

Reaction Systems and Enabling Equivalence

Jetty Kleijn*

LIACS, Leiden University

P.O.Box 9512, NL-2300 RA Leiden, The Netherlands

h.c.m.kleijn@liacs.leidenuniv.nl

Maciej Koutny

School of Computing, Newcastle University

Newcastle upon Tyne NE1 7RU, UK

maciej.koutny@ncl.ac.uk

Lukasz Mikulski

Faculty of Mathematics and Computer Science

Nicolaus Copernicus University, Toruń, Poland

lukasz.mikulski@mat.umk.pl

Abstract. Reaction systems were introduced in order to provide an abstract model for the study of the biochemical processes that take place in the living cell. Processes of this kind are the result of the interactions between reactions and may be influenced by the environment. Thus, reaction systems can be considered as a model of (interactive) computation. In previous works, various equivalences defined directly on reaction systems and processes had been proposed and compared. These equivalences were all based on functional equivalence that compares a system's behaviour at every stage of its execution. In this paper, in contrast, we investigate enabling equivalence which focuses on the system behaviour only in specific stages of its evolution, namely those where all of its reactions are active. We discuss the effect of such an approach and, in particular, its relationship to a transition system representation of the system's behaviour.

Keywords: Reaction system, living cell, model of computation, functional equivalence, interactive process, context-independent process, transition system, enabling equivalence

*Address for correspondence: LIACS, Leiden University, P.O.Box 9512, NL-2300 RA Leiden, The Netherlands

1. Introduction

Reaction systems [1] originated as a formal model for the investigation of the functioning of the living cell. They provide a means to investigate the biocomputations that drive natural processes within cells (see, e.g., [2, 3, 4, 5]). The model of reaction system presents a conceptual view of the processes resulting from interactions between biochemical reactions. These interactions are driven by two mechanisms, facilitation and inhibition: (products of) reactions may facilitate or inhibit each other. The basic model of reaction systems is a qualitative rather than a quantitative model that abstracts from various features of biochemical reactions. However, it takes into account that the living cell is an open system and its behaviour is influenced by the environment.

Reaction systems have turned out to be an interesting novel model of interactive computation (see, e.g., [6, 7, 8, 9, 10, 11, 12, 13, 14, 15]). Also, in this paper we take such a point of view. The dynamics of reaction systems defines interactive computations (processes) that transit from state to state. In [16], we have considered various notions of equivalence for reaction systems based directly on the reaction systems as well as on the transition systems that can be associated with their computations. In this paper, we follow up on a suggestion from [16]. There the (direct) equivalences were based on so-called *functional equivalence*. Here we incorporate the more subtle *enabling equivalence*, introduced in [17] in the context of evolving reaction systems where the set of reactions of a reaction system may change provided the original and the transformed system are enabling equivalent. Enabling equivalence treats the set of reactions of a reaction system as if it were a single reaction. It thus focuses on the system behaviour only in specific states namely those where all of its reactions are active. We discuss the effect of such an approach and, in particular, its relationship to a transition system representation of the system's behaviour.

The paper is organised as follows. In Section 2, we introduce reaction systems together with some examples that will also be of use later in the paper. In particular, we recall the concepts of interactive and context-independent processes and (context-independent) transition systems of reaction systems. The direct equivalences, based on the idea that a reaction system defines a function from states to states, are considered in Section 3. After discussing functional equivalence, process related equivalences, and (the new) transition systems equivalences we move in Section 4 to enabling equivalence. Rather than only relating to the result of the reactions on a state, this equivalence also takes into account whether or not it enables all reactions of the system. As it turns out, context-independence and process equivalence based on enabling equivalence are no longer the same (in contrast to the case that they are based on functional equivalence). We demonstrate how context-independent process equivalence can be characterised in terms of a new kind of transition systems (in which some states are invisible). Section 5 concludes the paper with a brief discussion of what has been achieved.

2. Preliminaries

We use the standard mathematical notation. Moreover, the *projection* of a sequence of sets $\xi = X_1 \dots X_n$ w.r.t. a set X is the sequence of sets $proj_X(\xi) = (X_1 \cap X) \dots (X_n \cap X)$. The empty sequence is denoted by λ . Note that λ is not the same as the sequence \emptyset consisting of a single occurrence of the empty set. In addition, $\lambda\xi_1\xi_2 = \xi_1\lambda\xi_2 = \xi_1\xi_2\lambda = \xi_1\xi_2$, for all sequences ξ_1 and ξ_2 .

A *transition system* is a pair $\mathcal{T} = (Q, V)$, where Q is a finite set and $V \subseteq Q \times Q$. The set of *state sequences* of \mathcal{T} , denoted by $\text{STS}(\mathcal{T})$, is the set of all finite sequences $\theta = q_0 q_1 \dots q_n$ ($n \geq 0$) of elements of Q such that $(q_i, q_{i+1}) \in V$ for all $0 \leq i < n$. Furthermore, $\text{STS}_q(\mathcal{T})$ are all state sequences of \mathcal{T} which start from a given $q \in Q$.

2.1. Reaction systems

Let S be a finite nonempty set with at least two elements. A *reaction* (in S) is a triple $a = (R, I, P)$ such that R, I , and P are nonempty subsets of S with $R \cap I = \emptyset$, respectively called the *reactants*, *inhibitors*, and *products* of a , and denoted by R_a, I_a , and P_a . In the examples, a will be denoted as:

$$a : R, I \rightarrow P$$

We refer to S as a *background set* while the elements of a background set are *entities*; $\text{rac}(S)$ denotes the set of all reactions in S .

A reaction $a \in \text{rac}(S)$ is *enabled* by $X \subseteq S$, denoted by $\text{en}_a(X)$, if $R_a \subseteq X$ and $I_a \cap X = \emptyset$.¹ The *result* of a on X is $\text{res}_a(X) = P_a$ if $\text{en}_a(X)$, and $\text{res}_a(X) = \emptyset$ otherwise. For $A \subseteq \text{rac}(S)$, the *result* of A on X is $\text{res}_A(X) = \bigcup_{a \in A} \text{res}_a(X)$. We also denote

$$R_A = \bigcup_{a \in A} R_a \quad I_A = \bigcup_{a \in A} I_a \quad P_A = \bigcup_{a \in A} P_a.$$

Moreover, A is *enabled* by $X \subseteq S$ if $R_A \subseteq X$ and $I_A \cap X = \emptyset$. In other words, all reactions from A are enabled by X . We denote this by $\text{en}_A(X)$.

A *reaction system* is a pair $\mathcal{A} = (S, A)$, where S is a background set and $A \subseteq \text{rac}(S)$ a nonempty set of reactions. The elements of S are called the *entities* of \mathcal{A} . The subsets of S are the *states* of \mathcal{A} . Given a state $X \subseteq S$, the *result* of \mathcal{A} on X is $\text{res}_{\mathcal{A}}(X) = \text{res}_A(X)$.

The running example of this paper illustrates *computational* applications of reaction systems.

Example 2.1. Consider reaction system $\mathcal{A} = (S, A)$ with background set $S = \{1, 2, 3, 4\}$ and reactions $A = \{a, b, c\}$ defined as follows:

$$\begin{aligned} a : \{4\} \quad , \{3\} &\rightarrow \{4\} \\ b : \{1, 4\} \quad , \{3\} &\rightarrow \{2\} \\ c : \{4\} \quad , \{2, 3\} &\rightarrow \{1\} \end{aligned}$$

In this reaction system, entities 3 and 4 are ‘dummies’, viz. 3 is a universal inhibitor (should never be present) while 4 is a universal reactant (should always be present). The other two entities, 1 and 2, can be seen as representing two bits with the first (least significant) bit being 0 iff 1 is not present in the current state and, similarly, the second bit being 0 iff 2 is not present. Then \mathcal{A} can be interpreted as a Gray code [18]. This can be seen by examining the states of \mathcal{A} and the results of \mathcal{A} on them.

