Generalized contexts for reaction systems: definition and study of dynamic causalities

Roberto Barbuti · **Roberta Gori** · **Francesca Levi** ·
Paolo Milazzo

Abstract *Reaction systems* are a qualitative formalism for the modelling of systems of biochemical reactions. In their original formulation, a reaction system executes in an environment (or context) that can supply it with new objects at each evolution step. The context drives the behaviour of a reaction system: it can provide different inputs to the system that can lead to different behaviours. In order to more faithfully deal with open systems, in this paper we propose a more powerful notion of context having not only the capability to provide objects, but also to absorb (or remove) objects at each evolution step.

For such reaction systems with generalized context we investigate properties of *dynamic causality* by revising the previously proposed concept of formula based predictor. A formula based predictor is a boolean formula characterising all contexts that lead to the production of a certain object after a given number of steps.

In this paper, we revise the theory of formula based predictors in order to deal with reaction systems executed in a context of the new kind. As applications, we show an example of interaction between biochemical pathways and a reaction system modelling cell metabolism and respiration.

1 Introduction

In the last decades, the mechanisms underlying the functioning of living cells have been the source of inspiration of many formalisms and notations in the field of Natural Computing

R. Barbuti
Dipartimento di Informatica, Università di Pisa
E-mail: barbuti@di.unipi.it

R. Gori
Dipartimento di Informatica, Università di Pisa
E-mail: gori@di.unipi.it

F. Levi
Dipartimento di Informatica, Università di Pisa
E-mail: levifran@di.unipi.it

P. Milazzo
Dipartimento di Informatica, Università di Pisa
E-mail: milazzo@di.unipi.it

[17,9]. Many of these formalisms are based on rewriting approaches (e.g. membrane systems [19,20]). This is due to the similarity between rewrite rules and chemical reactions (that govern the functioning of living cells) and to the fact that rewriting approaches can be studied from the viewpoint of computing power with techniques that are typical of the theory of formal languages.

Reaction systems [11,6] were introduced by Ehrenfeucht and Rozenberg as a novel model for the description of biochemical processes driven by the interaction among reactions in living cells. Reaction systems are based on two opposite mechanisms, namely *facilitation* and *inhibition*. Facilitation means that a reaction can occur only if all its reactants are present, while inhibition means that the reaction cannot occur if any of its inhibitors is present. A rewrite rule of a reaction system (called *reaction*) is hence a triple (R, I, P), where R, I and P are sets of objects representing reactants, inhibitors and products of the modeled chemical reaction. A *reaction system* is represented by a set of reactions having such a form, together with a (finite) support set S containing all of the objects that can appear in a reaction.

The state of a reaction system consists of a finite set of objects, describing the biological entities that are present in the real system being modeled. In particular, the presence of an object in the state expresses the fact that the corresponding biological entity, in the real system being modeled, is present in a number of copies as high as needed. This is the *threshold supply* assumption and characterizes reaction systems as a *qualitative* modeling formalism.

A reaction system evolves by means of the application of its reactions. A reaction is applicable if its reactants are present and its inhibitor are not present in the current state of the system. The threshold supply assumption ensures that the application of different reactions never compete for their reactants, and hence all the applicable reactions in a step are always applied. The result of the application of a set of reactions results in the introduction of all of their products in the next state of the system. Reaction systems assume the *non permanency* of the elements, namely unused elements are never carried over to the next state. In particular, the next state consists only of the products of the reactions applied in the current step. This is one of the most original bio-inspired features of reaction systems that distinguishes it from the other formalisms mentioned above.

The overall behavior of a reaction system model is driven by the (set of) contextual elements which are received from the external environment at each step. Such elements join the current state of the system and, as the other objects in the system state, can enable or disable reactions. The computation of the next state of a reaction system is a deterministic procedure. Consequently, if the contextual elements provided to the system at each step are know, then the whole execution of the reaction system is determinated. However, if the contextual elements provided to the system step by step are not known, the description of the overall system dynamics becomes non deterministic since the contextual elements that can be received at each step can be any subset of the support set S.

The role of the environment in reaction systems is really crucial since it is intended to model (through context sequences, sequences of object supplied at each step) the interaction that a biological system modelled in isolation may have with all the other surrounding biological processes. Indeed, environments are intended to model an open systems where the emphasis is on the particular behaviour we are modelling. Consider, for example, the case we are not interested in describing all possible reactions that compose a biological system, but we want to focus on a particular behaviour. This can happens when we are interested in modelling just one particular pathway between many different interacting ones. In this case we might want to describe the reactions of the chosen pathway in detail abstracting away from the description of all the other pathways that may interact with it. The effects of the interaction between the chosen pathway and the others is modelled in reaction systems by

the environment. It is worth noting that this capability of abstracting away from less relevant interactions w.r.t. the selected behaviour it is always necessary when modelling biological systems since any biological behaviour cannot be faithfully studied in its whole complexity.

In the original formulation of reaction systems, environments could only supply elements at a given evolution step. In order to be able to more faithfully model the possible interactions between the selected system and its whole surrounding environment, in this paper we propose to consider more powerful context sequences for reaction systems. Such sequences not only supply objects but have also the capability to express the absorption of objects at a given evolution step.

Generalized context sequences are useful for easily modelling open systems. For example, in Section 6.1 we consider the well-known Epidermial Growth Factor (EGF) pathway. The EGF pathway is one of the pathways that is most frequently involved in cancer development. As a consequence, the EGF pathway is the target of several cancer therapies. Such therapies may consist in drugs, that are molecules that interfere with the reactions of a target pathway. Drugs are usually also involved in other reactions, in which they are decomposed and become inactive. In order to model the EGF pathway and the effect of drugs on it, we can use suitable context sequences to model the reactions of drug decomposition in an abstract way. In more detail, the effect of the drug and the interference between the drug decomposition pathway and the EGF pathway can be modelled by context sequences that *absorb* a certain object as long as the drug is present. This model allows us to observe the effect of drugs administration.

Moreover, the capability of the generalized context sequences to absorb objects can be usefully exploited to more faithfully model the effects that reactions executing in some compartment may have on some other compartment. Indeed, such effects may be, under some conditions, the removal of objects from the modelled compartment. These effects can be easily described by using an environment able to absorb an object at some steps. In Section 6.2 we use our generalized context sequences to model cell metabolism and in particular the respiration inside the cell. While the glycolysis pathway, that transforms glucose received from the environment into pyruvate is activated inside the cell, Glucose can enter the cell through the GLUT channel. GLUT is a transmembrane protein that is specialized into transportation of glucose molecules from the external environment into the cell. Similarly, lactate can enter and exit the cell through specific channels (proteins MCT1 and MCT4, respectively). Hence, in this case, to faithfully model the biological behaviour, the context sequences have to be able to model on the one hand the supply of glucose and lactate at some steps but, on the other hand, also the absorption of lactate at some other steps.

