Interval Order Synthesis of EN-Systems with Read and Mutex Arcs

Maciej Koutny[1,*], Łukasz Mikulski[2] and 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

[2] Faculty of Mathematics and Computer Science, Nicolaus Copernicus University in Toruń, Chopina 12/18, Toruń, Poland

Abstract

Elementary net systems (EN-systems) are a fundamental Petri net model with simple markings and simple connections between places and transitions. To enhance their modelling power, a number of extensions have been proposed such as transition probabilities or inhibitor arcs. In this paper, we consider EN-systems extended with read and mutex arcs. The resulting ENRM-systems allow one to capture a wider range of computational/dynamic systems than the original EN-systems.

The standard semantical models of EN-systems and their extensions are based on sequences of executed transitions (or total orders) or sequences of sets of transitions executed simultaneously (or stratified orders). The first kind of semantics does not capture simultaneity of executed transitions, and the second kind only allows a restrictive form of (transitive) simultaneity.

In this paper, we introduce semantics of ENRM-systems based on interval (partial) orders which allows one to describe behaviours where transitions have non-atomic duration and the simultaneity of executed transitions does not need to be transitive. Assuming such a semantical model, we consider the net synthesis problem for ENRM-systems, and demonstrate that the standard notion of a region of a transition system (providing input to the synthesis procedure) can still be applied after suitable modifications.

Keywords

theory of concurrency, Petri net, elementary net system, interval order semantics, interval transition system, rm-regions, synthesis problem, structure and behaviour of Petri net

1. Introduction

Petri nets are a general model of concurrent systems which emerged as a counterpart to the state machines that were used so successfully to model sequential systems. In particular, 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]). A fundamental class of Petri nets in that respect are Elementary Net Systems (EN-systems) [4]. In this paper, we consider EN-systems extended with read and mutex arcs which allow two ways of testing for non-emptiness of places (ENRM-systems).

The execution semantics of Petri nets (i.e., the representation of individual runs or observations) is often 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 can be argued that any execution that can be observed by a single observer must be an interval order [5], where simultaneity is often non-transitive.

In this paper, extending the ideas presented in [6, 7], we first show how one can generate interval order executions of ENRM-systems in a direct way, without the need to modify the original system specification (e.g., by splitting transitions into explicit beginnings and endings) as it was done, for example, in [8, 9, 10]. The proposed semantical treatment of read arcs follows the way in which we approached the interval order semantics of inhibitor arcs in [7]. Intuitively, the latter allows a degree of 'fuzziness' when looking at the creation and consumption of tokens. When such an approach is applied to read arcs, it accepts that a token

International Workshop on Algorithms & Theories for the Analysis of Event Data, June, 2025, Paris, France
*Corresponding author.
maciej.koutny@ncl.ac.uk (M. Koutny); lukasz.mikulski@mat.umk.pl (Ł. Mikulski); marta.koutny@ncl.ac.uk (M. Pietkiewicz-Koutny)

is consumed gradually, and so the token only disappears completely after the transition connected with an ordinary arc and responsible for this is completed. This allows for overlapping of executions of two transitions connected to a shared place by a read arc and an ordinary directed arc. Dealing with mutex arcs follows a different, much stricter, policy as any overlap of two transitions when one of them is connected to a shared place by a mutex arc is prohibited. Thus, in the proposed interval semantics, one cannot simulate mutex arcs by read arcs. They cannot be simulated by self-loops either as our model excludes them. And, even if self-loops were available, the mutex arcs could not be simulated by them because self-loops and mutex arcs interact differently with read arcs as mentioned earlier. Hence, adding mutex arcs to the semantics based on interval orders increases the modelling expressiveness of the resulting model. It is also worth emphasizing that the proposed semantics is a variant of the 'a priori' approach as we capture conditions under which a transition can start its execution, but we do not stipulate when it should terminate even if there are other transitions waiting for the tokens it is supposed to produce.

We also define Interval Reachability Graphs (IR-graphs) which are finite generators of potentially infinite sets of interval orders defined by ENRM-systems. IR-graphs are a subclass of Interval Transition Systems (ITR-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 ENRM-systems, we consider the problem of synthesising ENRM-systems from given ITR-systems.

We approach the new synthesis problem using the standard synthesis approach based on the theory of regions [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]. 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. As in the existing literature about Petri net synthesis, we demonstrate that all ENRM-system realisable ITR-systems are characterised by suitably adapted State Separation and Forward Closure axioms.

Though in a majority of the existing works dealing with synthesis problem(s) it was assumed that the actions/transitions have atomic duration and are executed sequentially, there were also papers dealing with non-atomic durations and partial order executions, e.g., [6, 7, 22, 23].

The paper is organised as follows. In the next two sections, we recall basic facts about partial orders and introduce ENRM-systems. Section 4 presents our first contribution, viz. the interval order semantics of ENRM-systems which does not rely on transition splitting, and the resulting IR-graph representation. The latter is a subclass of ITR-systems generating interval orders discussed in Section 5. Section 6 comprises our second contribution, viz. a full characterisation of those ITR-systems which can be synthesised to ENRM-systems, and a procedure to do so. Section 7 concludes the paper.

2. Partial orders

Executions of concurrent systems can be represented by partial orders as acyclicity of events is a clear requirement resulting from physical considerations, and event precedence is transitive (this is true whether or not events are considered as instantaneous). Moreover, the simultaneity of events (corresponding to the overlapping of time intervals) amounts to the lack of ordering between events.

There are different partial order models of concurrent behaviours reflecting, e.g., the underlying hardware and/or software. In the literature, there are three main kinds of such models, namely the total, stratified, and interval orders, as recalled below.

Let X be a finite set (of events), $\prec$ be a binary (precedence) relation over X, and ℓ be a labelling function for X (in this paper, labels are net transitions). Then $po = \langle X, \prec, \ell\rangle$ is called (below $x,y,z,w \in X$):

- *partial order* if $x \not\prec x$ and $x \prec y \prec z \implies x \prec z$.
- *total order* if $x \not\prec x$, $x \neq y \implies x \prec y \vee y \prec x$, and $x \prec y \prec z \implies x \prec z$.
- *stratified order* if $x \not\prec x$, $x \not\prec y \not\prec x \wedge x \prec z \implies y \prec z$, $x \prec y \prec z \implies x \prec z$, and $x \not\prec y \not\prec x \wedge z \prec x \implies z \prec y$.
- *interval order* if $x \not\prec x$ and $x \prec y \wedge z \prec w \implies x \prec w \vee z \prec y$.

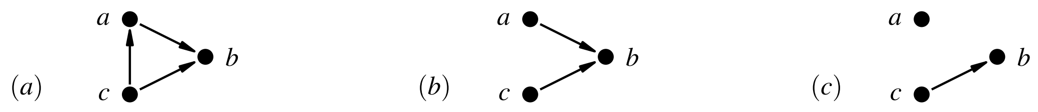

Figure 1: Diagram of (a) total order, (b) stratified order which is not total, and (c) interval order ipo_o which is not stratified.

All total orders are stratified, all stratified orders are interval, and all interval orders are partial. The *maximal* elements of $po = \langle X, \prec, \ell \rangle$ 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 $x \nprec y \nprec x$, i.e., $\frown$ relates *unordered* elements. Figure 1 shows examples of three different kinds of partial orders which are discussed in this paper.

The adjective in 'interval order' derives from an alternative definition (see [24]): *po* is an interval order if there exist real-valued mappings, β and ε, such that, for all $x, y \in X$, $\beta(x) < \varepsilon(x)$ and $x \prec y \iff \varepsilon(x) < \beta(y)$. Intuitively, β and ε represent the 'beginnings' and 'endings' of the event intervals.