¹Note that this means that $\emptyset \neq X \neq S$ as it is assumed that $R_a, I_a \neq \emptyset$.

Only states that contain 4 and do not contain 3 enable reactions. These four states are generated cyclically in the order $\{4\} \{1, 4\} \{1, 2, 4\} \{2, 4\}$, as shown in Figure 1. In this diagram, states are depicted as nonagons, while the reactions enabled by a given state are displayed on its outgoing arc. Note that two neighbouring states (sets) differ by exactly one entity coming from $\{1, 2\}$, and so they can be seen as representing two consecutive words in a Gray code. \diamond

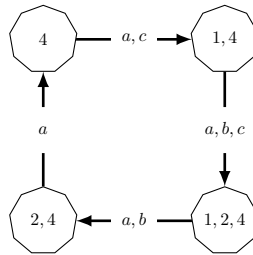


Figure 1. States $\{4\}$, $\{1, 4\}$, $\{1, 2, 4\}$, $\{2, 4\}$ represent cyclically the words 00, 01, 11, 10.

Processes and transition systems of reaction systems

A reaction system formalises the ‘static structure’ of a system. Its dynamics results from the interactions of the reactions. Starting from an initial state, the reactions enabled by the current state produce a result set which then, possibly with context thrown in, forms the next state etc.

An *interactive process* in a reaction system $\mathcal{A} = (S, A)$ is a pair $\pi = (\gamma, \delta)$ comprising a *context sequence* $\gamma = C_0 \dots C_n$ and *result sequence* $\delta = D_0 \dots D_n$, where $n \geq 0$, $C_0, D_0, \dots, C_n, D_n \subseteq S$ and $D_i = res_{\mathcal{A}}(D_{i-1} \cup C_{i-1})$, for every $1 \leq i \leq n$. Moreover, $STS(\pi) = W_0 \dots W_n$, where $W_i = C_i \cup D_i$, for every $0 \leq i \leq n$, is the *state sequence* of π and $W_0 = C_0 \cup D_0$ is the *initial state* of π . (This is illustrated in Figure 2.) For a subset X of S , $PROC_X(\mathcal{A})$ is the set of all interactive processes in \mathcal{A} with initial state X , and $PROC(\mathcal{A}) = \bigcup_{X \subseteq S} PROC_X(\mathcal{A})$.

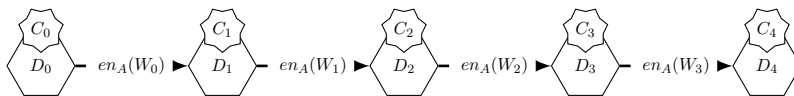


Figure 2. An interactive process. Hexagons depict its result sequence, while seven-pointed stars depict the context sequence. The labels on the arcs indicate which reactions are enabled in the preceding state.

Context sequences formalise the fact that the behaviour of an open system is influenced by its environment (the ‘rest’ of a bigger system). An interactive process π is *context-independent* if $C_i \subseteq D_i$, for every $i \leq n$. In that case, one may assume that no context is provided i.e., the C_i ’s are empty. $CIPROC(\mathcal{A})$ is the set of all context-independent interactive processes of \mathcal{A} and $CIPROC_X(\mathcal{A})$ is the set of context-independent interactive processes of \mathcal{A} with initial state X .

Interactive processes are one way of capturing behaviours of reaction systems. Another is provided through transition systems, by viewing the state space of a reaction system \mathcal{A} as a finite directed graph

with the states of \mathcal{A} as its nodes and its edges determined either by the result function $res_{\mathcal{A}}$ (in case of context-independent interactive processes) or by the joint effect of $res_{\mathcal{A}}$ and the context sets (in case of general interactive processes). This leads to the following definitions.

The *context-independent transition system* of reaction system $\mathcal{A} = (S, A)$ is the transition system $CITS(\mathcal{A}) = (Q, V)$ such that $Q = 2^S$ and $V = \{(X, res_{\mathcal{A}}(X)) \mid X \in 2^S\}$. The *(full) transition system* of \mathcal{A} is the pair $TS(\mathcal{A}) = (Q, V)$ such that $Q = 2^S$ and $V = \{(X, res_{\mathcal{A}}(X) \cup C) \mid X, C \in 2^S\}$. Note that $CITS(\mathcal{A})$ is a proper subgraph of $TS(\mathcal{A})$. Neither of the two transition systems is initialised, as any subset of the background set S can be the first state of an interactive process or a context-independent interactive process of \mathcal{A} .

The soundness of the definitions of $TS(\mathcal{A})$ and $CITS(\mathcal{A})$ is established by relating the state sequences of the interactive processes of \mathcal{A} to the state sequences of the two kinds of transition systems.

Proposition 2.2. ([16])

Let $\mathcal{A} = (S, A)$ be a reaction system and $X \subseteq S$. Then:

$$\begin{aligned} STS(TS(\mathcal{A})) &= STS(\text{PROC}(\mathcal{A})) & STS(CITS(\mathcal{A})) &= STS(\text{CIPROC}(\mathcal{A})) \\ STS_X(TS(\mathcal{A})) &= STS(\text{PROC}_X(\mathcal{A})) & STS_X(CITS(\mathcal{A})) &= STS(\text{CIPROC}_X(\mathcal{A})) . \end{aligned}$$

A *direct* relationship between reaction systems and transition systems has been studied in the literature, see e.g., [8, 19]. In particular, one can simulate finite transition systems by reaction systems and one can simulate reaction systems by finite transition systems, see e.g., [2, 8].

Example 2.3. Consider reaction system $\mathcal{A}' = (S, A')$ with background set $S = \{1, 2, 3, 4\}$ and reactions $A' = \{a, b, c, d, e, f\}$ defined as follows:

$$\begin{aligned} a : \{4\} \quad , \quad \{2, 3\} &\rightarrow \{1, 4\} & d : \{2\} \quad , \quad \{1\} &\rightarrow \{2, 3\} \\ b : \{1, 4\} \quad , \quad \{3\} &\rightarrow \{2, 4\} & e : \{1, 3\} \quad , \quad \{4\} &\rightarrow \{3\} \\ c : \{2, 3\} \quad , \quad \{4\} &\rightarrow \{1\} & f : \{3\} \quad , \quad \{1, 2\} &\rightarrow \{4\} \end{aligned}$$

The role of entities 1 and 2 is as before in Example 2.1, viz. they represent two bits in a Gray code. Entities 3 and 4, however, now have a new role. Similarly as Figure 1, Figure 3 depicts the states of \mathcal{A}' containing either 3 or 4 and transitions between them. Now one can think of the presence of 4 as indicating a forward listing of the consecutive words in a Gray code, while the presence of 3 is associated with a generation of the code in the reverse direction. The sequence $\{4\}\{1, 4\}\{1, 2, 4\}\{2, 4\}\{2, 3\}\{1, 2, 3\}$ is an example of a state sequence of a context-independent interactive process of \mathcal{A}' with initial state $\{4\}$.

Without the names of enabled reactions, Figure 3 shows a fragment of the context-independent transition system of \mathcal{A}' . What is missing are the states which contain neither 3 nor 4, or contain 3 and 4, together with their outgoing arcs.

Figure 4 depicts two context-independent interactive processes of \mathcal{A}' starting in state $\{4\}$. The first one is context-independent as the environment does not throw in anything, while the context sequence of the second interactive process is $\{4\}\{4\}\{4\}\{4\}\{3\}\{3\}\{3\}\{3\}\{4\}$ (and so each entity provided as context was also produced as a result of the reactions enabled by the previous state). \diamond

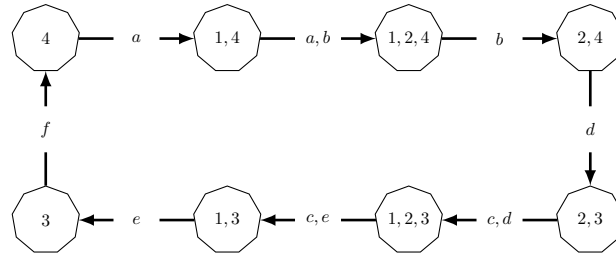


Figure 3. Forward and backward generation of a Gray code.

