Modelling Reaction Systems with Petri Nets

Jetty Kleijn, Maciej Koutny and Grzegorz Rozenberg
Modelling Reaction Systems with Petri Nets

J. Kleijn, M. Koutny and G. Rozenberg

Abstract

We investigate how Petri nets could be used to provide a faithful semantics of reaction systems, a formal framework for the investigation of processes carried by biochemical reactions. We propose and discuss possible approaches to this problem using some existing Petri net classes and concurrency concepts, such as maximal parallelism. After that we introduce a new class of Petri nets, called set-nets, which provide a computational model matching very closely that exhibited by reaction systems. The key difference between standard Petri nets and set-nets is that the former support multiset-based token arithmetic, whereas the latter support set-based operations on tokens.
Abstract

We investigate how Petri nets could be used to provide a faithful semantics of reaction systems, a formal framework for the investigation of processes carried by biochemical reactions. We propose and discuss possible approaches to this problem using some existing Petri net classes and concurrency concepts, such as maximal parallelism. After that we introduce a new class of Petri nets, called set-nets, which provide a computational model matching very closely that exhibited by reaction systems. The key difference between standard Petri nets and set-nets is that the former support multiset-based token arithmetic, whereas the latter support set-based operations on tokens.

About the authors

Jetty Kleijn is a visiting fellow within the School of Computing Science, Newcastle University.

Maciej Koutny is a Professor of Computing Science in the School of Computing Science at Newcastle University. He received his MSc (1982) and PhD (1984) in Applied Mathematics from the Warsaw University of Technology, Poland. In 1985 he joined the then Computing Laboratory of the University of Newcastle upon Tyne to work as a Research Associate. In 1986 he became a Lecturer in Computing Science at Newcastle, and from 1994 to 2000 he held an established Readership at Newcastle University. His research interests centre on the theory of distributed and concurrent systems, including both theoretical aspects of their semantics and application of formal techniques to the modelling and verification of such systems; in particular, model checking based on net unfoldings. He has also investigated non-interleaving semantics of priority systems, and the relationship between temporal logic and process algebras. Recently, he has been working on the development of a formal model combining Petri nets and process algebras as well as on Petri net based behavioural models of membrane systems.

Grzegorz Rozenberg is a Professor of Computer Science at Leiden University, The Netherlands, and an Adjunct Professor at the Department of Computer Science of University of Colorado at Boulder, U.S.A. He is the head of the Theoretical Computer Science group at Leiden Institute of Advanced Computer Science (LIACS), and the scientific director of Leiden Center for Natural Computing (LCNC). He has published more than 500 papers, 6 books, and is a (co-)editor of about 90 books. He is a member of an editorial board of about 20 journals and book series. He has been a member of the program committees and invited speaker for practically all major conferences in theoretical computer science in Europe. He is a Foreign Member of the Finnish Academy of Sciences and Letters, a member of Academia Europaea, and he is holder of Honorary Doctorates of the University of Turku, Finland, the Technical University of Berlin, Germany, and the University of Bologna, Italy. He has received the Distinguished Achievements Award of the European Association for Theoretical Computer Science "In recognition of his outstanding scientific contributions to theoretical computer science". He is a Highly Cited Researcher by ISI.

Suggested keywords

REACTION SYSTEMS
PETRI NETS
BIOCHEMISTRY
NATURAL COMPUTING
SET-NETS
MODEL TRANSLATION
Modelling Reaction Systems with Petri Nets

Jetty Kleijn\textsuperscript{1}, Maciej Koutny\textsuperscript{2}, and Grzegorz Rozenberg\textsuperscript{1,3}

\textsuperscript{1} LIACS, Leiden University
Leiden, 2300 RA The Netherlands
\{kleijn,rozenber\}@liacs.nl

\textsuperscript{2} School of Computing Science, Newcastle University
Newcastle upon Tyne, United Kingdom
maciej.koutny@ncl.ac.uk

\textsuperscript{3} Department of Computer Science, University of Colorado at Boulder
430 UCB Boulder, CO 80309-0430, U.S.A.

Abstract. We investigate how Petri nets could be used to provide a faithful semantics of reaction systems, a formal framework for the investigation of processes carried by biochemical reactions. We propose and discuss possible approaches to this problem using some existing Petri net classes and concurrency concepts, such as maximal parallelism. After that we introduce a new class of Petri nets, called SET-nets, which provide a computational model matching very closely that exhibited by reaction systems. The key difference between standard Petri nets and SET-nets is that the former support multiset-based token arithmetic, whereas the latter support set-based operations on tokens.

Keywords: reaction systems, Petri nets, biochemistry, natural computing, set-nets, model translation

1 Introduction

One of the key strands of Natural Computing — a fast growing interdisciplinary field of research — is the investigation of the computational nature of biochemical reactions. Its aim is to contribute to a computational understanding of the functioning of the living cell, which is based on interactions between (a huge number of) individual reactions. These reactions are regulated, and the main regulation mechanisms are facilitation/acceleration and inhibition/retardation. The interactions between individual reactions take place through their influence on each other, and this influence happens through these two mechanisms.

Reaction systems [5–7] are a recently proposed formal framework for the investigation of processes carried out by biochemical reactions in living cells. Thus the framework is inspired by the biochemistry of the living cell and its underlying ideas are motivated by facilitation/acceleration and inhibition/retardation, properties shared by a great number of biochemical reactions. Reaction systems constitute a computational approach inspired by nature and targeted at the investigation of ongoing dynamic changes occurring in biochemical systems through information processing. However, the model is based on principles remarkably different from those underlying other computational models in computer science.
The aim of this paper is to develop a faithful Petri net model of reaction systems. The main motivation behind this is to establish whether Petri net based concepts, such as causal processes (such as causal processes) and methods (such as synthesis of nets from a specification of their behaviour) could be used to provide analytical tools for reaction systems. It is not the intention of this paper to provide direct feedback to the area of biological applications. Rather, we are concerned here with establishing bridges between biology and Petri nets through the connection provided by reaction systems. The expected feedback is planned for our future work.

As a first step, we propose and discuss four different approaches to the modeling of reaction systems by using existing Petri net classes and concurrency concepts. However, as it turns out, in order to obtain a good match between reaction systems and Petri nets, it is necessary to re-evaluate one of the basic net principles, namely, token counting. This leads us to the introduction of a new class of Petri nets, called set-nets, which provide a net based computational model matching very closely the computations exhibited by reaction systems. The main difference between set-nets and standard Petri nets is that the latter support multiset-based token arithmetic, whereas the former support set-based (boolean) operations on tokens. Thus, the computational 'intuition' embedded in bioprocesses provides an inspiration to introduce a new class of nets with intriguing and yet to discover properties. Consequently, the main contribution of this paper is more than just providing a bridge between reaction systems and the world of Petri nets. In the future, after fully understanding and mastering the properties of the new set-nets, one would hope to provide also a new set of tools and analyses for biological applications.