The relevance of interval orders follows from an observation, credited to Wiener [5], that any execution of a physical system that can be observed by a single observer must be an interval order. Hence the most precise qualitative semantics is the one defined in terms of interval orders (cf. [25]). In the area of concurrency theory, the use of interval orders can be traced back to [26, 27, 28], and processes of concurrent systems with interval order semantics were studied in [29, 10]. Interval orders were used to investigate communication protocols in [30], using the approach of [31].

Interval order executions of, e.g., EN-systems can be generated by splitting each transition into a 'begin' followed by an 'end' transitions and, after executing the modified model using sequential semantics, deriving the corresponding interval orders. However, this can also be done directly, i.e., without splitting actions, as shown in [6].

3. ENRM-systems and their standard semantics

In this section, we present ENRM-systems and two semantical models based on total and stratified orders.

Definition 1 (ENRM-system)**.** *An* elementary net system with read and mutex arcs *(or* ENRM-*system) is a tuple $enrm = \langle P, T, F, R, M, m_0 \rangle$, 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*, $R \subseteq P \times T$ is the set of* read arcs*, $M \subseteq P \times T$ is the set of* mutex arcs*, and $m_0 \subseteq P$ is the* initial marking *(in general, any subset of places is a* marking*). For every node x and every set of nodes X, we denote:*

$$
\begin{array}{lcllcl}
{}^\bullet x & = & \{y \mid \langle y,x \rangle \in F\} & {}^\bullet X & = & \bigcup\{{}^\bullet x \mid x \in X\} \\
x^\bullet & = & \{y \mid \langle x,y \rangle \in F\} & X^\bullet & = & \bigcup\{x^\bullet \mid x \in X\} \\
{}^\odot x & = & \{y \mid \langle x,y \rangle \in R \;\vee\; \langle y,x \rangle \in R\} & {}^\odot X & = & \bigcup\{{}^\odot x \mid x \in X\} \\
{}^\otimes x & = & \{y \mid \langle x,y \rangle \in M \;\vee\; \langle y,x \rangle \in M\} & {}^\otimes X & = & \bigcup\{{}^\otimes x \mid x \in X\} \\
{}^\circledcirc x & = & {}^\bullet x \cup {}^\odot x \cup {}^\otimes x & {}^\circledcirc X & = & {}^\bullet X \cup {}^\odot X \cup {}^\otimes X \,.
\end{array}
$$

Moreover, $und(enrm) = \langle P, T, F, \varnothing, \varnothing, m_0 \rangle$. We then require, for all transitions t and places p:

1. ${}^\bullet t \neq \varnothing \neq t^\bullet$ *and* $t^\bullet \cap ({}^\bullet t \cup {}^\odot t \cup {}^\otimes t) = \varnothing$.
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 $q \neq p$ such that ${}^\bullet p = {}^\bullet q$, $p^\bullet = q^\bullet$, ${}^\odot p = {}^\odot q$, and ${}^\otimes p = {}^\otimes q$.*

The semantics of arcs $\langle x, y \rangle \in F$ in ENRM-systems is standard, i.e., as in EN-systems. The read arcs are formed when $p \in {}^\odot t$ and the mutex arcs are formed when $p \in {}^\otimes t$. Intuitively, $p \in {}^\odot t$ means that a marked p is needed for the execution of t, but such an execution does not change the marking of p. Similarly, $p \in {}^\otimes t$ means that a marked p is needed for the execution of t and the execution of t does not change the

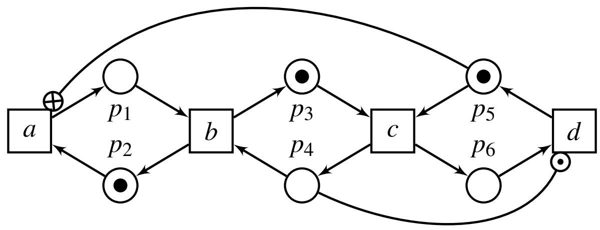

Figure 2: ENRM-system.

marking of p; moreover, in this second case, no other transition which needs a marked p for its execution can be simultaneous with t.

Note that Definition 1(2,3,4) is included in order to simplify Definition 2. Also, $und(enrm)$ can be regarded as an elementary net system (EN-system) *underlying enrm*, and some of the subsequent definitions introduced for *enrm* are conservative extensions of those provided for EN-systems in [6].

In diagrams, places are represented by circles, transitions by rectangles, the flow relation by directed arcs, the read arcs by edges with $\odot$ as an arrowhead, the mutex arcs by edges with $\otimes$ as an arrowhead, and a marking by small black dots drawn inside places belonging to the marking. Figure 2 shows an ENRM-system such that, e.g., ${}^\bullet b = \{p_1, p_4\}$, $c^\bullet = \{p_4, p_6\}$, ${}^\odot d = \{p_4\}$, and ${}^\otimes a = \{p_5\}$. Moreover, the initial marking is $m_0 = \{p_2, p_3, p_5\}$.

Until Section 5.1, we assume that $enrm = \langle P, T, F, R, M, m_0 \rangle$ is a fixed ENRM-system.

3.1. Firing sequences and total orders of ENRM-systems

The first kind of dynamic behaviour of ENRM-systems is given using sequences of executed transitions.

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

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

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

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

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

Figure 2 shows an ENRM-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 marked and a can only be executed if p_5 is marked (in addition, a cannot be simultaneous with c though this feature is not relevant for the purely sequential semantics). For example, $cdab$ is a firing sequence leading back to the initial marking.

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

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

- *$tpo_\varnothing = \langle \varnothing,\varnothing,\varnothing\rangle$ is a total order of enrm, and it leads to marking* $\mathrm{mar}_{tpo_\varnothing} = m_0$.
- *Let $tpo = \langle X,\prec,\ell\rangle$ be a total order of enrm leading to marking* mar_{tpo}*, and t be a transition such that ${}^{\circledcirc}t \subseteq \mathrm{mar}_{tpo}$. Then, $tpo' = \langle X\cup\{x\},\prec\cup(X\times\{x\}),\ell\cup\{\langle x,t\rangle\}\rangle$ is a total order of enrm leading to* $\mathrm{mar}_{tpo'} = (\mathrm{mar}_{tpo}\setminus{}^\bullet t)\cup t^\bullet$*, where $x = t^{(1+|\ell^{-1}(t)|)}$.*

Proposition 2. $\mathsf{TPO}_{enrm} \subseteq \mathsf{TPO}_{und(enrm)}$ *and* $\{\mathrm{mar}_{tpo} \mid tpo \in \mathsf{TPO}_{enrm}\} \subseteq \{\mathrm{mar}_{tpo} \mid tpo \in \mathsf{TPO}_{und(enrm)}\}$.

3.2. Step sequences and stratified orders of ENRM-systems

To capture concurrent behaviour of ENRM-systems one can use sequences of sets of executed transitions (steps).