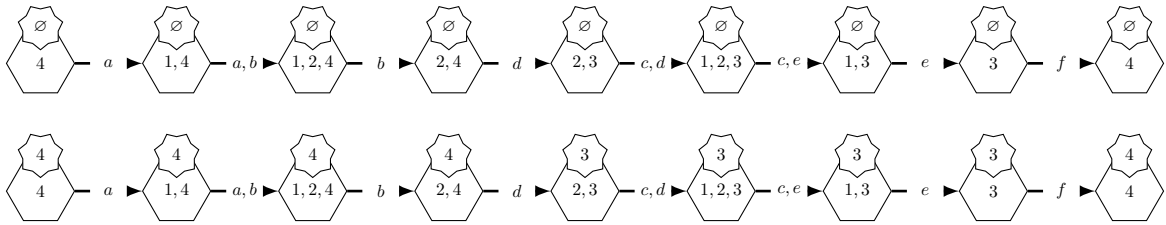


Figure 4. Two context-independent interactive processes of \mathcal{A}' that start from state 4.

Example 2.4. Consider reaction system $\mathcal{A}'' = (S, A'')$ with background set $S = \{1, 2, 3, 4\}$ and reactions $A'' = \{a, b, c, d\}$ defined as follows:

$$\begin{aligned}
 a &: \{4\} \quad , \quad \{2\} \rightarrow \{1\} & c &: \{2, 3\} , \{4\} \rightarrow \{1\} \\
 b &: \{1, 4\} , \{3\} \rightarrow \{2\} & d &: \{2\} \quad , \{1\} \rightarrow \{2\}
 \end{aligned}$$

This reaction system \mathcal{A}'' is derived from \mathcal{A}' from Example 2.3 under the assumption that entities 3 and 4 are not produced by the system itself but rather provided by the environment as context.

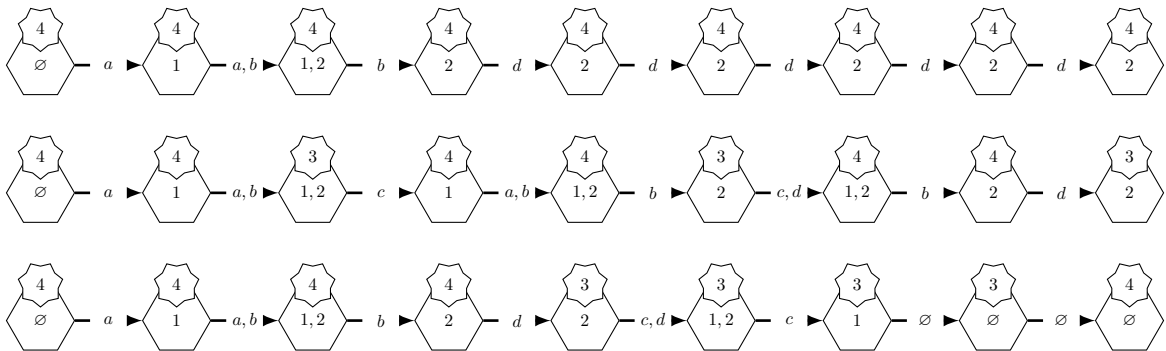


Figure 5. Interactive processes in reaction system \mathcal{A}'' with different context sequences starting from state $\{4\}$.

Figure 5 depicts three interactive processes in \mathcal{A}' : (i) in the first one, the environment throws in nine times entity 4 (for forward execution); (ii) in the second interactive process, the context sequence is $\{4\}\{4\}\{3\}\{4\}\{4\}\{3\}\{4\}\{4\}\{3\}$; and (iii) in the third one, after four 4's there are four 3's and another 4. Note that the state sequence of the last interactive process is identical with the state sequences of the two context-independent interactive processes depicted in Figure 4. \diamond

3. Direct equivalences and transition system equivalence

Let $\mathcal{A} = (S, A)$ and $\mathcal{B} = (S, B)$ be two reaction systems with the same background set. The (standard) notions of equivalence are based on the idea that a reaction system is a function from states to states, and that reaction systems are generators of interactive processes:

- \mathcal{A} and \mathcal{B} are (functionally) equivalent ([1]), denoted $\mathcal{A} \sim \mathcal{B}$, if they have the same result on every state, i.e., $res_{\mathcal{A}}(X) = res_{\mathcal{B}}(X)$ for all $X \subseteq S$.
- \mathcal{A} and \mathcal{B} are process equivalent ([16]), denoted $\mathcal{A} \sim_{\text{PROC}} \mathcal{B}$, if their interactive processes are the same, i.e., $\text{PROC}(\mathcal{A}) = \text{PROC}(\mathcal{B})$.
- \mathcal{A} and \mathcal{B} are context-independent process equivalent, denoted $\mathcal{A} \sim_{\text{CIPROC}} \mathcal{B}$, if their context-independent interactive processes are the same, i.e., $\text{CIPROC}(\mathcal{A}) = \text{CIPROC}(\mathcal{B})$.

Process equivalence is rather demanding since it relates reaction systems in all possible contexts and compares states in their entirety. For this reason, [16] considered also ways in which one might relax \sim_{PROC} by taking into account only those interactive processes that begin in certain designated states ($\emptyset \neq Z \subseteq 2^S$), and allowing only a part ($Y \subseteq S$) of the background set to be ‘visible’ to an observer:

- \mathcal{A} and \mathcal{B} are Y -projection equivalent ([16]), denoted $\mathcal{A} \sim_{\text{PROC}}^Y \mathcal{B}$, if $proj_Y(\text{STS}(\text{PROC}(\mathcal{A}))) = proj_Y(\text{STS}(\text{PROC}(\mathcal{B})))$.²
- \mathcal{A} and \mathcal{B} are context-independent Y -projection equivalent, denoted $\mathcal{A} \sim_{\text{CIPROC}}^Y \mathcal{B}$, if we have $proj_Y(\text{STS}(\text{CIPROC}(\mathcal{A}))) = proj_Y(\text{STS}(\text{CIPROC}(\mathcal{B})))$.
- \mathcal{A} and \mathcal{B} are Y -projection equivalent w.r.t. a set of initial states Z ([16]), denoted $\mathcal{A} \sim_{\text{PROC}}^{Y:Z} \mathcal{B}$, if we have $proj_Y(\text{STS}(\text{PROC}_X(\mathcal{A}))) = proj_Y(\text{STS}(\text{PROC}_X(\mathcal{B})))$, for every $X \in Z$.
- \mathcal{A} and \mathcal{B} are context-independent Y -projection equivalent w.r.t. a set of initial states Z , denoted $\mathcal{A} \sim_{\text{CIPROC}}^{Y:Z} \mathcal{B}$, if we have $proj_Y(\text{STS}(\text{CIPROC}_X(\mathcal{A}))) = proj_Y(\text{STS}(\text{CIPROC}_X(\mathcal{B})))$, for every $X \in Z$.

Example 3.1. Consider again two reaction systems: $\mathcal{A} = (S, A)$ from Example 2.1, and $\mathcal{A}' = (S, A')$ from Example 2.3.

Clearly, $\mathcal{A} \sim_{\text{CIPROC}}^{\{1\}:\{4\}} \mathcal{A}'$ since in both reaction systems the language of state sequences projected on $\{1\}$ for context-independent interactive processes starting in $\{4\}$ consists of all nonempty prefixes of $(\emptyset\{1\}\{1\}\emptyset)^*$.

