# Plug-in Context Providers for Reaction Systems

Jetty Kleijn [a] Maciej Koutny [b] Grzegorz Rozenberg [a]

[a] *LIACS, Leiden University,P.O.Box 9512, NL-2300 RA Leiden, The Netherlands*

[b] *School of Computing, Newcastle University, Newcastle upon Tyne, NE1 7RU, United Kingdom*

**Abstract**

Reaction systems originated as models of interactions between biochemical reactions in the living cell. Since then they were also successfully developed as models of interactive computation. Here, the interaction between a (reaction) system and its environment is modeled through context sequences provided by the environment – they influence the processes in the system. In this paper we introduce and investigate a 'plug-in' methodology for providing context sequences, where we view interactive processes from the perspective of the environment. The environment is modeled by a plug-in device, where a reaction system can be plugged in. When a reaction system is plugged in, then it receives context sequences from the plug-in device. Several sorts of such devices are investigated and compared (as influencers of behaviours of reaction systems).

*Key words:* reaction system, interactive computation, context sequences, interactive processes

## 1 Introduction

The original motivation for introducing reaction systems (see, e.g., [1,2,3,4]) was to model interactions between biochemical reactions taking place in the living cell. Two basic mechanisms behind these interactions are facilitation and inhibition: the product of one reaction $a$ may contain reactants of another reaction $b$ (hence $a$ is facilitating $b$), but also this product may contain inhibitors of reaction $c$ (hence $a$ is inhibiting $c$). These mechanisms are explicitly formalised within the model of reaction systems, so that the interactions between reactions are modelled by dynamic processes taking place in reaction

*Email addresses:* h.c.m.kleijn@liacs.leidenuniv.nl (Jetty Kleijn), maciej.koutny@ncl.ac.uk (Maciej Koutny), g.rozenberg@liacs.leidenuniv.nl (Grzegorz Rozenberg).

*Preprint submitted to Elsevier* *26 January 2020*

systems – such formalised dynamic processes are appropriately called *interactive processes* (and also model interactions with the environment).

Although the model of reaction systems was inspired by biology, since then research topics were guided by both biological motivations (see, e.g., [4,5,6,7,8]) and by the need to understand the underlying computations. In particular, by now reaction systems have become a novel and attractive model of interactive computation (see, e.g., [9,10,11,12,13,14,15,16,17,18,19,20,21,22]).

Formally, a *reaction* is a triplet $b = (R_b, I_b, P_b)$ of three finite nonempty sets, where $R_b$ is the set of all reactants that $b$ needs to take place, $I_b$ is the set of all inhibitors of $b$ (if any of them is present in the current state $T$, then $b$ will not take place in $T$), and $P_b$ is the product set of $b$ (if $b$ takes place in $T$, then it will contribute its product set $P_b$ to the successor state of $T$). If $Z$ is a finite set such that all three sets $R_b$, $I_b$, and $P_b$ are subsets of $Z$, then we say that $b$ is a reaction over $Z$.

Then a *reaction system* is defined as an ordered pair $\mathcal{A} = (S, A)$, where $S$ is a finite set (of at least two elements), called the *background set* of $\mathcal{A}$, and $A$ is a set of reactions over $S$. The background set is needed to define all reactions in $A$, as well as to define interactions with the environment of $\mathcal{A}$. States of $\mathcal{A}$ are subsets of $S$.

The dynamic behaviour of $\mathcal{A}$ is formalised through interactive processes in $\mathcal{A}$, where a transition from the current state $T$ of $\mathcal{A}$ to its successor state $T'$ is determined by: (i) the transformation of $T$ by the reactions of $\mathcal{A}$, which results in a set $D \subseteq S$, and (ii) the contribution of the environment in the form of a context set $C$. Consequently, the successor state $T'$ equals $D \cup C$.

Thus $\mathcal{A}$ is an *open system*: its behaviour (interactive processes in $\mathcal{A}$) is influenced by the environment. To understand the 'internal' behaviour of $\mathcal{A}$ (influenced by its reactions only) one also considers $\mathcal{A}$ as a *closed system*, where the interactive processes are not influenced by the environment – such processes are called *context-independent* interactive processes.

One of the central lines of research concerning reaction systems is to understand the nature of interactions with the environment. In particular, one attempts to define context sequences (fed into reaction systems by the environment) in some structural fashion, defining in this way classes of interactive processes more general than context-independent, but (much) more restrictive than arbitrary interactive processes.

In this paper we propose a 'plug-in' methodology for providing context sequences: one plugs in a reaction system into a device that will provide context sequences for it. In particular, we consider two types of such devices.

First, we formulate and analyse the notion of an *expander*, which itself is a reaction system. Then we reformulate the notion of a (context) *controller* (introduced within research on model checking for reaction systems, [14,23,24]) as a plug-in context provider. Subsequently, we analyse the relationship between these two sorts of devices for providing context sequences for reaction systems. The paper is organised as follows.

After fixing in Section 2 basic mathematical notation and terminology for this paper, in Section 3 we recall main notions of reaction systems to be used in this paper. In Section 4 we introduce expanders which are plug-in context providers, and are themselves reaction systems. We formulate and discuss the behaviour of reaction systems plugged into expanders. In Section 5 we recall the notion of (context) controller, formulate its use as a plug-in context provider, and discuss the behaviour of reaction systems plugged into controllers. In Section 6 we demonstrate that expanders and controllers are equivalent in the sense that they can (quite precisely) simulate each other. In Section 7 we consider a restricted version of controllers, the so-called state-oblivious controllers (while they provide context sequences for a reaction system $\mathcal{A}$, they are not aware of the current state of $\mathcal{A}$). We demonstrate that state-oblivious controllers are less powerful (in controlling the behaviour of reaction systems) than the general (state-aware) controllers. We also demonstrate that they are equivalent to a restricted use of expanders defined through the notion of an extension of reaction systems well-known from the literature (see, e.g., [25]). The discussion in Section 8, proposing several lines of research on plug-in context providers, concludes this paper.

## 2 Preliminaries

In order to fix notation and terminology for this paper, we recall in this section some basic mathematical notions concerning sets and graphs.

For a finite set $X$, $|X|$ denotes its cardinality, $2^X$ denotes the set of all subsets of $X$, and $\varnothing$ denotes the empty set. For sets $X$ and $Y$, $X \setminus Y$ denotes their difference, $X \cup Y$ denotes their union, $X \cap Y$ denotes their intersection, $X \times Y$ denotes their cartesian product, while $X \subseteq Y$ denotes the (not necessarily strict) inclusion of $X$ in $Y$. For a family $\mathcal{L}$ of sets, $\bigcup \mathcal{L}$ denotes the union of the sets in $\mathcal{L}$.

If $\tau = W_0, \ldots, W_n$, for some $n \geq 0$, is a sequence of sets and $Q$ is a set, then the $Q$-projection of $\tau$ is the sequence of sets $proj_Q(\tau) = W_0 \cap Q, \ldots, W_n \cap Q$.

A *directed edge-labelled graph* is a triplet $\mathcal{G} = (V, E, L)$, where $V$ is a finite set of nodes, $E \subseteq V \times L \times V$ is the set of labelled edges, and $L$ is a finite set of edge labels. An edge $e = (v, x, u)$ is *outgoing* from $v$ and *incoming* into $u$, while $x$ is the label of $e$.

## 3 Reactions and Reaction Systems

In this section we recall some basic notions of reaction systems that will be used in this paper; most of them are taken from [3,22].

**Definition 1** *A* reaction *is a triplet $b = (R, I, P)$ such that $R, I, P$ are finite* *nonempty sets with* $R \cap I = \varnothing$.

The sets $R, I, P$ are called the *reactant set of* $b$, the *inhibitor set of* $b$, and the *product set of* $b$, respectively – they are also denoted as $R_b$, $I_b$ and $P_b$, respectively. If $R, I, P \subseteq Z$ for a finite set $Z$, then we say that $b$ is a *reaction over* $Z$ and we use $rac(Z)$ to denote the set of all reactions over $Z$ – note that $rac(Z)$ is finite.

Since $R$ and $I$ are nonempty and disjoint, a finite set $Z$ as above must have at least 2 elements – we refer to such finite sets as *background sets.*

To describe the effect of a set of reactions on a state (e.g., of a biochemical system), we first define the effect of a single reaction. A state of a system is formalised as a subset of its background set.

**Definition 2** *Let* $Z$ *be a background set, let* $X \subseteq Z$*, and let* $b \in rac(Z)$*. Then* $b$ *is* enabled by $X$*, denoted by* $en_b(X)$*, if* $R_b \subseteq X$ *and* $I_b \cap X = \varnothing$*. The* result of $b$ on $X$*, denoted by* $res_b(X)$*, is defined by* $res_b(X) = P_b$ *if* $en_b(X)$*, and* $res_b(X) = \varnothing$ *otherwise.*

Here the finite set $X$ is a formal representation of a state (e.g., the set of biochemical entities currently present in the given biochemical system). Then, $b$ is enabled by $X$ if $X$ separates $R_b$ from $I_b$, meaning that all reactants from $R_b$ are present in $X$ and none of the inhibitors from $I_b$ is present in $X$. When $b$ is enabled by $X$, it contributes its product $P_b$ to the successor state; otherwise it does not contribute anything to the successor state of $X$.

The effect of a set of reactions on a state is formally defined as follows.

**Definition 3** *Let* $Z$ *be a background set, let* $X \subseteq Z$*, and let* $B \subseteq rac(Z)$*. The* result of $B$ on $X$*, denoted by* $res_B(X)$*, is defined by* $res_B(X) = \bigcup\{res_b(X) \mid b \in B\}$.

Thus applying sets of reactions is additive (cumulative): if $B = B_1 \cup B_2$, then $res_B(X) = res_{B_1}(X) \cup res_{B_2}(X)$.

With the formal notion of a reaction and its effect on states established, we can now proceed to formally define the main notion of this paper, viz., a reaction system (originally introduced as an abstract model of the interactions of biochemical reactions in the living cell).

**Definition 4** *A* reaction system*, abbreviated* rs*, is an ordered pair* $\mathcal{A} = (S, A)$*, where* $S$ *is a background set and* $A$ *is a nonempty subset of* $rac(S)$.

The set $S$ is called the *background set of* $\mathcal{A}$ and its elements are called the *entities of* $\mathcal{A}$ (in the original biochemical interpretation they represent molecular entities such as, e.g., atoms, ions, molecules) that may be present in the states of the reaction system under consideration. The set $A$ is called the *set of reactions of* $\mathcal{A}$; clearly $A$ is finite (as a finite $S$ implies a finite $rac(S)$). The subsets of $S$ are called the *states of* $\mathcal{A}$. Given a state $X \subseteq S$, the *result of* $\mathcal{A}$ on $X$, denoted by $res_{\mathcal{A}}(X)$, is defined by $res_{\mathcal{A}}(X) = res_A(X)$.

The dynamic behaviour of reaction systems is expressed through interactive processes, which originally were defined as follows.

**Definition 5** *Let $\mathcal{A} = (S, A)$ be a reaction system. An* interactive process *in $\mathcal{A}$ is a pair $\pi = (\gamma, \delta)$ of finite sequences such that $\gamma = C_0, \ldots, C_n$ and $\delta = D_0, \ldots, D_n$, for some $n \geq 1$, where $C_0, \ldots, C_n \subseteq S$, $D_0, \ldots, D_n \subseteq S$, and $D_i = res_{\mathcal{A}}(D_{i-1} \cup C_{i-1})$, for all $i \in \{1, \ldots, n\}$.*

We say that $\pi$ is a *n-step* interactive process. The sequence $\gamma$ is the *context sequence of* $\pi$, the sequence $\delta$ is the *result sequence of* $\pi$, and the sequence $\tau = W_0, \ldots, W_n$, where, for all $i \in \{0, \ldots, n\}$, $W_i = C_i \cup D_i$, is the *state sequence of* $\pi$, with $W_0 = C_0 \cup D_0$ the *initial state* of $\pi$. When we consider what happens after the first step of the process, then we get sequences $C_1, \ldots, C_n$; $D_1, \ldots, D_n$; and $W_1, \ldots, W_n$ which are referred to as the *strict* context sequence, *strict* result sequence, and *strict* state sequence, respectively.

If, for all $i \in \{1, \ldots, n\}$, $C_i \subseteq D_i$, then we say that $\pi$ is *context-independent*: whatever $C_i$ adds to $D_i$, has already been produced by the reaction system from the predecessor state ($C_i$ is included in the result $D_i$) or perhaps $C_i$ adds nothing at all ($C_i = \varnothing$). A special case of a context-independent interactive process is the case when $C_i = \varnothing$, for all $i \in \{1, \ldots, n\}$ (it is referred to as an *empty-context* process). If $\pi$ is context-independent, then the initial state $W_0$ determines the state sequence of $\pi$ by the repeated application of $res_{\mathcal{A}}$.

Note that the context sequence $\gamma$ together with the initial state $W_0$ uniquely determines $\pi$, because $\gamma$ and $D_0$ uniquely determine $\pi$ (through the result function $res_{\mathcal{A}}$). The context sequence formalises the fact that the *living cell is an open system* in the sense that its behaviour is influenced by its environment (the 'rest' of a bigger system).

An interactive process may be visualised by a three-row representation, where the first row represents the context sequence and is labelled by '$\gamma$', the second row represents the result sequence and is labelled by '$\delta$', and the third row represents the state sequence and is labelled by '$\tau$'. Such a visualisation looks as follows:

$$\begin{array}{rlllll} \gamma: & C_0 & C_1 & \ldots & C_{n-1} & C_n \\ \delta: & D_0 & D_1 & \ldots & D_{n-1} & D_n \\ \tau: & W_0 & W_1 & \ldots & W_{n-1} & W_n \end{array}$$

This visualisation leads to an equivalent formulation of an interactive process, based on the notion of configuration (see, e.g., [9,22]). This formulation will be used in our paper. To begin with, for a background set $S$, an *$S$-configuration* is a triple $f = (C, D, W)$ such that $C, D, W \subseteq S$ and $W = C \cup D$.

**Definition 6** *Let $\mathcal{A} = (S, A)$ be a rs. An* interactive process in $\mathcal{A}$ *is a sequence $\pi$ of $S$-configurations, $\pi = f_0, f_1, \ldots, f_n$ for some $n \geq 1$ such that, for each $i \in \{0, \ldots, n\}$, $f_i = (C_i, D_i, W_i)$, where for each $i \in \{1, \ldots, n\}$, $D_i = res_{\mathcal{A}}(W_{i-1})$.*
We refer to $f_0, f_1, \ldots, f_n$ as *configurations of* $\pi$ and to $f_0$ as the *initial configuration of* $\pi$.