It is worth noting that the previous systems could also be modelled using reaction systems with the ordinary notion of context sequences, by adding more reactions and by controlling them using the mechanism of inhibition. However, this would make the reaction system very complex and difficult to understand and use.

In this paper we define a new syntax and semantics for reaction systems with generalized context sequences and we propose an encoding that relates reaction systems working in generalized contexts with reaction systems working in ordinary contexts. In particular, we show that the effects of absorption of objects realized by a generalized environment can be simulated by the reactions of the reaction system. However, this encoding could cause an exponential growth in the number of reactions.

The second part of the paper is dedicated to study dynamic causal dependences in reaction systems working in generalized contexts. Addressing causal relationships among the actions performed by a system is a very relevant issue in system biology (see e.g. [13,4,5]).

In [7] Brijder, Ehrenfeucht and Rozenberg initiate an investigation of *causalities* in re-action systems, i.e. the ways that entities of a reaction system influence each other. Both static/structural causalities as well as dynamic causalities are discussed, introducing the idea of *predictor*. Assume that one is interested in a particular object $s \in S$ and in know-ing if that object s will be present after n steps of execution of the reaction system. Since the only source of non-determinism are the contextual elements received at each step, knowing which objects will be received at each step can allow the creation of s after n steps to be predicted. The concept of predictor is based on the idea that, in general, not all contextual elements are relevant for determining if s will be produced after n steps. Indeed, for given s and n, there might be a subset Q of S which is the part of S that it is essential to observe in contextual elements for predicting whether s will be produced after n steps or not. Such set Q can then be used to concentrate uniquely on the relevant part of the contextual elements received from the external environment at each step, ignoring all elements that are surely not involved in the production of s in n steps. If two different sequence of contextual sets become equal after dropping elements not in Q, we can be sure that they both determine either the presence or the absence of s after n steps. In other words, Q is a subset of S which is a cause for s to be *uniformly* either present or absent after n steps. Brijder et al. define such Q the predictor of s in n steps, since knowing the behavior of the system with a sequence of contextual element, allows us to predict the behavior of such system with any other sequence having the same sets of relevant contextual elements.

Predictors can be useful to simplify the observation and evaluation of a biological sys-tem. In particular, they can be helpful to single out the part of the context (molecules to be provided to the system) that have to be observed in order to obtain the production of a molecule of interest.

In [2] we have introduced the new notion of *formula based predictor*. A formula based predictor consists in a propositional logic formula to be satisfied by the sequence of (sets of) elements provided by the environment. Satisfaction of the logic formula precisely discrim-inates the cases in which s will be produced after n steps from those in which it will not. We have shown that minimal formula based predictors exist for a given object s and n and that they can be effectively computed. In this setting a formula based predictor is minimal provided that it contains the smallest number of propositional symbols. Minimal formula based predictors are simpler to verify and do contain spurious causal dependencies.

Formula based predictors (as well as standard predictors) do not assume anything about the elements provided by the environment. However, often the sequences of sets of objects provided by the environment follow specific patterns or, more generally, have specific dy-namical properties. For example, some objects may never be provided by the environment, or some objects may be provided only after some others.

In [1,3] we revised formula based predictors by introducing specialized formula based predictors. The revised notion is specialized with respect to a temporal logic formula ex-pressing the dynamical properties of the environment. More specifically, a specialized for-mula based predictor is a propositional logical formula that predicts the production of an object s after n steps, by considering only the subset of the context sequences satisfying the given temporal logic formula. A specialized predictor can be a substantially simpler for-mula than the corresponding (non-specialized) formula based predictor even if we loose the property of existence of a unique minimal predictor.

In this paper we revise these concepts adapting them to reaction systems with general-ized contexts and investigating many of their properties such as the existence of minimal predictors.

The paper is organized as follows. Section 2 presents the syntax and the semantics of reaction system with generalized context sequences and studies, through the definition of proper encodings, the relations between generalized and standard reaction systems. Section 3 presents some formal concepts that are used in following definitions of predictors. The revised notion of formula based predictor for reaction systems with generalized contexts and all the related properties can be found in Section 4. Section 5 revises the notion of specialized predictor for reaction systems with generalized contexts and proves its properties. Section 6 presents some applications of our approach to different biological systems: in Section 6.1 generalized context sequences are used to model the interaction of the EGF pathway in case of drug administration, while in Section 6.2 generalized context sequences are used to faithfully model cell metabolism and respiration. Finally, our conclusions can be found in Section 7.

2 Reaction Systems with Generalized Contexts

In this section we revise the definition of the syntax and the semantics of reaction systems as introduced by Ehrenfeucht, Brijder and Rozenberg [11]. The idea is to define reaction systems able to deal with more powerful environments that are able to supply objects, as in the original model, and also to absorb (or remove) objects.

2.1 Syntax and semantics

Let S be a finite set of symbols, called *objects*. A *reaction* is formally a triple (R,I,P) with $R,I,P \subseteq S$, composed of *reactants R*, *inhibitors I*, and *products P*. Reactants and inhibitors are assumed to be disjoint ($R \cap I = \emptyset$), otherwise the reaction would never be applicable. The reactants and the inhibitors $R \cup I$ of a reaction are collectively called the *resources* of such a reaction. The set of all possible reactions over a set S is denoted by $\mathrm{rac}(S)$. Finally, a *reaction system* is a pair $\mathscr{A} = (S,A)$, with S being a finite set of objects, and $A \subseteq \mathrm{rac}(S)$ being a finite set of reactions.

The state of a reaction system (S,A) is described by a set of objects that is a subset of S. The dynamics of a reaction system is based on the "threshold supply" assumption which, in words, means that if an object is present in a state, then it is assumed to be available in any arbitrary number of copies. In fact, reaction systems can be seen as a *qualitative* formalism, rather than a *quantitative* one.