²See the beginning of Section 2 for the definition of the projection function $proj$.

Also $\mathcal{A} \sim_{\text{CIPROC}}^{\{1\}} \mathcal{A}'$. In the case of \mathcal{A} there are 12 states not depicted in Figure 1. Some of them contain 1 and none contain 4, and so the result of \mathcal{A} on any of them is \emptyset . The case of \mathcal{A}' is somewhat more complicated. There are 8 states not depicted in Figure 3: (i) the result of \mathcal{A}' on $\{1, 2, 3, 4\}$, $\{1, 3, 4\}$, $\{1\}$, $\{1, 2\}$, and \emptyset is \emptyset ; (ii) the result of \mathcal{A}' on $\{3, 4\}$ is $\{4\}$; (iii) the result of \mathcal{A}' on $\{2\}$ and $\{2, 3, 4\}$ is $\{2, 3\}$. Hence, for both systems, the set of state sequences projected on $\{1\}$ consists of all nonempty prefixes of:

$$\emptyset^* \cup \{1\}\emptyset^* \cup (\emptyset\{1\}\{1\}\emptyset)^* \cup \emptyset(\emptyset\{1\}\{1\}\emptyset)^* \cup \{1\}\emptyset(\emptyset\{1\}\{1\}\emptyset)^* \cup \{1\}\{1\}\emptyset(\emptyset\{1\}\{1\}\emptyset)^* .$$

Moreover, $\mathcal{A}' \not\sim_{\text{CIPROC}} \mathcal{A} \not\sim_{\text{CIPROC}}^{\{1\}:\{3\}} \mathcal{A}' \not\sim_{\text{CIPROC}}^{\{2\}} \mathcal{A}$. \diamond

The transition systems associated with reaction systems also lead in a natural way to an equivalence notion. Reaction systems \mathcal{A} and \mathcal{B} are *transition system equivalent*, denoted $\mathcal{A} \sim_{\text{TS}} \mathcal{B}$, if they induce the same transition system, i.e., $\text{TS}(\mathcal{A}) = \text{TS}(\mathcal{B})$. Similarly, they are *context-independent transition system equivalent*, denoted by $\mathcal{A} \sim_{\text{CITS}} \mathcal{B}$, if they define the same context-independent transition system, namely $\text{CITS}(\mathcal{A}) = \text{CITS}(\mathcal{B})$.

Both \sim_{PROC} and \sim_{TS} are equivalences defined in terms of states. They are based on manifestations of behaviour expressed in terms of interactive processes and transition systems, respectively. For a given reaction system, there will normally be loss of behavioural information when one focusses solely on the state sequences of its interactive processes, or of its transition system. In general, it will not be clear for a given state, what has been produced from a previous state and what has been thrown in as context.

As expected, one can show that the equivalence relations $\sim_{\text{PROC}}^{Y:Z}$ form a ‘hierarchy’, as $\mathcal{A} \sim_{\text{PROC}}^{Y:Z} \mathcal{B}$ implies $\mathcal{A} \sim_{\text{PROC}}^{X:W} \mathcal{B}$, for all $X \subseteq Y \subseteq S$ and $\emptyset \neq W \subseteq Z \subseteq 2^S$ (see [16]). What is perhaps less expected, is that process equivalence holds whenever reaction systems are equivalent under full observability with respect to any nonempty set of initial states (in fact, even one initial state is sufficient). Moreover, the process, functional, and transition system equivalences coincide.

Proposition 3.2. Let \mathcal{A} and \mathcal{B} be reaction systems with the same background set S . Then $\mathcal{A} \sim \mathcal{B}$ iff $\mathcal{A} \sim_{\text{PROC}} \mathcal{B}$ iff $\mathcal{A} \sim_{\text{CIPROC}} \mathcal{B}$ iff $\mathcal{A} \sim_{\text{TS}} \mathcal{B}$ iff $\mathcal{A} \sim_{\text{CITS}} \mathcal{B}$ iff $\mathcal{A} \sim_{\text{PROC}}^{S:Z} \mathcal{B}$ for some nonempty $Z \subseteq 2^S$.

Proof:

Corollary 2 from [16] states that $\mathcal{A} \sim \mathcal{B} \iff \mathcal{A} \sim_{\text{PROC}} \mathcal{B} \iff (\exists \emptyset \neq Z \subseteq 2^S) \mathcal{A} \sim_{\text{PROC}}^{S:Z} \mathcal{B}$. Proposition 2.2 implies $\mathcal{A} \sim_{\text{PROC}} \mathcal{B} \iff \mathcal{A} \sim_{\text{TS}} \mathcal{B}$ and $\mathcal{A} \sim_{\text{CIPROC}} \mathcal{B} \iff \mathcal{A} \sim_{\text{CITS}} \mathcal{B}$. Moreover, $\mathcal{A} \sim_{\text{TS}} \mathcal{B} \iff \mathcal{A} \sim_{\text{CITS}} \mathcal{B}$ is immediate. Hence all the equivalences are satisfied. \square

The above result cannot be extended to arbitrary equivalence relations $\sim_{\text{PROC}}^{Y:Z}$ as we have $\mathcal{A} \sim_{\text{PROC}}^{\emptyset:Z} \mathcal{B}$, for all reaction systems \mathcal{A} and \mathcal{B} . Similarly, it cannot be extended to $\mathcal{A} \sim_{\text{CIPROC}}^{S:Z} \mathcal{B}$ for arbitrary nonempty Z , since $\mathcal{A} \sim_{\text{CIPROC}}^{S:\{\emptyset, S\}} \mathcal{B}$, for all reaction systems \mathcal{A} and \mathcal{B} .

4. Enabling equivalence

When considering the result of a single reaction on a given state, then this reaction is either enabled or not. However, in the case of a set of reactions, it is possible that some of the reactions are enabled and

some not. For this reason, [17] motivated, introduced, and investigated a notion of equivalence which treats a set of reactions as if it were a single reaction, and so this set is considered relevant only for those states that enable all its reactions.

Considering a set of reactions as one ‘block’ that acts in unison leads to a different notion of equivalence for reaction systems. Reaction systems $\mathcal{A} = (S, A)$ and $\mathcal{B} = (S, B)$ are *enabling equivalent* if, for every $X \subseteq S$, $en_{\mathcal{A}}(X)$ iff $en_{\mathcal{B}}(X)$, and $en_{\mathcal{A}}(X)$ implies $res_{\mathcal{A}}(X) = res_{\mathcal{B}}(X)$. We denote this by $\mathcal{A} \doteq \mathcal{B}$.

There is a syntactic characterisation of enabling equivalence of \mathcal{A} and \mathcal{B} which allows one to check whether $\mathcal{A} \doteq \mathcal{B}$ just by inspecting the respective sets of reactions. Here a reaction system (S, A) is said to be *consistent* if $R_A \cap I_A = \emptyset$. Note that there is a state $X \subseteq S$ that enables A iff (S, A) is consistent.

Proposition 4.1. ([17])

$\mathcal{A} \doteq \mathcal{B}$ iff $R_A = R_B$, $I_A = I_B$, and $P_A = P_B$, whenever A and B are consistent.

In other words, in order to establish the enabling equivalence of two reaction systems, it is sufficient to compare them as if they had a single reaction. Functional equivalence and enabling equivalence are incomparable notions (see Examples 3.3 and 3.4 in [17]), but for reaction systems with only a single reaction they coincide.

Enabling process equivalence

The process based equivalence notions discussed so far were all related to functional equivalence in the sense that they all were based on the combined result of the reactions of the reaction systems under consideration. Now we will rely on enabling equivalence for our comparisons which implies that we only consider the result of the reactions of a reaction system on a state when all its reactions are enabled by that state.