The paper is organised in the following way. In the next section, we describe basic notions of reaction systems. Section 3 describes two methods of modelling reaction system using low-level Petri nets, and the next one does the same using high-level Petri nets. The new class of set-nets is introduced in Section 5, and in Section 6 we explain why this new class of nets can faithfully and elegantly model reaction systems. Comparison with related work is presented in Section 7.

We use the standard mathematical notions and notation. A multiset over a set $X$ is a function $\mu : X \to \mathbb{N} = \{0, 1, 2, \ldots\}$, and its support is $|\mu| = \{x \in X \mid \mu(x) > 0\}$. The empty multiset $\emptyset$ satisfies $|\emptyset| = \emptyset$. A multiset may be represented, somewhat informally, by listing its elements with repetitions, e.g., $\mu = \{y, y, z\}$ is such that $\mu(y) = 2$, $\mu(z) = 1$, and $\mu(x) = 0$ otherwise. We treat sets as multisets without repetitions.

## 2 Reaction systems

In this section, we explain some notions relevant to reaction systems. It is our intention to introduce enough concepts to allow one to follow the subsequent discussion on the relationship between reaction systems and Petri nets. For a comprehensive description of reaction systems, including motivations, applications and examples, the reader is referred to [5-7].
**Definition 1 (reaction system [5-7]).** A reaction system $\mathcal{A} = (S, A)$ is a pair such that $S$ is a finite background set comprising the entities of $A$, and $A$ is the set of reactions of $\mathcal{A}$. Each reaction is a triple of the form $a = (R, I, P)$, where the three components are finite non-empty sets: $R \subseteq S$ is the set of reactants, $I \subseteq S$ is the set of inhibitors, and $P \subseteq S$ is the set of products.

The components of a reaction $a = (R, I, P)$ may be denoted by $R_a$, $I_a$ and $P_a$, respectively. Definition 1 describes static aspects of a reaction system $\mathcal{A} = (S, A)$. To capture its dynamic behaviour, we need additional notions.

**Definition 2 (state of reaction system).** A state of a reaction system is any set $C$ of its entities. Then an initialised reaction system $\mathcal{A} = (S, A, C_0)$ is a triple such that $(S, A)$ is a reaction system and $C_0 \subseteq S$ is the initial state.

In this and in the next section, we will consider as an example the initialised reaction system $\mathcal{A}_0 = (\{w, x, y, z\}, \{a, b, c\}, \{x, z\})$, with background set $\{w, x, y, z\}$, the initial state $\{x, z\}$, and three reactions:

$$a = (\{x\}, \{y\}, \{y, z\}) \quad b = (\{y\}, \{x\}, \{x, z\}) \quad c = (\{z\}, \{w\}, \{z\}).$$

A reaction system $\mathcal{A} = (S, A)$ with background set $S$ has exactly $2^{|S|}$ potential states. To describe possible moves between these states, we need to say first when a reaction is enabled.

**Definition 3 (state change).** A reaction $a$ is enabled at a state $C \subseteq S$ if $R_a \subseteq C$ and $I_a \cap C = \emptyset$; the result of a reaction $a$ at $C$ is defined by $\text{res}_a(C) = P_a$ if $a$ is enabled at $C$ and $\text{res}_a(C) = \emptyset$ otherwise. The result of $A$ on $C$, denoted by $\text{res}_A(C)$, consists of the products of all reactions from $A$ enabled at $C$, that is

$$\text{res}_A(C) = \bigcup_{a \in A} \text{res}_a(C).$$

This state change is denoted by $C \rightarrow \text{res}_A(C)$.

Note that the state changes captured by Definition 3 are deterministic. Moreover, all entities in $C \setminus \bigcup_{a \in A} \text{res}_a(C)$ disappear. As a result, and unlike in other formal models of dynamic systems, there is no persistency in a reaction system in the sense that an entity present in a state disappears unless it is sustained by at least one reaction.

For the example reaction system $\mathcal{A}_0$, we have:

$$\{x, z\} \rightarrow \{y, z\} \text{ and } \{y, z\} \rightarrow \{x, z\} \text{ and } \{w, x, y\} \rightarrow \emptyset.$$

In addition to the above, we may say that a set $\mathcal{R} \subseteq A$ is enabled at $C$ if each reaction of $\mathcal{R}$ is enabled at $C$. If $\mathcal{R} \subseteq A$ is enabled at $C$, then

$$C \xrightarrow{\mathcal{R}} \text{res}_\mathcal{R}(C) = \bigcup_{a \in \mathcal{R}} P_a.$$
denotes the effect of $R$ at $C$.

One may observe that there is no conflict between reactions in the sense that the occurrence of one reaction might imply that another reaction which is also enabled at the current state, cannot occur. This, again, is a feature not found in most other formal models of dynamic systems. In particular, it is worthwhile to point explicitly to the `non-counting’ features of reaction systems: entities are either present or not, and produced or not, and reactions can or cannot occur given the presence or absence of certain entities. There is no representation of multiple instances of entities or multiple occurrences of reactions. We also note that there is an alternative notion of conflict-freeness for a set of reactions, called consistency. A set of reactions $R$ is consistent if for any two reactions $a, b \in R$, $R_a \cap I_b = R_b \cap I_a = \emptyset$. Clearly, if a set of reactions is not consistent, then the reactions it comprise cannot be executed simultaneously.

In general, reaction systems may have an environment and then operate within a changing context which leads to the concept of an interactive process. Such an process is defined by a reaction system with an initial state, where each subsequent state is obtained as the result of reactions transforming the previous state together with possible additional entities coming from the environment.

**Definition 4 (interactive process).** An interactive process in a reaction system $A = (S, A)$ is a pair $\pi = (\gamma, \delta)$ of finite sequences $\gamma = C_0 \ldots C_n$ (called context sequence) and $\gamma = D_0 \ldots D_n$ (called result sequence), with $n \geq 1$, such that $C_0, \ldots, C_n, D_0, \ldots, D_n \subseteq S$, $D_0 = \emptyset$, and:

$$D_j = \text{res}_A(D_{j-1} \cup C_{j-1}),$$

for all $j \in \{1, \ldots, n\}$. (See Figure 1 for an illustration.)

\[\square\]

![Fig. 1. An interactive process where $D_j = \text{res}_A(D_{j-1} \cup C_{j-1})$.](image)

Given an interactive process $\pi$ as in Definition 4, the sequence $(W_0, \ldots, W_n)$ with $W_j = C_j \cup D_j$ for all $j \in \{0, \ldots, n\}$, is a state sequence of $A$. Moreover, if $C_j \subseteq D_j$, for each $j \in \{1, \ldots, n\}$, then $\pi$ is a context-independent process of $A$, and the corresponding state sequence is context-independent. Clearly, if $\pi$ is
context-independent, then we can assume that $C_1 = \ldots = C_n = \emptyset$ (the resulting state sequence will not change). If $\mathcal{A}$ is an initialised reaction system, we assume that $C_0$ is its initial state. The state sequences and context-independent state sequences of $\mathcal{A}$ are respectively denoted by $STS(\mathcal{A})$ and $CISTS(\mathcal{A})$. For the reaction system $\mathcal{A}_0$, the sequences $\langle \{x, z\} \rangle$ and $\langle \{x, z\}, \{y, z\}, \{x, z\} \rangle$ are examples of context-independent state sequences. It is thus easy to see that $CISTS(\mathcal{A}_0)$ comprises all sequences that are a prefix of $\{x, z\}, \{y, z\}, \ldots$

In this paper, we are concerned only with context-independent processes of initialised reaction systems, aiming at a faithful representation of their dynamic behaviours using Petri nets.