Let $a = (R_a, I_a, P_a)$ be a reaction and T a set of objects. The result $\mathrm{res}_a(T)$ of the application of a to T is either P_a, if T separates R_a from I_a (i.e. $R_a \subseteq T$ and $I_a \cap T = \emptyset$), or the empty set $\emptyset$ otherwise. Because of the threshold supply assumption, the application of multiple reactions at the same time occurs without any competition for the used reactants. Therefore, each reaction which is not inhibited can be applied, and the result of the application of multiple reactions is cumulative. Formally, given a reaction system $\mathscr{A} = (S,A)$, the result of application of $\mathscr{A}$ to a set $T \subseteq S$ is defined as $\mathrm{res}_{\mathscr{A}}(T) = \mathrm{res}_A(T) = \bigcup_{a \in A} \mathrm{res}_a(T)$.

The dynamics of a reaction system is driven by the interactions with the environment (or context). Our proposal is to assume that the environment can both supply and absorb objects. In order to have a non ambiguous semantics we only require that an environment does not supply and absorb the same object simultaneously. The objects supplied or absorbed by the environment at each step are called *contextual* objects.

An important characteristic of reaction systems, which distinguishes them from most other biologically-inspired computational models, is the assumption of the *non-permanency* of objects. Under such an assumption the objects carried over to the next step of the execution of the system are only those which are produced by reactions. All the other objects vanish, even if they are not "consumed" by any reaction, and are not included in the state reached by the system after executing the step.

Formally, we define the dynamics of a reaction system $\mathscr{A} = (S, A)$ working in a generalized context as a *generalized interactive process* $\pi = (\gamma, \delta)$, where γ is the *generalized context sequence* and δ is the *result sequence*. More specifically, the sequences are of the form $\gamma = (C_0, \overline{C}_0), (C_1, \overline{C}_1), \ldots, (C_n, \overline{C}_n)$ and $\delta = D_0, D_1, \ldots, D_n$ for some $n \geq 1$, with $C_i, \overline{C}_i, D_i \subseteq S$, $D_0 = \emptyset$ and $C_i \cap \overline{C}_i = \emptyset$, for each $0 \leq i \leq n$. In a generalized context sequence $\gamma = (C_0, \overline{C}_0), (C_1, \overline{C}_1), \ldots, (C_n, \overline{C}_n)$, the pair of sets of objects $(C_i, \overline{C}_i)$ indicates that the environment at the i-th computational step will supply the system with objects in C_i while it will absorb objects in $\overline{C}_i$. Thus, the condition $C_i \cap \overline{C}_i = \emptyset$ says that the same object cannot be supplied and absorbed at the same step. Each set D_i in the result sequence, for $i \geq 1$, is obtained from the application of reactions A to a state composed of both the results of the previous step D_{i-1} that are not absorbed by the context ($\overline{C}_{i-1}$), and the objects supplied by the context (C_{i-1}); formally $D_i = res_{\mathscr{A}}(C_{i-1} \cup (D_{i-1} \backslash \overline{C}_{i-1}))$ for all $1 \leq i \leq n$. Finally, the *state sequence* of π is defined as the sequence $W_0, W_1, \ldots, W_n$, where $W_i = C_i \cup (D_i \backslash \overline{C}_i)$ for all $1 \leq i \leq n$. In the following we say that $\gamma = (C_0, \overline{C}_0), (C_1, \overline{C}_1), \ldots, (C_n, \overline{C}_n)$ is a n-step generalized context sequence.

Example 1 Let $\mathscr{A} = (\{\texttt{A}, \texttt{B}, \texttt{C}, \texttt{D}, \texttt{E}, \texttt{F}, \texttt{G}\}, \{a_1, a_2, a_3, a_4, a_5\})$ be a *reaction system* with reaction rules

$$a_1 = (\{\texttt{A}, \texttt{B}\}, \{\texttt{C}\}, \{\texttt{D}\}) \quad a_2 = (\{\texttt{E}\}, \{\}, \{\texttt{B}\}) \quad a_3 = (\{\texttt{F}\}, \{\}, \{\texttt{B}\})$$
$$a_4 = (\{\texttt{E}\}, \{\}, \{\texttt{A}\}) \quad\quad a_5 = (\{\texttt{G}\}, \{\}, \{\texttt{C}, \texttt{A}\}).$$

In order to illustrate the behaviour of the reaction system we introduce two generalized context sequences:

$$\gamma_1 = (\{\texttt{E}\}, \emptyset), (\{\texttt{G}, \texttt{F}\}, \emptyset), (\emptyset, \emptyset), (\{\texttt{G}, \texttt{F}\}, \emptyset), (\emptyset, \{\texttt{C}\}), (\emptyset, \emptyset)$$
$$\gamma_2 = (\{\texttt{E}\}, \emptyset), (\{\texttt{G}, \texttt{F}\}, \{\texttt{A}\}), (\emptyset, \{\texttt{C}\}), (\{\texttt{G}, \texttt{E}\}, \emptyset), (\emptyset, \emptyset), (\emptyset, \emptyset)$$

The generalized context sequences γ_1 and γ_2 and the corresponding result sequences δ_1 and δ_2 are depicted in Table 1 (for simplicity we use $\bar{s}$ to indicate that the environment absorbs object s at the corresponding step). Note that in generalized reaction systems the effect of absorption of an object s may *enable* and *disable* reactions. More specifically, a reaction having s as an inhibitor may be enabled while a reaction having s as a reactant may be disabled. For example, considering the fifth step of the generalized context sequence γ_1 the reaction a_1 is enabled as a consequence of the absorption of the object $\texttt{C}$, given that $\texttt{C}$ is an inhibitor. By contrast, considering the second step of generalized context sequence γ_2, the reaction a_1 is disabled because of the absorption of the object $\texttt{A}$ that is a reactant of a_1.

2.2 Relations between reaction systems working in generalized and ordinary contexts

The dynamics of a reaction system working in an ordinary context can be obtained also by executing the same reaction system with the corresponding generalized context sequence where the second component (representing absorbed objects) is empty at all steps. Also the dynamics of a reaction system working in a generalized context can be related with the

	0	1	2	3	4	5
γ_1	E	G F		G F	$\overline{C}$	
δ_1		A B	D C A B		C A B	D

	0	1	2	3	4	5
γ_2	E	G F $\overline{A}$	$\overline{C}$	E G		
δ_2		A B	C A B	D	C A B	

Table 1 The dynamics of the generalized reaction system $\mathscr{A}$ obtained from two examples of context sequence γ_1 and γ_2.

dynamics of a corresponding reaction system working in an ordinary context, but in this case the latter system has to be obtained through an encoding.