We use the following notation: for a sequence $\xi = X_1 \dots X_n$ of subsets of the background set of a reaction system $\mathcal{A} = (S, A)$ we let $[\xi]_{en}^{\mathcal{A}}$ be the sequence obtained from ξ by deleting all X_i such that neither $en_{\mathcal{A}}(X_i)$ nor $en_{\mathcal{A}}(X_{i-1})$ (if $i > 1$). In other words, we retain only those states at which A is enabled, and the states immediately following them. This reflects the idea that we are interested only in states that enable all reactions from A and the result of A on these states. For a set Ξ of sequences of subsets of the background set of reaction system \mathcal{A} , we denote $[\Xi]_{en}^{\mathcal{A}} = \{[\xi]_{en}^{\mathcal{A}} \mid \xi \in \Xi\}$. Following this, reaction systems \mathcal{A} and \mathcal{B} with the same background set are:

- *ci-process enabling equivalent* if $[\text{STS}(\text{CIPROC}(\mathcal{A}))]_{en}^{\mathcal{A}} = [\text{STS}(\text{CIPROC}(\mathcal{B}))]_{en}^{\mathcal{B}}$.
- *process enabling equivalent* if $[\text{STS}(\text{PROC}(\mathcal{A}))]_{en}^{\mathcal{A}} = [\text{STS}(\text{PROC}(\mathcal{B}))]_{en}^{\mathcal{B}}$.

We denote this respectively by $\mathcal{A} \doteq_{\text{CIPROC}} \mathcal{B}$ and $\mathcal{A} \doteq_{\text{PROC}} \mathcal{B}$.

It turns out that ci-process enabling equivalence implies enabling equivalence.

Proposition 4.2. $\mathcal{A} \doteq_{\text{CIPROC}} \mathcal{B}$ implies $\mathcal{A} \doteq \mathcal{B}$, for all consistent reaction systems \mathcal{A} and \mathcal{B} over the same background set.

Proof:

Let $\mathcal{A} = (S, A)$, $\mathcal{B} = (S, B)$, and let X be any subset of S such that $en_{\mathcal{A}}(X)$ (such a subset exists since \mathcal{A} is consistent). Then there exists a context-independent process π of \mathcal{A} with initial state X satisfying $[STS(\pi)]_{en}^{\mathcal{A}} = XY$ where $Y = res_{\mathcal{A}}(X)$. Hence, by $\mathcal{A} \doteq_{\text{CIPROC}} \mathcal{B}$, there exists a context-independent process π' of \mathcal{B} such that $[STS(\pi')]_{en}^{\mathcal{B}} = XY$. Thus, by the definition of $[STS(\text{CIPROC}(\mathcal{B}))]_{en}^{\mathcal{B}}$, we have $en_{\mathcal{B}}(X)$ and $res_{\mathcal{B}}(X) = Y = res_{\mathcal{A}}(X)$. \square

As the next example demonstrates, the reverse implication does not hold.

Example 4.3. Consider reaction system $\mathcal{A}_1 = (S, \{a\})$ with background set $S = \{1, 2, 3\}$ and one reaction:

$$a : \{3\}, \{1, 2\} \rightarrow \{2, 3\}$$

and reaction system $\mathcal{A}'_1 = (S, \{a', b'\})$ with the same background set S and two reactions:

$$a' : \{3\}, \{2\} \rightarrow \{2\}$$

$$b' : \{3\}, \{1\} \rightarrow \{3\}$$

By Proposition 4.1, we have $\mathcal{A}_1 \doteq \mathcal{A}'_1$.

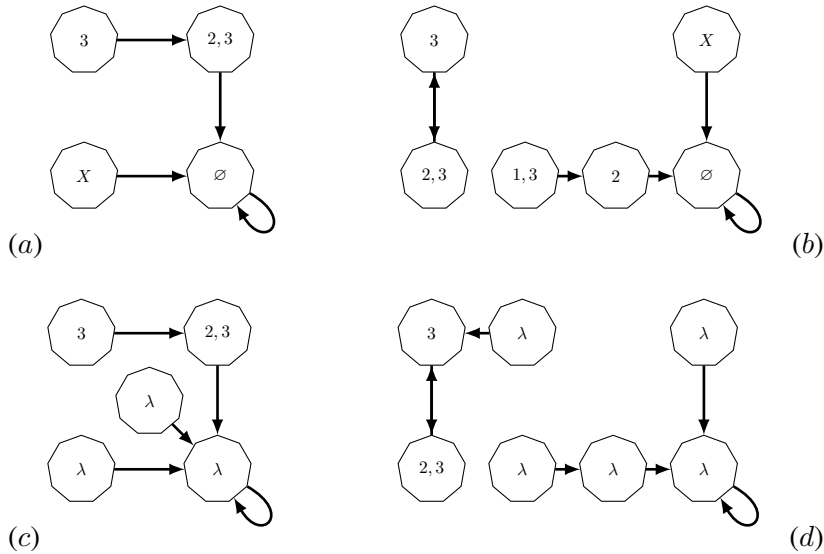


Figure 6. The context-independent transition system of reaction system \mathcal{A}_1 in Example 4.3 (a), and the context-independent transition system of reaction system \mathcal{A}'_1 (b), where the X stand for any subset of the background set other than those depicted explicitly. Also, the ci-enabling transition system of \mathcal{A}_1 (c), and the ci-enabling transition system of \mathcal{A}'_1 (d). Note that the bottom left state in (c), and the top right state in (d) have several omitted duplicates. The middle state in (c) and the middle state in the top row in (d) are initial states that correspond to $\{2, 3\}$ (which, as initial state, is not the state that directly follows the state at which the set of reactions of \mathcal{A}_1 and \mathcal{A}'_1 , respectively, is enabled).

The context-independent transition systems of \mathcal{A}_1 and \mathcal{A}'_1 are given in Figure 6(a, b). From these (see Proposition 2.2), it is easily seen that

$$\{3\}\{2, 3\}\{3\}\{2, 3\} \in [\text{STS}(\text{CIPROC}(\mathcal{A}'_1))]_{en}^{\mathcal{A}'_1} \setminus [\text{STS}(\text{CIPROC}(\mathcal{A}_1))]_{en}^{\mathcal{A}_1}.$$

Hence $\mathcal{A}_1 \not\equiv_{\text{CIPROC}} \mathcal{A}'_1$. ◇

The discriminating power of ci-process enabling equivalence is greater than that of process enabling equivalence.

Proposition 4.4. $\mathcal{A} \doteq_{\text{PROC}} \mathcal{B}$ iff $\mathcal{A} \doteq \mathcal{B}$, for all consistent reaction systems \mathcal{A} and \mathcal{B} over the same background set.

Proof:

Let $\mathcal{A} = (S, A)$, $\mathcal{B} = (S, B)$, and $\mathcal{D} = (S, D) \in \{\mathcal{A}, \mathcal{B}\}$. For $k = 1, 2$, let $\text{STS}_k^{\mathcal{D}}$ be the set of all sequences in $[\text{STS}(\text{PROC}(\mathcal{D}))]_{en}^{\mathcal{D}}$ of length k . Since \mathcal{D} is a consistent reaction system, $\text{STS}_1^{\mathcal{D}} \neq \emptyset \neq \text{STS}_2^{\mathcal{D}}$. We then observe that, by the definition of $[\text{STS}(\text{PROC}(\mathcal{D}))]_{en}^{\mathcal{D}}$, for every $X \subseteq S$ we have:

$$en_{\mathcal{D}}(X) \iff X \in \text{STS}_1^{\mathcal{D}}. \quad (1)$$

Moreover, by the definition of $[\text{STS}(\text{PROC}(\mathcal{D}))]_{en}^{\mathcal{D}}$, for every $X \subseteq S$ satisfying $en_{\mathcal{D}}(X)$ we have:

$$res_{\mathcal{D}}(X) = P_{\mathcal{D}} = \bigcap_{XY \in \text{STS}_2^{\mathcal{D}}} Y. \quad (2)$$

(\implies) Suppose that $\mathcal{A} \doteq_{\text{PROC}} \mathcal{B}$. Then $\text{STS}_k^{\mathcal{A}} = \text{STS}_k^{\mathcal{B}}$, for $k = 1, 2$.