Clearly, in the terminology following Definition 5, the sequence $C_0, \ldots, C_n$ is the context sequence of $\pi$, the sequence $D_0, \ldots, D_n$ is the result sequence of $\pi$, and the sequence $W_0, \ldots, W_n$ is the state sequence of $\pi$. Note that we have now included the state sequence directly in the definition of an interactive process – this is very convenient in the formal analysis of interactive processes.

Following the intuition of the use of $S$-configurations in defining interactive processes, given an $S$-configuration $f = (C, D, W)$ we refer to $C$, $D$, and $W$ as the *context of* $f$, the *result of* $f$, and the *state of* $f$, respectively.

We will use

- $PROC(\mathcal{A})$ to denote the set of all interactive processes of $\mathcal{A}$,
- $CIPROC(\mathcal{A})$ to denote the set of all context-independent interactive processes in $\mathcal{A}$,
- $STS(\mathcal{A})$ to denote the set of all state sequences of all interactive processes in $PROC(\mathcal{A})$, and
- $CISTS(\mathcal{A})$ to denote the set of all state sequences of all context-independent interactive processes in $CIPROC(\mathcal{A})$.

It is instructive to notice that $res_{\mathcal{A}}(S) = res_{\mathcal{A}}(\varnothing) = \varnothing$. This holds because, for each $a \in S$, $I_a \neq \varnothing$ and $R_a \neq \varnothing$.

One sort of reactions turns out to be very useful in defining functions (from $2^S$ to $2^S$) – they are called *complementary reactions.* A reaction $a \in A$ is complementary, if it is of the form $a = (Z, S \setminus Z, T)$, where $Z$ and $T$ are nonempty subsets of $S$. Note that $a$ is enabled in exactly one state of $\mathcal{A}$, viz., $Z$ (and then it contributes $T$ to the successor of $Z$). We will use complementary reactions in proofs in Section 6.

A convenient way of combining reaction systems is through their unions defined as follows.

**Definition 7** *Let* $\mathcal{A}_1 = (S_1, A_1)$ *and* $\mathcal{A}_2 = (S_2, A_2)$ *be reaction systems. The* union of $\mathcal{A}_1$ and $\mathcal{A}_2$*, denoted by* $\mathcal{A}_1 \oplus \mathcal{A}_2$*, is the reaction system defined by* $\mathcal{A}_1 \oplus \mathcal{A}_2 = (S_1 \cup S_2, A_1 \cup A_2)$.

We end this section by pointing out three important properties of state-sequences of context-independent processes. Let $\mathcal{A} = (S, A)$ be a reaction system, and $\tau \in CISTS(\mathcal{A})$, $\tau = W_0, \ldots, W_n$ for some $n \geq 1$.
(1) 'No resurrection': if $W_i = \varnothing$ for some $i \in \{0, \ldots, n-1\}$, then $W_j = \varnothing$ for all $j \in \{i+1, \ldots, n\}$ (because $res_{\mathcal{A}}(\varnothing) = \varnothing$),
(2) 'No saturation': if $W_i = S$ for some $i \in \{0, \ldots, n-1\}$, then $W_{i+1} = \varnothing$ (because $res_{\mathcal{A}}(S) = \varnothing$),
(3) 'Once repeated, always repeated': if $n \geq 2$ and $W_i = W_{i+1}$ for some $i \in \{0, \ldots, n-2\}$, then $W_{i+1} = W_{i+2}$ (because $W_{i+2} = res_{\mathcal{A}}(W_{i+1}) = res_{\mathcal{A}}(W_i) = W_{i+1}$).

## 4 Providing context through expanders

In this section we will consider reaction systems as providers of context sequences for other reaction systems. This is done as follows.

A reaction system $\mathcal{B}$ over a background set $S'$ can be used as a provider of context sequences for interactive processes in a rs $\mathcal{A}$ over a background set $S \subseteq S'$, by expanding $\mathcal{A}$ by $\mathcal{B}$ through the operation of union of reaction systems. One obtains in this way a reaction system $\mathcal{A}' = \mathcal{B} \oplus \mathcal{A}$ over $S'$ and the behaviour of $\mathcal{A}$ within $\mathcal{A}'$ is influenced by $\mathcal{B}$, as $\mathcal{B}$ (within context-independent behaviour of $\mathcal{A}'$) provides context sequences for $\mathcal{A}$. Since $\mathcal{A}$ is over $S$, in the current state $W'$ of $\mathcal{A}'$, $\mathcal{A}$ processes only $W' \cap S$ and it contributes to the successor state $res_{\mathcal{A}'}(W')$ of $W'$ entities from $S$ only. However $(res_{\mathcal{A}'}(W')) \cap S$ may contain entities which were not contributed by $\mathcal{A}$, because products of reactions in $\mathcal{B}$ may contain also entities from $S$ — this additional contribution by $\mathcal{B}$ to $(res_{\mathcal{A}'}(W')) \cap S$ is the context set provided by $\mathcal{B}$. This idea of $\mathcal{B}$ as a context provider for $\mathcal{A}$ will be formally introduced and discussed in this section.

To start with, we observe that given $\mathcal{A}'$ as above we cannot see from the definition of $\mathcal{A}'$ as a reaction system $\mathcal{A}' = (S', A')$ which reactions in $A'$ belong to $\mathcal{B}$ and which reactions in $\mathcal{A}'$ belong to $\mathcal{A}$; $\mathcal{A}'$ is just a set of reactions with no additional structure. Since we want to consider the behaviour of $\mathcal{A}$ in $\mathcal{A}'$ as the behaviour of $\mathcal{A}$ 'plugged into $\mathcal{B}$', we begin with formalising the concept of one reaction system ($\mathcal{A}$) being plugged into another reaction system ($\mathcal{B}$).

**Definition 8** *An* expander-plug pair*, abbreviated* ep pair*, is an ordered pair $\mathcal{Z} = (\mathcal{B}, \mathcal{A})$ such that $\mathcal{B}$ is a reaction system over a background set $S'$ and $\mathcal{A}$ is a reaction system over a background set $S$, where $S \subseteq S'$. The reaction system $\mathcal{A}' = \mathcal{B} \oplus \mathcal{A}$ is the* expansion of $\mathcal{A}$ by $\mathcal{B}$.

We refer to $\mathcal{B}$ as the *expander of $\mathcal{A}$ (in $\mathcal{Z}$)*, or just the *expander of $\mathcal{Z}$*, and to $\mathcal{A}$ as the *plug-into $\mathcal{B}$ (in $\mathcal{Z}$)*, or just the *plug of $\mathcal{Z}$*. We also say that $\mathcal{Z}$ *is over* $(S', S)$ and that $\mathcal{A}'$ is the *$\mathcal{Z}$-expansion*; note that $\mathcal{A}'$ is over $S'$.

For a rs $\mathcal{B}$ over a background set $S'$ and a background set $S$ such that $S \subseteq S'$, we use

- $\mathcal{P}_{\mathtt{ext}}(\mathcal{B}, S)$ to denote the family of all ep pairs $\mathcal{Z}$ such that $\mathcal{B}$ is the expander of $\mathcal{Z}$ and the plug of $\mathcal{Z}$ is a reaction system over $S$, and
- $\mathcal{F}_{\mathtt{ext}}(\mathcal{B}, S)$ to denote the family of all $\mathcal{Z}$-expansions of all $\mathcal{Z} \in \mathcal{P}_{\mathtt{ext}}(\mathcal{B}, S)$.

Thus, for a given rs $\mathcal{B}$ over a background set $S'$ and a background set $S \subseteq S'$, $\mathcal{B}$ is a 'universal' expander for all reaction systems over $S$. It is a 'plug-in' rs, where one can plug any rs $\mathcal{A}$ over $S$ obtaining in this way all ep pairs $\mathcal{Z} = (\mathcal{B}, \mathcal{A})$ in $\mathcal{P}_{\mathtt{ext}}(\mathcal{B}, S)$, which yield then all the expansions $\mathcal{A}'$ of $\mathcal{Z}$ in $\mathcal{F}_{\mathtt{ext}}(\mathcal{B}, S)$ (as illustrated in Figure 1).

The structure of the successor state $res_{\mathcal{A}'}(W')$ of a state $W'$ of the expansion $\mathcal{A}'$ of $\mathcal{A}$ by $\mathcal{B}$ is given by the following result, and it is illustrated in Figure 2.

**Theorem 9** *Let $\mathcal{Z} = (\mathcal{B}, \mathcal{A})$ be an ep pair over $(S', S)$ and let $\mathcal{A}'$ be the $\mathcal{Z}$-expansion. Then, for each $W' \subseteq S'$,*

*(1)* $res_{\mathcal{A}'}(W') = res_{\mathcal{B}}(W') \cup res_{\mathcal{A}}(W')$,
*(2)* $res_{\mathcal{A}}(W') = res_{\mathcal{A}}(W' \cap S)$ *and* $res_{\mathcal{A}}(W') \subseteq (res_{\mathcal{A}'}(W')) \cap S$, *and*
*(3)* $(res_{\mathcal{A}'}(W')) \cap S = ((res_{\mathcal{B}}(W') \cap S) \cup res_{\mathcal{A}}(W' \cap S)$.

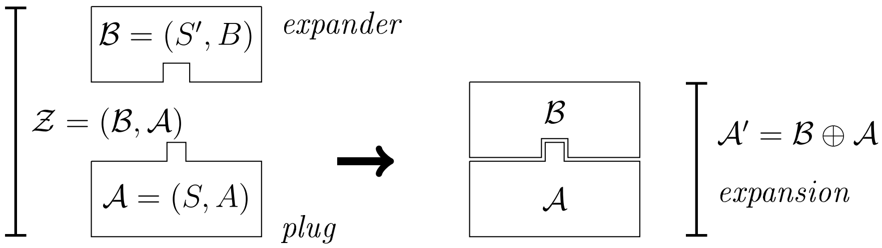


Fig. 1. An expander-plug pair $\mathcal{Z} = (\mathcal{B}, \mathcal{A})$ and its expansion.

**PROOF.** Let $\mathcal{B} = (S', B)$, $\mathcal{A} = (S, A)$, and $\mathcal{A}' = (S', A')$.

(1) This follows directly from the fact that $A' = B \cup A$.
(2) This follows from the fact that for each $a \in A$, $R_a \subseteq S$, $I_a \subseteq S$, and $P_a \subseteq S$.
(3) This follows from 1 and 2. □

Thus, if $W'$ is the current state of $\mathcal{A}'$, then

- by (1), the successor state $res_{\mathcal{A}'}(W')$ consists of (is the union of) contributions of reactions of $\mathcal{B}$ and contributions of reactions of $\mathcal{A}$,
- by (2), the reactions of $\mathcal{A}$ act only on $W' \cap S$ and they contribute only to $res_{\mathcal{A}'}(W')) \cap S$, and
- by (3), the part of the successor state which is included in $S$ consists of the contribution of $\mathcal{B}$ (applied to $W'$) and the contribution of $\mathcal{A}$ (applied to $W' \cap S$); it is important to notice that, in general, the latter does not have to be a subset of the former (i.e., $res_{\mathcal{A}}(W' \cap S) \subseteq (res_{\mathcal{B}}(W')) \cap S$ does not have to hold).

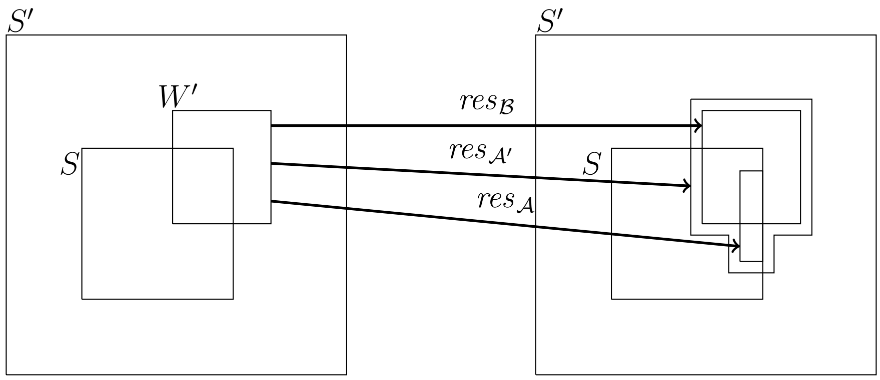


Fig. 2. The structure of the successor state $res_{\mathcal{A}'}(W')$ of a state $W'$ of the expansion $\mathcal{A}'$ of $\mathcal{A}$ by $\mathcal{B}$.

We proceed now to define interactive processes of ep pairs, beginning with defining their configurations.

Let $\mathcal{Z} = (\mathcal{B}, \mathcal{A})$ be an ep pair over $(S', S)$, for some background sets $S'$ and $S$, and let $\mathcal{A}' = (S', A')$ be the $\mathcal{Z}$-extension, i.e., $\mathcal{A}' = \mathcal{B} \oplus \mathcal{A}$. An $(S', S)$-*configuration* is a 4-tuple $f = (W', C, D, W)$ such that $W' \subseteq S'$ and $(C, D, W)$

is an $S$-configuration. Then, *configurations of* $\mathcal{Z}$ are just $(S', S)$-configurations, they depend *only* on the background sets of $\mathcal{B}$ and $\mathcal{A}$.

We interpret $W'$ and $D$ as the results of applying $res_{\mathcal{B}}$ and $res_{\mathcal{A}}$, respectively, to the previous state of the interactive process under consideration (if there was a previous state which is not the case for the initial configuration of the process), and $C$ as the current context for $\mathcal{A}$ (within $\mathcal{Z}$) provided by $\mathcal{B}$. Also, we interpret $W$ as the current state of $\mathcal{A}$ (within $\mathcal{Z}$). Since from the point of view of interactive processes the current state of a reaction system is the union of the result of applying its reactions and of the outside context, it is natural to require that $W = C \cup D$, as is the case for $S$-configurations.

This interpretation is formalised in the following definition of an interactive process in $\mathcal{Z}$.