### 3 Reaction systems and low-level Petri nets

In this section, we discuss two possible ways of modelling context-independent processes of reaction systems using low-level Petri nets (\textit{PT}-nets with inhibitor and activator arcs). First, however, we provide the necessary formal definitions which will be used throughout the paper.

**Definition 5 (\textit{PT}-nets with inhibitor and activator arcs [11]).** A \textit{PT}-net with inhibitor and activator arcs (or \textit{\textit{PT}IAA}-net) $N = (Pl, Tr, Flw, Inh, Act, M_0)$ is a tuple such that $Pl$ and $Tr$ are finite, disjoint sets of respectively places and transitions, and: $Flw \subseteq (Pl \times Tr) \cup (Tr \times Pl)$, $Inh \subseteq Pl \times Tr$, $Act \subseteq Pl \times Tr$ are respectively the sets of flow, inhibitor and activator arcs. Moreover, $M_0$ is a multiset of places, the initial marking of $N$; in general, any multiset of places is called a marking.

In diagrams, places are drawn as circles and transitions as rectangles. Markings are the possible global configurations (states) of $N$. We say that a place $q$ is \textit{marked} under a marking $M$ if $M(q) > 0$. In diagrams, markings are indicated by putting $M(q)$ \textit{tokens} inside the circle representing $q$. If $(x, y) \in Flw$, then $(x, y)$ is an arc leading from \textit{node} $x$ to \textit{node} $y$. A double headed arrow between $q$ and $t$ indicates that $(q, t), (t, q) \in Flw$. An inhibitor arc ends with a small open circle, while an activator arc ends with a small black circle.

Given a node $x$, we denote by $\cdot x$ the set of \textit{input nodes} of $x$, i.e., those $y$ for which $(y, x) \in Flw$, and by $x^\cdot$ the set of \textit{output nodes} of $x$, i.e., those $y$ for which $(x, y) \in Flw$. For a transition $t$ we use $^\circ t = \{ q \mid (q, t) \in Inh \}$ and $^\cdot t = \{ q \mid (q, t) \in Act \}$ to respectively denote the inhibitor and activator places of $t$. All four notations extend in the usual way to sets of nodes. As in the case of reaction systems, we now formalise the notion of marking (state) change.

**Definition 6 (marking change).** A multiset of transitions $U$ (also called a \textit{step}) is enabled at a marking $M$ if $^\circ U \cap \|M\| = \emptyset$, $^\cdot U \subseteq \|M\|$ and, for every place $q$, $M(q) \geq \sum_{t \in q^\cdot} U(t)$. In such a case, $U$ can be fired with its effect on $M$ being given by the resulting marking $M'$ such that, for every place $q$:

$$M'(q) = M(q) - \sum_{t \in q^\cdot} U(t) + \sum_{t \in q^\circ} U(t).$$
We denote this by \( M[U]M' \). Moreover, if \( U \) is a maximal (w.r.t. multiset inclusion) step of transitions enabled at \( M \), then we may denote this marking change also by \( M[U]_{max}M' \).

Note that whenever a step \( U \) is enabled at marking \( M \) it must be the case that all activator places of transitions in \( |U| \) are marked (are in \( |M| \)) and none of the inhibitor places of transitions in \( |U| \) are marked.

Before proceeding with the task of defining a formal model of reaction systems using PTIA-nets, we make some general observations and assumptions about the relationship between reaction systems and nets.

- Entities can be represented by places, and reactions by net transitions.
- Since there are no conflicts between reactions, activator arcs can be used to test for the presence of reactants (rather than claiming resources for the exclusive use as with ordinary arcs and input places).
- All reactions that can occur in a reaction system at a given state do occur, and the only entities left after a state change are the newly generated products. In the Petri net framework, these features correspond to maximal parallelism described at the end of Definition 6, and place resetting [4] described later on.

**Method I.** The first attempt is illustrated in Figure 2(a) for the example reaction system \( \mathcal{A}_0 \). Method I produces a PTIA-net \( N_I(\mathcal{A}_0) \) such that:

- Transitions \( a, b \) and \( c \) use activator arcs and inhibitor arcs to test respectively for the presence and absence of tokens in the places \( w, x, y \) and \( z \).
- Places \( q_a, q_b \) and \( q_c \) ensure that the three transitions modelling reactions, i.e., \( a, b \) and \( c \), fire at most once in any step. This corresponds to the ‘non-counting’ of occurrence instances of the same reaction in a reaction system.
- Transitions \( r_x, r_y \) and \( r_z \) (in a maximal step) empty the four places modelling entities \( w, x, y \) and \( z \). This does not have any influence on the firing of the transitions \( a, b \) and \( c \).
- In a single maximal step, \( M[U]_{max}M' \), the net fires a maximal multiset of transitions \( U \) enabled at marking \( M \) and then produces a new marking \( M' \).

For the net in Figure 2(a), such a firing rule gives:

\[
  \{x, z, q_a, q_b, q_c\} \rightarrow_{\text{max}} \{y, z, q_a, q_b, q_c\} \\
  \{x, x, z, q_a, q_b, q_c\} \rightarrow_{\text{max}} \{y, z, z, q_a, q_b, q_c\}.
\]

Formally, given an initialised reaction system \( \mathcal{A} = (S, A) \), Method I yields a PTIA-net \( N_I(A) \) such that the places, transitions and the initial marking are:

- \( Pl = \{q_a | a \in A\} \cup S \), \( Tr = \{r_s | s \in S\} \cup A \) and \( M_0 = \{q_a | a \in A\} + C_0 \), respectively.
- Moreover, the sets of flow, inhibitor and activator arcs are:

\[
  Flw = \{(s, r_s) | s \in S\} \cup \{(a, q_a), (q_a, a) | a \in A\} \cup \{(a, s) | a \in A \land s \in P_a\}
\]

\[
  Inh = \{(s, a) | a \in A \land s \in I_a\}
\]

\[
  Act = \{(s, a) | a \in A \land s \in R_a\}.
\]
In order to relate the behaviour of the original reaction system \( A \) and the \( \text{PTIA}-\text{net} \) representation just introduced, we need two mappings. The first one takes a marking \( M \) of \( N_I(A) \) and returns a state of \( A \), and the other takes a step \( U \) of transitions of \( N_I(A) \) and returns a set of reactions of \( A \), as follows:

\[
\nu_I(M) = S \cap |M| \quad \text{and} \quad \varphi_I(U) = A \cap |U|.
\]

It is then possible to show the following three results, where a marking \( M \) of the \( \text{PTIA}-\text{net} \) \( N_I(A) \) is called \textit{well-formed} if \( M(q_a) = 1 \), for every \( a \in A \).