Definition 4 (step sequences of ENRM-system)**.** *The* step sequences *of enrm, denoted by* SSEQ_{enrm}*, are generated as follows.*

- *The empty sequence λ is a step sequence of enrm, and it leads to marking* $\mathrm{mar}_\lambda = m_0$.
- *Let ξ be a step sequence of enrm leading to marking* mar_ξ*, and t be a transition such that ${}^{\circledcirc}t \subseteq \mathrm{mar}_\xi$. Then $\xi\{t\}$ is a step sequence of enrm leading to marking* $\mathrm{mar}_{\xi\{t\}} = (\mathrm{mar}_\xi\setminus{}^\bullet t)\cup t^\bullet$.
- *Let $\xi = \xi' U$ be a step sequence of enrm leading to marking* mar_ξ*, and t be a transition such that*

$$\begin{array}{llll} (1) & {}^\bullet t\cup{}^{\otimes}t \subseteq \mathrm{mar}_\xi\setminus U^\bullet & (3) & {}^{\circledcirc}t\cap{}^{\otimes}U = \varnothing \\ (2) & {}^{\odot}t \subseteq (\mathrm{mar}_\xi\setminus U^\bullet)\cup{}^\bullet U & (4) & {}^{\otimes}t\cap{}^{\circledcirc}U = \varnothing\,. \end{array}$$

Then $\xi'(U\cup\{t\})$ is a step sequence of enrm leading to marking $\mathrm{mar}_{\xi'(U\cup\{t\})} = (\mathrm{mar}_\xi\setminus{}^\bullet t)\cup t^\bullet$.

Observe that the third item of Definition 4 contains the conditions that must be satisfied by t to be added to the step U to make a bigger step, $U\cup\{t\}$, enabled at $\mathrm{mar}_{\xi'}$. Tokens from the places in ${}^\bullet U$ are already 'reserved' for transitions from U (i.e., $\mathrm{mar}_\xi\cap{}^\bullet U = \varnothing$) and tokens in the places in $U^\bullet$ are not available at $\mathrm{mar}_{\xi'}$. So, after the step sequence ξ', the pre-places of t and the mutex places of t must rely on the tokens that are not needed for the execution of any of the transitions from U to exclude conflicts for resources. This is captured by the inclusion in (1). The second inclusion, in (2), reflects the fact that the tokens from the places of ${}^\bullet U$, at the marking $\mathrm{mar}_{\xi'}$, can be used by t if it is connected to them by read arcs. Conflicts resulting from mutex arcs related to t or transitions from U are prevented by (3) and (4). Observe also that the condition (4) can be replaced by ${}^{\otimes}t\cap({}^{\otimes}U\cup{}^{\odot}U) = \varnothing$ as ${}^{\otimes}t\cap{}^\bullet U = \varnothing$ is implied by (1).

Proposition 3. *Let $\xi \in \mathsf{SSEQ}_{enrm}$, $t \in T$, and $p,q \in P$.*

1. $\mathsf{SSEQ}_{enrm} \subseteq \mathsf{SSEQ}_{und(enrm)}$.
2. *If ${}^\bullet p = q^\bullet$ and $p^\bullet = {}^\bullet q$, then $p \in \mathrm{mar}_\xi \iff q \notin \mathrm{mar}_\xi$.*
3. *If ${}^{\circledcirc}t \subseteq \mathrm{mar}_\xi$, then $t^\bullet\cap\mathrm{mar}_\xi = \varnothing$.*

One can also associate stratified orders of events with step sequences.

Definition 5 (stratified orders of ENRM-system)**.** *The* stratified orders *of enrm, denoted by* SPO_{enrm}*, are generated as follows.*

- *$spo_\varnothing = \langle\varnothing,\varnothing,\varnothing\rangle$ is a stratified order of enrm, and it leads to marking* $\mathrm{mar}_{spo_\varnothing} = m_0$.
- *Let $spo = \langle X,\prec,\ell\rangle$ be a stratified order of enrm leading to marking* mar_{spo}*, and t be a transition such that ${}^{\circledcirc}t \subseteq \mathrm{mar}_{spo}$. Then $spo' = \langle X\cup\{x\},\prec\cup(X\times\{x\}),\ell\cup\{\langle x,t\rangle\}\rangle$ is a stratified order of enrm leading to* $\mathrm{mar}_{spo'} = (\mathrm{mar}_{spo}\setminus{}^\bullet t)\cup t^\bullet$*, where $x = t^{(1+|\ell^{-1}(t)|)}$.*
- *Let $spo = \langle X,\prec,\ell\rangle \neq spo_\varnothing$ be a stratified order of enrm leading to marking* mar_{spo}*, and t be a transition such that*

$$\begin{array}{rclrcl} {}^\bullet t\cup{}^{\otimes}t & \subseteq & \mathrm{mar}_{spo}\setminus\ell(\max_{spo})^\bullet & {}^{\circledcirc}t\cap{}^{\otimes}\ell(\max_{spo}) & = & \varnothing \\ {}^{\odot}t & \subseteq & (\mathrm{mar}_{spo}\setminus\ell(\max_{spo})^\bullet)\cup{}^\bullet\ell(\max_{spo}) & {}^{\otimes}t\cap{}^{\circledcirc}\ell(\max_{spo}) & = & \varnothing\,. \end{array}$$

Then $spo' = \langle X\cup\{x\},\prec\cup((X\setminus\max_{spo})\times\{x\}),\ell\cup\{\langle x,t\rangle\}\rangle$ is a stratified order of enrm leading to $\mathrm{mar}_{spo'} = (\mathrm{mar}_{spo}\setminus{}^\bullet t)\cup t^\bullet$*, where $x = t^{(1+|\ell^{-1}(t)|)}$.*

Proposition 4. $\mathsf{SPO}_{enrm} \subseteq \mathsf{SPO}_{und(enrm)}$ *and* $\{\mathrm{mar}_{spo} \mid spo \in \mathsf{SPO}_{enrm}\} \subseteq \{\mathrm{mar}_{spo} \mid spo \in \mathsf{SPO}_{und(enrm)}\}$.

4. Interval order semantics of ENRM-systems

The above standard execution semantics of ENRM-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 consumes 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. Moreover, the read and mutex arcs impose additional constraints on the relationships between events. In particular, if event z based on transition v consumes a token from a read place of t, then z cannot directly precede event x (based on t), but when v adds a token to a read place of t, then z must finish before x starts.

Following the approach initiated in [6] and then followed in [7], we now define an interval order semantics for *enrm*. Similarly as in Definitions 3 and 5, we will use an inductive approach. This leads to the following question:

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

Our answer is based on the following key observations:

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

(ii) No maximal event in *ipo* has consumed a token that t wants to consume or exclusively reserve (i.e. $z \in \max_{ipo} \Rightarrow {}^\bullet\ell(z) \cap ({}^\bullet t \cup {}^\otimes t) = \varnothing$).

(iii) The maximal events in *ipo* belong to three categories: *Cntd* comprises events which must be continued when x starts, *Fin* events which must be finished before x starts, and any remaining events which may be continued or may be finished. More precisely,

  (a) $z \in Cntd$ if ${}^\bullet\ell(z) \cap {}^\odot t \neq \varnothing$, and then $z \frown_{ipo'} x$.

  (b) $z \in Fin$ if $(\ell(z)^\bullet \cup {}^\otimes\ell(z)) \cap {}^\circledcirc t \neq \varnothing$ or ${}^\odot\ell(z) \cap {}^\otimes t \neq \varnothing$, and then $z \prec_{ipo'} x$.

  (c) $z \in \max_{ipo} \setminus (Cntd \cup Fin)$, and then $z \prec_{ipo'} x$ or $z \frown_{ipo'} x$.

(iv) $Cntd \cap Fin = \varnothing$ 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, $z \in Cntd$ has to continue, as its finishing would finish removing a token from a place which acts as a read place for t.

