, we assume that $its = \langle S, \rightarrow, s_0, \iota \rangle$ is a fixed IT-system over T.

The regions we are going to introduce will be called *inh-regions*.

Definition 10 (inh-region of IT-system)**.** *An* inh-region *of its is* $\mathfrak{r} = \langle In_\mathfrak{r}, Out_\mathfrak{r}, Inh_\mathfrak{r}, S_\mathfrak{r} \rangle$*, where* $In_\mathfrak{r}$*,* $Out_\mathfrak{r}$*,* $Inh_\mathfrak{r} \subseteq T$*, and* $S_\mathfrak{r} \subseteq S$ *are such that the following hold, for every* $s \xrightarrow{t:V} r$*:*

1. $t \in In_\mathfrak{r}$ *iff* $s \notin S_\mathfrak{r}$ *and* $r \in S_\mathfrak{r}$*.*
2. $t \in Out_\mathfrak{r}$ *iff* $s \in S_\mathfrak{r}$ *and* $r \notin S_\mathfrak{r}$*.*
3. *If* $t \in Out_\mathfrak{r}$ *and* $v \in In_\mathfrak{r} \cap \iota(s)$*, then* $v \notin \iota(r)$*.*
4. *If* $t \in In_\mathfrak{r}$ *and* $v \in Out_\mathfrak{r} \cap \iota(s)$*, then* $v \notin \iota(r)$*.*
5. *If* $t \in Inh_\mathfrak{r}$ *and* $s \in S_\mathfrak{r}$*, then* $In_\mathfrak{r} \cap \iota(s) \neq \varnothing$*.*
6. *If* $t \in Inh_\mathfrak{r}$ *and* $v \in In_\mathfrak{r} \cap \iota(s)$*, then* $v \in \iota(r)$*.*
7. *If* $t \in Inh_\mathfrak{r}$ *and* $v \in Out_\mathfrak{r} \cap \iota(s)$*, then* $v \notin \iota(r)$*.* ◇

There are two kinds of trivial inh-regions, $\langle \varnothing, \varnothing, \varnothing, S \rangle$ and $\langle \varnothing, \varnothing, T', \varnothing \rangle$ (for $T' \subseteq T$). The set of all non-trivial inh-regions of *its* is denoted by $\mathfrak{R}_{its}$, and $\mathfrak{R}_s = \{\mathfrak{r} \in \mathfrak{R}_{its} \mid s \in S_\mathfrak{r}\}$ are the non-trivial inh-regions comprising a state $s \in S$. We also denote, for all $t \in T$ and $U \subseteq T$:

$$\begin{array}{ll} {}^\blacklozenge t = \{\mathfrak{r} \in \mathfrak{R}_{its} \mid t \in Out_\mathfrak{r}\} & {}^\blacklozenge U = \bigcup \{{}^\blacklozenge t \mid t \in U\} \\ t^\blacklozenge = \{\mathfrak{r} \in \mathfrak{R}_{its} \mid t \in In_\mathfrak{r}\} & U^\blacklozenge = \bigcup \{t^\blacklozenge \mid t \in U\} \\ {}^\lozenge t = \{\mathfrak{r} \in \mathfrak{R}_{its} \mid t \in Inh_\mathfrak{r}\} \ . & \end{array} \tag{1}$$

Proposition 14. *If* $\mathfrak{r} \in \mathfrak{R}_{its}$ *then* $\overline{\mathfrak{r}} = \langle Out_\mathfrak{r}, In_\mathfrak{r}, \varnothing, S \setminus S_\mathfrak{r} \rangle \in \mathfrak{R}_{its}$*.*

Proof. It follows from Definition 10. (Note that Definition 10(5,6,7) are trivially satisfied by $\overline{\mathfrak{r}}$.) □

The next two results relate the inh-regions involved in a transition between two states of *its*.

Proposition 15. *Let* $s \xrightarrow{t:V} r$*. Then:*

1. ${}^\blacklozenge t \cap (t^\blacklozenge \cup {}^\lozenge t) = \varnothing$*.*
2. ${}^\blacklozenge t \subseteq \mathfrak{R}_s$ *and* ${}^\blacklozenge t \cap \mathfrak{R}_r = \varnothing$*.*
3. $\mathfrak{R}_s \setminus \mathfrak{R}_r = {}^\blacklozenge t$ *and* $\mathfrak{R}_r \setminus \mathfrak{R}_s = t^\blacklozenge$*.*

Proof. (1) Suppose that $\mathfrak{r} \in {}^\blacklozenge t \cap t^\blacklozenge$. Then $t \in Out_\mathfrak{r} \cap In_\mathfrak{r}$. Hence, by Definition 10(1,2), we have $s \in S_\mathfrak{r}$ and $s \notin S_\mathfrak{r}$, yielding a contradiction.