**Proposition 1.** \( M_0 \) is a well-formed marking satisfying \( \nu_I(M_0) = C_0 \).

**Proof.** We have \( \nu_I(M_0) = S \cap |M_0| = S \cap \{q_a \mid a \in A\} + C_0 \| = C_0 \). Moreover, \( M_0 \) is well-formed as \( M_0 = \{q_a \mid a \in A\} + C_0 \). \( \square \)

**Proposition 2.** If \( M \) is a well-formed marking and \( M[U] \) \( \text{max} \) \( M' \) then \( M' \) is also well-formed.

**Proof.** Follows from \( ^*q_a = q_a^* \), for every \( a \in A \). \( \square \)

**Proposition 3.** If \( M \) is a well-formed marking then, for every reaction \( a \in A \), \( a \) is enabled at \( M \) iff \( \{a\} \) is enabled at state \( \nu_I(M) \).

**Proof.** We first observe that, for every reaction \( a \in A \):

\[
^*a = R_a \quad \circ a = I_a \quad \circ a = P_a \cup \{q_a\} \quad \circ a = \{q_a\}.
\]

This and \( M \) being well-formed implies that:

\[
^*a \cup \circ a \subseteq |M| \iff R_a \subseteq |M| \iff R_a \subseteq S \cap |M| \iff I_a \cap |M| = \emptyset \iff I_a \cap (S \cap |M|) \ni
\]

\[
^\circ a \cap |M| = \emptyset \iff I_a \cap |M| \iff I_a \cap (S \cap |M|) \ni.
\]
Hence \( a \) is enabled at \( M \) iff \( \{ a \} \) is enabled at \( \nu_t(M) \).

We then can show that the translation is sound.

**Theorem 1.** If \( M \) is a well-formed marking then:

1. \( M[U]M' \) implies \( \nu_t(M) \overset{\varphi_t(U)}{\leadsto} \nu_t(M') \). Moreover, if \( M[U]_{\text{max}}M' \), then \( \varphi_t(U) \) comprises all reactions enabled at \( \nu_t(M) \).

2. \( \nu_t(M) \overset{R}{\rightarrow} C \) implies \( M[U]M' \) for some \( U \) and \( M' \) satisfying \( \varphi_t(U) = R \) and \( \nu_t(M') = C \). Moreover, if \( R \) comprises all reactions enabled at \( \nu_t(M) \), then \( M[U]_{\text{max}}M' \).

**Proof.** (1) We first observe that, by Proposition 3, \( \varphi_t(U) = A \cap \|U\| \) is enabled in state \( \nu_t(M) = S \cap \|M\| \), and if \( M[U]_{\text{max}}M' \) then \( \varphi_t(U) \) comprises all reactions enabled in state \( \nu_t(M) \). To show that \( \nu_t(M') = \text{res}_{\varphi_t(U)}(\nu_t(M)) \), we note that:

- For every \( s \in S \), \( U(r_s) = M(s) \). (i).
- For every \( a \in A \), \( U(a) \leq 1 \). (ii).

We then have, for every \( s \in S \):

\[
M'(s) = M(s) - \sum_{t \in e^*} U(t) + \sum_{t \in e} U(t) \\
= \sum_{t \in e^*} U(t) \quad \text{(by (i))}
\]

\[
= \sum_{a \in S \cap A} U(a) \quad \text{(by } e^* \subseteq A) \]

\[
= |(s \cap (A \cap \|U\|) | \\
= |\{a \in A \cap \|U\| \cap S \} | \\
= |\{a \in \varphi_t(U) \cap S = P_a \} | \\
\]

Hence \( s \in \nu_t(M') \) iff \( s \in \text{res}_{\varphi_t(U)}(\nu_t(M)) \), and so \( \nu_t(M') = \text{res}_{\varphi_t(U)}(\nu_t(M)) \).

(2) Let \( U = R + \sum_{s \in S} M(s) \cdot \{ r_s \} \). Clearly \( \varphi_t(U) = R \). One can then use Proposition 3 to show that \( U \) is enabled at \( M \), and if \( R \) comprises all reactions enabled at \( \nu_t(M) \), then \( U \) is max-enabled at \( M \). Finally, by proceeding similarly as in part (1), one can show that \( \nu_t(M') = \text{res}_R(\nu_t(M)) \), where \( M' \) is the marking satisfying \( M[U]M' \).

Thus, each maximal computational step in the Petri net corresponds to a unique execution of the reaction system, and each execution in the reaction system corresponds to at least one maximal step in the Petri net. For example, the two executions given above for the Petri net in Figure 2(a) both correspond to \( \{(x, z) \overset{q_2}{\rightarrow} \{y, z\} \} \) in the reaction system \( A_0 \).

Note that in Figure 2(a) one cannot simply delete the auxiliary places of the form \( q_0 \), as then each of the transitions representing reactions could be enabled an unbounded number of times. To address this problem one could change the activator arcs from places representing entities into flow arcs. Then, however, it would be necessary to add weights \( |R| \) to the arcs corresponding to the production of new entities in order to avoid conflicts on the places representing the reactants. This would not prevent steps from having multiple occurrences of a single transition.
Method II. The first attempt to model context-independent reaction systems provides a sound translation, but it is not simple as it employs features which can make formal analysis and verification far from easy. One way of improving Method I could be to replace multisets of fired transitions by sets of fired transitions leading to a maximal set-semantics. This can be achieved by using reset arcs [4], connecting places to transitions and indicated by *’s in the diagrams, which always empty their source place. Formally, reset arcs Reset \( \subseteq Pl \times Tr \) do not have any influence on the enabledness of a step \( U \), but the calculation of the marking of a place \( q \) after the firing of \( U \) (now a set) at marking \( M \) changes to:

\[
M'(q) = \begin{cases} 
M(q) - |q^* \cap U| + |q \cap U| & \text{if } (\{q\} \times U) \cap \text{Reset} = \emptyset \\
|q \cap U| & \text{otherwise}.
\end{cases}
\]

The resulting \textsc{ptn}\textsc{a}-net with reset arcs \( N_H(A_0) \) is shown in Figure 2(b). Transition \( r \) is always enabled and, when fired, removes all the tokens from the places modelling the entities. For the net in Figure 2(b), the new firing rule gives

\[
\{x, z\} \{(r, a, c)\}_{\max} \{y, z, z\} \\
\{x, x, z\} \{(r, a, c)\}_{\max} \{y, z, z\}.
\]

One can then show that a counterpart of Theorem 1 holds also in this case, with \( \nu_H \) defined as \( \nu_I \) before and \( \varphi_H(U) = U \setminus \{r\} \). As transition \( r \) is always enabled, we now have a one-to-one correspondence between groups of executed reactions and transitions, at the price of introducing non-standard reset arcs.