Let $X \subseteq S$. Then, by (1) and $\text{STS}_1^{\mathcal{A}} = \text{STS}_1^{\mathcal{B}}$, we have $en_{\mathcal{A}}(X)$ iff $en_{\mathcal{B}}(X)$. Moreover, if $en_{\mathcal{A}}(X)$ (and so also $en_{\mathcal{B}}(X)$), then $res_{\mathcal{A}}(X) = res_{\mathcal{B}}(X)$, which follows from (2) and $\text{STS}_2^{\mathcal{A}} = \text{STS}_2^{\mathcal{B}}$. Hence $\mathcal{A} \doteq \mathcal{B}$.

(\impliedby) We first observe that the empty sequence belongs to $[\text{STS}(\text{PROC}(\mathcal{B}))]_{en}^{\mathcal{B}}$, as $\pi = (\emptyset, \emptyset)$ is an interactive process of \mathcal{B} . Suppose then that $\mathcal{A} \doteq \mathcal{B}$ and $X_1 \dots X_n \in [\text{STS}(\text{PROC}(\mathcal{A}))]_{en}^{\mathcal{A}}$ ($n \geq 1$). We proceed by induction on n to show that $X_1 \dots X_n \in [\text{STS}(\text{PROC}(\mathcal{B}))]_{en}^{\mathcal{B}}$.

In the base case, we have $X_1 \in \text{STS}_1^{\mathcal{A}}$. Hence, by (1), $en_{\mathcal{A}}(X_1)$. Thus, by $\mathcal{A} \doteq \mathcal{B}$, $en_{\mathcal{B}}(X_1)$. We then observe that $\pi = (X_1, \emptyset)$ is an interactive process of \mathcal{B} such that $\text{STS}(\pi) = X_1$. This and $en_{\mathcal{B}}(X_1)$ means that $[\text{STS}(\pi)]_{en}^{\mathcal{B}} = X_1$.

In the induction step we assume that $X_1 \dots X_n X_{n+1} \in [\text{STS}(\text{PROC}(\mathcal{A}))]_{en}^{\mathcal{A}}$ and $X_1 \dots X_n \in [\text{STS}(\text{PROC}(\mathcal{B}))]_{en}^{\mathcal{B}}$. The latter means that there is an interactive process $\pi = (\gamma, \delta)$ of \mathcal{B} such that $[\text{STS}(\pi)]_{en}^{\mathcal{B}} = X_1 \dots X_n$ and $\text{STS}(\pi) = \xi X_n$, for some sequence ξ . We then consider two cases.

Case 1: $en_{\mathcal{A}}(X_n)$. Then, by the definition of $[\text{STS}(\text{PROC}(\mathcal{A}))]_{en}^{\mathcal{A}}$, there is an interactive process $\pi' = (\gamma', \delta')$ of \mathcal{A} such that $\text{STS}(\pi') = \xi' X_n X_{n+1}$. Hence $res_{\mathcal{A}}(X_n) \subseteq X_{n+1}$. We then observe that, by $\mathcal{A} \doteq \mathcal{B}$, $en_{\mathcal{B}}(X_n)$ as well as $res_{\mathcal{B}}(X_n) = res_{\mathcal{A}}(X_n) \subseteq X_{n+1}$. Let $\pi'' = (\gamma X_{n+1}, \delta res_{\mathcal{B}}(X_n))$. We observe that π'' is an interactive process of \mathcal{B} such that

$$\text{STS}(\pi'') = \xi X_n (X_{n+1} \cup res_{\mathcal{B}}(X_n)).$$

Hence, by $res_B(X_n) \subseteq X_{n+1}$ and $en_B(X_n)$ and the induction hypothesis, we obtain:

$$[\text{STS}(\pi'')]_{en}^{\mathcal{B}} = [\text{STS}(\pi)]_{en}^{\mathcal{B}} X_{n+1} = X_1 \dots X_n X_{n+1} .$$

Case 2: $\neg en_A(X_n)$. Then, by the definition of $[\text{STS}(\text{PROC}(\mathcal{A}))]_{en}^{\mathcal{A}}$, we have $en_A(X_{n+1})$. Hence, by $\mathcal{A} \doteq \mathcal{B}$, we also have $en_A(X_{n+1})$. Let $\pi'' = (\gamma S X_{n+1}, \delta res_B(X_n) \emptyset)$. We observe that π'' is a valid interactive process of \mathcal{B} . Moreover, we have $\text{STS}(\pi'') = \text{STS}(\pi) S X_{n+1}$, and so, by $\neg en_B(S)$ and $en_B(X_{n+1})$ and the induction hypothesis, we obtain:

$$[\text{STS}(\pi'')]_{en}^{\mathcal{B}} = [\text{STS}(\pi)]_{en}^{\mathcal{B}} X_{n+1} = X_1 \dots X_n X_{n+1} .$$

In this way, we have shown that $[\text{STS}(\text{PROC}(\mathcal{A}))]_{en}^{\mathcal{A}} \subseteq [\text{STS}(\text{PROC}(\mathcal{A}))]_{en}^{\mathcal{B}}$. And, similarly, we can show that $[\text{STS}(\text{PROC}(\mathcal{A}))]_{en}^{\mathcal{A}} \supseteq [\text{STS}(\text{PROC}(\mathcal{A}))]_{en}^{\mathcal{B}}$. Hence $\mathcal{A} \doteq_{\text{PROC}} \mathcal{B}$. \square

Enabling transition system equivalence

Next we relate enabling equivalence to the transition systems of reaction systems.

Example 4.5. Consider reaction system $\mathcal{A} = (S, \{a, b, c\})$, as in Example 2.1, with background set $S = \{1, 2, 3, 4\}$ and three reactions:

$$\begin{aligned} a : \{4\} \quad , \{3\} &\rightarrow \{4\} \\ b : \{1, 4\} \quad , \{3\} &\rightarrow \{2\} \\ c : \{4\} \quad , \{2, 3\} &\rightarrow \{1\} \end{aligned}$$

and reaction system $\widehat{\mathcal{A}} = (S, \{\widehat{a}, \widehat{b}, \widehat{c}\})$ with the same background set and three reactions:

$$\begin{aligned} \widehat{a} : \{4\} \quad , \{3\} &\rightarrow \{4\} \\ \widehat{b} : \{1, 4\} \quad , \{3\} &\rightarrow \{1\} \\ \widehat{c} : \{4\} \quad , \{2, 3\} &\rightarrow \{2\} \end{aligned}$$

Then we have $\mathcal{A} \doteq \widehat{\mathcal{A}}$ and $\mathcal{A} \doteq_{\text{CIPROC}} \widehat{\mathcal{A}}$ (as the only state where both sets of reactions are enabled is $\{1, 4\}$), but the context-independent transition systems of \mathcal{A} and $\widehat{\mathcal{A}}$, depicted in Figure 7, are not only different, but structurally different. Note also that if we could see the initial state of the context-independent processes of reaction systems, then the reaction systems \mathcal{A} and $\widehat{\mathcal{A}}$ would no longer be equivalent. This distinction can easily be seen by starting both reaction systems in state $\{4\}$. \diamond

As a result, providing a characterisation of ci-process enabling equivalence using transition systems calls for a new transition system representation of ci-enabling processes, as described next.