Suppose now that $\mathfrak{r} \in {}^\blacklozenge t \cap {}^\lozenge t$. Then $t \in Out_\mathfrak{r} \cap Inh_\mathfrak{r}$. From $t \in Out_\mathfrak{r}$ and Definition 10(2), we have $s \in S_\mathfrak{r}$. Hence, as $t \in Inh_\mathfrak{r}$, by Definition 10(5) we have $In_\mathfrak{r} \cap \iota(s) \neq \varnothing$. Let $v \in In_\mathfrak{r} \cap \iota(s) \neq \varnothing$.

Then, by Definition 10(6), we have $v \in \iota(r)$. On the other hand, by $t \in Out_{\mathfrak{r}}$ and $v \in In_{\mathfrak{r}} \cap \iota(s) \neq \varnothing$, we have, from Definition 10(3), that $v \notin \iota(r)$. Hence we obtained a contradiction.

(2) It follows from Definition 10(2).

(3) We only show $\mathfrak{R}_s \setminus \mathfrak{R}_r = {}^{\blacklozenge}t$ as the second part can be shown in a similar way.

By part (2), ${}^{\blacklozenge}t \subseteq \mathfrak{R}_s \setminus \mathfrak{R}_r$. Suppose that $\mathfrak{r} \in \mathfrak{R}_s \setminus \mathfrak{R}_r$, i.e., $s \in S_{\mathfrak{r}}$ and $r \notin S_{\mathfrak{r}}$. Then, by Definition 10(2), we have $t \in Out_{\mathfrak{r}}$, and so $\mathfrak{r} \in {}^{\blacklozenge}t$. Thus, $\mathfrak{R}_s \setminus \mathfrak{R}_r \subseteq {}^{\blacklozenge}t$. $\square$

Proposition 16. *Let $s \xrightarrow{t:V}$. Then:*

1. ${}^{\blacklozenge}t \subseteq \mathfrak{R}_s \setminus cntd^{\blacklozenge}$
2. ${}^{\lozenge}t \cap (\mathfrak{R}_s \setminus cntd^{\blacklozenge}) = \varnothing$
3. $cntd \cap fin = \varnothing$
4. $cntd \subseteq V \subseteq \iota(s) \setminus fin$

where $cntd = \{v \in \iota(s) \mid v^{\blacklozenge} \cap {}^{\lozenge}t \neq \varnothing\}$ *and* $fin = \{v \in \iota(s) \mid v^{\blacklozenge} \cap {}^{\blacklozenge}t \neq \varnothing \vee {}^{\blacklozenge}v \cap {}^{\lozenge}t \neq \varnothing\}$.

Proof. Let r be such that $s \xrightarrow{t:V} r$.

(1) By Proposition 15(2), we have ${}^{\blacklozenge}t \subseteq \mathfrak{R}_s$. Suppose now, to the contrary, that there is $v \in cntd$ (i.e., $v \in \iota(s)$ and $v^{\blacklozenge} \cap {}^{\lozenge}t \neq \varnothing$) such that $v^{\blacklozenge} \cap {}^{\blacklozenge}t \neq \varnothing$. Then, by $v \in \iota(s)$, $v^{\blacklozenge} \cap {}^{\lozenge}t \neq \varnothing$ and Definition 10(6), $v \in \iota(r)$. On the other hand, by $v \in \iota(s)$, $v^{\blacklozenge} \cap {}^{\blacklozenge}t \neq \varnothing$ and Definition 10(3), $v \notin \iota(r)$. Hence we obtained a contradiction.

(2) If $\mathfrak{r} \in {}^{\lozenge}t \cap \mathfrak{R}_s$ then, by Definition 10(5), there is $v \in \iota(s)$ such that $\mathfrak{r} \in v^{\blacklozenge}$. Thus $v \in cntd$ and $\mathfrak{r} \in cntd^{\blacklozenge}$. Hence ${}^{\lozenge}t \cap \mathfrak{R}_s \subseteq cntd^{\blacklozenge}$, and so part (2) is satisfied.

(3) It follows from Definition 10(3,6,7).