To remove the need to have reset arcs or, equivalently, to obtain a one-to-one correspondence between states and markings, one could change the rules for inserting tokens into places, by basically applying an OR-treatment for arriving tokens. This would, of course, be a radical departure from the standard Petri net approach, but one worth investigating. The resulting model of \textsc{set-nets} will be described in Section 5.

4 Reaction systems and high-level Petri nets

The two translations described in the previous section use low-level \textsc{pt}-nets extended with reset arcs in addition to inhibitor and activator arcs as well as maximal parallelism. Reset arcs are a non-standard mechanism and, in particular, they do not as yet support a causal process semantics. Moreover, the effect of a reset arc depends on the current marking rather than on a fixed input/output relation with its neighbourhood. To cope with this problem, we will now outline two translations from context-independent reaction systems to high-level Petri nets. We assume that the reader is familiar with the basic concepts of high-level Petri nets [10].
Method III. The first translation is illustrated by the high-level net $N_{III}(A_0)$ shown in Figure 3(a). In this case, tokens are positive integers acting as though they were time-stamps. Intuitively, a token $n$ is active only in the $n$-th execution cycle of the reaction system. Because the same token cannot be accessed more than once in a step sequence evolution, reset arcs are not needed anymore. Since the □ transition fires in each maximal step, the cycle number $n$ held in the ‘clock’ place $clk$ is known to all transitions representing reactions. In the places representing entities, they check only for tokens $n$, ignoring all the other tokens produced in previous cycles, and then produce tokens with value $n+1$ to be used in the next cycle. The initial marking $M_0$ is formed by inserting a single token 1 into place $clk$ and all the places $s$ such that $s \in C_0$. Note that the resulting net may be unbounded as the tokens in places representing entities are not ‘garbage collected’. For the high-level net $N_{III}(A_0)$ in Figure 3(b), we have:

$$
\{ x \mapsto \{1\}, y \mapsto \varnothing, z \mapsto \{1\}, w \mapsto \varnothing, clk \mapsto \{1\} \\
\{ y \mapsto \{2\}, z \mapsto \{1,2,2\}, w \mapsto \varnothing, clk \mapsto \{2\} \\
\{ z \mapsto \{1,3\}, y \mapsto \{2\}, z \mapsto \{1,2,3,3\}, w \mapsto \varnothing, clk \mapsto \{3\} \}.
$$

Fig. 3. Method III and IV representations of reaction system $A_0$. Note that $n$ and $m$ are net variables, and that to avoid chitter not all arcs have been annotated: all the flow (thicker) arcs to places $x$, $y$, $z$ are in fact annotated with $n + 1$, and all the unannotated inhibitor and activator arcs are annotated with $n$. In (b), the auxiliary places for transitions $b$ and $c$ are omitted. Note that $\langle m \geq n \rangle$ is the guard of transition $a_n$, and all other transitions have the trivial true guard.
As in the case of Method I, not every marking \( M \) of \( \mathcal{N}_{\text{III}}(A) \) can represent a valid state of the reaction system \( A \). We say that \( M \) is clock-consistent if there is a single token \( k \) in place \( \text{clk} \), and all the tokens \( l \) in other places satisfy \( l \leq k \).

Relating the resulting net and the original reaction system can be done using the following two mappings:

\[
\nu_{\text{III}}(M) = \{ s \in S \mid \| M(\text{clk}) \| \cap \| M(s) \| \neq \emptyset \} \quad \text{and} \quad \varphi_{\text{III}}(U) = U \setminus \{ \checkmark \}.
\]

**Proposition 4.** \( M_0 \) is a clock-consistent marking satisfying \( \nu_{\text{III}}(M_0) = C_0 \).

*Proof.* Follows directly from the definitions. \( \square \)

**Proposition 5.** If \( M \) is a clock-consistent marking and \( M[U]\max M' \) then \( M' \) is also clock-consistent.

*Proof.* That there is always a single token \( k \) in the place \( \text{clk} \) follows from the design of the flow arcs adjacent to it. Let \( M(\text{clk}) = \{ k \} \). Then \( M'(\text{clk}) = \{ k+1 \} \) as transition \( \checkmark \) is always enabled and so belongs to a maximally parallel step \( U \). Moreover, any new token which is inserted into a place \( s \in \), has also the value is \( k+1 \). Hence no token occurring in \( M' \) is greater than \( k+1 \). \( \square \)

We then can show that the translation is sound.

**Theorem 2.** If \( M \) is a clock-consistent marking then:

1. \( M[U]\max M' \) implies that \( \nu_{\text{III}}(M) \xrightarrow{\nu_{\text{III}}(U)} \nu_{\text{III}}(M') \) and \( \varphi_{\text{III}}(U) \) comprises all reactions enabled at \( \nu_{\text{III}}(M) \).
2. \( \nu_{\text{III}}(M) \xrightarrow{\mathcal{R}} C \), with \( \mathcal{R} \) comprising all reactions enabled at \( \nu_{\text{III}}(M) \), implies \( M[U]\max M' \) for some \( U \) and \( M' \) satisfying \( \varphi_{\text{III}}(U) = \mathcal{R} \) and \( \nu_{\text{III}}(M') = C \).

*Proof.* (sketch) Since \( M \) is a clock-consistent, we can assume below that \( M(\text{clk}) = \{ k \} \).

(1) Suppose that \( a \in U \setminus \{ \checkmark \} \). Then, for each \( s \in \bullet a \), we have \( k \in M(s) \). Moreover, for each \( s \in \circ a \), we have \( k \notin M(s) \). Thus, since \( \bullet a = R_a \) and \( \circ a = I_a \), we have that \( R_a \subseteq \nu_{\text{III}}(M) \) and \( I_a \cap \nu_{\text{III}}(M) = \emptyset \). As a result, \( \{ a \} \) is enabled in the state \( \nu_{\text{III}}(M) \). It is then easy check that the reverse implication holds as well, i.e., if \( \{ a \} \) is enabled in state \( \nu_{\text{III}}(M) \) then \( a \in U \setminus \{ \checkmark \} \). Hence we obtain that \( \varphi_{\text{III}}(U) = U \setminus \{ \checkmark \} \) is enabled in state \( \nu_{\text{III}}(M) \) and, moreover, it comprises all reactions enabled at \( \nu_{\text{III}}(M) \).

What still remains to be shown is that \( \nu_{\text{III}}(M') = \text{res}_{\nu_{\text{III}}(U)}(\nu_{\text{III}}(M)) \). This, however, can be easily check after observing that \( M'(\text{clk}) = \{ k+1 \} \) and that all tokens with the value \( k+1 \) occurring in \( M' \) must have been created in the currently executed step \( U \) as all the tokens occurring in \( M \) had their values not greater than \( M(\text{clk}) = \{ k \} \) (recall that \( M \) was assumed to be clock-consistent).