The *ci-enabling transition system* of a reaction system $\mathcal{A} = (S, A)$ is a tuple $\text{CITS}_{en}(\mathcal{A}) = (Q, V, \ell, Q')$, where Q is a finite set of nodes, $V \subseteq Q \times Q$ is a set of arcs, $\ell : Q \rightarrow 2^S \cup \{\lambda\}$ is a labelling of the nodes, and $Q' \subseteq Q$ is a nonempty set of initial nodes, such that the following hold:

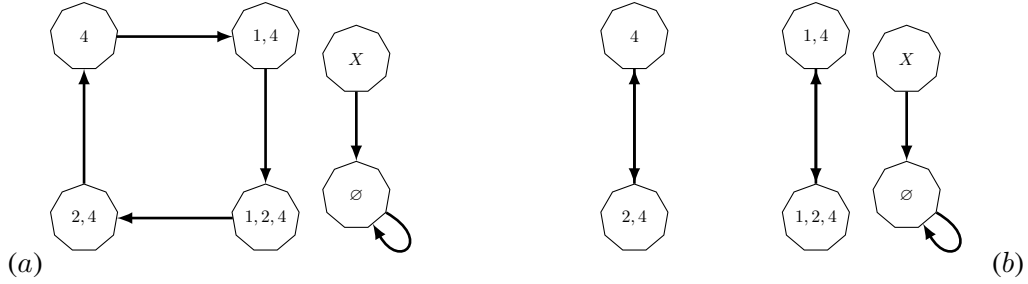


Figure 7. The context-independent transition system of \mathcal{A} (a), and the context-independent transition system of $\widehat{\mathcal{A}}$ (b). The X stands for any subset of the background set other than those depicted explicitly.

- If $en_A(P_A)$ then:
 - $Q = Q' = 2^S$
 - $V = \{(X, res_A(X)) \mid X \subseteq S\}$
 - $\ell(X) = X$, for all X such that $en_A(X)$; otherwise $\ell(X) = \lambda$.
- If $\neg en_A(P_A)$ then:
 - $Q = 2^S \uplus \{\perp\}$ and $Q' = 2^S$
 - $V = \{(X, res_A(X)) \mid X \subseteq S \wedge \neg en_A(X)\} \cup \{(X, \perp) \mid X \subseteq S \wedge en_A(X)\} \cup \{(\perp, res_A(P_A))\}$
 - $\ell(X) = X$, for all X such that $en_A(X)$; $\ell(\perp) = P_A$; otherwise $\ell(X) = \lambda$.

The *state sequences* $STS(\text{CITS}_{en}(\mathcal{A}))$ of $\text{CITS}_{en}(\mathcal{A})$ are defined by

$$STS(\text{CITS}_{en}(\mathcal{A})) = \{\ell(Y_1 \cdots Y_n) \mid Y_1 \cdots Y_n \text{ is a path in } \text{CITS}_{en}(\mathcal{A}) \text{ with } Y_1 \in Q'\}.$$

Basically, $\text{CITS}_{en}(\mathcal{A})$ is the context-independent transition system of \mathcal{A} enriched with labels (and in the second case an additional state \perp). The empty sequence λ is used as a ‘silent’ label to hide those states that do not enable the full set of reactions A (or are not the followers of such states). Moreover, a distinction is made between whether or not the set consisting of all products of the reactions of \mathcal{A} (i.e., P_A) enables all reactions of \mathcal{A} . If not (the second case above), V connects all states X of \mathcal{A} with the result of A on X provided that not all reactions of \mathcal{A} are enabled by X ; otherwise X is connected to \perp (that represents $res_A(X) = P_A$) which in turn is connected to $res_A(P_A)$.

Note that in this (second) case P_A is labelled with λ , while \perp is labelled with P_A . Having the intermediate state \perp makes it possible to distinguish between on the one hand the situation when P_A is the initial state or has been obtained from only a subset of the available reactions, and on the other hand the situation when P_A has been the result of executing all the reactions of \mathcal{A} .

The soundness of the definition of ci-enabling transition system follows from the next result, where $\mathcal{A} \doteq_{\text{CITS}} \mathcal{B}$ means that $STS(\text{CITS}_{en}(\mathcal{A})) = STS(\text{CITS}_{en}(\mathcal{B}))$.

Proposition 4.6. $\mathcal{A} \doteq_{\text{CIPROC}} \mathcal{B}$ iff $\mathcal{A} \doteq_{\text{CITS}} \mathcal{B}$, for all consistent reaction systems \mathcal{A} and \mathcal{B} over the same background set.

Proof:

It suffices to observe that $\text{STS}(\text{CITS}_{en}(\mathcal{D})) = [\text{STS}(\text{PROC}(\mathcal{D}))]_{en}^{\mathcal{D}}$, for $\mathcal{D} \in \{\mathcal{A}, \mathcal{B}\}$, which follows directly from the definitions. \square

Thus this result provides also a finite means for checking whether two infinite sets of sequences are the same.

Example 4.7. Figure 8 shows the ci-enabling transition systems of the reaction systems from Example 4.5. Clearly, we have $\text{STS}(\text{CITS}_{en}(\mathcal{A})) = \text{STS}(\text{CITS}_{en}(\widehat{\mathcal{A}}))$. Moreover, Figure 6(c, d) shows the ci-enabling transition systems of the reaction systems from Example 4.3. \diamond

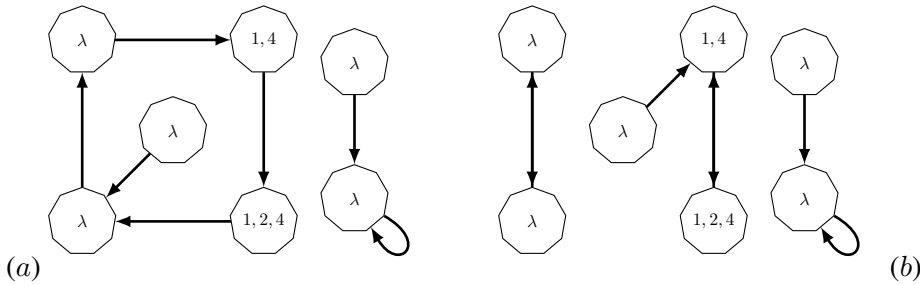


Figure 8. The ci-enabling transition system of reaction system \mathcal{A} in Example 4.5 (a), and the ci-enabling transition system of $\widehat{\mathcal{A}}$ (b). The rightmost top states stand for several duplicates. The middle state in (a) and the state in the second row in (b) are initial states that correspond to $\{1, 2, 4\}$ (which, as initial state, is not the state that directly follows the state at which the set of reactions of \mathcal{A} and $\widehat{\mathcal{A}}$, respectively, is enabled).

So far it was implicitly assumed that both the initial states and context sequences are unknown to any observers of processes of reaction systems. A detailed analysis of the implications of relaxing these constraints is beyond the scope of the present paper. That such analysis is justified follows from our last result that the enabling process equivalence with visible context sequences has more discriminating power than other equivalences defined so far.

To show this, let \mathcal{A} and \mathcal{B} be two reaction systems with the same background set. Then $\mathcal{A} \doteq_{\text{PROC}}^{\text{viscxt}} \mathcal{B}$ if for all interactive processes, $\pi = (\gamma, \delta)$ and $\pi' = (\gamma, \delta')$ with the same initial state, of respectively \mathcal{A} and \mathcal{B} , we have $[\text{STS}(\pi)]_{en}^{\mathcal{A}} = [\text{STS}(\pi')]_{en}^{\mathcal{B}}$.

Theorem 4.8. Let \mathcal{A} and \mathcal{B} be two reaction systems with the same background set. Then:

$$\mathcal{A} \doteq_{\text{PROC}}^{\text{viscxt}} \mathcal{B} \implies \mathcal{A} \doteq_{\text{CIPROC}} \mathcal{B} \iff \mathcal{A} \doteq_{\text{CITS}} \mathcal{B} \implies \mathcal{A} \doteq \mathcal{B} \iff \mathcal{A} \doteq_{\text{PROC}} \mathcal{B}$$

and no new implications can be added.