In case (b) above, we have different reasons for finishing $z \in Fin$. First, if $\ell(z)^\bullet \cap {}^\circledcirc t \neq \varnothing$, then x needs a token produced by z for its execution (this is similar to the treatments in [6] and [7]). Second, if ${}^\otimes\ell(z) \cap {}^\circledcirc t \neq \varnothing$, then z cannot be simultaneous with x, and it has to finish before x starts. Third, if ${}^\odot\ell(z) \cap {}^\otimes t \neq \varnothing$ then there is a mutex place p for t which is also needed for the enabling of $\ell(z)$. If we allowed $z \frown_{ipo'} x$, then x and z would become overlapping events violating the mutex arc constraint. Hence $z \prec_{ipo'} x$.

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 or step sequence execution semantics of *enrm*.

4.1. Interval orders generated by ENRM-systems

The observations we have just made lead to the following definition.

Definition 6 (interval orders of ENRM-system)**.** *The* interval orders *of enrm, denoted by* IPO_{enrm}, *are generated as follows.*

- $ipo_\varnothing = \langle \varnothing, \varnothing, \varnothing \rangle$ *is an interval order of enrm, and it leads to marking* $\mathrm{mar}_{ipo_\varnothing} = m_0$.

- *Let* $ipo = \langle X, \prec, \ell\rangle$ *be an interval order of enrm leading to marking* mar_{ipo}, *and t be a transition such that*

$$^{\bullet}t \cup {}^{\otimes}t \subseteq \mathrm{mar}_{ipo} \setminus \ell(Cntd)^{\bullet} \;\; and \;\; ^{\odot}t \subseteq (\mathrm{mar}_{ipo} \setminus \ell(Cntd)^{\bullet}) \cup {}^{\bullet}\ell(Cntd) \;\; and \;\; Cntd \cap Fin = \varnothing \ ,$$

where $Fin = \{z \in \max_{ipo} \mid (\ell(z)^{\bullet} \cup {}^{\otimes}\ell(z)) \cap {}^{\circledcirc}t \neq \varnothing \ \vee \ {}^{\odot}\ell(z) \cap {}^{\otimes}t \neq \varnothing\}$ *and* $Cntd = \{z \in \max_{ipo} \mid {}^{\bullet}\ell(z) \cap {}^{\odot}t \neq \varnothing\}$. *Then, assuming that* $x = t^{(1+|\ell^{-1}(t)|)}$ *and* $Cntd \subseteq Exec \subseteq \max_{ipo} \setminus Fin$,

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

is an interval order of enrm leading to marking $\mathrm{mar}_{ipo'} = (\mathrm{mar}_{ipo} \setminus {}^{\bullet}t) \cup t^{\bullet}$.
We also denote $ipo \rightarrow_{enrm} ipo'$ *and* $ipo \xrightarrow{t:\ell(Exec)}_{enrm} ipo'$.

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 6 extends conservatively a corresponding definition of interval order semantics introduced for EN-systems in [6]. We can therefore carry over a number of suitably adapted results established in [6].

Proposition 5. *Assume the notation as in Definition 6. 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* $ipo \xrightarrow{t:V}_{enrm} ipo'$ *and* $ipo \xrightarrow{t:V}_{enrm} ipo''$, *then* $ipo' = ipo''$ *and* $\mathrm{mar}_{ipo'} = \mathrm{mar}_{ipo''}$.

Proposition 6. IPO_{enrm} *is a set of interval orders such that* $\mathsf{TPO}_{enrm} \subseteq \mathsf{SPO}_{enrm} \subseteq \mathsf{IPO}_{enrm} \subseteq \mathsf{IPO}_{und(enrm)}$ *and* $\{\mathrm{mar}_{ipo} \mid ipo \in \mathsf{IPO}_{enrm}\} \subseteq \{\mathrm{mar}_{ipo} \mid ipo \in \mathsf{IPO}_{und(enrm)}\}$.

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 7 (extension equivalent interval orders of ENRM-system)**.** *Two interval orders of enrm, 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 $ipo \sim_{enrm} ipo'$.

Clearly, $\sim_{enrm}$ is an equivalence relation. Moreover, the following result is needed to define states of ENRM-systems.

Proposition 7. *If* $ipo \sim_{enrm} ipo'$ *and* $ipo \xrightarrow{t:V}_{enrm} ipo_o$, *then there is* ipo'_o *such that* $ipo' \xrightarrow{t:V}_{enrm} ipo'_o$ *and* $ipo_o \sim_{enrm} ipo'_o$.

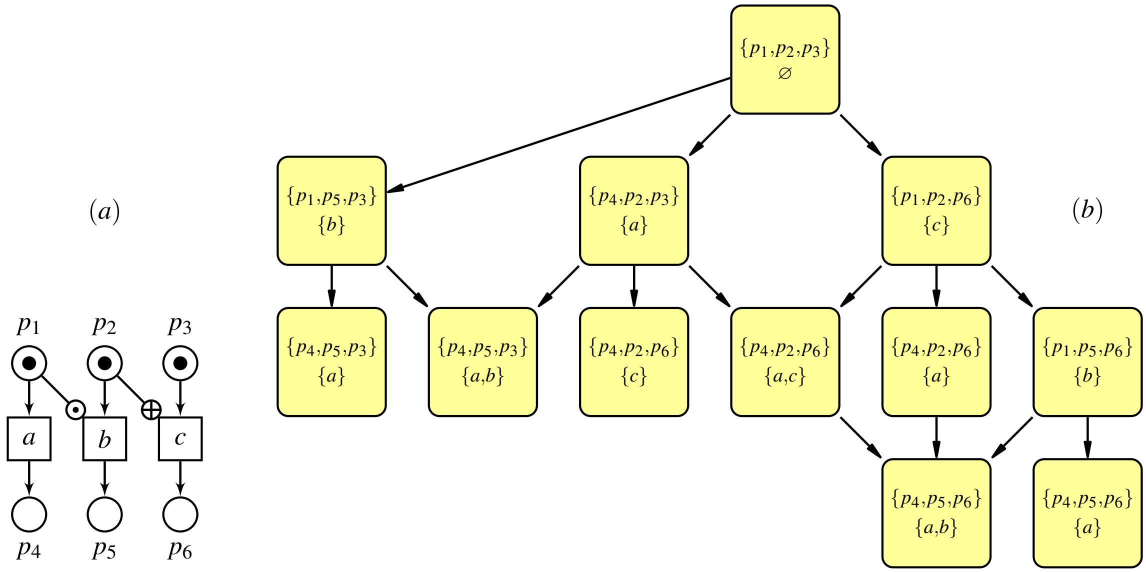

Figure 3: (a) ENRM-system and (b) its IR-graph.

4.2. Reachable states and interval reachability graphs

As observed in [6], 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 here, is to associate a state of *enrm* 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 (see Proposition 7).