(4) By Definition 10(6), $cntd \subseteq \iota(r)$. Hence, by Definition 7(3), we have $cntd \subseteq V$. Moreover, by Definition 10(3,7), we have $fin \subseteq \iota(s) \setminus \iota(r)$. Hence, by Definition 7(3), we have $V \subseteq \iota(s) \setminus fin$. $\square$

We can now provide a precise definition of all those IT-systems which can be translated into semantically equivalent ENI-systems.

Definition 11 (ENI-IT-system)**.** *its is an* ENI-IT-*system if the following hold, for all $t \in T$, $V \subseteq T$, and $s \neq r \in S$:*

1. $t^{\blacklozenge} \neq \varnothing \neq {}^{\blacklozenge}t$.
2. *If $\iota(s) = \iota(r)$, then there is $\mathfrak{r} \in \mathfrak{R}_{its}$ such that $|S_{\mathfrak{r}} \cap \{s, r\}| = 1$.* *(state separation)*
3. *If $s \stackrel{t:V}{\not\longrightarrow}$, then at least one of the following holds:* *(forward closure)*
   - $t \in \iota(s)$
   - $V \not\subseteq \iota(s)$
   - ${}^{\blacklozenge}t \not\subseteq \mathfrak{R}_s \setminus cntd^{\blacklozenge}$
   - ${}^{\lozenge}t \cap (\mathfrak{R}_s \setminus cntd^{\blacklozenge}) \neq \varnothing$
   - $cntd \cap fin \neq \varnothing$
   - $cntd \not\subseteq V$
   - $V \not\subseteq \iota(s) \setminus fin$

   where cntd and fin are as in Proposition 16. $\lozenge$

The above three 'axioms' characterise the ENI-system realisable IT-systems. 'State separation' requires that if two distinct states of *its* are not distinguished by at least one inh-region, then they are distinguished by the labels of the maximal elements of their associated interval orders. 'Forward closure' is a variation of similar axioms that can be found in the literature for solving synthesis problems, e.g., [18, 30, 31, 32, 7, 9]. Note, however, that both state separation and forward closure axioms for ENI-IT-systems differ from their standard formalisation as they do not rely only on inh-regions, but also on sets of transitions labelling the states.

Definition 12. *The tuple associated with an* ENI-IT-*system its is given by* $eni_{its} = \langle \mathfrak{R}_{its}, T, F_{its}, I_{its}, \mathfrak{R}_{s_0}\rangle$, *where:*

$$\begin{array}{rcl} F_{its} & = & \{\langle \mathfrak{r}, t\rangle \in \mathfrak{R}_{its} \times T \mid t \in Out_{\mathfrak{r}}\} \cup \{\langle t, \mathfrak{r}\rangle \in T \times \mathfrak{R}_{its} \mid t \in In_{\mathfrak{r}}\} \\ I_{its} & = & \{\langle \mathfrak{r}, t\rangle \in \mathfrak{R}_{its} \times T \mid t \in Inh_{\mathfrak{r}}\} . \end{array}$$

◇

Until the end of this section, we assume that $eni = eni_{its} = \langle \mathfrak{R}_{its}, T, F_{its}, I_{its}, \mathfrak{R}_{s_0}\rangle$ is the tuple associated with an ENI-IT-system $its = \langle S, \rightarrow, s_0, \iota\rangle$, and $irg_{eni} = \langle Q, \rightarrow_o, q_0, \iota_o\rangle$ is the IR-graph of eni_{its}. Moreover, we use the dot-notation in the context of eni_{its}, and the diamond-notation in the context of *its*.

Referring to irg_{eni} as the 'IR-graph of eni_{its}' is justified as eni_{its} is a valid ENI-system.

Proposition 17.

1. eni_{its} *is an* ENI-*system.*
2. ${}^{\bullet}t = {}^{\blacklozenge}t$, $t^{\bullet} = t^{\blacklozenge}$, *and* ${}^{\circ}t = {}^{\Diamond}t$, *for every* $t \in T$.
3. ${}^{\bullet}\mathfrak{r} = In_{\mathfrak{r}}$ *and* $\mathfrak{r}^{\bullet} = Out_{\mathfrak{r}}$, *for every* $\mathfrak{r} \in \mathfrak{R}_{its}$.