**Definition 10** *Let* $\mathcal{Z} = (\mathcal{B}, \mathcal{A})$ *be an ep pair over* $(S', S)$ *and let* $\mathcal{A}'$ *be the* $\mathcal{Z}$*-expansion.*

*(1) An* interactive process in $\mathcal{Z}$ *is a sequence* $\pi' = f'_0, f'_1, \ldots, f'_n$ *of configurations of* $\mathcal{Z}$ *such that* $n \geq 1$*, and*

$$f'_0 = \begin{pmatrix} W'_0 \\ C_0 \\ D_0 \\ W_0 \end{pmatrix}, f'_1 = \begin{pmatrix} W'_1 & = res_{\mathcal{B}}(W'_0 \cup W_0) \\ C_1 & = W'_1 \cap S \\ D_1 & = res_{\mathcal{A}}(W_0) \\ W_1 & = C_1 \cup D_1 \end{pmatrix}, \ldots, f'_n = \begin{pmatrix} W'_n & = res_{\mathcal{B}}(W'_{n-1} \cup W_{n-1}) \\ C_n & = W'_n \cap S \\ D_n & = res_{\mathcal{A}}(W_{n-1}) \\ W_n & = C_n \cup D_n \end{pmatrix}.$$

*(2) The* interactive process $\pi$ in $\mathcal{A}$ induced by $\pi'$ *(also referred to as an* interactive process in $\mathcal{A}$ within $\mathcal{Z}$ and as an *interactive process of* $\mathcal{A}$ plugged into $\mathcal{B}$*) is the following sequence of configurations of* $\mathcal{A}$*:*

$$f_0 = \begin{pmatrix} C_0 \\ D_0 \\ W_0 \end{pmatrix}, f_1 = \begin{pmatrix} C_1 \\ D_1 \\ W_1 \end{pmatrix}, \ldots, f_n = \begin{pmatrix} C_n \\ D_n \\ W_n \end{pmatrix}.$$

*where, for each* $i \in \{0, \ldots, n\}$*,* $C_i$*,* $D_i$*, and* $W_i$ *are as in* $\pi'$*.*

The configuration $f'_0$ of $\pi'$ is referred to as the *initial configuration of* $\pi'$.

Note that

- a configuration of $\mathcal{Z}$ depends on the background sets $S'$ and $S$ only; however the configurations $f'_1, \ldots, f'_n$ in an interactive process $\pi'$ in $\mathcal{Z}$ depend also on the reactions of $\mathcal{B}$ and $\mathcal{A}$, as, for each $i \in \{1, \ldots, n\}$, $C_i = (res_{\mathcal{B}}(W'_{i-1} \cup W_{i-1})) \cap S$ and $D_i = res_{\mathcal{A}}(W_{i-1})$.
- it follows from the definition of configurations of $\mathcal{Z}$, that, for each $i \in \{0, \ldots, n\}$, $W_i \subseteq S$ and $W_i = C_i \cup D_i$, so that indeed $\pi$ is a sequence of configurations of $\mathcal{A}$,
- the proper context sequence of $\pi$ is $C_1 = W'_1 \cap S = (res_{\mathcal{B}}(W'_0 \cup W_0)) \cap S$, $C_2 = W'_2 \cap S = (res_{\mathcal{B}}(W'_1 \cup W_1)) \cap S, \ldots, C_n = W'_n \cap S = (res_{\mathcal{B}}(W'_{n-1} \cup W_{n-1})) \cap S$, so that indeed it is provided by $\mathcal{B}$, and
- since the initial configuration $f'_0$ can be an arbitrary configuration of $\mathcal{Z}$ (it

is not obtained in $\mathcal{Z}$ from a previous configuration), in general, the equality $C_0 = W'_0 \cap S$ does not have to hold.

It is important to realise that interactive processes in $\mathcal{Z}$ are *strongly deterministic* in the sense that for each configuration $f$ of $\mathcal{Z}$ there exists *exactly one* configuration $h$ of $\mathcal{Z}$ such that $f, h$ is an interactive process in $\mathcal{Z}$. Consequently, each interactive process $\pi = f_0, \ldots, f_n$ is uniquely determined by $f_0$ and the number of steps $n$. Clearly this strong determinism is a consequence of the deterministic character of reaction systems (the interactive processes in $\mathcal{Z}$ are determined by the result *functions* $res_{\mathcal{A}}$ and $res_{\mathcal{B}}$). We will use

- $PROC(\mathcal{Z})$ to denote the set of interactive processes in $\mathcal{Z}$,
- $PROC(\mathcal{A} \to \mathcal{Z})$ to denote the set of interactive processes in $\mathcal{A}$ within $\mathcal{Z}$,
- for a set $G$ of configurations of $\mathcal{Z}$, $PROC_G(\mathcal{Z})$ to denote the set of interactive processes in $\mathcal{Z}$ for which the initial configurations belong to $G$,
- for a set $G$ of configurations of $\mathcal{Z}$, $PROC_G(\mathcal{A} \to \mathcal{Z})$ to denote the set of interactive processes in $\mathcal{A}$ within $\mathcal{Z}$ induced by the interactive processes in $PROC_G(\mathcal{Z})$.

Also, for a rs $\mathcal{B}$ over a background set $S'$ and a background set $S \subseteq S'$, $INPROC(\mathcal{B}, S) = \{PROC(\mathcal{A} \to (\mathcal{B}, \mathcal{A})) \mid \mathcal{A} \text{ is a rs over } S\}$. Thus $INPROC(\mathcal{B}, S)$ is a family of sets of interactive processes in reaction systems over $S$ within ep pairs where $\mathcal{B}$ is the expander (hence the provider of context sequences).

Hence, from the point of view of regulating (limiting) control sequences of interactive processes of reaction systems, $\mathcal{B}$ is a universal provider of context sequences for all reaction systems over any subset $S$ of $S'$. These context sequences are 'well-structured', as they are provided (generated) by a reaction system, viz., $\mathcal{B}$.

We have defined interactive processes for ep pairs even though they are not reaction systems. But, considering interactive processes in $\mathcal{Z} = (\mathcal{B}, \mathcal{A})$ is a convenient way of distinguishing between the roles of $\mathcal{B}$ and $\mathcal{A}$ in the (context-independent) interactive processes in the $\mathcal{Z}$-expansion. In particular, this allows us to investigate the behaviour of $\mathcal{A}$ influenced by context sequences provided by $\mathcal{B}$. This is reflected in our terminology 'an interactive process in $\mathcal{Z}$' (rather than 'a context-independent interactive process in $\mathcal{Z}$', which would be justified by the fact that we do not provide *external* context sequences which would influence the behaviour of $\mathcal{Z}$), which really means that such a process demonstrates how the behaviour of $\mathcal{A}$ is influenced by (the context sequences provided by) $\mathcal{B}$ within $\mathcal{Z}$.

We conclude this section by pointing out that by using ep pairs in modelling the interaction of a rs with its environment (context) we obtain a 2*-way interaction*. For an ep pair $\mathcal{Z} = (\mathcal{B}, \mathcal{A})$, $\mathcal{B}$ influences $\mathcal{A}$ by providing context sequences for (interactive processes in) $\mathcal{A}$, but the functioning of $\mathcal{B}$ as a context provider is influenced by $\mathcal{A}$, as the contribution of $\mathcal{A}$ to the successor state of the current state of the $\mathcal{Z}$-expander does not have to be included in the contribution of $\mathcal{B}$ – see Theorem 9 and its illustration by Figure 2.

## 5 Providing context through controllers

In this section we consider context controllers as plug-in context providers for reaction systems. We begin by recalling from [26] the notion of a state-aware context controller.

**Definition 11** *A* state-aware context controller over a background set $S$ *is an edge-labelled graph* $\mathcal{E} = (Q, E, L)$ *such that* $Q$ *is a finite nonempty set,* $E \subseteq Q \times L \times Q$*,* $L = 2^S \times 2^S$*, and, for every* $q \in Q$ *and* $W \subseteq S$*, there are* $C \subseteq S$ *and* $q' \in Q$ *such that* $(q, (W, C), q') \in E$*.*

We refer to $Q$ as the set of *states of* $\mathcal{E}$, to $E$ as the set of *transitions of* $\mathcal{E}$, and to $L$ as the set of *labels of* $\mathcal{E}$. If, for every $q \in Q$ and $W \subseteq S$, there exists exactly one $C \subseteq S$ and one $q' \in Q$ such that $(q, (W, C), q') \in E$, then $\mathcal{E}$ is *strictly deterministic.*

We formalise now the intuitive notion of a rs plugged into a state-aware controller.

**Definition 12** *A* state-aware context controller, plug pair*, abbreviated sacp pair, is an ordered pair* $\mathcal{U} = (\mathcal{E}, \mathcal{A})$ *such that* $\mathcal{E}$ *is a state-aware controller over a background set* $S$ *and* $\mathcal{A}$ *is a rs over* $S$*.*

We refer to $\mathcal{E}$ as the *controller of* $\mathcal{A}$ *(in* $\mathcal{U}$*)*, or just as the *controller of* $\mathcal{U}$, and to $\mathcal{A}$ as the *plug-into* $\mathcal{E}$ *(in* $\mathcal{U}$*)*, or just the *plug of* $\mathcal{U}$. We also say that $\mathcal{U}$ *is over* $S$. The structure of a sacp pair is illustrated in Figure 3.

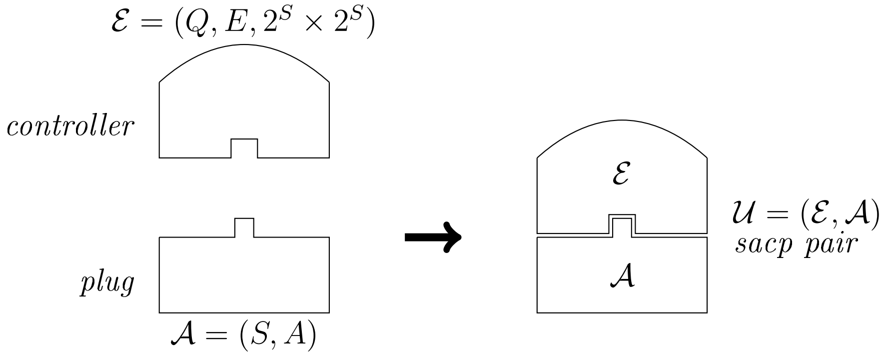


Fig. 3. The structure of a sacp pair.

A *configuration of* $\mathcal{E}$ is a 4-tuple $\varphi = (q, C, D, W)$ such that $q$ is a state of $\mathcal{E}$ and $(C, D, W)$ is an $S$-configuration. We proceed now to define interactive processes of sacp pairs, which will be sequences of configurations of their controllers.

Let $\mathcal{U} = (\mathcal{E}, \mathcal{A})$ be a sacp pair over a background set $S$. *Configurations* of $\mathcal{U}$ are just configurations of $\mathcal{E}$. For a configuration $\varphi = (q, C, D, W)$ of $\mathcal{U}$, we interpret $q$ as the current state of $\mathcal{E}$ controlling $\mathcal{A}$ while $\mathcal{A}$ is in its current state $W$. Also, we interpret $C$ as the current context of $\mathcal{A}$ provided by $\mathcal{E}$ when it made the transition to the current state from its previous state, and $D$ as the result of applying $res_{\mathcal{A}}$ to the previous state of $\mathcal{A}$ (obviously this interpretation of $C$ and $D$ applies only if $\varphi$ is not an initial configuration). This intuition corresponds well with the use of configurations of $\mathcal{U}$ in the definition of interactive processes in $\mathcal{U}$.

**Definition 13** *Let $\mathcal{U} = (\mathcal{E}, \mathcal{A})$ be a sacp pair over a background set $S$, where $\mathcal{E} = (Q, E, 2^S \times 2^S)$.*
*1. An* interactive process in $\mathcal{U}$ *is a sequence $\overline{\lambda} = \overline{h}_0, \overline{h}_1, \ldots, \overline{h}_n$ of configurations of $\mathcal{U}$, such that $n \geq 1$ and*

$$\overline{h}_0 = \begin{pmatrix} q_0 \\ C_0 \\ D_0 \\ W_0 \end{pmatrix}, \overline{h}_1 = \begin{pmatrix} q_1 \\ C_1 \\ D_1 \\ W_1 \end{pmatrix}, \ldots, \overline{h}_n = \begin{pmatrix} q_n \\ C_n \\ D_n \\ W_n \end{pmatrix},$$

*where for each $i \in \{1, \ldots, n\}$, $D_i = res_{\mathcal{A}}(W_{i-1})$ and $(q_{i-1}, (W_{i-1}, C_i), q_i) \in E$.*
*2. The* interactive process $\lambda$ in $\mathcal{A}$ induced by $\overline{\lambda}$ *(also referred to as an* interactive process in $\mathcal{A}$ within $\mathcal{U}$ *and as* an interactive process of $\mathcal{A}$ plugged-into $\mathcal{E}$*) is the sequence of configurations of $\mathcal{A}$*

$$h_0 = \begin{pmatrix} C_0 \\ D_0 \\ W_0 \end{pmatrix}, h_1 = \begin{pmatrix} C_1 \\ D_1 \\ W_1 \end{pmatrix}, \ldots, h_n = \begin{pmatrix} C_n \\ D_n \\ W_n \end{pmatrix},$$

*where, for each $i \in \{0, \ldots, n\}$, $C_i$, $D_i$, and $W_i$ are as in $\overline{\lambda}$.*

Note that

- it follows from the definition of configurations of $\mathcal{U}$ that, for each $i \in \{0, \ldots, n\}$, $W_i \subseteq S$ and $W_i = C_i \cup D_i$, so that indeed $\lambda$ is a sequence of configurations of $\mathcal{A}$,
- the proper context sequence $C_1, \ldots, C_n$ is provided by $\mathcal{E}$ (through consecutive transitions $e_1, \ldots, e_n$ of $\mathcal{E}$ applied to consecutive configurations $\overline{h}_0, \ldots, \overline{h}_{n-1}$ of $\overline{\lambda}$), and
- the condition from Definition 11 requiring that for each $q \in Q$ and $W \subseteq S$ there are $C \subseteq S$ and $q' \in Q$ such that $(q, (W, C), q')$ is a transition of $\mathcal{E}$, ensures that $\mathcal{E}$ cannot block $\mathcal{A}$ within $\mathcal{U}$: at each state $q \in Q$ and each state $W$ of $\mathcal{A}$, $\mathcal{E}$ provides at least one context set for $\mathcal{A}$.