Proof:

The first implication follows directly from the definition $\stackrel{\text{visext}}{\equiv}_{\text{PROC}}$, and the second from Proposition 4.2. The first equivalence follows from Proposition 4.6, and the second from Proposition 4.4. Example 4.3 shows that that the second implication cannot be reversed. Finally, the first implication cannot be reversed, as for the reaction systems in Example 4.5, we have $\mathcal{A} \stackrel{\text{CIPROC}}{\equiv} \widehat{\mathcal{A}}$ and $\mathcal{A} \not\stackrel{\text{visext}}{\equiv}_{\text{PROC}} \widehat{\mathcal{A}}$. Indeed, it suffices to take the context sequence $\emptyset\emptyset\emptyset$ and consider the interactive process $\pi = (\emptyset\emptyset\emptyset, \{1, 4\}\{1, 2, 4\}\{1, 4\})$ in $\widehat{\mathcal{A}}$ for which we have $[\text{STS}(\pi)]_{en}^{\widehat{\mathcal{A}}} = \{1, 4\}\{1, 2, 4\}\{1, 4\}$. However, there is no interactive process $\pi' = (\emptyset\emptyset\emptyset, \delta)$ in \mathcal{A} such that $[\text{STS}(\pi')]_{en}^{\mathcal{A}} = \{1, 4\}\{1, 2, 4\}\{1, 4\}$, which can be shown by checking each potential first set in δ . \square

5. Concluding remarks

The notion of both static and dynamic equivalence in the domain of reaction systems had been investigated in, e.g., [17, 19, 16, 12]. In this paper, we investigated the concept of enabling equivalence introduced in [17], which provided a novel way of capturing equivalent evolutions in the living systems. In particular, we investigated possible ways of incorporating such an equivalence as a replacement of functional equivalence in the range of equivalence notions introduced in [16].

We have demonstrated that enabling equivalence has clearly different discrimination power when compared with functional equivalence which played a central role in [16]. We believe that a further comparison and contrasting of these two approaches to behavioural equivalence in reaction systems is needed. In particular, we plan to investigate this when context sequences are constrained rather than arbitrary and are provided, for instance, by the context controllers introduced in [16].

Acknowledgments

First of all we would like to express our gratitude to Giancarlo Mauri for providing us with an overview of the field as well as pleasant circumstances to pursue our joint endeavours into the realm of reaction systems. We are grateful to the anonymous reviewers for their constructive feedback leading to an improved presentation of our results. Finally, the research presented in this paper was partially supported by the Polish National Agency for Academic Exchange.

References

- [1] Ehrenfeucht A, Rozenberg G. Reaction systems. *Fundam. Inform.*, 2007. **75**(1-4):263–280. URL <http://content.iospress.com/articles/fundamenta-informaticae/fi75-1-4-15>.
- [2] Brijder R, Ehrenfeucht A, Main MG, Rozenberg G. A tour of reaction systems. *Int. J. Found. Comput. Sci.*, 2011. **22**(7):1499–1517. doi:10.1142/S0129054111008842.
- [3] Corolli L, Maj C, Marini F, Besozzi D, Mauri G. An excursion in reaction systems: From computer science to biology. *Theor. Comput. Sci.*, 2012. **454**:95–108. doi:10.1016/j.tcs.2012.04.003.

- [4] Dennunzio A, Formenti E, Manzoni L. Extremal combinatorics of reaction systems. In: Dediu A, Martín-Vide C, Sierra-Rodríguez JL, Truthe B (eds.), Language and Automata Theory and Applications - 8th International Conference, LATA 2014, Madrid, Spain, March 10-14, 2014. Proceedings, volume 8370 of *Lecture Notes in Computer Science*. Springer, 2014 pp. 297–307. URL https://doi.org/10.1007/978-3-319-04921-2_24.
- [5] Ehrenfeucht A, Kleijn J, Koutny M, Rozenberg G. Reaction systems: A natural computing approach to the functioning of living cells. In: *A Computable Universe: Understanding and Exploring Nature As Computation*. World Scientific, 2012 pp. 189–208.
- [6] Dennunzio A, Formenti E, Manzoni L, Porreca AE. Ancestors, descendants, and gardens of Eden in reaction systems. *Theor. Comput. Sci.*, 2015. **608**:16–26. doi:10.1016/j.tcs.2015.05.046. URL <https://doi.org/10.1016/j.tcs.2015.05.046>.
- [7] Ehrenfeucht A, Kleijn J, Koutny M, Rozenberg G. Minimal reaction systems. *Trans. Computational Systems Biology*, 2012. **14**:102–122. doi:10.1007/978-3-642-35524-0_5.
- [8] Ehrenfeucht A, Petre I, Rozenberg G. Reaction systems: A model of computation inspired by the functioning of the living cell. In: *The Role of Theory in Computer Science: Essays Dedicated to Janusz Brzozowski*. World Scientific, 2017 pp. 1–32. URL https://doi.org/10.1142/9789813148208_0001.
- [9] Ehrenfeucht A, Rozenberg G. Events and modules in reaction systems. *Theor. Comput. Sci.*, 2007. **376**(1-2):3–16. doi:10.1016/j.tcs.2007.01.008.
- [10] Ehrenfeucht A, Rozenberg G. Zoom structures and reaction systems yield exploration systems. *Int. J. Found. Comput. Sci.*, 2014. **25**(3):275–306. doi:10.1142/S0129054114500142.
- [11] Formenti E, Manzoni L, Porreca AE. Fixed points and attractors of reaction systems. In: Beckmann A, Csuhaj-Varjú E, Meer K (eds.), Language, Life, Limits - 10th Conference on Computability in Europe, CiE 2014, Budapest, Hungary, June 23-27, 2014. Proceedings, volume 8493 of *Lecture Notes in Computer Science*. Springer, 2014 pp. 194–203. URL https://doi.org/10.1007/978-3-319-08019-2_20.
- [12] Pardini G, Barbuti R, Maggiolo-Schettini A, Milazzo P, Tini S. Compositional semantics and behavioural equivalences for reaction systems with restriction. *Theor. Comput. Sci.*, 2014. **551**:1–21. doi:10.1016/j.tcs.2014.04.010.
- [13] Salomaa A. Functions and sequences generated by reaction systems. *Theor. Comput. Sci.*, 2012. **466**:87–96. doi:10.1016/j.tcs.2012.07.022.
- [14] Salomaa A. On state sequences defined by reaction systems. In: Constable RL, Silva A (eds.), Logic and Program Semantics - Essays Dedicated to Dexter Kozen on the Occasion of His 60th Birthday, volume 7230 of *Lecture Notes in Computer Science*. Springer, 2012 pp. 271–282. doi:10.1007/978-3-642-29485-3_17.
- [15] Salomaa A. Minimal and almost minimal reaction systems. *Natural Computing*, 2013. **12**(3):369–376. URL <https://doi.org/10.1007/s11047-013-9372-y>.
- [16] Kleijn J, Koutny M, Mikulski L, Rozenberg G. Reaction systems, transition systems, and equivalences. In: Böckenhauer H, Komm D, Unger W (eds.), Adventures Between Lower Bounds and Higher Altitudes - Essays Dedicated to Juraj Hromkovič on the Occasion of His 60th Birthday, volume 11011 of *Lecture Notes in Computer Science*. Springer, 2018 pp. 63–84. doi:10.1007/978-3-319-98355-4_5.
- [17] Ehrenfeucht A, Kleijn J, Koutny M, Rozenberg G. Evolving reaction systems. *Theor. Comput. Sci.*, 2017. **682**:79–99. doi:10.1016/j.tcs.2016.12.031.

- [18] Gray F. Pulse code communications, 1953. US Patent 2 632 058.
- [19] Genova D, Hoogeboom HJ, Jonoska N. A graph isomorphism condition and equivalence of reaction systems. *Theor. Comput. Sci.*, 2017. **701**:109–119. URL <https://doi.org/10.1016/j.tcs.2017.05.019>.