Proof. Parts (2) and (3) follow from Definition 12 and Eq.(1). With this in mind, we show part (1) in the following way:
Definition 1(1) follows from Definition 11(1) and Propositions 15(1).
Definition 1(2) follows from Propositions 14.
Definition 1(3) follows from Definitions 10(1,2) and 7(1,4) as well as the fact that the places of eni_{its} are nontrivial regions of *its*.
Definition 1(4) follows from part (3) and Definition 10. □

We then obtain the second major contribution of this paper.

Theorem 1. $its \approx_{\mathfrak{s}} irg_{eni}$, *where* $\mathfrak{s} : S \rightarrow Q$ *is such that* $\mathfrak{s}(s) = \langle \mathfrak{R}_s, \iota(s)\rangle$, *for every* $s \in S$.

Proof. (The aim is to show that Proposition 13 can be applied.)

By Definition 11(2), $\mathfrak{s}$ is an injective mapping.

By Proposition 17, *eni* is an ENI-system and, by Definition 7(1), all the states of *its* are reachable from s_0. Also all the states of irg_{eni} are reachable from q_0, which follows from the construction of irg_{eni} and the inductive approach of Definition 4. Furthermore, from Definition 6(3) we have $q_0 = \langle \mathfrak{R}_{s_0}, \varnothing\rangle$, and from Definition 7(2) we have $\iota(s_0) = \varnothing$. Hence, $\mathfrak{s}(s_0) = q_0$.

Suppose now that $s \in S$ and $q \in Q$ are such that $q = \mathfrak{s}(s)$. Then there is $ipo = \langle X, \prec, \ell\rangle \in \mathsf{IPO}_{eni}$ such that

$$q = state_{eni}(ipo) = \langle \mathrm{mar}_{ipo}, \ell_{ipo}(\mathrm{max}_{ipo})\rangle = \langle \mathfrak{R}_s, \iota(s)\rangle = \mathfrak{s}(s) .$$

Hence $\mathrm{mar}_{ipo} = \mathfrak{R}_s$ and $\ell_{ipo}(\mathrm{max}_{ipo}) = \iota(s)$. We then prove two lemmas.

Lemma 2. *If $q \xrightarrow{t:V}_o q'$, then there is $s' \in S$ such that $s \xrightarrow{t:V} s'$ and $q' = \mathfrak{s}(s')$.*

To prove the lemma, we observe that, by $q \xrightarrow{t:V}_o q'$, there is $ipo' \in \mathsf{IPO}_{eni}$ such that $ipo \xrightarrow{t:V}_{eni} ipo'$ and

$$q' = state_{eni}(ipo') = \langle \mathrm{mar}_{ipo'}, \ell_{ipo'}(\max_{ipo'})\rangle$$

and, by Definition 4, the following hold:

- ${}^\bullet t \subseteq \mathrm{mar}_{ipo} \setminus \ell(Cntd)^\bullet$
- ${}^\circ t \cap (\mathrm{mar}_{ipo} \setminus \ell(Cntd)^\bullet) = Cntd \cap Fin = \varnothing$
- $ipo' = \langle X \cup \{x\}, \prec \cup ((X \setminus Exec) \times \{x\}), \ell \cup \{\langle x, t\rangle\}\rangle$
- $\mathrm{mar}_{ipo'} = (\mathrm{mar}_{ipo} \setminus {}^\bullet t) \cup t^\bullet$
- $V = \ell(Exec)$
- $Cntd \subseteq Exec \subseteq \max_{ipo} \setminus Fin$
  where:
- $Cntd = \{z \in \max_{ipo} \mid \ell(z)^\bullet \cap {}^\circ t \neq \varnothing\}$
- $Fin = \{z \in \max_{ipo} \mid \ell(z)^\bullet \cap {}^\bullet t \neq \varnothing \vee {}^\bullet\ell(z) \cap {}^\circ t \neq \varnothing\}$
- $x = t^{(1+|\ell^{-1}(t)|)}$.

Hence, by Proposition 17, $\mathrm{mar}_{ipo} = \mathfrak{R}_s$, and $\ell_{ipo}(\max_{ipo}) = \iota(s)$, we have:

- ${}^\blacklozenge t \subseteq \mathfrak{R}_s \setminus cntd^\blacklozenge$
- ${}^\lozenge t \cap (\mathfrak{R}_s \setminus cntd^\blacklozenge) = cntd \cap fin = \varnothing$ (the second part relies also on Proposition 4(7))
- $\ell_{ipo'}(\max_{ipo'}) = V \cup \{t\}$
- $\mathrm{mar}_{ipo'} = (\mathfrak{R}_s \setminus {}^\blacklozenge t) \cup t^\blacklozenge$
- $cntd \subseteq V \subseteq \iota(s) \setminus fin$ (relies also on Proposition 4(7))
  where *cntd* and *fin* are as in Proposition 16.