Thus, indeed, $\mathcal{E}$ is a universal provider of context sequences for all reaction systems over $S$ (plugged into $\mathcal{E}$). These context-sequences are 'well-structured', as they are generated by a labelled graph (labelled transition system), viz., $\mathcal{E}$. The role of $\mathcal{E}$ as a context provider for a rs $\mathcal{A}$ within the sacp pair $\mathcal{U} = (\mathcal{E}, \mathcal{A})$ is illustrated in Figure 4.

Obviously, since the identities/names of states of $\mathcal{E}$ are not essential for $\mathcal{E}$ in its role as context provider for $\mathcal{A}$ (one can change $Q$ to an 'isomorphic' set of states $Q'$), *we may assume that the sets $Q$ and $S$ are disjoint.*

Note that also (see the last paragraph of Section 4) sacp pairs $\mathcal{U} = (\mathcal{E}, \mathcal{A})$ model a 2-*way interaction* with the environment: $\mathcal{E}$ influences (the behaviour

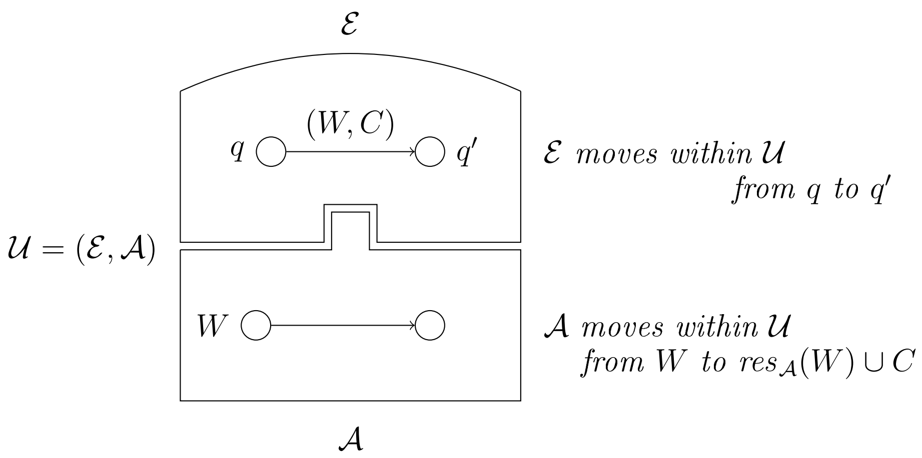


Fig. 4. $\mathcal{E}$ as a plug-in context-provider for $\mathcal{A}$ in $\mathcal{U}$.

of) $\mathcal{A}$ by providing context sequences for $\mathcal{A}$, but the functioning of $\mathcal{E}$ as a context provider is influenced by $\mathcal{A}$ which through its current state ($W$) influences transitions that $\mathcal{E}$ can take in its current state ($q$).

We will use $PROC(\mathcal{A} \to \mathcal{U})$ to denote the set of all interactive processes in $\mathcal{A}$ within $\mathcal{U}$ and $STS(\mathcal{A} \to \mathcal{U})$ to denote the set of all state sequences of all interactive processes in $PROC(\mathcal{A} \to \mathcal{U})$.

Also, $INPROC(\mathcal{E}) = \{PROC(\mathcal{A} \to (\mathcal{E}, \mathcal{A})) \mid \mathcal{A}$ is a rs over $S\}$. Thus, $INPROC(\mathcal{E})$ is a family of sets of interactive processes in reaction systems over $S$ within sacp pairs where $\mathcal{E}$ is the controller (hence the provider of context sequences).

When a state-aware controller $\mathcal{E} = (Q, E, 2^S \times 2^S)$ over $S$ is such that, for all $q, q' \in Q$, each $C \subseteq S$, and all $W_1, W_2 \subseteq S$, $(q, (W_1, C), q') \in E$ if and only if $(q, (W_2, C), q') \in E$, then, in fact, for all reaction systems $\mathcal{A}$, transitions from the current configuration of the sacp pair $\mathcal{U} = (\mathcal{E}, \mathcal{A})$ to the successor configuration in any interactive process of $\mathcal{U}$ are *independent* of the current state of $\mathcal{A}$. Therefore we call such controllers *state-oblivious*. In fact, in this case we can consider the set of edges $E$ to be a subset of $Q \times 2^S \times Q$, and so state oblivious controllers can be (re)defined as follows.

**Definition 14** *A* state-oblivious context controller *over a background set $S$ is an edge-labelled directed graph $\mathcal{C} = (Q, E, 2^S)$ such that $Q$ is a finite nonempty set of* states of $\mathcal{C}$, *$2^S$ is the set of labels of $\mathcal{C}$, and $E \subseteq Q \times 2^S \times Q$ is the set of labelled edges, the* transitions of $\mathcal{C}$, *satisfying the following condition: for every node $q \in Q$, there exists a labelled edge outgoing from $q$.*

Now, when we consider an interactive process $\overline{\lambda}$ in a sacp pair $\mathcal{U} = (\mathcal{C}, \mathcal{A})$, see Definition 12, where $\mathcal{C}$ is a state-oblivious controller, then for each $i \in \{1, \ldots, n\}$, $D_i = res_{\mathcal{A}}(W_{i-1})$ and $(q_{i-1}, C_i, q_i) \in E$ – thus $q_i$ is not dependent on $W_{i-1}$ anymore.

Note that sacp pairs $\mathcal{U} = (\mathcal{C}, \mathcal{A})$ model only 1-*way interaction* with the environment, as $\mathcal{A}$ does not influence $\mathcal{C}$ in its functioning as provider of context sequences for $\mathcal{A}$.

Retaining the terminology for state-aware controllers, we call $\mathcal{C}$ *strictly deterministic* if for each $q \in Q$ there exists *exactly one* labelled edge in $E$ outgoing from $q$.

## 6 Expanders vs state-aware context controllers

In this section we demonstrate that expanders and strictly deterministic state-aware context controllers are equivalent as plug-in context providers, as they can simulate each other in the way stated in the two theorems of this section.

As we have pointed out in Section 3, interactive processes in ep pairs are strongly deterministic in the sense that, for each ep pair $\mathcal{Z}$, each configuration of $\mathcal{Z}$ has a unique successor (and so each $n$-step interactive process in $\mathcal{Z}$ is uniquely determined by its initial configuration and $n$). This property does not hold for sacp pairs $(\mathcal{E}, \mathcal{A})$, because $\mathcal{E} = (Q, E, 2^S \times 2^S)$ may have, for a given $q \in Q$ and $W \subseteq S$, two *different* edges $e_1 = (q, (W, C_1), q_1)$ and $e_2 = (q, (W, C_2), q_2)$. However, if $\mathcal{E}$ is strictly deterministic, then this cannot happen. Therefore, in order to have 'fair' comparisons of expanders and controllers, we will consider now only strictly deterministic state-aware controllers.

We begin by demonstrating that expanders can be simulated by state-aware context controllers.

**Theorem 15** *Let $\mathcal{B}$ be a rs over a background set $S'$ and let $S$ be a background set such that $S \subseteq S'$. There exists a strictly deterministic state-aware context controller $\mathcal{E}$ over $S$, such that, for every rs $\mathcal{A}$ over $S$, $PROC(\mathcal{A} \to (\mathcal{B}, \mathcal{A})) = PROC(\mathcal{A} \to (\mathcal{E}, \mathcal{A}))$.*

**PROOF.** Let $\mathcal{B} = (S', B)$ be a reaction system and let $S$ be a background set such that $S \subseteq S'$. Then let $\mathcal{E} = (Q, E, 2^S \times 2^S)$ be the strictly deterministic state-aware context controller over $S$ such that:
$Q = \{[Z] \mid Z \in 2^{S'}\}$ and
$E = \{([Z], (W, C), [Z']) \mid Z' = res_{\mathcal{B}}(Z \cup W) \text{ and } C = Z' \cap S\}$.

Since $res_{\mathcal{B}}$ is a function, for each $[Z] \in Q$ and $W \subseteq S$, there exists exactly one $Z'$ such that $([Z], (W, C), [Z']) \in E$ for some $C \subseteq S$, and since $C$ is defined by $C = Z' \cap S$, there exists exactly one such $C$. Thus, indeed, $\mathcal{E}$ is strictly deterministic.

Let $\mathcal{A} = (S, A)$ be a rs over $S$. Then, let $\mathcal{Z}$ be the ep pair $\mathcal{Z} = (\mathcal{B}, \mathcal{A})$ and let $\mathcal{U}$ be the sacp pair $\mathcal{U} = (\mathcal{E}, \mathcal{A})$.

The intuition behind the construction of $\mathcal{E}$ is that the states of $\mathcal{E}$ are 'names' for the states of $\mathcal{B}$ (i.e., the state $[Z]$ of $\mathcal{E}$ 'remembers' the state $Z$ of $\mathcal{B}$) and so the transitions of $\mathcal{E}$ simulate the functioning of $\mathcal{B}$ within $\mathcal{Z}$.

We prove now the statement of the theorem by proving two inclusions.

(I)$PROC(\mathcal{A} \to \mathcal{Z}) \subseteq PROC(\mathcal{A} \to \mathcal{U})$.

Let $\pi \in PROC(\mathcal{A} \to \mathcal{Z})$ and let $\pi'$ be an interactive process of $\mathcal{Z}$ such that $\pi$

is induced by $\pi'$. Thus, $\pi' = f'_0, f'_1, \ldots, f'_n$, where $n \geq 1$, and the consecutive configurations of $\pi'$ are:

$$f'_0 = \begin{pmatrix} W'_0 \\ C_0 \\ D_0 \\ W_0 \end{pmatrix}, f'_1 = \begin{pmatrix} W'_1 = res_{\mathcal{B}}(W'_0 \cup W_0) \\ C_1 \; = W'_1 \cap S \\ D_1 \; = res_{\mathcal{A}}(W_0) \\ W_1 = C_1 \cup D_1 \end{pmatrix}, \ldots, f'_n = \begin{pmatrix} W'_n = res_{\mathcal{B}}(W'_{n-1} \cup W_{n-1}) \\ C_n \; = W'_n \cap S \\ D_n \; = res_{\mathcal{A}}(W_{n-1}) \\ W_n = C_n \cup D_n \end{pmatrix}.$$

Consequently $\pi = f_0, f_1, \ldots, f_n$, with

$$f_0 = \begin{pmatrix} C_0 \\ D_0 \\ W_0 \end{pmatrix}, f_1 = \begin{pmatrix} C_1 \\ D_1 \\ W_1 \end{pmatrix}, \ldots, f_n = \begin{pmatrix} C_n \\ D_n \\ W_n \end{pmatrix},$$

where, for each $i \in \{1, \ldots, n\}$, $C_i$, $D_i$, and $W_i$ are as in $\pi'$.

Let, then $\overline{\lambda} = \overline{h}_0, \overline{h}_1, \ldots, \overline{h}_n$ be the interactive process in $\mathcal{U}$, where, for each $i \in \{1, \ldots, n\}$, $\overline{h}_i = (\overline{q}_i, \overline{C}_i, \overline{D}_i, \overline{W}_i)$ and $\overline{h}_0 = ([W'_0], C_0, D_0, W_0)$.

It follows then from the definition of $\mathcal{E}$ (and from the definition of an interactive process in an sacp pair) that the consecutive configurations of $\overline{\lambda}$ are of the form

$$\overline{h}_0 = \begin{pmatrix} \overline{q}_0 \; = [W'_0] \\ \overline{C}_0 \; = C_0 \\ \overline{D}_0 \; = D_0 \\ \overline{W}_0 = W_0 \end{pmatrix}, \overline{h}_1 = \begin{pmatrix} \overline{q}_1 \; = [W'_1] \\ \overline{C}_1 \; = W'_1 \cap S \\ \overline{D}_1 \; = res_{\mathcal{A}}(W_0) \\ \overline{W}_1 = C_1 \cup D_1 \end{pmatrix}, \ldots, \overline{h}_n = \begin{pmatrix} \overline{q}_n \; = [W'_n] \\ \overline{C}_n \; = W'_n \cap S \\ \overline{D}_n \; = res_{\mathcal{A}}(W_{n-1}) \\ \overline{W}_n = C_n \cup D_n \end{pmatrix}.$$

Hence the interactive process $\lambda$ in $\mathcal{A}$ induced by $\overline{\lambda}$ is of the form $\lambda = \overline{h}_0, \overline{h}_1, \ldots, \overline{h}_n$, where, for each $i \in \{0, \ldots, n\}$, $\overline{h}_i = (\overline{C}_i, \overline{D}_i, \overline{W}_i)$ with $\overline{C}_i$, $\overline{D}_i$, and $\overline{W}_i$ defined as above for $\overline{\lambda}$. Since, for each $i \in \{0, \ldots, n\}$, $\overline{C}_i = C_i$, $\overline{D}_i = D_i$, and $\overline{W}_i = W_i$, we obtain $\lambda = \pi$. Consequently, $\pi \in PROC(\mathcal{A} \to \mathcal{U})$.

Since $\pi$ was an arbitrary interactive process in $\mathcal{A}$ within $\mathcal{Z}$, it follows that $PROC(\mathcal{A} \to \mathcal{Z}) \subseteq PROC(\mathcal{A} \to \mathcal{U})$.

(II) $PROC(\mathcal{A} \to \mathcal{U}) \subseteq PROC(\mathcal{A} \to \mathcal{Z})$.

Let $\lambda \in PROC(\mathcal{A} \to \mathcal{U})$ and let $\overline{\lambda}$ be an interactive process of $\mathcal{U}$ such that $\lambda$ is induced by $\overline{\lambda}$. Thus, $\overline{\lambda} = \overline{h}_0, \overline{h}_1, \ldots, \overline{h}_n$, where $n \geq 1$, and the consecutive configurations of $\overline{\lambda}$ are:

$$\overline{h}_0 = \begin{pmatrix} q_0 = [W'_0] \\ C_0 \\ D_0 \\ W_0 = C_0 \cup D_0 \end{pmatrix}, \overline{h}_1 = \begin{pmatrix} q_1 \; = [W'_1 = res_{\mathcal{B}}(W'_0 \cup W_0)] \\ C_1 \; = W'_1 \cap S \\ D_1 \; = res_{\mathcal{A}}(W_0) \\ W_1 = C_1 \cup D_1 \end{pmatrix}$$

$$, \ldots, \overline{h}_n = \begin{pmatrix} q_n \; = [W'_n = res_{\mathcal{B}}(W'_{n-1} \cup W_{n-1})] \\ C_n \; = W'_n \cap S \\ D_n \; = res_{\mathcal{A}}(W_{n-1}) \\ W_n = C_n \cup D_n \end{pmatrix}$$

for some $W_0' \subseteq S'$ and $C_0, D_0 \subseteq S$. Consequently, $\lambda = h_0, h_1, \ldots, h_n$, with

$$h_0 = \begin{pmatrix} C_0 \\ D_0 \\ W_0 \end{pmatrix}, h_1 = \begin{pmatrix} C_1 \\ D_1 \\ W_1 \end{pmatrix}, \ldots, h_n = \begin{pmatrix} C_n \\ D_n \\ W_n \end{pmatrix},$$

where, for each $i \in \{0, \ldots, n\}$, $C_i$, $D_i$, and $W_i$ are as in $\overline{\lambda}$ above.