In order to formally relate reaction systems working in generalized and ordinary contexts we recall the standard semantics of reaction systems. The dynamics of a reaction system $\mathscr{A} = (S,A)$ working in an ordinary context is defined as an *ordinary interactive process* $\tilde{\pi} = (\tilde{\gamma}, \tilde{\delta})$, with $\tilde{\gamma}$ and $\tilde{\delta}$ being finite sequences of sets of objects, called the *context sequence* and the *result sequence*, respectively. The sequences are of the form $\tilde{\gamma} = \tilde{C}_0, \tilde{C}_1, \ldots, \tilde{C}_n$ and $\tilde{\delta} = \tilde{D}_0, \tilde{D}_1, \ldots, \tilde{D}_n$ for some $n \geq 1$, with $\tilde{C}_i, \tilde{D}_i \subseteq S$, and $\tilde{D}_0 = \emptyset$. Each set $\tilde{D}_i$, for $i \geq 1$, in the result sequence is obtained from the application of reactions A to a state composed of both the results of the previous step $\tilde{D}_{i-1}$ and the objects $\tilde{C}_{i-1}$ from the context; formally $\tilde{D}_i = res_{\mathscr{A}}(\tilde{C}_{i-1} \cup \tilde{D}_{i-1})$ for all $1 \leq i \leq n$. Finally, the *state sequence* of $\tilde{\pi}$ is defined accordingly.

The following theorem shows that the dynamics of any reaction system working in an ordinary context is included in the dynamics of the same reaction system in a generalized context.

Theorem 1 *Let $\mathscr{A} = (S,A)$ be a reaction system and $\tilde{\delta} = \tilde{D}_0, \tilde{D}_1, \ldots, \tilde{D}_n$ be the result sequence corresponding to the ordinary context sequence $\tilde{\gamma} = \tilde{C}_0, \tilde{C}_1, \ldots, \tilde{C}_n$. We have that $\tilde{\delta} = \delta$, where $\delta = D_0, D_1, \ldots, D_n$ is the result sequence of the same reaction system $\mathscr{A} = (S,A)$, obtained by considering the generalized context sequence $\gamma = (\tilde{C}_0, \emptyset), (\tilde{C}_1, \emptyset), \ldots, (\tilde{C}_n, \emptyset)$.*

Proof It is easy to show that $\tilde{D}_i = D_i$ for each $i \in \{1, \ldots, n\}$, by induction on i. We omit the details of the proof that are trivial. □

On the other hand, we can also establish a simulation between reaction systems working in generalized and ordinary contexts. This correspondence is more difficult to show given that ordinary contexts cannot absorb objects. Nevertheless, we can define an *encoding* that transforms a reaction systems into another one able to simulate in an ordinary context the dynamics of former in a generalized context. This means that for any generalized context sequence there exists a corresponding standard context sequence such that both systems exhibit the same behaviour.

The definition of the encoding requires the introduction of new objects and a transformation of the reaction rules. Since in ordinary contexts the environment cannot absorb objects, such a feature has to be simulated. The effect of absorption of an object $s \in S$ is twofold: on the one hand, it may disable some reactions; on the other hand, it may enable some other reactions. In particular, due to the removal of object s, all the reactions having s as a reactant are disabled, while the reactions having s as an inhibitor may be enabled. In order to model the absorption of an object s we introduce a new object X_s representing the removal of s realized by the environment. The new symbol X_s is added as an inhibitor to all reactions having s as a reactant. In this way, these reaction are disabled when the environment supplies

X_s, that simulates absorption of s. Moreover, as regards reactions having s as inhibitor, they have to be handled by introducing a set of new reactions dealing with the case in which the environment supplies the symbol X_s. In particular, for each reaction having s as inhibitor it is necessary to add a new reaction in which s is removed from the set of inhibitors and X_s is added among reactants. It is worth noting that if the reaction has more than one inhibitor, we have to add new reactions representing all the possible ways for the inhibitors to be absent in the state (that is, either they are actually not present, or they are absorbed).

In the following, given a finite a set of objects S we define $\mathscr{E}(S) = S \cup \{X_s \mid s \in S\}$, where we assume $X_s \notin S$ and for all $s' \in S$ s.t. $s' \neq s$ it holds $X_{s'} \neq X_s$. The same properties of X_s will be assumed to hold in any further definition that uses the X_s notation. We can now introduce the definition of encoding of a reaction that, given a reaction on S, gives a set of reactions on the set of objects $\mathscr{E}(S)$.

Definition 1 Let $a = (R, I, P)$ be a reaction with $R, I, P \subseteq S$ for a finite set S. We define the *encoding* of a as follows:

$$\mathscr{E}(a) = \bigcup_{I' \subseteq I} \{(R \cup \{X_s \mid s \in I'\}, (I \backslash I') \cup \{X_s \mid s \in R\}, P)\}.$$

The *encoding of a reaction system* $\mathscr{A} = (S, A)$ can be obtained from the set of objects $\mathscr{E}(S)$ and the encoding of reactions as follows:

$$\mathscr{E}(\mathscr{A}) = (\mathscr{E}(S), \mathscr{E}(A))$$

where $\mathscr{E}(A) = \bigcup_{a \in A} \mathscr{E}(a)$.

The following theorem formalizes the main property relating the behaviour of a reaction system working in a generalized context with the behaviour of its encoding working in an ordinary context. The property shows that the reaction system obtained by the encoding is able to simulate the dynamics of the encoded system for any generalized context sequence. That is, for any generalized context sequence there exists a corresponding context sequence producing the same result sequence. Given a generalized context sequence $\gamma = (C_0, \overline{C}_0), (C_1, \overline{C}_1), \dots, (C_n, \overline{C}_n)$ we define the translation of γ, denoted by $\mathscr{E}(\gamma)$, as:

$$\mathscr{E}(\gamma) = \tilde{C}_0, \tilde{C}_1, \dots, \tilde{C}_n$$

where $\tilde{C}_i = C_i \cup \{X_s \mid s \in \overline{C}_i\}$, for each $i \in \{1, \dots, n\}$. As expected $\mathscr{E}(\gamma)$ is a context sequence that at the i-th step contains the symbol X_s whenever s is absorbed in the corresponding generalized context sequence γ at the same step. Hence, at each step the absorption of s is rendered by the new symbol X_s.

Theorem 2 *Let $\mathscr{A} = (S, A)$ be a reaction system and $\delta = D_0, D_1, \dots, D_n$ be the result sequence given by the generalized context sequence $\gamma = (C_0, \overline{C}_0), (C_1, \overline{C}_1), \dots, (C_n, \overline{C}_n)$. We have that $\delta = \tilde{\delta}$ where $\tilde{\delta} = \tilde{D}_0, \tilde{D}_1, \dots, \tilde{D}_n$ is the result sequence of the reaction system $\mathscr{E}(\mathscr{A})$ corresponding to the context sequence $\mathscr{E}(\gamma) = \tilde{C}_0, \tilde{C}_1, \dots, \tilde{C}_n$.*

Proof We have to prove that $\tilde{D}_i = D_i$ for each $i \in \{1, \dots, n\}$. The proof is by induction on i:

 – $i = 0$. In this case, by definition, $\tilde{D}_0 = D_0 = \emptyset$.