Hence, by Proposition 4(1) and Definition 11, there is $s' \in S$, such that $s \xrightarrow{t:V} s'$. Then, by Propositions 15(3) and 17,

$$\mathrm{mar}_{ipo'} = (\mathrm{mar}_{ipo} \setminus {}^\bullet t) \cup t^\bullet = (\mathfrak{R}_s \setminus {}^\blacklozenge t) \cup t^\blacklozenge = \mathfrak{R}_{s'} .$$

Moreover, $\ell_{ipo'}(\max_{ipo'}) = V \cup \{t\} = \iota(s')$. Hence, $q' = state_{eni}(ipo') = \langle \mathfrak{R}_{s'}, \iota(s')\rangle = \mathfrak{s}(s')$ This concludes the proof of Lemma 2.

Lemma 3. *If $s \xrightarrow{t:V}$ then $q \xrightarrow{t:V}_o$.*

To prove the lemma, we observe that, as $s \xrightarrow{t:V}$, by Definition 7(3) and Proposition 16, we have:

- $t \notin \iota(s)$
- $V \subseteq \iota(s)$
- ${}^\blacklozenge t \subseteq \mathfrak{R}_s \setminus cntd^\blacklozenge$
- ${}^\lozenge t \cap (\mathfrak{R}_s \setminus cntd^\blacklozenge) = \varnothing$
- $cntd \subseteq V \subseteq \iota(s) \setminus fin$
- $cntd \cap fin = \varnothing$

where *cntd* and *fin* are as in Proposition 16.

Hence, by Proposition 17, $\mathrm{mar}_{ipo} = \mathfrak{R}_s$, and $\ell_{ipo}(\max_{ipo}) = \iota(s)$, we have:

- $t \notin \ell_{ipo}(\max_{ipo})$
- $V \subseteq \ell_{ipo}(\max_{ipo})$
- ${}^{\bullet}t \subseteq \mathrm{mar}_{ipo} \setminus \ell_{ipo}(Cntd)^{\bullet}$
- ${}^{\circ}t \cap (\mathrm{mar}_{ipo} \setminus \ell_{ipo}(Cntd)^{\bullet}) = \varnothing$
- $\ell_{ipo}(Cntd) \subseteq V$ (which implies $Cntd \subseteq \ell_{ipo}^{-1}(V)$)
- $V \subseteq \ell_{ipo}(\max_{ipo}) \setminus \ell_{ipo}(Fin)$ (which implies $\ell_{ipo}^{-1}(V) \subseteq \max_{ipo} \setminus Fin$)
- $\ell_{ipo}(Cntd) \cap \ell_{ipo}(Fin) = \varnothing$ (which implies $Cntd \cap Fin = \varnothing$)

  where:
- $Cntd = \{z \in \max_{ipo} \mid \ell_{ipo}(z)^{\bullet} \cap {}^{\circ}t \neq \varnothing\}$
- $Fin = \{z \in \max_{ipo} \mid \ell_{ipo}(z)^{\bullet} \cap {}^{\bullet}t \neq \varnothing \vee {}^{\bullet}\ell_{ipo}(z) \cap {}^{\circ}t \neq \varnothing\}$.

Let $Exec = \ell_{ipo}^{-1}(V) \cap \max_{ipo}$ and $x = t^{(1+|\ell_{ipo}^{-1}(t)|)}$. We then have $ipo \xrightarrow{t:V}_{eni} ipo'$, where:

$$ipo' = \langle X \cup \{x\}, \prec \cup ((X \setminus Exec) \times \{x\}), \ell_{ipo} \cup \{\langle x, t\rangle\}\rangle \ .$$

Hence $ipo \xrightarrow{t:V}_{eni}$, and so $q \xrightarrow{t:V}_{o}$. This concludes the proof of Lemma 3.

We then observe that, from $\mathfrak{s}(s_0) = q_0$ and Lemma 2 and the fact that all states in Q are reachable from q_0, it follows that $Q \subseteq \mathfrak{s}(S)$. Hence $\mathfrak{s}$ is well-defined mapping. It is also injective as *its* is an ENI-IT-system that satisfies Definition 11 (in particular, Definition 11(2)).