Consider now the sequence $\pi' = f_0', f_1', f_1', \ldots, f_n'$ of configurations of $\mathcal{Z}$:

$$f_0' = \begin{pmatrix} W_0' \\ C_0 \\ D_0 \\ W_0 \end{pmatrix}, f_1' = \begin{pmatrix} W_1' &= res_{\mathcal{B}}(W_0' \cup W_0) \\ C_1 &= W_1' \cap S \\ D_1 &= res_{\mathcal{A}}(W_0) \\ W_1 &= C_1 \cup D_1 \end{pmatrix}, \ldots, f_n' = \begin{pmatrix} W_n' &= res_{\mathcal{B}}(W_{n-1}' \cup W_{n-1}) \\ C_n &= W_n' \cap S \\ D_n &= res_{\mathcal{A}}(W_{n-1}) \\ W_n &= C_n \cup D_n \end{pmatrix}.$$

It follows from the definition of $\mathcal{E}$ that $\pi' \in PROC(\mathcal{Z})$. Clearly, the interactive process $\pi$ in $\mathcal{A}$ induced by $\pi'$ is such that $\pi = \lambda$. Thus $\lambda \in PROC(\mathcal{A} \to \mathcal{Z})$.

Since $\lambda$ was an arbitrary interactive process in $PROC(\mathcal{A} \to \mathcal{U})$, it follows that $PROC(\mathcal{A} \to \mathcal{U}) \subseteq PROC(\mathcal{A} \to \mathcal{Z})$.

The theorem follows now from (I) and (II). □

We move now to demonstrate that strictly deterministic state-aware controllers can be simulated by expanders.

**Theorem 16** *Let $\mathcal{E}$ be a strictly-deterministic state-aware context controller over a background set $S$. There exists a rs $\mathcal{B}$ over a background set $S'$ with $S \subseteq S'$ and a set $G$ of $(S', S)$-configurations such that for every rs $\mathcal{A}$ over $S$, $PROC(\mathcal{A} \to (\mathcal{E}, \mathcal{A})) = PROC_G(\mathcal{A} \to (\mathcal{B}, \mathcal{A}))$.*

**PROOF.** Let $\mathcal{E} = (Q, E, 2^S \times 2^S)$ be a strictly deterministic state-aware controller over a background set $S$ – we assume that $Q \cap S = \varnothing$. Then:
(1) let $\mathcal{B} = (S', B)$ be a rs such that $S' = Q \cup S$ and $B = \{b(e) \mid e \in E\}$, where, for each $e = (q, (W, C), q') \in E$, $b(e) = (\{q\} \cup W, S' \setminus (\{q\} \cup W), \{q'\} \cup C)$, and
(2) let $G$ be the set of $(S', S)$-configurations $(W', C_0, D_0, W_0)$ of $\mathcal{B}$ such that each $W'$ is a singleton subset of $Q$, i.e., each $W' = \{q\}$ for some $q \in Q$.

Let now $\mathcal{A} = (S, A)$ be a rs over $S$, and accordingly, let $\mathcal{U}$ be the sacp pair $\mathcal{U} = (\mathcal{E}, \mathcal{A})$ and let $\mathcal{Z}$ be the ep pair $\mathcal{Z} = (\mathcal{B}, \mathcal{A})$.

Note the use of complementary reactions in $B$. For $e = (q, (W, C), q') \in E$, the reaction $b(e)$ is complementary: it is enabled only in the state $\{q\} \cup W$ of the $\mathcal{Z}$-extension $\mathcal{A}'$ and contributes $\{q'\} \cup C$ to the successor of this state in $\mathcal{A}'$, simulating in this way exactly the transition of $\mathcal{E}$ in its state $q$ when $\mathcal{A}$ (plugged into $\mathcal{E}$) is in state $W$.

We prove the statement of the theorem by proving two inclusions.

(I) $PROC(\mathcal{A} \to \mathcal{U}) \subseteq PROC_G(\mathcal{A} \to \mathcal{Z})$.

Let $\lambda \in PROC(\mathcal{A} \to \mathcal{U})$ and let $\overline{\lambda}$ be an interactive process of $\mathcal{U}$ such that $\lambda$ is induced by $\overline{\lambda}$. Thus $\overline{\lambda} = \overline{h}_0, \overline{h}_1, \ldots, \overline{h}_n$ for some $n \geq 1$, with the consecutive configurations of $\overline{\lambda}$ of the form:

$$\overline{h}_0 = \begin{pmatrix} \overline{q}_0 & & \\ \overline{C}_0 & & \\ \overline{D}_0 & & \\ \overline{W}_0 = \overline{C}_0 \cup \overline{D}_0 & & \end{pmatrix}, \overline{h}_1 = \begin{pmatrix} \overline{q}_1 & \\ \overline{C}_1 & \\ \overline{D}_1 & = res_{\mathcal{A}}(\overline{W}_0) \\ \overline{W}_1 & = \overline{C}_1 \cup \overline{D}_1 \end{pmatrix}, \ldots, \overline{h}_n = \begin{pmatrix} \overline{q}_n & \\ \overline{C}_n & \\ \overline{D}_n & = res_{\mathcal{A}}(\overline{W}_{n-1}) \\ \overline{W}_n & = \overline{C}_n \cup \overline{D}_n \end{pmatrix}.$$

where, for each $i \in \{1, \ldots, n\}$, $(\overline{q}_{i-1}, (\overline{W}_{i-1}, \overline{C}_i), \overline{q}_i) \in E$.

Thus $\lambda = h_0, h_1, \ldots, h_n$ with

$$h_0 = \begin{pmatrix} \overline{C}_0 \\ \overline{D}_0 \\ \overline{W}_0 \end{pmatrix}, h_1 = \begin{pmatrix} \overline{C}_1 \\ \overline{D}_1 \\ \overline{W}_1 \end{pmatrix}, \ldots, h_n = \begin{pmatrix} \overline{C}_n \\ \overline{D}_n \\ \overline{W}_n \end{pmatrix},$$

where for each $i \in \{0, \ldots, n\}$, $\overline{C}_i$, $\overline{D}_i$, and $\overline{W}_i$ are as in $\overline{\lambda}$ above.

Let then $\pi' = f'_0, f'_1, \ldots, f'_n$ be the interactive process in $\mathcal{Z}$ such that $f'_0 = (\{\overline{q}_0\}, \overline{C}_0, \overline{D}_0, \overline{W}_0)$.

Since, for each $i \in \{1, \ldots, n\}$, $(\overline{q}_{i-1}, (\overline{W}_{i-1}, \overline{C}_i), \overline{q}_i) \in E$, it follows then from the definition of reactions in $B$ that the consecutive configurations of $\pi'$ are:

$$f'_0 = \begin{pmatrix} \{\overline{q}_0\} \\ \overline{C}_0 \\ \overline{D}_0 \\ \overline{W}_0 \end{pmatrix}, f'_1 = \begin{pmatrix} \{\overline{q}_1\} \cup \overline{C}_1 \\ \overline{C}_1 \\ \overline{D}_1 \\ \overline{W}_1 \end{pmatrix}, \ldots, f'_n = \begin{pmatrix} \{\overline{q}_n\} \cup \overline{C}_n \\ \overline{C}_n \\ \overline{D}_n \\ \overline{W}_n \end{pmatrix}.$$

Hence the interactive process $\pi$ in $\mathcal{A}$ induced by $\pi'$ is of the form $\pi = f_0, f_1, \ldots, f_n$, where for each $i \in \{0, \ldots, n\}$, $f_i = (C_i, D_i, W_i)$ with $C_i = \overline{C}_i$, $D_i = \overline{D}_i$, and $W_i = \overline{W}_i$, where $\overline{C}_i$, $\overline{D}_i$, and $\overline{W}_i$ are as defined above for $\pi'$. Thus $\pi = \lambda$, and since $f'_0 \in G$, we obtain $\lambda \in PROC_G(\mathcal{A} \to \mathcal{Z})$.

Since $\pi$ was an arbitrary interactive process in $\mathcal{A}$ within $\mathcal{U}$, it follows that $PROC(\mathcal{A} \to \mathcal{U}) \subseteq PROC_G(\mathcal{A} \to \mathcal{Z})$.

(II) $PROC_G(\mathcal{A} \to \mathcal{Z}) \subseteq PROC(\mathcal{A} \to \mathcal{U})$.

Let $\pi \in PROC_G(\mathcal{A} \to \mathcal{Z})$ and let $\pi' \in PROC_G(\mathcal{Z})$ be such that $\pi$ is induced by $\pi'$ in $\mathcal{Z}$. Thus, $\pi' = f'_0, f'_1, \ldots, f'_n$, where $n \geq 1$, $f'_0 = (\{q_0\}, C_0, D_0, W_0)$ for some $q_0 \in Q$ (because $f'_0 \in G$), and the consecutive configurations of $\pi'$ are:

$$f'_0 = \begin{pmatrix} \{q_0\} \\ C_0 \\ D_0 \\ W_0 = C_0 \cup D_0 \end{pmatrix}, f'_1 = \begin{pmatrix} \{q_1\} \cup C_1 \\ C_1 \\ D_1 = res_{\mathcal{A}}(W_0) \\ W_1 = C_1 \cup D_1 \end{pmatrix}, \ldots, f'_n = \begin{pmatrix} \{q_n\} \cup C_n \\ C_n \\ D_n = res_{\mathcal{A}}(W_{n-1}) \\ W_n = C_n \cup D_n \end{pmatrix},$$

where, for each $i \in \{1, \ldots, n\}$, $(q_{i-1}, (W_{i-1}, C_i), q_i) \in E$.

Consequently $\pi = f_0, f_1, \ldots, f_n$, with

$$f_0 = \begin{pmatrix} C_0 \\ D_0 \\ W_0 \end{pmatrix}, f_1 = \begin{pmatrix} C_1 \\ D_1 \\ W_1 \end{pmatrix}, \ldots, f_n = \begin{pmatrix} C_n \\ D_n \\ W_n \end{pmatrix},$$

where, for each $i \in \{0, \ldots, n\}$, $C_i$, $D_i$, and $W_i$ are as in $\pi'$.

Let then $\overline{\lambda} = \overline{h}_0, \overline{h}_1, \ldots, \overline{h}_n$, be the interactive process in $\mathcal{U}$, where $\overline{h}_0 = (q_0, C_0, D_0, W_0)$. Since, for each $i \in \{1, \ldots, n\}$, $(q_{i-1}, (W_{i-1}, C_i), q_i) \in E$, the consecutive configurations of $\overline{\lambda}$ are:

$$\overline{h}_0 = \begin{pmatrix} q_0 \\ C_0 \\ D_0 \\ W_0 = C_0 \cup D_0 \end{pmatrix}, \overline{h}_1 = \begin{pmatrix} q_1 \\ C_1 \\ D_1 = res_{\mathcal{A}}(W_0) \\ W_1 \end{pmatrix}, \ldots, \overline{h}_n = \begin{pmatrix} q_n \\ C_n \\ D_n = res_{\mathcal{A}}(W_{n-1}) \\ W_n \end{pmatrix}.$$

Hence, the interactive process $\lambda$ in $\mathcal{A}$ induced by $\overline{\lambda}$ is of the form $\lambda = h_0, h_1, \ldots, h_n$, where for each $i \in \{1, \ldots, n\}$, $h_i = (C_i, D_i, W_i)$. Thus, $\lambda = \pi$ and consequently $\pi \in PROC(\mathcal{A} \to \mathcal{U})$.

Since $\pi$ was an arbitrary interactive process in $PROC_G(\mathcal{A} \to \mathcal{Z})$, it follows that $PROC_G(\mathcal{A} \to \mathcal{Z}) \subseteq PROC(\mathcal{A} \to \mathcal{U})$.

The theorem now follows from (I) and (II). □

Considering $PROC_G(\mathcal{A} \to \mathcal{Z})$ (rather than just $PROC(\mathcal{A} \to \mathcal{Z})$) in the above theorem was necessary, as an arbitrary state of $\mathcal{B}$ (a subset of $S'$) can contain more than one state of $Q$ (or none at all), while $\mathcal{E}$ is always in one state.

## 7 Comparison with state-oblivious context controllers

In this section we investigate state-oblivious context controllers as plug-in context providers.

We begin by relating them to state-aware context controllers.

Now the notion of a sacp pair is modified as follows. An ordered pair $\mathcal{U} = (\mathcal{C}, \mathcal{A})$ such that $\mathcal{C}$ is a state-oblivious controller over a background set $S$ and $\mathcal{A}$ is a rs over $S$, is referred to as a *state-oblivious context controller, plug pair*, abbreviated socp pair.

Since for a state-oblivious context controller $\mathcal{C}$ and each $\mathcal{A}$, controller $\mathcal{C}$ moves within the socp pair $\mathcal{U} = (\mathcal{C}, \mathcal{A})$ from its current state to the successor state *independently* of the current state of $\mathcal{A}$, the following result holds.

**Lemma 17** *Let $\mathcal{C}$ be a strictly deterministic state-oblivious controller over a background set $S$. Let $\mathcal{A}$ be a rs over $S$ and let $\mathcal{U}$ be the socp pair $(\mathcal{C}, \mathcal{A})$. Let $f_0 = (q_0, C_0, D_0, W_0)$ and $h_0 = (\overline{q}_0, \overline{C}_0, \overline{D}_0, \overline{W}_0)$ be $\mathcal{U}$-configurations such* *that* $q_0 = \overline{q}_0$ *and* $W_0 \neq \overline{W}_0$*. If* $\pi \in PROC(\mathcal{U})$ *is such that* $\pi = f_0, f_1$*, where* $f_1 = (q_1, C_1, D_1, W_1)$*, then also* $\overline{\pi} \in PROC(\mathcal{U})$*, where* $\overline{\pi} = h_0, h_1$ *with* $h_1 = (\overline{q}_1, \overline{C}_1, \overline{D}_1, \overline{W}_1)$ *such that* $q_1 = \overline{q}_1$ *and* $\overline{C}_1 = C_1$*.*

Thus even though the predecessor states ($W_0$ and $\overline{W}_0$) are not equal, the current contexts ($C_1$ and $\overline{C}_1$) are equal.