– $i > 0$. In this case we apply the definition of the semantics of reaction system $\mathscr{A}$ and of its encoding $\mathscr{E}(\mathscr{A})$. For $\mathscr{A}$ we have,

$$D_i = res_{\mathscr{A}}(C_{i-1} \cup (D_{i-1} \backslash \overline{C}_{i-1})) = res_A(C_{i-1} \cup (D_{i-1} \backslash \overline{C}_{i-1})).$$

For $\mathscr{E}(\mathscr{A})$ we have

$$\tilde{D}_i = res_{\mathscr{E}(\mathscr{A})}(\tilde{C}_{i-1} \cup \tilde{D}_{i-1}) = res_{\mathscr{E}(A)}(\tilde{C}_{i-1} \cup \tilde{D}_{i-1})$$

where $\mathscr{E}(A) = \bigcup_{a \in A} \mathscr{E}(a)$ by definition of the encoding.
By definition of $\mathscr{E}(\gamma)$ we have $\tilde{C}_{i-1} = C_{i-1} \cup \{X_s \mid s \in \overline{C}_{i-1}\}$. Therefore,

$$\tilde{D}_i = res_{\mathscr{E}(A)}(\tilde{C}_{i-1} \cup \tilde{D}_{i-1}) = res_{\mathscr{E}(A)}(C_{i-1} \cup \{X_s \mid s \in \overline{C}_{i-1}\} \cup \tilde{D}_{i-1})$$

By applying the induction hypothesis ($\tilde{D}_{i-1} = D_{i-1}$), we are left to prove that

$$res_A(C_{i-1} \cup (D_{i-1} \setminus \overline{C}_{i-1})) = res_{\mathscr{E}(A)}((C_{i-1} \cup \{X_s \mid s \in \overline{C}_{i-1}\}) \cup D_{i-1}) \qquad (1)$$

Note that by Definition 1 any reaction rule $\tilde{a}$ such that $\tilde{a} \in \mathscr{E}(a)$ produces the same products as reaction a. Hence, in order to prove (1) it is enough to show that for any reaction $a \in A$ which is enabled in $C_{i-1} \cup (D_{i-1} \backslash \overline{C}_{i-1})$ there exists a corresponding reaction $\tilde{a} \in \mathscr{E}(a)$ (and therefore $\tilde{a} \in \mathscr{E}(A)$) which is enabled in $\tilde{C}_{i-1} \cup D_{i-1}$ (and viceversa).

– We first prove the right implication. Let us consider $a \in A$ with $a = (R, I, P)$. By definition of the semantics of reaction systems, a is enabled in $C_{i-1} \cup (D_{i-1} \backslash \overline{C}_{i-1})$ iff $R \subseteq C_{i-1} \cup (D_{i-1} \backslash \overline{C}_{i-1})$ and $I \cap (C_{i-1} \cup (D_{i-1} \backslash \overline{C}_{i-1})) = \emptyset$. If the reaction a does not have inhibitors (that is $I = \emptyset$) we have that $\mathscr{E}(a) = \{\tilde{a}\}$ with $\tilde{a} = (R, \{X_s \mid s \in R\}, P)$. In this case we have that $R \subseteq C_{i-1} \cup (D_{i-1} \backslash \overline{C}_{i-1})$ implies $R \subseteq C_{i-1} \cup \{X_s \mid s \in \overline{C}_{i-1}\} \cup D_{i-1}$. Hence, $\tilde{a}$ is enabled since $\{X_s \mid s \in R\} \cap (C_{i-1} \cup \{X_s \mid s \in \overline{C}_{i-1}\} \cup D_{i-1}) = \emptyset$. We observe that for any reactant $s \in R$ we have that $s \in C_{i-1} \cup (D_{i-1} \backslash \overline{C}_{i-1})$ and that $C_{i-1} \cap \overline{C}_{i-1} = \emptyset$. As a consequence, it must be the case that $s \notin \overline{C}_{i-1}$ and therefore $\{X_s \mid s \in R\} \cap (\tilde{C}_{i-1} \cup D_{i-1}) = \emptyset$.
The reasoning can be generalized by considering a reaction $a = (R, I, P)$ with $I \neq \emptyset$. In this case, by Definition 1 we have

$$\mathscr{E}(a) = \bigcup_{I' \subseteq I} \{(R \cup \{X_{s_i} \mid s_i \in I'\}, (I \backslash I') \cup \{X_{s_r} \mid s_r \in R\}, P)\}.$$

Given that a is enabled, $R \subseteq C_{i-1} \cup (D_{i-1} \backslash \overline{C}_{i-1})$ and $I \cap (C_{i-1} \cup (D_{i-1} \backslash \overline{C}_{i-1})) = \emptyset$. Similarly as in the previous case we prove that there exists a reaction $\tilde{a} \in \mathscr{E}(a)$ that is enabled in $C_{i-1} \cup \{X_s \mid s \in \overline{C}_{i-1}\} \cup D_{i-1}$. Given that $I \cap (C_{i-1} \cup (D_{i-1} \backslash \overline{C}_{i-1})) = \emptyset$ we have also $I \cap C_{i-1} = \emptyset$ and $I \cap (D_{i-1} \backslash \overline{C}_{i-1}) = \emptyset$. In this case we consider a subset of inhibitors $I' \subseteq I$ such that $I' = D_{i-1} \cap \overline{C}_{i-1}$. Note that $(I \backslash I') \cap C_{i-1} = \emptyset$ and $(I \backslash I') \cap D_{i-1} = \emptyset$ hold.
Then we consider the reaction $\tilde{a}$ corresponding to I' that is $\tilde{a} = (R \cup \{X_{s_i} \mid s_i \in I'\}, (I \backslash I') \cup \{X_{s_r} \mid s_r \in R\}, P)$. The reaction $\tilde{a}$ is enabled iff (i) $R \cup \{X_{s_i} \mid s_i \in I'\} \subseteq (C_{i-1} \cup \{X_s \mid s \in \overline{C}_{i-1}\}) \cup D_{i-1}$ and (ii) $((I \backslash I') \cup \{X_{s_r} \mid s_r \in R\}) \cap ((C_{i-1} \cup \{X_s \mid s \in \overline{C}_{i-1}\}) \cup D_{i-1}) = \emptyset$ hold.
We first examine (i). We have that $I' \subseteq D_{i-1} \cap \overline{C}_{i-1}$ and therefore $I' \subseteq \overline{C}_{i-1}$. As a consequence, for any inhibitor $s_i \in I'$ we have that $X_{s_i} \in \{X_s \mid s \in \overline{C}_{i-1}\}$. Hence, we can derive that $\{X_{s_i} \mid s_i \in I'\} \subseteq \{X_s \mid s \in \overline{C}_{i-1}\}$ holds. Moreover, analogously as in the previous case we have also $R \subseteq D_{i-1} \cap C_{i-1}$.