The theorem then follows from Proposition 13, Lemmas 2 and 3, $\mathfrak{s}(s_0) = q_0$, and the fact that $\mathfrak{s}$ is a well-defined injective mapping. □

6. Concluding remarks

As already mentioned, our treatment of interval order semantics of ENI-systems is consistent with the semantics of inhibitor arcs proposed in, e.g., [15], where transitions are split and each inhibitor arc is replaced by a pair of inhibitor arcs. Consider again the ENI-system *eni* shown in Figure 3(a). The construction in [15], depicted in Figure 5(a), replaces each transition t by two transitions, B_t (the beginning of t) and E_t (the end of t), connected by a place p_t. Moreover, the inhibitor arc between p_2 and c is replaced by two inhibitor arcs: one between p_2 and B_c, and the other between p_b and B_c. It can then be checked that all the firing sequences of the resulting ENI-system eni_{split}, from the initial marking $\{p_1, p_4\}$ to the marking $\{p_3, p_5\}$, generate interval orders which can also be generated by *eni* (the rule applied here is that $t \prec u$ iff E_t precedes B_u in the firing sequence). For example, the firing sequence $\sigma = B_aB_cE_aB_bE_cE_b$ generates the interval order in Figure 4(d) since E_a precedes B_b and this is the only relationship of this kind in σ. And, conversely, each interval order generated by *eni* can also be generated by eni_{split}. One might be tempted to remove the inhibitor arc between p_b and B_c, yielding the ENI-system eni'_{split} depicted in Figure 5(b). This however, would not work as eni'_{split} generates the firing sequence $B_aE_aB_bB_cE_cE_b$ which corresponds to the interval order such that $a \prec b$, $a \prec b$ and $b \frown c$. Such

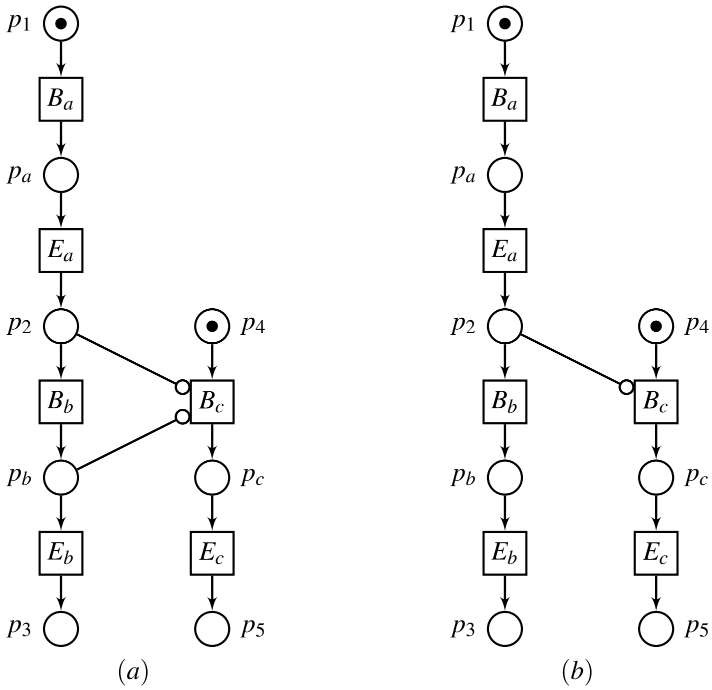

Figure 5: (*a*) ENI-system with split transitions corresponding to the ENI-system in Figure 3(*a*); and (*b*) ENI-system with split transitions which does not correspond to the ENI-system in Figure 3(*a*).

an interval order cannot be generated by eni_{split} as it corresponds also to the firing sequence $B_aE_aB_cB_bE_cE_b$ which is inconsistent with the inhibitor arc between p_b and B_c. Note finally that an underlying assumption of the approach in [15] is that if $\sigma B_t B_u \sigma'$ is a firing sequence then so is $\sigma B_u B_t \sigma'$.

There are two main contributions of this paper extending results first presented in [12]. First, we introduced a direct way of generating interval order executions of ENI-systems. We then demonstrated that such a semantics can be succintly captured using a suitable class of transition systems (ENI-IT-systems), where paths are associated with interval orders and states are labelled with the maximal elements of these interval orders. As in [12], ENI-systems are executed sequentially, but there is an assumption that every transition execution takes time and may (partially) overlap with transitions that started earlier, but have not yet finished. We then introduced and proved correct a translation from ENI-IT-systems to ENI-systems with interval order semantics generating isomorphic transition systems. In doing so, we demonstrated that one can adapt the synthesis approach based on the region concept developed for the standard transition systems to work also for the ENI-IT-systems.