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

Definition 8 (interval reachability graph of ENRM-system)**.** *The* interval reachability graph (or IR-graph) *of enrm is* $irg_{enrm} = \langle Q, \to, q_0, \iota\rangle$*, where:*

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

Figure 3(b) shows the IR-graph of *enrm*, the ENRM-system in Figure 3(a). It has 12 states given as follows, where ipo_o is the interval partial order as in Figure 1(c):

$$\begin{array}{llll}
\langle\{p_1,p_2,p_3\},\varnothing\rangle & = state_{enrm}(po_\lambda) & \langle\{p_1,p_5,p_3\},\{b\}\rangle & = state_{enrm}(po_b)\\
\langle\{p_4,p_2,p_3\},\{a\}\rangle & = state_{enrm}(po_a) & \langle\{p_1,p_2,p_6\},\{c\}\rangle & = state_{enrm}(po_c)\\
\langle\{p_4,p_5,p_3\},\{a\}\rangle & = state_{enrm}(po_{ba}) & \langle\{p_3,p_4,p_5\},\{a,b\}\rangle & = state_{enrm}(po_{\{a,b\}})\\
\langle\{p_4,p_2,p_6\},\{c\}\rangle & = state_{enrm}(po_{ac}) & \langle\{p_4,p_2,p_6\},\{a,c\}\rangle & = state_{enrm}(po_{\{a,c\}})\\
\langle\{p_4,p_2,p_6\},\{a\}\rangle & = state_{enrm}(po_{ca}) & \langle\{p_1,p_5,p_6\},\{b\}\rangle & = state_{enrm}(po_{cb})\\
\langle\{p_4,p_5,p_6\},\{a,b\}\rangle & = state_{enrm}(ipo_o) & \langle\{p_4,p_5,p_6\},\{a\}\rangle & = state_{enrm}(po_{cba}).
\end{array}$$

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 4. E.g., at the state $\langle\{p_2,p_4,p_6\},\{a\}\rangle$ of the IR-graph in Figure 4(b), transitions b, c, and d are enabled. At this state, if we want to execute b, $a^{(1)} \in \mathit{Fin}$ $(a^\bullet \cap {}^\bullet b \neq \varnothing)$ and therefore b must wait for a to finish in order to start its execution leading to $\langle\{p_3,p_4,p_6\},\{b\}\rangle$, where only b is active. However, if we want to execute c at $\langle\{p_2,p_4,p_6\},\{a\}\rangle$, then $a^{(1)} \in \mathit{Cntd}$ $({}^\bullet a \cap {}^\odot c \neq \varnothing)$ meaning that according to Definition 6

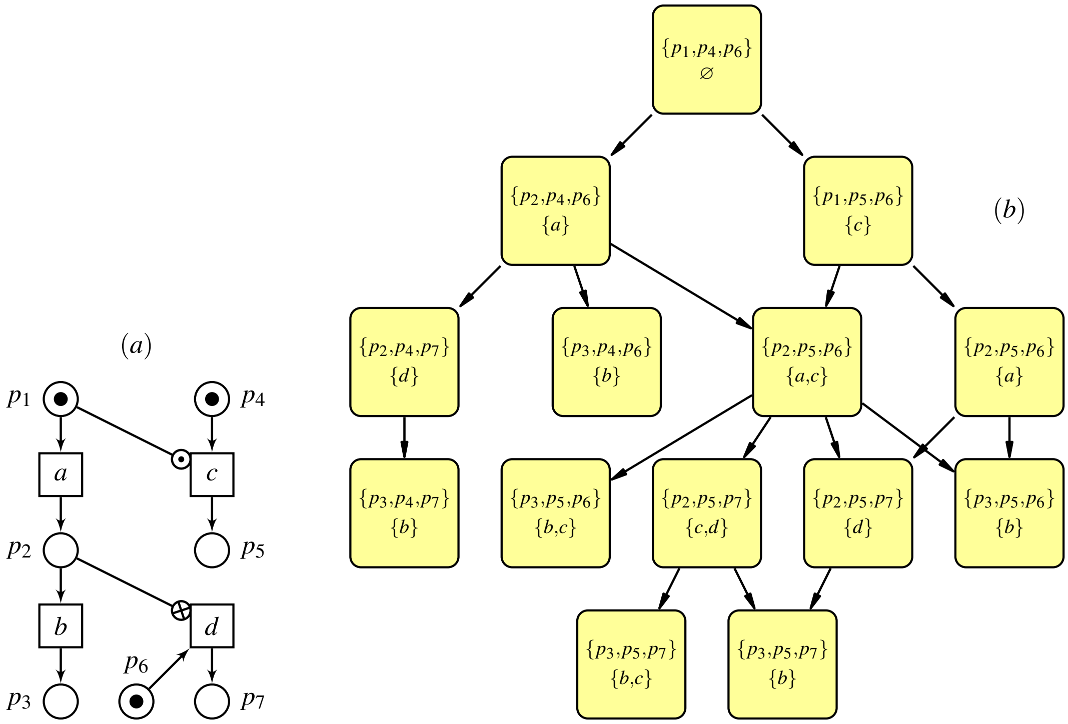

Figure 4: (a) ENRM-system (some redundant places needed by Definition 1 are omitted); and (b) its IR-graph.

transition c is indeed enabled at this state, but c can only be executed at $\langle\{p_2,p_4,p_6\},\{a\}\rangle$ by joining a. Therefore, executing c at $\langle\{p_2,p_4,p_6\},\{a\}\rangle$ leads to $\langle\{p_2,p_5,p_6\},\{a,c\}\rangle$, where both a and c are active. At the state $\langle\{p_2,p_4,p_6\},\{a\}\rangle$, d is also enabled. Its execution requires that $a^{(1)} \in Fin$ ($a^\bullet \cap {}^\otimes d \neq \varnothing$) and executing it leads to $\langle\{p_2,p_4,p_7\},\{d\}\rangle$, where only d is active.

Consider then $\langle\{p_2,p_5,p_6\},\{a,c\}\rangle$. 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: it *must* wait for a to finish, but it may wait for c to finish (leading to $\langle\{p_3,p_5,p_6\},\{b\}\rangle$) or may overlap with c (leading to $\langle\{p_3,p_5,p_6\},\{b,c\}\rangle$).

In the next section, we will show that $irg_{enrm} = \langle Q, \rightarrow, q_0, \iota\rangle$ is a generator of all the interval orders of *enrm*. For that we need yet another result and an extension to the notation of Definition 8(2), where the annotation in $\xrightarrow{t:V}$ is used explicitly for $ipo \xrightarrow{t:V}_{enrm} ipo'$.

Proposition 8. *For every* $ipo \in \mathsf{IPO}_{enrm}$:

$$\begin{array}{rcl} ipo \xrightarrow{t:V}_{enrm} ipo' & \Longrightarrow & state_{enrm}(ipo) \xrightarrow{t:V} state_{enrm}(ipo') \\ state_{enrm}(ipo) \xrightarrow{t:V} q & \Longrightarrow & \exists ipo' \in \mathsf{IPO}_{enrm} : ipo \xrightarrow{t:V}_{enrm} ipo' \wedge q = state_{enrm}(ipo') . \end{array} \tag{1}$$

5. Transition systems generating interval orders

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

Definition 9 (interval transition system)**.** *An* interval transition system over T (or ITR-system) *is* $itrs = \langle S, \rightarrow, s_0, \iota\rangle$, *where S is a finite set of states, $\rightarrow \subseteq S \times S$ is the set of arcs, $s_0 \in S$ is the initial state, and $\iota : S \rightarrow 2^T$ 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 $s \rightarrow r$, 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 $s \xrightarrow{t:V} r$.

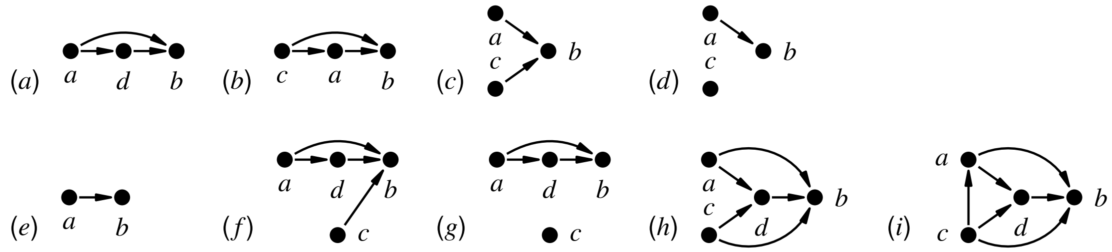

Figure 5: Interval orders generated by the maximal paths in the transition system of Figure 4(b).