We then prove (ii). Note that similarly as in the previous case we have $\{X_s \mid s \in R\} \cap (C_{i-1} \cup \{X_s \mid s \in \overline{C}_{i-1}\} \cup D_{i-1}) = \emptyset$. We have that $(I\backslash I') \cap C_{i-1} = \emptyset$ and $(I \backslash I') \cap D_{i-1} = \emptyset$ and therefore it remains to show that $(I\backslash I') \cap \{X_s \mid s \in \overline{C}_{i-1}\} = \emptyset$. For this it is enough to observe that $I \subseteq S$ and therefore $I \cap \{X_s \mid s \in \overline{C}_{i-1}\} = \emptyset$.

– We prove the left implication. Thus, we show that for a reaction $\tilde{a} \in \mathscr{E}(A)$ such that $\tilde{a}$ is enabled in $\tilde{C}_{i-1} \cup D_{i-1}$ there exists a reaction $a \in A$ which is enabled in $C_{i-1} \cup (D_{i-1}\backslash\overline{C}_{i-1})$. Note that by definition of the encoding for any $\tilde{a}$ such that $\tilde{a} \in \mathscr{E}(A)$ there exists $a \in A$ such that $\tilde{a} = \mathscr{E}(a)$. Similarly as in the previous case we prove that if $\tilde{a}$ is enabled then also a is enabled. By Definition 1 we have

$$\mathscr{E}(a) = \bigcup_{I' \subseteq I} \{(R \cup \{X_{s_i} \mid s_i \in I'\}, (I\backslash I') \cup \{X_{s_r} \mid s_r \in R\}, P)\}.$$

where $a = (R, I, P)$ is a reaction with $a \in A$. Hence, it must be the case that $\tilde{a} = (R \cup \{X_{s_i} \mid s_i \in I'\}, (I\backslash I') \cup \{X_{s_r} \mid s_r \in R\}, P)$ for some subset of inhibitors $I' \subseteq I$.
By definition of the standard semantics of reaction systems, $\tilde{a}$ is enabled in $\tilde{C}_{i-1} \cup D_{i-1}$ iff $(R \cup \{X_{s_i} \mid s_i \in I'\}) \subseteq \tilde{C}_{i-1} \cup D_{i-1}$ and $((I\backslash I') \cup \{X_{s_r} \mid s_r \in R\}) \cap (\tilde{C}_{i-1} \cup D_{i-1}) = \emptyset$.
The reaction a is enabled in $C_{i-1} \cup (D_{i-1}\backslash\overline{C}_{i-1})$ iff (i) $R \subseteq C_{i-1} \cup (D_{i-1}\backslash\overline{C}_{i-1})$ and (ii) $I \cap (C_{i-1} \cup (D_{i-1}\backslash\overline{C}_{i-1})) = \emptyset$.
We first examine (i). We have that $R \subseteq \tilde{C}_{i-1} \cup D_{i-1}$ where $\tilde{C}_{i-1} = C_{i-1} \cup \{X_s \mid s \in \overline{C}_{i-1}\}$). Moreover, $R \cap \{X_s \mid s \in \overline{C}_{i-1}\} = \emptyset$ given that $R \subseteq S$ and $S \cap \{X_s \mid s \in \overline{C}_{i-1}\} = \emptyset$. Thus, we obtain that $R \subseteq C_{i-1} \cup D_{i-1}$. We now show that for each reactant $s_r \in R$ it is not the case that $s_r \in \overline{C}_{i-1}$. Let us assume that $s_r \in \overline{C}_{i-1}$ then by definition $X_{s_r} \in \tilde{C}_{i-1}$. Moreover, by definition X_{s_r} is an inhibitor of the reaction $\tilde{a}$ and we have $((I\backslash I') \cup \{X_{s_r} \mid s_r \in R\}) \cap (\tilde{C}_{i-1} \cup D_{i-1}) = \emptyset$. Hence, we would obtain that $X_{s_r} \in ((I\backslash I') \cup \{X_{s_r} \mid s_r \in R\}) \cap (\tilde{C}_{i-1} \cup D_{i-1})$ which is a contradiction. Thus we have $R \subseteq C_{i-1} \cup D_{i-1}$ and $R \cap \overline{C}_{i-1} = \emptyset$ and therefore we can derive $R \subseteq C_{i-1} \cup (D_{i-1}\backslash\overline{C}_{i-1})$.
We now examine (ii). We have that $((I\backslash I') \cup \{X_{s_r} \mid s_r \in R\}) \cap (\tilde{C}_{i-1} \cup D_{i-1}) = \emptyset$ and therefore $(I\backslash I') \cap (\tilde{C}_{i-1} \cup D_{i-1}) = \emptyset$ where $\tilde{C}_{i-1} = C_{i-1} \cup \{X_s \mid s \in \overline{C}_{i-1}\}$. Therefore we derive $(I\backslash I') \cap (C_{i-1} \cup D_{i-1}) = \emptyset$ and also $(I\backslash I') \cap (C_{i-1} \cup (D_{i-1}\backslash\overline{C}_{i-1})) = \emptyset$.
It remains to show that $I' \cap (C_{i-1} \cup (D_{i-1}\backslash\overline{C}_{i-1})) = \emptyset$. We observe that for each $s_i \in I'$, by definition of the encoding, X_{s_i} is a reactant of reaction $\tilde{a}$. In this case we have that $\{X_{s_i} \mid s_i \in I'\} \subseteq (C_{i-1} \cup \{X_s \mid s \in \overline{C}_{i-1}\}) \cup D_{i-1}$ given that $\tilde{a}$ is enabled. Moreover, we have that $\{X_{s_i} \mid s_i \in I'\} \cap C_{i-1} = \emptyset$ and $\{X_{s_i} \mid s_i \in I'\} \cap D_{i-1} = \emptyset$ hold given that $C_{i-1}, D_{i-1} \subseteq S$. Thus, it must be the case that $\{X_{s_i} \mid s_i \in I'\} \subseteq \{X_s \mid s \in \overline{C}_{i-1}\}$. Hence, if $s_i \in I'$ then $X_{s_i} \in \{X_s \mid s \in \overline{C}_{i-1}\}$ and by definition $s_i \in \overline{C}_{i-1}$. Finally we can conclude that $I' \cap (C_{i-1} \cup (D_{i-1}\backslash\overline{C}_{i-1})) = \emptyset$ since $s_i \in \overline{C}_{i-1}$ for each $s_i \in I'$ and $C_{i-1} \cap \overline{C}_{i-1} = \emptyset$. $\qquad\square$