It is worth mentioning that the presented approach to constructing a system from ‘interval data’ differs from other approaches pursued in the area of Petri net synthesis and process mining.

The IT-systems (including IR-graphs) *do not* directly show the temporal relationships between transitions/actions, but these relationships can be inferred from them during the synthesis procedure and become evident in the synthesised ENI-systems. However, these relationships are not as precise as in other approaches found in the literature, where systems are discovered/synthesised from behavioural information about the activities that are treated as non-instantaneous (i.e., taking some time to complete). For example, Context-Aware Temporal Network Representation (TNR) graphs of [33] that are extracted from event logs capture the global relationships between different non-instantaneous activities/actions and use 13 relationships to relate the intervals of any two activities as described by Allen's Interval Algebra [34]. In our approach, we use an abstraction that recognises only two relationships between the intervals related to two transitions, viz. one can precede the other or they can overlap.

In essence, the approaches of [33, 34] are semantically close to real-time semantics whereas the approach pursued in this paper is more abstract. For similar reasons, the interval order semantics used in this paper and the 'interval semantics' or 'interval time semantics' of, e.g., [35, 36], are incomparable.

Acknowledgments

The authors gratefully acknowledge three anonymous referees, whose comments contributed to the final version of this paper.

References

[1] J. Desel, W. Reisig, Place/transition Petri nets, in: W. Reisig, G. Rozenberg (Eds.), Lectures on Petri Nets I: Basic Models, Advances in Petri Nets, the volumes are based on the Advanced Course on Petri Nets, held in Dagstuhl, September 1996, volume 1491 of *Lecture Notes in Computer Science*, Springer, 1996, pp. 122–173.

[2] K. Jensen, Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1, Basic Concepts, Monographs in Theoretical Computer Science, Springer-Verlag, 1997.

[3] C. A. Petri, Concepts of net theory, in: Mathematical Foundations of Computer Science: Proceedings of Symposium and Summer School, Strbské Pleso, High Tatras, Czechoslovakia, September 3-8, 1973, Mathematical Institute of the Slovak Academy of Sciences, 1973, pp. 137–146.

[4] W. Reisig, Understanding Petri Nets - Modeling Techniques, Analysis Methods, Case Studies, Springer, 2013.

[5] G. Rozenberg, J. Engelfriet, Elementary net systems, in: W. Reisig, G. Rozenberg (Eds.), Petri Nets, volume 1491 of *Lecture Notes in Computer Science*, Springer, 1996, pp. 12–121.

[6] R. Janicki, M. Koutny, Semantics of inhibitor nets, Information and Computation 123 (1995) 1–16.

[7] N. Busi, G. M. Pinna, Synthesis of nets with inhibitor arcs, in: A. W. Mazurkiewicz, J. Winkowski (Eds.), CONCUR '97: Concurrency Theory, 8th International Conference, Warsaw, Poland, July 1-4, 1997, Proceedings, volume 1243 of *Lecture Notes in Computer Science*, Springer, 1997, pp. 151–165.

[8] M. Pietkiewicz-Koutny, Synthesis of eni-systems using minimal regions, in: D. Sangiorgi, R. de Simone (Eds.), CONCUR '98: Concurrency Theory, 9th International Conference,

Nice, France, September 8-11, 1998, Proceedings, volume 1466 of *Lecture Notes in Computer Science*, Springer, 1998, pp. 565–580.

[9] M. Pietkiewicz-Koutny, The synthesis problem for elementary net systems with inhibitor arcs, Fundamenta Informaticae 40 (1999) 251–283.

[10] N. Wiener, A contribution to the theory of relative position, Proc. of the Cambridge Philosophical Society 17 (1914) 441–449.

[11] R. Janicki, M. Koutny, Structure of concurrency, Theoretical Computer Science 112 (1993) 5–52.

[12] M. Pietkiewicz-Koutny, M. Koutny, Synthesising elementary net systems with interval order semantics, in: L. Gomes, P. Leitão, R. Lorenz, J. M. E. M. van der Werf, S. J. van Zelst (Eds.), Joint Proceedings of the Workshop on Algorithms & Theories for the Analysis of Event Data and the International Workshop on Petri Nets for Twin Transition co-located with the 44th International Conference on Application and Theory of Petri Nets and Concurrency (Petri Nets 2023), Caparica, Portugal, June 25-30, 2023, volume 3424 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023.