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 9. *irg_{enrm} is an* ITR*-system.*

The above proposition justifies the use of the same notations for the relations and labelling functions in irg_{enrm} and *itrs* graphs.

Definition 10 (interval orders of ITR-system)**.** *Let $itrs = \langle S, \rightarrow, s_0, \iota\rangle$ be an* ITR*-system. Its* interval orders*, denoted by IPO_{itrs}, 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 $\pi = s_0 \dots s_k$ be a path such that $ipo = ipo_{s_0 \dots s_{k-1}} = \langle X, \prec, \ell\rangle$ 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 $x = t^{(1+|\ell^{-1}(t)|)}$.

Figure 5 shows interval orders generated by different paths in the transition system of Figure 4(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 4(b), we obtain interval orders depicted in Figure 5(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 ENRM-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 5(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 5(d)). Once a was executed in the net of Figure 4(a) and b consumed the delivered token from p_2, both c and d cannot be executed (Figure 5(e)). Figure 5(f) shows an interesting case. The transitions a and c of the net in Figure 4(a) can start their executions, in whatever order, and overlap. In this case, a finishes first allowing d to start and overlap with c (c, which was started before, can finish whenever it wants; the enabling conditions are checked only at the beginning of its execution). After d completes b can be executed (b and d are mutually exclusive). In the case of Figure 5(f), b also waits for c to complete, but it does not need to. Transitions c and b can overlap. This alternative is illustrated in Figure 5(g). Figure 5(h) shows a possibility of a and c overlapping in their executions at first, and then, after they both finished (c does not need to finish, but it may) d can be executed followed by b (b and d are mutually exclusive). If we execute b first, d will not have a chance to fire (Figure 5(c)). Finally, Figure 5(i) shows the possibility of sequential execution of all the transitions in the net of Figure 4(a): $cadb$.

Proposition 10. $\mathsf{IPO}_{enrm} = \mathsf{IPO}_{irg_{enrm}}$.

5.1. ITRS-isomorphism

The standard definition of transition system isomorphism can be adapted for ITR-systems as shown below, where $itrs = \langle S, \rightarrow, s_0, \iota\rangle$ and $itrs' = \langle S', \Rightarrow, s'_0, \iota'\rangle$ are fixed ITR-systems over T.

Definition 11 (isomorphism of ITR-systems)**.** *itrs and* $itrs'$ *are* isomorphic *if there is a bijection* $\psi : S \rightarrow S'$ *such that* $\psi(s_0) = s'_0$, $\iota = \iota' \circ \psi$, *and* $s \rightarrow r \iff \psi(s) \Rightarrow \psi(r)$, *for all* $s, r \in S$.
We denote this by $itrs \approx_\psi itrs'$ *and* $itrs \approx itrs'$. $\diamond$

Proposition 11. $\approx$ *is an equivalence relation such that* $itrs \approx itrs'$ *implies* $\mathsf{IPO}_{itrs} = \mathsf{IPO}_{itrs'}$.

The next result is consequence of the fact that if in an ITR-system we replace each arc $s \rightarrow r$ by a labelled arc $s \xrightarrow{t:V} r$, where $\{t\} = \iota(r) \setminus \iota(s)$ and $V = \iota(r) \cap \iota(s)$, 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 12. *If* $itrs \approx itrs'$ *then there is exactly one* ψ *such that* $itrs \approx_\psi itrs'$. *Also, if* $s \in S$ *then:*

1. $s \xrightarrow{t:V} r$ *implies that there is exactly one* $r' \in S'$ *such that* $\psi(s) \xRightarrow{t:V} r'$; *moreover,* $\psi(r) = r'$.
2. $\psi(s) \xRightarrow{t:V} r'$ *implies that there is exactly one* $r \in S$ *such that* $s \xrightarrow{t:V} r$; *moreover,* $\psi(r) = r'$.

6. Synthesis

The synthesis procedure introduced in this section follows the standard approach applied in [11, 13, 15, 18, 19, 21], where a transition system with its global states is used as an initial specification from which local states (places of Petri nets) are inferred in the form of regions. In our case, transition systems are ITR-systems. The verification that a given ITR-system is realisable by an ENRM-system with interval order semantics is essentially done by checking whether the derived regions satisfy suitable state separation and forward closure properties.

Until Definition 13, we assume that $itrs = \langle S, \rightarrow, s_0, \iota\rangle$ is a fixed ITR-system over T. The regions we are going to introduce will be called RM-*regions*.

Definition 12 (RM-region of ITR-system)**.** *An* RM-region *of itrs is* $\mathfrak{r} = \langle In_\mathfrak{r}, Out_\mathfrak{r}, Read_\mathfrak{r}, Mut_\mathfrak{r}, S_\mathfrak{r}\rangle$, *where* $In_\mathfrak{r}, Out_\mathfrak{r}, Read_\mathfrak{r}, Mut_\mathfrak{r} \subseteq T$, *and* $S_\mathfrak{r} \subseteq S$ *are such that the following hold, for all* $s \xrightarrow{t:V} r$ *and* $v \in T$:

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* $t \in Read_\mathfrak{r}$ *and* $s \notin S_\mathfrak{r}$, *then* $Out_\mathfrak{r} \cap \iota(s) \neq \varnothing$.
6. *If* $t \in Read_\mathfrak{r}$ *and* $v \in Out_\mathfrak{r} \cap \iota(s)$, *then* $v \in \iota(r)$.
7. *If* $t \in Read_\mathfrak{r}$ *and* $v \in (In_\mathfrak{r} \cup Mut_\mathfrak{r}) \cap \iota(s)$, *then* $v \notin \iota(r)$.
8. *If* $t \in Mut_\mathfrak{r}$ *and* $v \in (In_\mathfrak{r} \cup Read_\mathfrak{r} \cup Mut_\mathfrak{r}) \cap \iota(s)$, *then* $v \notin \iota(r)$.
9. *If* $t \in Out_\mathfrak{r}$ *and* $v \in Mut_\mathfrak{r} \cap \iota(s)$, *then* $v \notin \iota(r)$.
10. *If* $t \in Mut_\mathfrak{r}$ *then* $s \in S_\mathfrak{r}$.