(2) It suffices to take \( U = \mathcal{R} \cup \{ \text{clk} \} \), and then use arguments similar as in the proof of part (1). \( \square \)
Method IV. In the second high-level net construction the aim is to eliminate the need for maximal parallelism using information present in the time-stamped tokens. We replace the global \( clk \) place by individual \( clk_a \) places, which are incremented by transitions representing reactions. Moreover, whenever a is blocked from firing in a certain cycle one of the auxiliary transitions corresponding to the possible `reasons` for the blocking a is fired to increment the token in \( clk_a \).

This results in an increment of the cycle number for this transition (in case there is more than one reason for blocking, an auxiliary transition is chosen non-deterministically).

There are two possible reasons why a might be blocked in cycle \( n \). One is the presence of a token \( n \) in the place representing an inhibitor of \( a \), and to check for this we use a transition with an activator arc, e.g., \( a_u \) in Figure 3(b). The other is more complicated as it is a lack of token \( n \) in the place representing a reactant \( s \) for \( a \), and to check for this we use a transition with an inhibitor arc. However, we also need to ensure that all transitions which feed tokens to \( s \) have already had a chance to do so, and we check this using extra activator arcs together with a transition guard which evaluates to true if all such feeding transitions have their local cycle sufficiently high, e.g., transition \( a_x \) in Figure 3(b). The overall result for the reaction system \( A_0 \) is a high-level net \( N_{IV}(A_0) \) shown in Figure 3(b).

The resulting high-level net is executed according to the standard sequential (interleaving) firing rule and its behaviour closely simulates that of the net obtained by Method III, and so also the behaviour of the original reaction system. We skip the full description of the relationship between these two nets. Intuitively, a marking \( M \) of the second translation corresponds directly to a marking of the first one if all the places of the form \( clk_a \) contain the same single token \( k \), and all the tokens \( l \) in other places satisfy \( l \leq k \). (Note that from each reachable marking of the second translation one can execute a sequence of transitions leading to a marking with this property.)

5 SET-nets

In our attempts to obtain a direct and elegant translation from reaction systems into Petri nets, a major and as far as we can tell insurmountable problem was the fact that several transitions may insert tokens into a place representing the presence of a single entity. In this section, we introduce SET-nets, a model that resulted from closer investigations into the possibilities of an OR-treatment of arriving tokens representing the production of entities by reactions. Note that OR-treatment of causality has been considered in [16], but the underlying principle there was completely different from what we are going to propose.

The main idea is that in a SET-net there is no concept of counting. Places are marked or not marked and arcs have no weights. SET-nets resemble elementary net systems (EN-systems) [15] which is a fundamental model to study basic features of concurrent systems, including conflict, causality and independence. However, their execution semantics is different. In SET-nets, a marked place indicates the presence of a resource without any quantification. Hence any number
of transitions that take input from this place can be fired at the same time. Moreover, firing a transition empties all its input places. Thus there are no conflicts over tokens in SET-nets, unlike in EN-systems or PT-nets. Similarly, places do not count the tokens, and the firing of a transition simply marks each of its output places (whether or not they were already marked). We will build up the
new model in two stages, introducing first SET-nets with only flow arcs.

**Definition 7 (basic SET-net).** A tuple \( SN = (Pl, Tr, Flw, M_0) \) is a (basic) SET-net if the first three components are as in Definition 5, and \( M_0 \subseteq Pl \) is the initial marking (in general, any set of places is a marking).

The graphical representation of SET-nets is the same as in the case of Petri nets. We now formalise the firing rule for SET-nets.

**Definition 8 (marking change).** A set of transitions \( U \) (also called a step) is enabled at a marking \( M \) if \( *U \subseteq M \). In such a case, \( U \) can be fired with its effect on \( M \) being given by the resulting marking
\( M' = (M \setminus *U) \cup U^* \). We denote this by \( M[U]M' \). Moreover, if \( U \) is the set of all transitions enabled at \( M \) (i.e., all transitions \( t \) satisfying \( *t \subseteq M \)), then we may write \( M[U]_{max} M' \).

Hence a step \( U \) enabled at a marking \( M \) may contain two distinct transitions \( t \) and \( u \) for which \( *t \cap *u = \emptyset \) or \( t^* \cap u^* = \emptyset \) and yet the common places will never contain more than one token. Since tokens are manipulated using set-based arithmetic

we have chosen the name ‘SET-nets’ for the new class of Petri nets.

We have introduced first basic SET-nets (without inhibitor and activator arcs), as it seems that one can attempt to develop for them a counterpart of ‘structure theory’ of PT-nets. To illustrate our point, let us consider a basic SET-net \( SN = (Pl, Tr, Flw, M_0) \) with at least one transition. A non-empty set of places \( Sphn \subseteq Pl \) is called a siphon if \( *Sphn \subseteq Sphn^* \). Similarly, a non-empty set of places \( Trap \subseteq Pl \) is called a trap if \( *Trap \subseteq *Trap \). It can be easily seen that an empty siphon cannot acquire a token by firing any transition, and a marked trap cannot become empty by firing any transition. Both type of sets of places can be used to provide a sufficient condition for deadlock-freeness in PT-nets which was a major motivation behind the development of their structure theory. As it turns out, the same can be done in case of SET-nets.

**Theorem 3.** If in the initial marking, every siphon contains a marked trap, then the SET-net is deadlock free.

**Proof.** We first observe that if a trap \( Trap \) is marked in the initial marking, \( Trap \cap M_0 \neq \emptyset \), then it is also marked in any reachable marking \( M \), \( Trap \cap M \neq \emptyset \). This follows directly from the fact that if \( *t \cap Trap \neq \emptyset \) than also \( t^* \cap Trap \neq \emptyset \).

Suppose now that there is a reachable marking \( M \) such that no transition is enabled at \( M \). Then, for each \( t \in T \), there must exist a place \( p_t \in *t \) such that
\[ M(p_0) = 0. \] Clearly, \( \text{Sphn} = \{ p_t \mid t \in T \} \) is a siphon (as \( \text{Sphn}^* = T \)) satisfying \( \text{Sphn} \cap M = \emptyset \). On the other hand, it was assumed that there exists a trap \( \text{Trap} \subseteq \text{Sphn} \) satisfying \( \text{Trap} \cap M_0 \neq \emptyset \). Hence, according to what we already proved, \( \text{Trap} \cap M \neq \emptyset \) producing a contradiction with \( \text{Sphn} \cap M = \emptyset \). \( \Box \)

We next introduce \text{SET}-nets with inhibitor and activator arcs.

**Definition 9 (SET-net).** A tuple \( \text{SNIA} = (P, T, Fwu, \text{Inh}, \text{Act}, M_0) \) is a \text{SET}-net if the first five components are as in Definition 5, and the last one as in Definition 7.