We demonstrate now that even one-state strictly deterministic state-aware controllers are 'stronger' than arbitrary state-oblivious controllers.

**Theorem 18** *There exists a one-state strictly deterministic state-aware context controller* $\mathcal{E}$ *over* $S$ *such that, for no state-oblivious controller* $\mathcal{C}$ *over* $S$*,* $INPROC(\mathcal{E}) = INPROC(\mathcal{C})$*.*

**PROOF.** Let $\mathcal{E} = (Q, E, 2^S \times 2^S)$ be the strictly deterministic state-aware controller such that $|Q| = 1$ with $Q = \{q\}$ and $E = \{(q, (Z, Z), q) \mid Z \subseteq S\}$. Thus for each rs $\mathcal{A}$ over $S$, all interactive processes in $PROC(\mathcal{A} \to (\mathcal{E}, \mathcal{A}))$ are of the form $\pi = f_0, \ldots, f_n$, for some $n \geq 1$, where

$$f_0 = \begin{pmatrix} q \\ \\ C_0 \\ D_0 \\ W_0 = C_0 \cup D_0 \end{pmatrix}, f_1 = \begin{pmatrix} q \\ \\ C_1 = W_0 \\ D_1 = res_{\mathcal{A}}(W_0) \\ W_1 = D_1 \cup C_1 \end{pmatrix}, \ldots, f_n = \begin{pmatrix} q \\ \\ C_n = W_{n-1} \\ D_n = res_{\mathcal{A}}(W_{n-1}) \\ W_n = D_n \cup C_n \end{pmatrix},$$

for some $C_0, D_0 \subseteq S$. Thus, each subsequent context set ($C_i$) of $\pi$ equals the previous state ($W_{i-1}$) of $\pi$ – such interactive processes could be called *context repeats state* interactive processes.

By Lemma 17, for no strictly deterministic state-oblivious controller $\mathcal{C}$, $\pi \in INPROC(\mathcal{C})$. □

We recall now the notion of an extension of reaction system (see, e.g., [25]) and then discuss its use as plug-in context provider.

**Definition 19** *Let* $\mathcal{A} = (S, A)$ *and* $\mathcal{A}' = (S', A')$ *be reaction systems. Then* $\mathcal{A}'$ *is an* extension of $\mathcal{A}$ *(or* $\mathcal{A}$ *is* embedded in $\mathcal{A}'$*) if* $S \subseteq S'$ *and* $A \subseteq A'$*.*

A natural way to view an extension of $\mathcal{A}$ by $\mathcal{A}'$ as plugging $\mathcal{A}$ into $\mathcal{A}'$ is as follows. A *subset ep pair* is an ep pair $\mathcal{Z} = (\mathcal{B}, \mathcal{A})$, where $\mathcal{B} = (S', B)$ and $\mathcal{A} = (S, A)$ are such that $A \subseteq B$. In this case, the expansion of $\mathcal{A}$ by $\mathcal{B}$ is $\mathcal{A}' = \mathcal{B} \oplus \mathcal{A} = \mathcal{B}$ and it is called simply *an extension* of $\mathcal{A}$.

Hence, in considering subset ep pairs one restricts the use of the expander $\mathcal{B}$ (as the context provider) by allowing to plug into it *only* reaction systems $\mathcal{A}$ which are already embedded in it. Thus, the expansion of $\mathcal{A}$ by $\mathcal{B}$ is $\mathcal{B}$ itself. Hence, using the notation introduced after Definition 8, we consider now

- a restriction of $\mathcal{P}_{ext}(\mathcal{B}, S)$, viz., $\overline{\mathcal{P}}_{ext}(\mathcal{B}, S)$, which is the family of all subset ep pairs $\mathcal{Z}$ such that $\mathcal{B}$ is the expander of $\mathcal{Z}$ and the plug of $\mathcal{Z}$ is a rs over $S$, and accordingly

- $\overline{\mathcal{F}}_{ext}(\mathcal{B}, S)$, which is the family of all $\mathcal{Z}$-expansions of all $\mathcal{Z} \in \overline{\mathcal{P}}_{ext}(\mathcal{B}, S)$; clearly, now $\overline{\mathcal{F}}_{ext}(\mathcal{B}, S)$ contains only one rs, viz., $\mathcal{B}$.

From the technical point of view of analysing interactive processes of $\mathcal{A}$ within $\mathcal{Z} = (\mathcal{B}, \mathcal{A})$, where $\mathcal{Z}$ is a subset ep pair, Theorem 9 from Section 4 describing the structure of the successor states becomes much simpler (it follows directly from the definition of a subset ep pair). This simplified situation is stated in the easily verifiable Theorem 20 and illustrated in Figure 5, the reader should compare it with Figure 2.

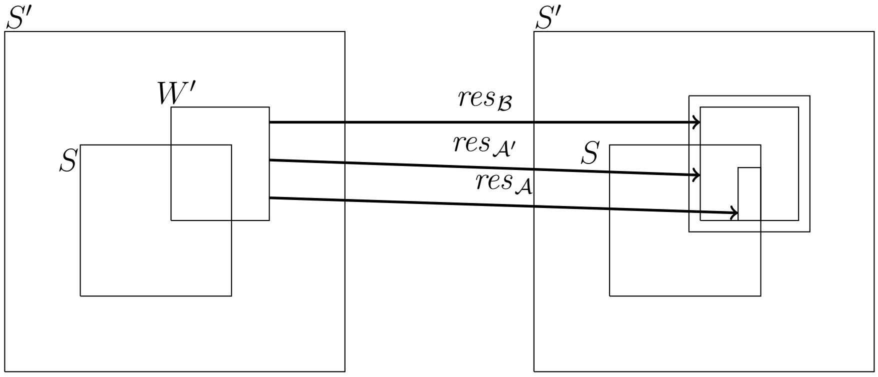


Fig. 5. The structure of the successor state $res_{\mathcal{A}'}(W')$ of a state $W'$ of the expansion $\mathcal{A}'$ of $\mathcal{A}$ by $\mathcal{B}$ for $\mathcal{A}$ embedded in $\mathcal{B}$.

**Theorem 20** *Let $\mathcal{Z} = (\mathcal{B}, \mathcal{A})$ be a subset ep pair over $(S', S)$ and let $\mathcal{A}'$ be the $\mathcal{Z}$-expansion. Then, $\mathcal{A}' = \mathcal{B}$, and, for each $W' \subseteq S'$, $res_{\mathcal{A}}(W') = res_{\mathcal{A}}(W' \cap S)$ and $res_{\mathcal{A}}(W') \subseteq (res_{\mathcal{B}}(W')) \cap S$.*

Since $\mathcal{A}$ is embedded in $\mathcal{B}$, we will also assume that, for each interactive process in $\mathcal{Z}$, its initial configuration $(W', C, D, W)$ is such that $W = W' \cap S$. This assumption together with Theorem 20 implies now that an interactive process in $\mathcal{Z}$ (see Definition 19) is of the form $\pi' = f'_0, f'_1, \ldots f'_n$, where

$$f'_0 = \begin{pmatrix} W'_0 \\ C_0 \\ D_0 \\ W_0 \end{pmatrix}, f'_1 = \begin{pmatrix} W'_1 & = res_{\mathcal{B}}(W'_0) \\ C_1 & = W'_1 \cap S \\ D_1 & = res_{\mathcal{A}}(W_0) \\ W_1 & = C_1 \cup D_1 \end{pmatrix}, \ldots, f'_n = \begin{pmatrix} W'_n & = res_{\mathcal{B}}(W'_{n-1}) \\ C_n & = W'_n \cap S \\ D_n & = res_{\mathcal{A}}(W_{n-1}) \\ W_n & = C_n \cup D_n \end{pmatrix}$$

are such that, for $i \in \{1, \ldots, n\}$, $D_i = res_{\mathcal{A}}(W_{i-1}) \subseteq (res_{\mathcal{B}}(W'_{i-1})) \cap S = C_i$ and so $W_i = W'_i \cap S$. Therefore, the interactive process $\pi$ in $\mathcal{A}$ induced by $\pi'$ is of the form $\pi = f_0, \ldots, f_n$ where

$$f_0 = \begin{pmatrix} C_0 \\ D_0 \\ W_0 \end{pmatrix}, f_1 = \begin{pmatrix} C_1 \\ D_1 \\ W_1 \end{pmatrix}, \ldots, f_n = \begin{pmatrix} C_n \\ D_n \\ W_n \end{pmatrix}$$

are such that, for each $i \in \{1, \ldots, n\}$, $D_i \subseteq C_i$ and thus $W_i = C_i$. Interestingly, this is a sort of a dual condition to the context-independent condition for interactive processes of reaction systems; such interactive processes could be called *result-independent*.

Thus the strict context sequence $C_1, \ldots, C_n$ solely determines the strict state sequence $W_1 = C_1, \ldots, W_n = C_n$. Since $C_1 = W_1' \cap S, \ldots, C_n = W_n' \cap S$, the strict state sequence $W_1, \ldots, W_n$ of $\pi$ is uniquely determined by the strict state sequence $W_1', \ldots, W_n'$ of $\mathcal{A}' = \mathcal{B}$. Since also $W_0 = W_0' \cap S$, we obtain $W_0, W_1, \ldots, W_n = proj_S(W_0', W_1', \ldots W_n')$. This yields the following result.

**Corollary 21** *Let $\mathcal{B}$ be a rs over $S'$ and let $S$ be a background set such that $S \subseteq S'$. For every subset ep pair $\mathcal{Z} = (\mathcal{B}, \mathcal{A}) \in \overline{\mathcal{P}}_{ext}(\mathcal{B}, S)$, $STS(\mathcal{A} \to \mathcal{Z}) = proj_S(STS(\mathcal{B}))$.*

This means that all reaction systems $\mathcal{A}$ over $S$ embedded in $\mathcal{B}$ have the same set of state sequences when plugged into $\mathcal{B}$ (viz., projections on $S$ of all state sequences of $\mathcal{B}$). Note that they still may have different sets of interactive processes, as, in each interactive process $\pi$ induced by $\pi'$ (in the notation as above), the sequence $D_1, \ldots, D_n$ is determined by $\mathcal{A}$.

Thus, even though the notion of an extension of a reaction system turned out to be useful in a number of research lines (see, e.g., [25,10]), it is quite limited as a plug-in provider of context sequences for reaction systems.

We will now demonstrate that reaction systems as plug-in context providers under the restriction that one can plug in only reaction systems that are embedded in the expanders are equivalent with strictly deterministic state-oblivious context controllers, in the sense that they can simulate each other.

**Theorem 22** *Let $\mathcal{B}$ be a reaction system over a background set $S'$ and let $S \subseteq S'$ be a background set. There exists a strictly deterministic state-oblivious context controller $\mathcal{C}$ and a set $G$ of $(S', S)$-configurations such that for each rs $\mathcal{A}$ over $S$ embedded in $\mathcal{B}$, $PROC(\mathcal{A} \to (\mathcal{B}, \mathcal{A})) = PROC_G(\mathcal{A} \to (\mathcal{C}, \mathcal{A}))$.*

**PROOF.** Let $\mathcal{B} = (S', B)$ be a rs and let $S \subseteq S'$ be a background set.

Let then $\mathcal{C} = (Q, E, 2^S)$ be the strictly deterministic state-oblivious context controller such that $Q = \{[Z] \mid Z \subseteq S'\}$ and $E = \{([Z], (res_{\mathcal{B}}(Z)) \cap S, [res_{\mathcal{B}}(Z)]) \mid Z \subseteq S'\}$. Since $res_{\mathcal{B}}$ is a function, $\mathcal{C}$ is strictly deterministic.

Let $G$ be the set of $(S', S)$-configurations $f$ of the form $f = ([Z], C, D, W = Z \cap S)$, where $Z \subseteq S'$.

Let $\mathcal{A} = (S, A)$ be a rs over $S$ such that $\mathcal{A}$ is embedded in $\mathcal{B}$. Then, let $\mathcal{Z}$ be the subset ep pair $\mathcal{Z} = (\mathcal{B}, \mathcal{A})$ and let $\mathcal{U}$ be the socp pair $\mathcal{U} = (\mathcal{C}, \mathcal{A})$.

We prove the statement of the theorem by proving two inclusions.

(I) $PROC(\mathcal{A} \to \mathcal{Z}) \subseteq PROC_G(\mathcal{A} \to \mathcal{U})$.

Let $\pi \in PROC(\mathcal{A} \to \mathcal{Z})$ and let $\pi'$ be an interactive process of $\mathcal{Z}$ such that $\pi$ is induced by $\pi'$. Thus $\pi' = f_0', \ldots, f_n'$, where $n \geq 1$ and the consecutive configurations of $\pi'$ are

$$f'_0 = \begin{pmatrix} W'_0 \\ C_0 \\ D_0 \\ W_0 = W'_0 \cap S \end{pmatrix}, f'_1 = \begin{pmatrix} W'_1 = res_{\mathcal{B}}(W'_0) \\ C_1 = W'_1 \cap S \\ D_1 = res_{\mathcal{A}}(W_0) \\ W_1 = D_1 \cup C_1 \end{pmatrix}, \ldots, f'_n = \begin{pmatrix} W'_n = res_{\mathcal{B}}(W'_{n-1}) \\ C_n = W'_n \cap S \\ D_n = res_{\mathcal{A}}(W_{n-1}) \\ W_n = D_n \cup C_n \end{pmatrix},$$

for some $W'_0 \subseteq S'$ and $C_0, D_0 \subseteq S$. It follows directly from Theorem 20 that, for each $i \in \{1, \ldots, n\}$, $W_i = W'_i \cap S$.