There are two trivial RM-regions, $\langle \varnothing, \varnothing, \varnothing, \varnothing, S\rangle$ and $\langle \varnothing, \varnothing, \varnothing, \varnothing, \varnothing\rangle$. The set of all non-trivial RM-regions of *itrs* is denoted by $\mathfrak{R}_{itrs}$, and $\mathfrak{R}_s = \{\mathfrak{r} \in \mathfrak{R}_{itrs} \mid s \in S_\mathfrak{r}\}$ are the non-trivial RM-regions comprising a state $s \in S$. We also denote, for all $t \in T$ and $U \subseteq T$:

$$\begin{array}{lll} {}^{\blacksquare}t = \{\mathfrak{r} \in \mathfrak{R}_{itrs} \mid t \in Out_\mathfrak{r}\} & t^{\blacksquare} = \{\mathfrak{r} \in \mathfrak{R}_{itrs} \mid t \in In_\mathfrak{r}\} & {}^{\boxcircle}t = {}^{\blacksquare}t \cup {}^{\boxdot}t \cup {}^{\boxtimes}t \\ {}^{\boxtimes}t = \{\mathfrak{r} \in \mathfrak{R}_{itrs} \mid t \in Mut_\mathfrak{r}\} & {}^{\boxdot}t = \{\mathfrak{r} \in \mathfrak{R}_{itrs} \mid t \in Read_\mathfrak{r}\} & {}^{\boxdot}U = \bigcup\{{}^{\boxdot}t \mid t \in U\} \\ {}^{\blacksquare}U = \bigcup\{{}^{\blacksquare}t \mid t \in U\} & U^{\blacksquare} = \bigcup\{t^{\blacksquare} \mid t \in U\} & {}^{\boxtimes}U = \bigcup\{{}^{\boxtimes}t \mid t \in U\}\,. \end{array} \tag{2}$$

Proposition 13. *If* $\mathfrak{r} = \langle In_\mathfrak{r}, Out_\mathfrak{r}, Read_\mathfrak{r}, Mut_\mathfrak{r}, S_\mathfrak{r}\rangle \in \mathfrak{R}_{itrs}$ *then* $\overline{\mathfrak{r}} = \langle Out_\mathfrak{r}, In_\mathfrak{r}, \varnothing, \varnothing, S \setminus S_\mathfrak{r}\rangle \in \mathfrak{R}_{itrs}$.

To provide some intuition for Definition 12, we need to look ahead and imagine that regions will become places of the synthesised net. Then, $S_\mathfrak{r}$ is the set of states where region/place $\mathfrak{r}$ is marked or waiting to be marked. The transitions from $In_\mathfrak{r}$ will deposit tokens in $\mathfrak{r}$ (Definition 12(1)), and transitions from $Out_\mathfrak{r}$ will be consuming tokens from $\mathfrak{r}$ (Definition 12(2)). As the transition t will be executing in the context of some currently active transitions, Definition 12($3,4,7,8,9$) must make sure that some previously active transitions, say v, which share parts of their environment with t, should finish their executions before t starts, respecting the properties of places in ENRM-systems and/or the need of some of the transitions to have mutually exclusive access to the token of the place/region $\mathfrak{r}$. Definition 12(5) captures the situation where t is connected to place/region $\mathfrak{r}$ by a read arc. As t is enabled at s and $s \notin S_\mathfrak{r}$ ($\mathfrak{r}$ is not marked at s), the only possibility of t 'seeing' a token in $\mathfrak{r}$ is the existence of some other active transition from $Out_\mathfrak{r}$, say v, which is in the process of consuming this token. Definition 12(6) is then making sure that v continues its execution as t starts, so the token is not yet consumed. Finally, Definition 12(10) guarantees that if a transition t is enabled at s and is connected to $\mathfrak{r}$ by a mutex arc, then $\mathfrak{r}$ must be marked at s ($s \in S_\mathfrak{r}$) as, with respect to place $\mathfrak{r}$, t cannot be 'helped' by any other active transition connected to $\mathfrak{r}$ as it was done when $t \in Read_\mathfrak{r}$ (see Definition 12(5,6)).

The next two results relate the RM-regions involved in a move between two states of *itrs*.

Proposition 14. *Let $s \xrightarrow{t:V} r$. Then:*

$$t^{\blacksquare} \cap {}^{\boxbox}t = \varnothing \qquad {}^{\blacksquare}t \subseteq \mathfrak{R}_s \qquad {}^{\blacksquare}t \cap \mathfrak{R}_r = \varnothing \qquad \mathfrak{R}_s \setminus \mathfrak{R}_r = {}^{\blacksquare}t \qquad \mathfrak{R}_r \setminus \mathfrak{R}_s = t^{\blacksquare}\ .$$

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

$${}^{\blacksquare}t \cup {}^{\boxtimes}t \subseteq \mathfrak{R}_s \setminus cntd^{\blacksquare} \quad {}^{\boxdot}t \subseteq (\mathfrak{R}_s \setminus cntd^{\blacksquare}) \cup {}^{\blacksquare}cntd \quad cntd \cap fin = \varnothing \quad cntd \subseteq V \subseteq \iota(s) \setminus fin$$

where $cntd = \{v \in \iota(s) \mid {}^{\blacksquare}v \cap {}^{\boxdot}t \neq \varnothing\}$ and $fin = \{v \in \iota(s) \mid (v^{\blacksquare} \cup {}^{\boxtimes}v) \cap {}^{\boxbox}t \neq \varnothing \vee {}^{\boxdot}v \cap {}^{\boxtimes}t \neq \varnothing\}$.

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

Definition 13 (ENRM-ITR-system)**.** *Let $itrs = \langle S, \rightarrow, s_0, \iota\rangle$ be an* ITR-*system over T. Then itrs is an* ENRM-ITR-*system if the following hold, for all $t \in T$, $V \subseteq T$, and $s \neq r \in S$:*

1. $t^{\blacksquare} \neq \varnothing \neq {}^{\blacksquare}t$.
2. *If $\iota(s) = \iota(r)$, then there is $\mathfrak{r} \in \mathfrak{R}_{itrs}$ such that $|S_\mathfrak{r} \cap \{s, r\}| = 1$.* *(state separation)*
3. *If $s \overset{t:V}{\not\longrightarrow}$, then at least one of the following holds:* *(forward closure)*

$$\begin{array}{llll} {}^{\blacksquare}t \cup {}^{\boxtimes}t \not\subseteq \mathfrak{R}_s \setminus cntd^{\blacksquare} & t \in \iota(s) & cntd \cap fin \neq \varnothing & V \not\subseteq \iota(s) \setminus fin \\ {}^{\boxdot}t \not\subseteq (\mathfrak{R}_s \setminus cntd^{\blacksquare}) \cup {}^{\blacksquare}cntd & V \not\subseteq \iota(s) & cntd \not\subseteq V & \end{array}$$

where cntd and fin are as in Proposition 15.

The above three 'axioms' characterise the ENRM-system realisable ITR-systems. 'State separation' requires that if two distinct states of *itrs* are not distinguished by at least one RM-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., [11, 21, 15, 18, 19]. Note, however, that both state separation and forward closure axioms for ENRM-ITR-systems differ from their standard formalisation as they do not rely only on RM-regions, but also on sets of transitions labelling the states.

Definition 14. *The tuple associated with an* ENRM-ITR-*system itrs is given by $enrm_{itrs} = \langle \mathfrak{R}_{itrs}, T, F_{itrs}, R_{itrs}, M_{itrs}, \mathfrak{R}_{s_0}\rangle$, where:*

$$\begin{array}{lcl} F_{itrs} & = & \{\langle \mathfrak{r}, t\rangle \in \mathfrak{R}_{itrs} \times T \mid t \in Out_\mathfrak{r}\} \cup \{\langle t, \mathfrak{r}\rangle \in T \times \mathfrak{R}_{itrs} \mid t \in In_\mathfrak{r}\} \\ R_{itrs} & = & \{\langle \mathfrak{r}, t\rangle \in \mathfrak{R}_{itrs} \times T \mid t \in Read_\mathfrak{r}\} \\ M_{itrs} & = & \{\langle \mathfrak{r}, t\rangle \in \mathfrak{R}_{itrs} \times T \mid t \in Mut_\mathfrak{r}\}\ . \end{array}$$

Until the end of this section, we assume that $enrm = enrm_{itrs} = \langle \mathfrak{R}_{itrs}, T, F_{itrs}, R_{itrs}, M_{itrs}, \mathfrak{R}_{s_0} \rangle$ is the tuple associated with an ENRM-ITR-system $itrs = \langle S, \to, s_0, \iota \rangle$, and $irg_{enrm} = \langle Q, \to_o, q_0, \iota_o \rangle$ is the IR-graph of $enrm_{itrs}$. Moreover, we use the circle-notation in the context of $enrm_{itrs}$, and the box-notation in the context of *itrs*.