The definitions and notations concerning the marking change in \text{SNIA} are the same as for \text{SN} in Definition 8 with one exception, namely a set of transitions \( U \) is enabled at a marking \( M \) if \( \text{Sphn} \subseteq \text{Sphn}^* \) and \( \text{Sphn} \cap M = \emptyset \). It is interesting to observe that an enabled step \( U \) is always consistent in the sense that \( (\text{Sphn} \cup \text{Sphn}^*) \cap \text{Sphn} = \emptyset \). Such a property has a natural and direct (as we will see) connection with the notion of consistency introduced for reaction systems.

As before, given a transition \( t \) representing a reaction, the sets \( \text{Sphn}^* \), \( \text{Sphn}^\circ \) and \( \text{Sphn}^\downarrow \) correspond to the reactants, inhibitors and products of this reaction. However, we do not require that these sets be non-empty in a \text{SET}-net (at least at this point) as such an assumption is not necessary.

### 6 Reaction systems and \text{SET}-nets

Reaction systems and \text{SET}-nets fit together well in the sense that both do not count tokens and both change states on the basis of the presence/absence of resources, represented by sets. Moreover, under the \text{SET}-net semantics, ordinary arcs (transitions) can be used to empty places. In this semantics, reset arcs with their effect depending on the current number of tokens in a place are meaningless. Finally, following the assumption that all reactions that can take place do take place, the maximal set-semantics can be employed.

Figure 4(a) depicts a \text{SET}-net corresponding to a context-independent initialised reaction system \( A_1 = (\{ r, q, s \}, \{ a, b \}, \{ q, s \}) \), where:

\[
a = (\{ r, q \}, \emptyset, \{ r \}) \quad \text{and} \quad b = (\{ q \}, \emptyset, \{ r, q \}).
\]

(For reasons of clarity, we allow in this section reactions without any inhibitors.) As before, places represent entities. Transitions \( r_\downarrow, q_\downarrow \) and \( s_\downarrow \) ensure that once the \text{SET}-net is active only tokens produced in the last maximal step are present in the current marking. For example, we have:

\[
\{ q, s \} \{ b, q_\downarrow, s_\downarrow \} \max \{ r, q \} \{ a, b, r_\downarrow, q_\downarrow \} \max \{ r, q \},
\]

and so \( \sigma = \{ b, q_\downarrow, s_\downarrow \} \{ a, b, r_\downarrow, q_\downarrow \} \) is a maxi-step sequence. Relating the behaviour of the \text{SET}-net model and the original reaction system is straightforward and we obtain a counterpart of Theorem 1 using the mappings:

\[
\nu(M) = M \quad \text{and} \quad \varphi(U) = U \setminus \{ s_\downarrow \mid s \in S \}.
\]
For a SET-net without inhibitor and activator arcs as in Figure 4(a), one can investigate the causality semantics of reaction systems based on the unfoldings of the corresponding SET-nets. Figure 4(b) shows how such an occurrence net could be derived for the SET-net in Figure 4(a) and its step sequence \(\{b, q_1, s_1\}\{a, b, r_1, q_1\}\) which corresponds of the state sequence \(\{b\}\{a, b\}\) of the original reaction system. It is worth observing that the process has branching places which is not possible, in the case of processes of EN-systems or PT-nets. This, however, is fully consistent with the execution semantics of SET-nets.

**Fig. 4.** A SET-net representing reaction system \(A_1\) (a); and an occurrence net constructed for its step sequence \(\{b, q_1, s_1\}\{a, b, r_1, q_1\}\) (b).

Modelling inhibition aspects of reactions is rather straightforward using inhibitor arcs, as illustrated by the SET-net in Figure 5(a), representing the context-independent initialised reaction system \(A_2 = (\{r, q, s\}, \{a, b\}, \{q\})\), where:

\[
a = (\{r, q\}, \emptyset, \{r\}) \text{ and } b = (\{q\}, \{s\}, \{r, q\}) \text{ and } c = (\{q\}, \emptyset, \{s\}).
\]

Using inhibitor arcs gives a compact translation of reaction systems which is in a sense minimal w.r.t. the number of places, arcs and transitions. Moreover, relating the behaviour of the resulting SET-nets and the original reaction systems can be done as before. Formally, the places, transitions and initial marking of the translation are given by: \(Pl = S, Tr = A \cup \{s_1 \mid s \in S\}\) and \(M_0 = C_0\). There are no activator arcs, and the flow and inhibitor arcs are as follows:

\[
Flw = \{(s, s_1) \mid s \in S\} \cup \{(s, a) \mid a \in A \land s \in R_a\} \cup \{(a, s) \mid a \in A \land s \in P_a\}
\]

\[
Inh = \{(s, a) \mid a \in A \land s \in I_a\}.
\]

The development of a causal process semantics of SET-nets with inhibitor arcs is more difficult. It is therefore interesting to consider models of reaction systems using SET-nets without any inhibitor arcs, as outlined next.

Figure 5(b) shows a SET-net without inhibitor arcs modelling \(A_2\). The way in which it does it is now more involved. More precisely, each execution step of the reaction system is simulated in two phases by the SET-net operating according to
the maximal parallelism execution semantics. To keep these two phases clearly separated, they are controlled by an additional cyclic subnet with two places. The key aspect of the construction is the use of a ‘complement’ \( s^{\text{cp}} \) of the ‘regular’ place \( s \) which at the time of checking whether \( s \) is empty by reaction \( b \) contains a token iff \( s \) is empty.

\[
\begin{array}{c}
\text{(a)}
\end{array}
\]

\[
\begin{array}{c}
\text{(b)}
\end{array}
\]

\[
\begin{array}{c}
\text{(c)}
\end{array}
\]

**Fig. 5.** Two set-nets representing \( A_2 (a, b) \). Generic translation without inhibitor arcs: here \( r \) is a reactant for reaction \( a \), product for \( b \), and inhibitor for \( c \) (c). Note that not all places and arcs are shown; in particular, each reaction has at least one reactant and hence transitions like \( c \) can only fire in the first phase.
Figure 5(c) provides a generic picture of how, in the proposed construction, a set-net (without inhibitor arcs) handles an entity $r$ in its role as a reactant, inhibitor, and product. Note that $r$ is represented by two places, $r$ and $r^{\text{pl}}$, and if $r^{\text{pl}}$ is marked then the entity $r$ in absent in the current state. Moreover, each reaction $d$ is represented by two transitions, $d$ and $d'$. The first corresponds to the enabling stage of $d$, and the second to the generation of its products.

The first phase of the simulation always starts in a consistent marking $M$ in which there is a token in place $\text{phI}$; for every $s \in S$, $s \in M$ $\Leftrightarrow$ $s^{\text{pl}} \notin M$, and otherwise all places are empty. In this phase transitions corresponding to reactions become active on the basis of the presence and absence of their reactants and inhibitors. Simultaneously, transitions of the form $r_\downarrow$ and $r_\uparrow$ take care that all the entities present in the current state cease to exist (their corresponding places are emptied and the complement places filled). In the second phase, each enabled transition $d'$ finishes the execution of the corresponding reaction, and marks the places corresponding to the entities produced by reaction $d$ and empties their complements.