The previous result shows that the effects of absorption of objects realized by the environment can be simulated by reaction systems with standard context sequences. However, the encoding into reaction system with standard context sequences could cause an exponential growth in the number of reactions. In fact, according to Definition 1 the encoding of a reaction having a set of inhibitors I requires to introduce a new reaction for each subset of I.

The following example illustrates the application of the encoding to the reaction system of Example 1.

Example 2 We consider the reaction system $\mathscr{A} = (\{\texttt{A}, \texttt{B}, \texttt{C}, \texttt{D}, \texttt{E}, \texttt{F}, \texttt{G}\}, \{a_1, a_2, a_3, a_4, a_5\})$ of Example 1 with reactions

$$a_1 = (\{\mathtt{A},\mathtt{B}\},\{\mathtt{C}\},\{\mathtt{D}\}) \quad a_2 = (\{\mathtt{E}\},\{\},\{\mathtt{B}\}) \qquad a_3 = (\{\mathtt{F}\},\{\},\{\mathtt{B}\})$$
$$a_4 = (\{\mathtt{E}\},\{\},\{\mathtt{A}\}) \qquad a_5 = (\{\mathtt{G}\},\{\},\{\mathtt{C},\mathtt{A}\}).$$

By applying the encoding we obtain the following reaction system

$$\mathscr{E}(\mathscr{A}) = (\mathscr{E}(\{\mathtt{A},\mathtt{B},\mathtt{C},\mathtt{D},\mathtt{E},\mathtt{F},\mathtt{G}\}),\{\tilde{a}_{1,1},\tilde{a}_{1,2},\tilde{a}_2,\tilde{a}_3,\tilde{a}_4,\tilde{a}_5\})$$

where

$$\tilde{a}_{1,1} = (\{\mathtt{A},\mathtt{B}\},\{X_{\mathtt{A}},X_{\mathtt{B}},\mathtt{C}\},\{\mathtt{D}\}) \quad \tilde{a}_{1,2} = (\{\mathtt{A},\mathtt{B},X_{\mathtt{C}}\},\{X_{\mathtt{A}},X_{\mathtt{B}}\},\{\mathtt{D}\})$$
$$\tilde{a}_2 = (\{\mathtt{E}\},\{X_{\mathtt{E}}\},\{\mathtt{B}\}) \qquad\qquad \tilde{a}_3 = (\{\mathtt{F}\},\{X_{\mathtt{F}}\},\{\mathtt{B}\})$$
$$\tilde{a}_4 = (\{\mathtt{E}\},\{X_{\mathtt{E}}\},\{\mathtt{A}\}) \qquad\qquad \tilde{a}_5 = (\{\mathtt{G}\},\{X_{\mathtt{G}}\}\{\mathtt{C},\mathtt{A}\})$$

Note that for any reaction a_i with $i \in \{2,\dots 5\}$ we have $\mathscr{E}(a_i) = \{\tilde{a}_i\}$ given that there are no inhibitors that have to be simulated. Instead, we have $\mathscr{E}(a_2) = \{\tilde{a}_{1,1},\tilde{a}_{1,2}\}$ given that the reaction a_2 has the object C as inhibitor. In particular, the reaction $\tilde{a}_{1,1}$ has C as inhibitor and thus captures the case in which C is not present while the reaction $\tilde{a}_{1,2}$ has X_C as reactant and thus captures the case in which C is absorbed by the environment.

To illustrate the encoding it is convenient to describe the dynamics of reaction system $\mathscr{E}(\mathscr{A})$ with respect to the context sequences corresponding to the two generalized context sequences γ_1 and γ_2, considered in Example 1 (see Table 1). By translating the generalized context sequences γ_1 and γ_2 we obtain:

$$\mathscr{E}(\gamma_1) = \tilde{\gamma}_1 = \{\mathtt{E}\},\{\mathtt{G},\mathtt{F}\},\emptyset,\{\mathtt{G},\mathtt{F}\},\{X_{\mathtt{C}}\},\emptyset$$
$$\mathscr{E}(\gamma_2) = \tilde{\gamma}_2 = \{\mathtt{E}\},\{\mathtt{G},\mathtt{F},X_{\mathtt{A}}\},\{X_{\mathtt{C}}\},\{\mathtt{G},\mathtt{E}\},\emptyset,\emptyset$$

The context sequences $\tilde{\gamma}_1$ and $\tilde{\gamma}_2$ and the corresponding result sequences $\tilde{\delta}_1$ and $\tilde{\delta}_2$ are depicted in Table 2. In both cases, the result sequences $\tilde{\delta}_1$ and $\tilde{\delta}_2$ coincide with the result sequences δ_1 and δ_2 depicted in Table 1. The reaction system $\mathscr{E}(\mathscr{A})$ exhibits the same behaviour of $\mathscr{A}$ since the generalized context sequences γ_1 and γ_2 are translated accordingly. In the encoding of the context sequences, $X_{\mathtt{A}}$, $X_{\mathtt{B}}$ and $X_{\mathtt{C}}$ represents the absorption of objects A, B and C in the corresponding generalized context sequences, respectively. For instance, let us consider the context sequence $\tilde{\gamma}_1$ and the reactions $\tilde{a}_{1,1}$ and $\tilde{a}_{1,2}$ having the product D. At the second step of the computation, reaction $\tilde{a}_{1,1}$ is applied because both reactants A and B are present while the inhibithor C is absent. Instead, at the fifth step the $\tilde{a}_{1,2}$ is applied because the reactants A, B and $X_{\mathtt{C}}$ are present. The presence of object $X_{\mathtt{C}}$ models the absorption of the object C in the corresponding generalized context sequence γ_1, at the same step.