Let then $\overline{\lambda} = \overline{h}_0, \overline{h}_1, \ldots, \overline{h}_n$, be the interactive process in $\mathcal{U}$, where, for each $i \in \{1, \ldots, n\}$, $\overline{h}_i = (\overline{q}_i, \overline{C}_i, \overline{D}_i, \overline{W}_i)$ with $\overline{h}_0 = (\overline{q}_0 = [W'_0], \overline{C}_0 = C_0, \overline{D}_0 = D_0, \overline{W}_0 = W_0)$. It follows then from the definition of $\mathcal{C}$ that the configurations $\overline{h}_0, \overline{h}_1, \ldots, \overline{h}_n$ are of the form.

$$\overline{h}_0 = \begin{pmatrix} \overline{q}_0 = [W'_0] \\ \overline{C}_0 = C_0 \\ \overline{D}_0 = D_0 \\ \overline{W}_0 = W'_0 \cap S \end{pmatrix}, \overline{h}_1 = \begin{pmatrix} \overline{q}_1 = [W'_1] \\ \overline{C}_1 = W'_1 \cap S \\ \overline{D}_1 = res_{\mathcal{A}}(W_0) \\ \overline{W}_1 = W'_1 \cap S \end{pmatrix}, \ldots, \overline{h}_n = \begin{pmatrix} \overline{q}_n = [W'_n] \\ \overline{C}_n = W'_n \cap S \\ \overline{D}_n = res_{\mathcal{A}}(W_{n-1}) \\ \overline{W}_n = W'_n \cap S \end{pmatrix},$$

Thus the interactive process $\lambda$ in $\mathcal{A}$ induced by $\overline{\lambda}$ is of the form $\lambda = h_0, h_1, \ldots, h_n$, where for each $i \in \{1, \ldots, n\}$, $h_i = (C_i, D_i, W_i)$ with $\overline{C}_i$, $\overline{D}_i$, $\overline{W}_i$ defined as above for $\overline{\lambda}$. Since, for each $i \in \{1, \ldots, n\}$, $\overline{C}_i = C_i$, $\overline{D}_i = D_i$, and $\overline{W}_i = W_i$, we obtain $\lambda = \pi$. Since $h_0 \in G$, we obtain $\pi \in PROC_G(\mathcal{A} \to \mathcal{U})$.

Since $\pi$ was an arbitrary process in $PROC(\mathcal{A} \to \mathcal{Z})$, it follows that $PROC(\mathcal{A} \to \mathcal{Z}) \subseteq PROC_G(\mathcal{A} \to \mathcal{U})$.

(II) $PROC_G(\mathcal{A} \to \mathcal{U}) \subseteq PROC(\mathcal{A} \to \mathcal{Z})$.

Let $\lambda \in PROC_G(\mathcal{A} \to \mathcal{U})$ and let $\overline{\lambda}$ be an interactive process of $\mathcal{U}$ such that $\lambda$ is induced by $\overline{\lambda}$. Thus, from the definitions of $\mathcal{C}$ and $G$, it follows that $\overline{\lambda} = \overline{h}_0, \overline{h}_1, \ldots, \overline{h}_n$, where $n \geq 1$ and the consecutive configurations of $\overline{\lambda}$ are

$$\overline{h}_0 = \begin{pmatrix} \overline{q}_0 = [W'_0] \\ \overline{C}_0 \\ \overline{D}_0 \\ \overline{W}_0 = W'_0 \cap S \end{pmatrix}, \overline{h}_1 = \begin{pmatrix} \overline{q}_1 & = [W'_1 = res_{\mathcal{B}}(W'_0)] \\ \overline{C}_1 & = W'_1 \cap S \\ \overline{D}_1 & = res_{\mathcal{A}}(\overline{W}_0) \\ \overline{W}_1 & = \overline{C}_1 \cup \overline{D}_1 \end{pmatrix}$$

$$, \ldots, \overline{h}_n = \begin{pmatrix} \overline{q}_n & = [W'_n = res_{\mathcal{B}}(W'_{n-1})] \\ \overline{C}_n & = W'_n \cap S \\ \overline{D}_n & = res_{\mathcal{A}}(\overline{W}_{n-1}) \\ \overline{W}_n & = \overline{C}_n \cup \overline{D}_n \end{pmatrix},$$

for some $W'_0 \subseteq S'$ and $\overline{C}_0, \overline{D}_0 \subseteq S$. Thus, $\lambda = h_0, h_1, \ldots, h_n$, with

$$h_0 = \begin{pmatrix} \overline{C}_0 \\ \overline{D}_0 \\ \overline{W}_0 \end{pmatrix}, h_1 = \begin{pmatrix} \overline{C}_1 \\ \overline{D}_1 \\ \overline{W}_1 \end{pmatrix}, \ldots, h_n = \begin{pmatrix} \overline{C}_n \\ \overline{D}_n \\ \overline{W}_n \end{pmatrix},$$

where, for each $i \in \{0, \ldots, n\}$, $\overline{C}_i$, $\overline{D}_i$, and $\overline{W}_i$ are defined as in $\overline{\lambda}$ above.

Consider now the interactive process $\pi' = f'_0, f'_1, f'_1 \ldots f'_n$ in $\mathcal{Z}$ such that $f'_0 = (W'_0, C_0 = \overline{C}_0, D_0 = \overline{D}_0, W_0 = \overline{W}_0)$. Hence, the consecutive configurations of $\pi'$ are:

$$f'_0 = \begin{pmatrix} W'_0 \\ C_0 \\ D_0 \\ W_0 \end{pmatrix}, f'_1 = \begin{pmatrix} W'_1 &= res_{\mathcal{B}}(W'_0) \\ \overline{C}_1 &= W'_1 \cap S \\ \overline{D}_1 &= res_{\mathcal{A}}(W_0) \\ \overline{W}_1 &= \overline{C}_1 \cup \overline{D}_1 \end{pmatrix}, \ldots, f'_n = \begin{pmatrix} W'_n &= res_{\mathcal{B}}(W'_{n-1}) \\ \overline{C}_n &= W'_n \cap S \\ \overline{D}_n &= res_{\mathcal{A}}(\overline{W}_{n-1}) \\ \overline{W}_n &= \overline{C}_n \cup \overline{D}_n \end{pmatrix}.$$

Note that since $\overline{h}_0 \in G$, $f'_0$ is a valid initial configuration for an interactive process in $\mathcal{Z}$ and so, indeed, $\pi'$ is an interactive process in $\mathcal{Z}$.

Obviously, the interactive process $\pi \in \mathcal{A}$ induced by $\pi'$ is such that $\pi = \lambda$. Thus $\lambda \in PROC(\mathcal{A} \to \mathcal{Z})$.

Since $\lambda$ was an arbitrary interactive process in $PROC_G(\mathcal{A} \to \mathcal{U})$, it follows that $PROC_G(\mathcal{A} \to \mathcal{U}) \subseteq PROC(\mathcal{A} \to \mathcal{Z})$.

The theorem follows now from (I) and (II). □

Note that considering $PROC_G(\mathcal{A} \to \mathcal{U})$ (rather than just $PROC(\mathcal{A} \to \mathcal{U})$ was necessary, because if $\mathcal{Z} = (\mathcal{B}, \mathcal{A})$ is a subset ep pair over $(S', S)$, then the initial configurations of interactive processes in $\mathcal{Z}$ are of the form $(W', C, D, W)$, where $W = W' \cap S$.

**Theorem 23** *Let $\mathcal{C}$ be a strictly deterministic state-oblivious context controller over a background set $S$. There exists a rs $\mathcal{B}$ over a background set $S'$ such that $S \subseteq S'$ and a set $G$ of $(S', S)$-configurations such that for each rs $\mathcal{A}$ over $S$, $PROC(\mathcal{A} \to (\mathcal{C}, \mathcal{A})) = PROC_G(\mathcal{A} \to (\mathcal{B}, \mathcal{A}))$.*

**PROOF.** Let $\mathcal{C} = (Q, E, 2^S)$ be a strictly deterministic state-oblivious context controller over a background set $S$. Again, we assume that $Q \cap S = \varnothing$.

Let then $\mathcal{B} = (S', B)$ be the rs such that $S' = S \cup Q$ and $B = \{b(e) \mid e \in E\}$, where, for each $e = (q, C, q') \in E$, $b(e) = (\{q\}, Q \setminus \{q\}, \{q'\} \cup C)$.

Let $G$ be the set of all $(S', S)$-configurations $(W', C, D, W)$ such that $W' \subseteq Q$ and $|W'| = 1$.

Let now $\mathcal{A} = (S, A)$ be a rs over $S$, and, accordingly, let $\mathcal{U}$ be the socp pair $(\mathcal{C}, \mathcal{A})$ and let $\mathcal{Z}$ be the cp pair $(\mathcal{B}, \mathcal{A})$.

We prove the statement of the theorem by proving two inclusions.

(I) $PROC(\mathcal{A} \to \mathcal{U}) \subseteq PROC_G(\mathcal{A} \to \mathcal{Z})$.

Let $\lambda \in PROC(\mathcal{A} \to \mathcal{U})$ and let $\overline{\lambda} \in PROC(\mathcal{U})$ be such that $\lambda$ is induced by $\overline{\lambda}$. Thus $\overline{\lambda} = \overline{h}_0, \overline{h}_1, \ldots, \overline{h}_n$, for some $n \geq 1$, with the consecutive configurations of $\overline{\lambda}$ of the form:

$$\overline{h}_0 = \begin{pmatrix} \overline{q}_0 \\ \overline{C}_0 \\ \overline{D}_0 \\ \overline{W}_0 = \overline{C}_0 \cup \overline{D}_0 \end{pmatrix}, \overline{h}_1 = \begin{pmatrix} \overline{q}_1 \\ \overline{C}_1 \\ \overline{D}_1 \ = res_{\mathcal{A}}(\overline{W}_0) \\ \overline{W}_1 = \overline{C}_1 \cup \overline{D}_1 \end{pmatrix}, \ldots, \overline{h}_n = \begin{pmatrix} \overline{q}_n \\ \overline{C}_n \\ \overline{D}_n \ = res_{\mathcal{A}}(\overline{W}_{n-1}) \\ \overline{W}_n = \overline{C}_n \cup \overline{D}_n \end{pmatrix},$$

where for each $i \in \{1, \ldots, n\}$, $(\overline{q}_{i-1}, \overline{C}_i, \overline{q}_i) \in E$. Thus $\lambda = h_0, h_1, \ldots, h_n$ with

$$h_0 = \begin{pmatrix} \overline{C}_0 \\ \overline{D}_0 \\ \overline{W}_0 \end{pmatrix}, h_1 = \begin{pmatrix} \overline{C}_1 \\ \overline{D}_1 \\ \overline{W}_1 \end{pmatrix}, \ldots, h_n = \begin{pmatrix} \overline{C}_n \\ \overline{D}_n \\ \overline{W}_n \end{pmatrix},$$

where, for each $i \in \{0, \ldots, n\}$, $\overline{C}_i$, $\overline{D}_i$, and $\overline{W}_i$ are as defined above for $\overline{\lambda}$.

Let then $\pi' = f'_0, f'_1, \ldots, f'_n$ be the interactive process in $\mathcal{Z}$ such that $f'_0 = (\{\overline{q}_0\}, \overline{C}_0, \overline{D}_0, \overline{W}_0)$.

Since, for each $i \in \{1, \ldots, n\}$, $(\overline{q}_{i-1}, \overline{C}_i, \overline{q}_i) \in E$, it follows then from the definition of $B$ (and from the definition of an interactive process of a socp pair) that the consecutive configurations of $\pi'$ are:

$$f'_0 = \begin{pmatrix} \{\overline{q}_0\} \\ \overline{C}_0 \\ \overline{D}_0 \\ \overline{W}_0 \end{pmatrix}, f'_1 = \begin{pmatrix} \{\overline{q}_1\} \cup \overline{C}_1 \\ \overline{C}_1 \\ \overline{D}_1 = res_{\mathcal{A}}(\overline{W}_0) \\ \overline{W}_1 = \overline{C}_1 \cup \overline{D}_1 \end{pmatrix}, \ldots, f'_n = \begin{pmatrix} \{\overline{q}_n\} \cup \overline{C}_n \\ \overline{C}_n \\ \overline{D}_n = res_{\mathcal{A}}(\overline{W}_{n-1}) \\ \overline{W}_n = \overline{C}_n \cup \overline{D}_n \end{pmatrix}.$$

Hence the interactive process $\pi$ in $\mathcal{A}$ induced by $\pi'$ is of the form $\pi = f_0, f_1, \ldots, f_n$, where

$$f_0 = \begin{pmatrix} \overline{C}_0 \\ \overline{D}_0 \\ \overline{W}_0 \end{pmatrix}, f_1 = \begin{pmatrix} \overline{C}_1 \\ \overline{D}_1 \\ \overline{W}_1 \end{pmatrix}, \ldots, f_n = \begin{pmatrix} \overline{C}_n \\ \overline{D}_n \\ \overline{W}_n \end{pmatrix},$$

Thus $\pi = \lambda$ and since $f'_0 \in G$, we obtain $\lambda \in PROC_G(\mathcal{A} \to \mathcal{Z})$.

Since $\pi$ was an arbitrary interactive process in $PROC(\mathcal{A} \to \mathcal{U})$, it follows that $PROC(\mathcal{A} \to \mathcal{U}) \subseteq PROC_G(\mathcal{A} \to \mathcal{Z})$.

(II) $PROC_G(\mathcal{A} \to \mathcal{Z}) \subseteq PROC(\mathcal{A} \to \mathcal{U})$.