[13] E. Best, M. Koutny, Petri net semantics of priority systems, Theoretical Computer Science 96 (1992) 175–174.

[14] W. M. Zuberek, Timed Petri nets and preliminary performance evaluation, in: J. Lenfant, B. R. Borgerson, D. E. Atkins, K. B. Irani, D. Kinniment, H. Aiso (Eds.), Proceedings of the 7th Annual Symposium on Computer Architecture, La Baule, France, May 6-8, 1980, ACM, 1980, pp. 88–96.

[15] R. Janicki, X. Yin, Modeling concurrency with interval traces, Information and Computation 253 (2017) 78–108.

[16] A. Ehrenfeucht, G. Rozenberg, Theory of 2-structures, part I: clans, basic subclasses, and morphisms, Theoretical Computer Science 70 (1990) 277–303.

[17] A. Ehrenfeucht, G. Rozenberg, Theory of 2-structures, part II: representation through labeled tree families, Theoretical Computer Science 70 (1990) 305–342.

[18] É. Badouel, L. Bernardinello, P. Darondeau, Petri Net Synthesis, Texts in Theoretical Computer Science. An EATCS Series, Springer, 2015.

[19] P. C. Fishburn, Intransitive indifference with unequal indifference intervals, Journal of Mathematical Psychology 7 (1970) 144–149.

[20] L. Lamport, The mutual exclusion problem: part I - a theory of interprocess communication, Journal of ACM 33 (1986) 313–326.

[21] L. Lamport, On interprocess communication: part i: basic formalism, Distributed Computing 1 (1986) 77–85.

[22] V. R. Pratt, Modeling concurrency with partial orders, International Journal of Parallel Programming 15 (1986) 33–71.

[23] R. Janicki, M. Koutny, Structure of concurrency, Theoretical Computer Science 112 (1993) 5–52.

[24] R. Janicki, M. Koutny, Fundamentals of modelling concurrency using discrete relational structures, Acta Informatica 34 (1997) 367–388.

[25] R. Janicki, X. Yin, Modeling concurrency with interval traces, Information and Computation 253 (2017) 78–108.

[26] U. Abraham, S. Ben-David, M. Magidor, On global-time and inter-process communication,

in: M. Kwiatkowska, M. W. Shields, R. M. Thomas (Eds.), Semantics for Concurrency, Proceedings, Workshops in Computing, Springer, 1990, pp. 311–323.

[27] L. Lamport, Time, clocks, and the ordering of events in a distributed system, Communications of the ACM 21 (1978) 558–565.

[28] W. Vogler, Partial order semantics and read arcs, Theoretical Computer Science 286 (2002) 33–63.

[29] R. J. v. Glabbeek, U. Goltz, J.-W. Schicke-Uffmann, On characterising distributability, Logical Methods in Computer Science 9 (2013) 1–58.

[30] É. Badouel, L. Bernardinello, P. Darondeau, Polynomial algorithms for the synthesis of bounded nets, in: P. D. Mosses, M. Nielsen, M. I. Schwartzbach (Eds.), TAPSOFT'95: Theory and Practice of Software Development, 6th International Joint Conference CAAP/FASE, Aarhus, Denmark, May 22-26, 1995, Proceedings, volume 915 of *Lecture Notes in Computer Science*, Springer, 1995, pp. 364–378.

[31] L. Bernardinello, G. D. Michelis, K. Petruni, S. Vigna, On the synchronic structure of transition systems, in: J. Desel (Ed.), Proceedings of the International Workshop on Structures in Concurrency Theory, STRICT 1995, Berlin, Germany, May 11-13, 1995, Workshops in Computing, Springer, 1995, pp. 69–84.

[32] M. Mukund, Petri nets and step transition systems, International Journal of Foundations of Computer Science 3 (1992) 443–478.

[33] A. Senderovich, M. Weidlich, A. Gal, Context-aware temporal network representation of event logs: Model and methods for process performance analysis, Information Systems 84 (2019) 240–254.

[34] J. F. Allen, Maintaining knowledge about temporal intervals, Communications of ACM 26 (1983) 832–843.

[35] E. Pelz, Full axiomatisation of timed processes of interval-timed Petri nets, Fundamenta Informaticae 157 (2018) 427–442.

[36] L. Popova-Zeugmann, E. Pelz, Algebraical characterisation of interval-timed Petri nets with discrete delays, Fundamenta Informaticae 120 (2012) 341–357.