Relating the behaviour of the set-net model and the original reaction system is more complicated, using the following two mappings:

$$\nu(M) = M \setminus (\{ \text{phI} \} \cup \{ s^{\text{pl}} \mid s \in S \})$$
$$\varphi(U) = U \setminus (\{ I \} \cup \{ s_\downarrow \mid s \in S \} \cup \{ s_\uparrow \mid s \in S \}).$$

One can then show that $M_0$ is a consistent marking satisfying $\nu(M_0) = C_0$, and if $M$ is a consistent marking and $M(U)_{\text{max}}M''(U')_{\text{max}}M'$ then $M'$ is also consistent.

The soundness result is now more complicated as it has to take into account two-phase simulation of the execution of reaction rules. First, we have that if $M$ is a consistent marking then $M(U)M''(U')M'$ implies $\nu(M) \xrightarrow{\varphi(U)} \nu(M')$. Second, if $M$ is a consistent marking then $\nu(M) \xrightarrow{R} C$ implies $M(U)M''(U')M'$ for some $U, U', M'$ and $M''$ satisfying $\varphi(U) = R$ and $\nu(M') = C$.

7 Related work

When introducing a new class of Petri nets, especially a fundamental one, it is necessary to put it in the context of existing formalisations. To make comparison fair, we will now drop the assumption about maximal parallelism in the execution of set-nets (which is implied by the execution mode of reaction systems), and consider semantics which allows any set of enabled transitions to be fired.

Set-nets are so simple when it comes to their definition, that it is reasonable to expect that there were in the past net classes with similar features. Indeed, the fundamental class of EN-systems [13] extended with inhibitor as well as activator arcs [9, 13, 14] basically have the same static structure as set-nets. However, their treatment of conflicts between transitions accessing the same token, as well blocking a transition which could add a token to a marked place, are totally
different. The latter issue has been noted in the past, and the constraint relaxed.
For example, there are variations of Petri nets, such as Boolean Petri nets, where
adding a token to an already marked place does not add another token [2, 3, 8].
Also, behaviour of this kind was mentioned in [1] in the context of net synthesis.
Having said that, the semantics considered in prior works known to us was
based on single transition firings, rather than (maximal) steps as is the case
for set-nets. Therefore, the previous models were not concerned with multiple
inputs of tokens to a single place something which is essential if one wants to
faithfully model reaction systems. Furthermore, by aiming at a set-semantics,
we had to introduce the non-conflict feature on the flow arcs consuming the
tokens. Therefore, as far as we are aware, the model of set-nets is an original
contribution to the field of Petri nets.

As we already mentioned, set-nets with interleaving semantics are nothing
but Boolean nets used, for instance, in [3]. In such a case, the lack of conflict
when firing two transitions sharing an input place is an irrelevant issue, and the
only non-standard aspects is that firing a transition with a marked output place
does not increase the token count in that place. Such a feature, moreover, can
easily be modelled using ordinary 1-safe PT-nets, according to the following idea.
First, one splits each place \( q \) into places \( q^0 \) and \( q^1 \), respectively representing
the lack and presence of a token in \( q \). Then, each transition \( t \) adding tokens
to place \( q \) is split into \( t^0 \) and \( t^1 \) to account for two different states the place
\( q \) can be in represented by \( q^0 \) and \( q^1 \). Figure 6 illustrates this construction.
It can be easily seen that both nets generate the same sequential reachability
graphs assuming that \( a^0 \) and \( a^1 \) are instances of \( a \), and \( b^0 \) and \( b^1 \) are instances of
\( b \). However, once we start treating the net in Figure 6(a) as set-net, the
situation changes radically. The reason is that we then have three firings of the following form: \( \varnothing\{\{a\}\}\{q\} \), \( \varnothing\{\{b\}\}\{q\} \) and \( \varnothing\{\{a, b\}\}\{q\} \). Now, the standard
classes of Petri nets enjoy the so-called subset property which means that if a
step \( U \) is enabled at marking \( M \), then also any of its subsets is enabled as well.
Suppose then, that there is a Petri net \( N \) satisfying this property and such that
its step reachability graph is the same as that of the set-net in Figure 6(a),
perhaps after renaming \( \lambda \) being applied to the transitions of the former. Then
we have to have two transitions, \( t \) and \( u \), in \( N \) such that \( \lambda(t) = a \), \( \lambda(u) = b \) and
\( M_0\{\{t, u\}\}M \). Then, by the subset closure property, we also have \( M_0\{\{t\}\}M' \)
and \( M_0\{\{u\}\}M'' \). Hence, by the reachability graph isomorphism, we must have
\( M = M' = M'' \) as well as \( M_0 \neq M \). Hence we have: \( M_0\{\{t, u\}\}M \) and
\( M_0\{\{t\}\}M \) and \( M_0\{\{u\}\}M \) and \( M_0 \neq M \). In the standard Petri nets, including various
extensions of PT-nets, \( M_0\{\{t, u\}\}M \) and \( M_0\{\{t\}\}M \) would imply that \( u \) does not change
the current marking. Similarly, \( M_0\{\{t, u\}\}M \) and \( M_0\{\{u\}\}M \) would imply
that \( t \) does not change the current marking. Yet the simultaneous firing of \( t \) and
\( u \) does change the marking as \( M_0 \neq M \). This would produce a contradiction.
What we just presented is intuition rather than proof, however, we expect that
detailed arguments can be developed for any of the standard net classes. An
important consequence, however, is that set-nets are semantically different from
the existing net classes and therefore deserve to be recognised as an original contribution.

\[ a \rightarrow b \]

\[ c \]

**Fig. 6.** Boolean net (set-net with sequential semantics) (a), and 1-safe PT-net simulating its (sequential) behaviour (b).

8 Concluding remarks

The main initial motivation of our investigation was to see how Petri net based concepts could be deployed to analyse reaction systems. In particular, we wanted to discover methods for checking properties of reaction systems by relating them to the properties of the corresponding Petri nets and causal processes.

We proposed modelling methods resulting both in low-level and high-level nets. In all four cases, we established a close correspondence between the markings of Petri nets and states of the original reaction systems. The same was true of the evolutions of two corresponding models. In fact, we established that they have essentially isomorphic state spaces. All these net models, however, exhibited deficiencies w.r.t. simplicity and/or elegance and/or tractability of the translation. For example, both high-level net models are intrinsically unbounded, and the second of the low-level translations uses reset arcs. We therefore proposed a new class of Petri nets, called set-nets, which we feel provide a strong match with the reaction systems and their semantics.

In this way we think we derived new interesting notions and contributions to Petri net theory based on our experiences with reaction systems in a similar way as the concepts of localities and locally maximal concurrency were derived from our previous investigation of a Petri net semantics of membrane systems [12].

References