Let $\pi \in PROC_G(\mathcal{A} \to \mathcal{Z})$ and $\pi' \in PROC(\mathcal{Z})$ be such that $\pi$ is induced by $\pi'$ within $\mathcal{Z}$. Thus $\pi' = f'_0, f'_1, \ldots, f'_n$, where $n \geq 1$, $f'_0 = (\{q_0\}, C_0, D_0, W_0)$ for some $q_0 \in Q$ (because $f'_0 \in G$), and the consecutive configurations of $\pi'$ are:

$$f'_0 = \begin{pmatrix} \{q_0\} \\ C_0 \\ D_0 \\ W_0 \end{pmatrix}, f'_1 = \begin{pmatrix} \{q_1\} \cup C_1 \\ C_1 \\ D_1 = res_{\mathcal{A}}(W_0) \\ W_1 = C_1 \cup D_1 \end{pmatrix}, \ldots, f'_n = \begin{pmatrix} \{q_n\} \cup C_n \\ C_n \\ D_n = res_{\mathcal{A}}(W_{n-1}) \\ W_n = C_n \cup D_n \end{pmatrix},$$

where, for each $i \in \{1, \ldots, n\}$, $(q_{i-1}, C_i, q_i) \in E$. Thus $\pi = f_0, f_1, \ldots, f_n$, with

$$f_0 = \begin{pmatrix} C_0 \\ D_0 \\ W_0 \end{pmatrix}, f_1 = \begin{pmatrix} C_1 \\ D_1 \\ W_1 \end{pmatrix}, \ldots, f_n = \begin{pmatrix} C_n \\ D_n \\ W_n \end{pmatrix}.$$

Let then $\overline{\lambda} \in PROC(\mathcal{U})$ be such that $\overline{\lambda} = \overline{h}_0, \overline{h}_1, \ldots, \overline{h}_n$, where $\overline{h_0} = (q_0, C_0, D_0, W_0)$ and, for each $i \in \{1, \ldots, n\}$, $\overline{h}_i = (\overline{q}_i, \overline{C}_i, \overline{D}_i, \overline{W}_i)$. Since, for each $i \in \{1, \ldots, n\}$, $(q_{i-1}, C_i, q_i) \in E$, the consecutive configurations of $\overline{\lambda}$ are:

$$\overline{h}_0 = \begin{pmatrix} q_0 \\ C_0 \\ D_0 \\ W_0 \end{pmatrix}, \overline{h}_1 = \begin{pmatrix} q_1 \\ C_1 \\ D_1 \\ W_1 \end{pmatrix}, \ldots, \overline{h}_n = \begin{pmatrix} q_n \\ C_n \\ D_n \\ W_n \end{pmatrix}.$$

Hence the interactive process $\lambda$ in $\mathcal{A}$ induced by $\overline{\lambda}$ is of the form $\lambda = h_0, h_1, \ldots, h_n$, where, for each $i \in \{0, \ldots, n\}$, $h_i = (C_i, D_i, W_i)$. This implies that $\lambda = \pi$ and consequently $\pi \in PROC(\mathcal{A} \to \mathcal{U})$.

Since $\pi$ was an arbitrary interactive process in $PROC_G(\mathcal{A} \to \mathcal{Z})$, it follows that $PROC_G(\mathcal{A} \to \mathcal{Z}) \subseteq PROC(\mathcal{A} \to \mathcal{U})$.

The theorem follows now from (I) and (II). $\square$

Considering $PROC_G(\mathcal{A} \to \mathcal{Z})$ (rather than just $PROC(\mathcal{A} \to \mathcal{Z})$) in the above theorem was necessary, as an arbitrary state of $\mathcal{B}$ (a subset of $S'$) can contain more than one state of $Q$ (or none at all), while $\mathcal{C}$ is always in one state.

## 8 Discussion

In this paper we have studied plug-in context providers for reaction systems, viewing in this way interactive processes from the perspective of the environment. First, we have introduced extenders, which are themselves reaction systems. Secondly, we have considered context controllers (which were already introduced in the literature in the context of model checking) as plug-in providers of context sequences. Then, we have demonstrated that extenders and state-aware context controllers are equivalent as plug-in context providers, in the sense that they can simulate each other.

We have also reformulated extensions of a reaction system (considered already in the literature) in terms of plug-in context providers and demonstrated that their use (as such providers) is rather limited. Also, we have shown that, as context providers, they are equivalent to state-oblivious context controllers.

Altogether we have established a 2-level hierarchy of plug-in context providers.

It is worthwhile to point out that the research presented in this paper is genuinely concerned with properties of *interactive processes* as such, rather than only with properties of their *state sequences* (which is mostly done in the literature).

This research fits into an important line of research concerned with finding classes of context sequences (for interactive processes of reaction systems) which are more general than in the case of context-independent interactive processes and (much) more restrictive than in the case of arbitrary interactive processes. Therefore it is important to notice that (as is easily seen) even the subset ep pairs $(\mathcal{B}, \mathcal{A})$ over $(S', S)$ can generate interactive processes with state sequences which do not obey the three basic properties of state sequences generated by context-independent interactive processes (listed at the end of Section 3): 'No resurrection', 'No saturation', and 'Once repeated, always repeated'.

We see this paper as a beginning of a systematic research into plug-in context providers for reaction systems. This (plug-in) approach leads to some novel research lines that should be exploited. Here are some examples of such research lines.

Each rs $\mathcal{B}$ over $S'$ used as an expander generates 'its own' family of sets of interactive processes over a background set $S \subseteq S'$, viz., $\mathit{INPROC}(\mathcal{B}, S)$, where each set in this family is the set of interactive processes for one rs over $S$ with context sequences provided by $\mathcal{B}$.

An important issue to be investigated is how various properties of the rs $\mathcal{B}$ are (can be) reflected in the properties of $\mathit{INPROC}(\mathcal{B}, S)$. Many interesting properties of reaction systems, especially of their state sequences (which are crucial when one considers a rs as a plug-in context provider) were investigated in the literature.

A natural question is: how the cardinality of $S' \setminus S$ influences the 'richness' of $\mathit{INPROC}(\mathcal{B}, S)$. One could call an expander $\mathcal{B}$ 'minimal' if $|S' \setminus S| = 1$. How 'simple' are the families $\mathit{INPROC}(\mathcal{B}, S)$?

Yet another natural problem is to define and investigate relationships between reaction systems $\mathcal{B}_1$, $\mathcal{B}_2$ (used as expanders), which can be reflected in the relationships between $\mathit{INPROC}(\mathcal{B}_1, S)$ and $\mathit{INPROC}(\mathcal{B}_2, S)$.

One of the very fruitful research lines concerning reaction systems is the investigation of properties of state sequences of context-independent interactive processes (see, e.g., [13,16,17,19]). As pointed out above, reaction systems plugged into the plug-in devices considered in this paper are more powerful generators of state sequences. The investigation of properties of these new classes of state sequences is certainly a novel and relevant research topic. In particular, we propose to investigate properties of state sequences generated by reaction systems plugged into strictly deterministic context-aware state controllers.

**Acknowledgements**

The authors are indebted to the referee for the useful comments.

## References


[1] A. Ehrenfeucht, G. Rozenberg, Reaction systems, Fundam. Inform. 75 (1-4) (2007) 263–280.
URL http://content.iospress.com/articles/fundamenta-informaticae/fi75-1-4-15

[2] A. Ehrenfeucht, J. Kleijn, M. Koutny, G. Rozenberg, Reaction Systems: A Natural Computing Approach to the Functioning of Living Cells, pp. 189–208. arXiv:https://www.worldscientific.com/doi/pdf/10.1142/9789814374309_0010, doi:10.1142/9789814374309_0010.
URL https://www.worldscientific.com/doi/abs/10.1142/9789814374309_0010

[3] A. Ehrenfeucht, I. Petre, G. Rozenberg, Reaction systems: A model of computation inspired by the functioning of the living cell, in: S. Konstantinidis, N. Moreira, R. Reis, J. Shallit (Eds.), The Role of Theory in Computer Science - Essays Dedicated to Janusz Brzozowski, World Scientific, 2017, pp. 1–32. doi:10.1142/9789813148208_0001.
URL https://doi.org/10.1142/9789813148208_0001

[4] A. Ehrenfeucht, J. Kleijn, M. Koutny, G. Rozenberg, Qualitative and Quantitative Aspects of a Model for Processes Inspired by the Functioning of the Living Cell, John Wiley & Sons, Ltd, 2012, Ch. 16, pp. 303–321. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/9783527645480.ch16, doi:10.1002/9783527645480.ch16.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9783527645480.ch16

[5] L. Corolli, C. Maj, F. Marini, D. Besozzi, G. Mauri, An excursion in reaction systems: From computer science to biology, Theor. Comput. Sci. 454 (2012) 95–108. doi:10.1016/j.tcs.2012.04.003.
URL https://doi.org/10.1016/j.tcs.2012.04.003

[6] I. Petre, A. Mizera, C. L. Hyder, A. Meinander, A. Mikhailov, R. I. Morimoto, L. Sistonen, J. E. Eriksson, R. Back, A simple mass-action model for the eukaryotic heat shock response and its mathematical validation, Natural Computing 10 (1) (2011) 595–612. doi:10.1007/s11047-010-9216-y.
URL https://doi.org/10.1007/s11047-010-9216-y

[7] S. Azimi, B. Iancu, I. Petre, Reaction system models for the heat shock response, Fundam. Inform. 131 (3-4) (2014) 299–312. doi:10.3233/FI-2014-1016.
URL https://doi.org/10.3233/FI-2014-1016

[8] R. Barbuti, P. Bove, R. Gori, F. Levi, P. Milazzo, Simulating gene regulatory networks using reaction systems, in: B. Schlingloff, S. Akili (Eds.), Proceedings of the 27th International Workshop on Concurrency, Specification and Programming, Berlin, Germany, September 24-26, 2018., Vol. 2240 of CEUR Workshop Proceedings, CEUR-WS.org, 2018.
URL http://ceur-ws.org/Vol-2240/paper11.pdf

[9] P. Bottoni, A. Labella, G. Rozenberg, Reaction systems with influence on environment, Journal of Membrane Computing 1 (1) (2019) 3–19. doi:10.1007/s41965-018-00005-8.
URL https://doi.org/10.1007/s41965-018-00005-8

[10] P. Bottoni, A. Labella, G. Rozenberg, Networks of reaction systems, Int. J. Found. Comput. Sci. to appear.

[11] R. Barbuti, R. Gori, F. Levi, P. Milazzo, Generalized contexts for reaction systems: definition and study of dynamic causalities, Acta Inf. 55 (3) (2018) 227–267. doi:10.1007/s00236-017-0296-3.
URL https://doi.org/10.1007/s00236-017-0296-3

[12] M. Hirvensalo, On probabilistic and quantum reaction systems, Theor. Comput. Sci. 429 (2012) 134–143. doi:10.1016/j.tcs.2011.12.032.
URL https://doi.org/10.1016/j.tcs.2011.12.032

[13] A. Salomaa, Functions and sequences generated by reaction systems, Theor. Comput. Sci. 466 (2012) 87–96. doi:10.1016/j.tcs.2012.07.022.
URL https://doi.org/10.1016/j.tcs.2012.07.022

[14] A. Meski, W. Penczek, G. Rozenberg, Model checking temporal properties of reaction systems, Inf. Sci. 313 (2015) 22–42. doi:10.1016/j.ins.2015.03.048.
URL https://doi.org/10.1016/j.ins.2015.03.048

[15] A. Ehrenfeucht, M. G. Main, G. Rozenberg, A. T. Brown, Stability and chaos in reaction systems, Int. J. Found. Comput. Sci. 23 (5) (2012) 1173. doi:10.1142/S0129054112500177.
URL https://doi.org/10.1142/S0129054112500177

[16] A. Salomaa, Functional constructions between reaction systems and propositional logic, Int. J. Found. Comput. Sci. 24 (1) (2013) 147–160. doi:10.1142/S0129054113500044.
URL https://doi.org/10.1142/S0129054113500044

[17] E. Formenti, L. Manzoni, A. E. Porreca, Fixed points and attractors of reaction systems, in: A. Beckmann, E. Csuhaj-Varjú, K. Meer (Eds.), Language, Life, Limits - 10th Conference on Computability in Europe, CiE 2014, Budapest, Hungary, June 23-27, 2014. Proceedings, Vol. 8493 of Lecture Notes in Computer Science, Springer, 2014, pp. 194–203. doi:10.1007/978-3-319-08019-2_20.
URL https://doi.org/10.1007/978-3-319-08019-2_20

[18] A. Ehrenfeucht, G. Rozenberg, Events and modules in reaction systems, Theor. Comput. Sci. 376 (1-2) (2007) 3–16. doi:10.1016/j.tcs.2007.01.008.
URL https://doi.org/10.1016/j.tcs.2007.01.008

[19] E. Formenti, L. Manzoni, A. E. Porreca, On the complexity of occurrence and convergence problems in reaction systems, Natural Computing 14 (1) (2015) 185–191. doi:10.1007/s11047-014-9456-3.
URL https://doi.org/10.1007/s11047-014-9456-3

[20] A. Ehrenfeucht, G. Rozenberg, Zoom structures and reaction systems yield exploration systems, Int. J. Found. Comput. Sci. 25 (3) (2014) 275–306. doi:10.1142/S0129054114500142.
URL https://doi.org/10.1142/S0129054114500142

[21] A. Ehrenfeucht, J. Kleijn, M. Koutny, G. Rozenberg, Minimal reaction systems, Trans. Computational Systems Biology 14 (2012) 102–122. doi:10.1007/978-3-642-35524-0_5.
URL https://doi.org/10.1007/978-3-642-35524-0_5

[22] A. Ehrenfeucht, J. Kleijn, M. Koutny, G. Rozenberg, Evolving reaction systems, Theor. Comput. Sci. 682 (2017) 79–99. doi:10.1016/j.tcs.2016.12.031.
URL https://doi.org/10.1016/j.tcs.2016.12.031

[23] A. Meski, M. Koutny, W. Penczek, Verification of linear-time temporal properties for reaction systems with discrete concentrations, Fundam. Inform. 154 (1-4) (2017) 289–306. doi:10.3233/FI-2017-1567.
URL https://doi.org/10.3233/FI-2017-1567

[24] A. Meski, M. Koutny, W. Penczek, Reaction mining for reaction systems, in: S. Stepney, S. Verlan (Eds.), Unconventional Computation and Natural Computation - 17th International Conference, UCNC 2018, Fontainebleau, France, June 25-29, 2018, Proceedings, Vol. 10867 of Lecture Notes in Computer Science, Springer, 2018, pp. 131–144. doi:10.1007/978-3-319-92435-9_10.
URL https://doi.org/10.1007/978-3-319-92435-9_10

[25] R. Brijder, A. Ehrenfeucht, G. Rozenberg, Reaction systems with duration, in: J. Kelemen, A. Kelemenová (Eds.), Computation, Cooperation, and Life - Essays Dedicated to Gheorghe Paun on the Occasion of His 60th Birthday, Vol. 6610 of Lecture Notes in Computer Science, Springer, 2011, pp. 191–202. doi:10.1007/978-3-642-20000-7_16.
URL https://doi.org/10.1007/978-3-642-20000-7_16

[26] J. Kleijn, M. Koutny, L. Mikulski, G. Rozenberg, Reaction systems, transition systems, and equivalences, in: H. Böckenhauer, D. Komm, W. Unger (Eds.), Adventures Between Lower Bounds and Higher Altitudes - Essays Dedicated to Juraj Hromkovič on the Occasion of His 60th Birthday, Vol. 11011 of Lecture Notes in Computer Science, Springer, 2018, pp. 63–84. doi:10.1007/978-3-319-98355-4_5.
URL https://doi.org/10.1007/978-3-319-98355-4_5