Referring to irg_{enrm} as the 'IR-graph of $enrm_{itrs}$' is justified as $enrm_{itrs}$ is a valid ENRM-system.

Proposition 16.

1. $enrm_{itrs}$ *is an* ENRM*-system.*
2. ${}^{\bullet}t = {}^{\blacksquare}t$, $t^{\bullet} = t^{\blacksquare}$, ${}^{\odot}t = {}^{\boxdot}t$, *and* ${}^{\otimes}t = {}^{\boxtimes}t$, *for every* $t \in T$.
3. ${}^{\bullet}\mathfrak{r} = In_{\mathfrak{r}}$ *and* $\mathfrak{r}^{\bullet} = Out_{\mathfrak{r}}$, *for every* $\mathfrak{r} \in \mathfrak{R}_{itrs}$.

We then obtain the second major contribution of this paper.

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

7. Concluding remarks

As already mentioned, our treatment in interval order semantics of ENRM-systems of read arcs is consistent with that of inhibitor arcs introduced in [7] (which, in turn, agrees with the semantics proposed in, e.g., [10], where transitions are split). The treatment of mutex arcs, however, does not seem to have a close predecessor. In system models, the read and mutex arcs can be used to check the availability of resources. Including both read and mutex arcs in the proposed model allowed us to express two types of conflict for resources. We have the 'soft' conflict when read and ordinary arcs are involved allowing, in some situations, more than one transition to access a shared resource for a period of time. We also can express the 'hard' conflict when mutex arcs are involved and exclusive access for checking the availability of a shared resource is required.

We expect that the theoretical concepts and results presented in this paper and its predecessors [6, 7] can provide a foundation to develop practical methods and tools for synthesising nets operating according to the interval order semantics. In particular, since many algorithms developed in the area of process mining (process discovery) were inspired by the results obtained for synthesising Petri nets from regions of the standard transition systems, we feel that a similar development is possible in the case of event logs that record events with duration. Such event logs could be derived, e.g., from records of transaction-like executions in distributed environments, where the start and finish of a transaction indicate its duration, and the overlapping of transactions is possible. Such event logs could be represented by interval orders (intuitively corresponding to paths in ITR-systems). We feel that the approach outlined in this paper can provide a line of work in the area of process discovery which is an alternative to the existing approaches pursued in, e.g., [32, 33, 34, 35, 36].

The ITR-systems (including IR-graphs) *do not* directly show all the relationships between transitions/actions, but these relationships can be inferred from them during the synthesis procedure and become evident in the synthesised ENRM-systems. One can also find other treatments of non-atomic actions in the literature. For example, Context-Aware Temporal Network Representation (TNR) graphs of [37] 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 [38]. 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. Moreover, assuming that it is not possible to observe the beginnings (or endings) of two intervals simultaneously, the relationships expressible in Allen's Interval Algebra can be embedded in the present framework using additional intervals. For example, we can express the fact that x and y overlap and x started before y started, provided that there is z such that $z \prec y$ and $z \frown x \frown y$ (recall that the relation $\frown$ does not need to be transitive). In essence, the approaches of [37, 38] 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., [39, 40], are incomparable.

Acknowledgments

This research was supported by the Leverhulme Trust funded grant RPG-2022-025. The authors are grateful to the anonymous referees, whose comments contributed to the final version of this paper.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] J. Desel, W. Reisig, Place/transition Petri nets, in: Advances in Petri Nets, 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] W. Reisig, Understanding Petri Nets - Modeling Techniques, Analysis Methods, Case Studies, Springer, 2013.

[4] G. Rozenberg, J. Engelfriet, Elementary net systems, in: Advances in Petri Nets, volume 1491 of *Lecture Notes in Computer Science*, Springer, 1996, pp. 12–121.

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

[6] M. Koutny, M. Pietkiewicz-Koutny, Synthesising elementary net systems with interval order semantics, in: ATAED/PN4TT, volume 3424 of *CEUR Workshop Proc.*, CEUR-WS.org, 2023.

[7] M. Koutny, M. Pietkiewicz-Koutny, Synthesising ENI-systems with interval order semantics, in: PNSE, volume 3730 of *CEUR Workshop Proc.*, CEUR-WS.org, 2024, pp. 33–52.

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

[9] W. M. Zuberek, Timed Petri nets and preliminary performance evaluation, in: Proc. 7th Annual Symposium on Computer Architecture, ACM, 1980, pp. 88–96.

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

[11] N. Busi, G. M. Pinna, Synthesis of nets with inhibitor arcs, in: CONCUR, volume 1243 of *Lecture Notes in Computer Science*, Springer, 1997, pp. 151–165.

[12] A. Ehrenfeucht, G. Rozenberg, Partial (set) 2-structures. part II: state spaces of concurrent systems, Acta Informatica 27 (1990) 343–368.

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

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

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

[16] E. Badouel, P. Darondeau, Theory of regions, in: Advances in Petri Nets, volume 1491 of *Lecture Notes in Computer Science*, Springer, 1996, pp. 529–586.

[17] L. Bernardinello, Synthesis of net systems, in: PETRI NETS, volume 691 of *Lecture Notes in Computer Science*, Springer, 1993, pp. 89–105.

[18] L. Bernardinello, G. D. Michelis, K. Petruni, S. Vigna, On the synchronic structure of transition systems, in: STRICT, Workshops in Computing, Springer, 1995, pp. 69–84.

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

[20] M. Nielsen, G. Rozenberg, P. S. Thiagarajan, Elementary transition systems, Theoretical Computer Science 96 (1992) 3–33.

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

[22] R. Bergenthum, Synthesizing Petri nets from Hasse diagrams, in: BPM, volume 10445 of *Lecture Notes in Computer Science*, Springer, 2017, pp. 22–39.

[23] R. Bergenthum, J. Desel, R. Lorenz, S. Mauser, Synthesis of Petri nets from finite partial languages, Fundamenta Informaticae 88 (2008) 437–468.

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

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

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

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

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

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

[30] U. Abraham, S. Ben-David, M. Magidor, On global-time and inter-process communication, in: Proc. Semantics for Concurrency, Workshops in Computing, Springer, 1990, pp. 311–323.

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

[32] J. Carmona, J. Cortadella, M. Kishinevsky, A region-based algorithm for discovering Petri nets from event logs, in: BPM, volume 5240 of *Lecture Notes in Computer Science*, Springer, 2008, pp. 358–373.

[33] J. M. E. M. van der Werf, B. F. van Dongen, C. A. J. Hurkens, A. Serebrenik, Process discovery using integer linear programming, in: PETRI NETS, volume 5062 of *Lecture Notes in Computer Science*, Springer, 2008, pp. 368–387.

[34] R. Bergenthum, Prime miner - process discovery using prime event structures, in: ICPM, IEEE, 2019, pp. 41–48.

[35] S. J. J. Leemans, D. Fahland, W. M. P. van der Aalst, Using life cycle information in process discovery, in: BPM, volume 256 of *Lecture Notes in Business Information Processing*, Springer, 2015, pp. 204–217.

[36] M. Dumas, L. García-Bañuelos, Process mining reloaded: Event structures as a unified representation of process models and event logs, in: PETRI NETS, volume 9115 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 33–48.

[37] 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.

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

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

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