Moreover, let us consider the second step of the context sequence $\tilde{\gamma}_2$. In this case, both reactions $\tilde{a}_{1,1}$ and $\tilde{a}_{1,2}$ are disabled beacuse the object $X_{\mathtt{A}}$ is present. In fact, both reactions have $X_{\mathtt{A}}$ as inhibitor. The presence of the object $X_{\mathtt{A}}$ models the absorption of the object A in the corresponding generalized context sequence γ_2, at the same step, which causes the corresponding reaction a_1 to be disabled.

3 Preliminaries

We start by defining some notions that will be used in the definitions of predictor we will give. We adopt propositional logic formulas parametric with respect to a set of propositional symbols $\mathscr{P}$.

Definition 2 (Logic Formulas) Let $\mathscr{P}$ be a set of propositional symbols. The set $F_{\mathscr{P}}$ of *propositional logic formulas* on $\mathscr{P}$ is inductively defined as follows:

	0	1	2	3	4	5
$\tilde{\gamma}_1$	E	G F		G F	X_C	
$\tilde{\delta}_1$		A B	D C A B		C A B	D

	0	1	2	3	4	5
$\tilde{\gamma}_2$	E	G F X_A	X_C	E G		
$\tilde{\delta}_2$		A B	C A B	D	C A B	

Table 2 The dynamics of the reaction system $\mathscr{E}(\mathscr{A})$ obtained from context sequences $\tilde{\gamma}_1$ and $\tilde{\gamma}_2$.

- $true, false \in F_{\mathscr{P}}$,
- if $p \in \mathscr{P}$ then $p \in F_{\mathscr{P}}$,
- if $f \in F_{\mathscr{P}}$ then $\neg f, (f) \in F_{\mathscr{P}}$,
- if $f_1, f_2 \in F_{\mathscr{P}}$ then $f_1 \vee f_2, f_1 \wedge f_2 \in F_{\mathscr{P}}$.

We assume the standard precedence rules for the logical operators. The propositional formulas on $\mathscr{P}$ are interpreted with respect to an *interpretation* on $\mathscr{P}$, defined in as $I : \mathscr{P} \to \{\mathbf{T}, \mathbf{F}\}$. The complete definition of the satisfaction is as follows.

Definition 3 Let $f \in F_{\mathscr{P}}$ be a propositional formula and $I : \mathscr{P} \to \{\mathbf{T}, \mathbf{F}\}$ be an interpretation. The *satisfaction relation* $I \models f$ is the smallest relation inductively defined as follows:

$$I \models p \text{ iff } I(p) = \mathbf{T}, \qquad I \models true,$$
$$I \models \neg f' \text{ iff } I \not\models f', \qquad I \models (f') \text{ iff } I \models f'$$
$$I \models f_1 \wedge f_2 \text{ iff } I \models f_1 \text{ and } I \models f_2, \quad I \models f_1 \vee f_2 \text{ iff either } I \models f_1 \text{ or } I \models f_2.$$

Given a set of propositional symbols $\mathscr{P}$ and a formula $f \in F_{\mathscr{P}}$ the truth value of the formula f with respect to an interpretation I on $\mathscr{P}$ can be denoted by $I(f)$. In the following $\equiv$ stands for the logical equivalence on propositional formulas $F_{\mathscr{P}}$. Moreover, given a formula $f \in F_{\mathscr{P}}$ we use $atom(f)$ to denote the set of propositional symbols that appear in f.

We introduce a more general notion of *equivalence* on propositional formulas which is parametric with respect to an hypothesis.

Definition 4 (Equivalence with hypothesis) Let $f_1, f_2 \in F_{\mathscr{P}}$ be two propositional formulas. Given a formula $h \in F_{\mathscr{P}}$ we say that f_1 and f_2 are *equivalent under the hypothesis h*, denoted by $f_1 \equiv_h f_2$, iff

$$h \implies f_1 \equiv f_2.$$

It should be clear that the notion of equivalence under the hypothesis is weaker than standard logical equivalence: if $f_1 \equiv f_2$ then it is also the case that $f_1 \equiv_h f_2$ for any hypothesis h (but not necessarily vice-versa).

Given a formula $f \in F_{\mathscr{P}}$ and an hypothesis $h \in F_{\mathscr{P}}$ we are interested in the properties of the equivalence class of f under the hypothesis h. More specifically, we are interested in formulas, equivalent to f under the hypothesis h, that contain the minimal number of propositional symbols. A formula which contains a smaller number of propositional symbols is easier to be verified. In order to compare two formulas equivalent under an hypothesis h we define an *approximation order* on formulas (denoted by $\sqsubseteq_h$).

Definition 5 (Order on formulas) Let $f_1, f_2 \in F_{\mathscr{P}}$ be two propositional formulas. Given an hypothesis $h \in F_{\mathscr{P}}$ we say that

- $f_1 \sqsubseteq_h f_2$ iff $f_1 \equiv_h f_2$ and $atom(f_1) \subseteq atom(f_2)$;

– $f_1 \sim_h f_2$ iff $f_1 \sqsubseteq_h f_2$ and $f_2 \sqsubseteq_h f_1$.

Note that $f_1 \sim_h f_2$ holds iff f_1 and f_2 are equivalent under the hypothesis h and they contain the same propositional symbols (that is $f_1 \equiv_h f_2$ and $atom(f_1) = atom(f_2)$). Moreover, we use $f_1 \sqsubset_h f_2$ iff $f_1 \equiv_h f_2$ and $atom(f_1) \subset atom(f_2)$.

Based on the previous definitions we also introduce the notion of minimal formula in the equivalence class of a given formula f under an hypothesis h.

Definition 6 Let $f, f' \in F_{\mathscr{P}}$ be a propositional formula. Given an hypothesis $h \in F_{\mathscr{P}}$ we say that f' is a *minimal form* of f *under the hypothesis* h iff: (i) $f' \sqsubseteq_h f$ and (ii) for any formula $f'' \in F_{\mathscr{P}}$ such that $f'' \not\sim_h f'$ and $f'' \sqsubseteq_h f$, we have that $f'' \not\sqsubseteq_h f'$.

In the following, given a formula $f \in F_{\mathscr{P}}$ and an hypothesis $h \in F_{\mathscr{P}}$, $min_h(f)$ stands for the set of formulas that are *minimal forms* of f under the hypothesis h. It is worth mentioning that in general there is no *unique* $\sim_h$-class of minimal forms of a formula f under an hypothesis h. Hence, they might exist two different formulas f_1 and f_2 which are minimal forms of f under the hypothesis h such that $f_1 \not\sim_h f_2$. This means that $f \equiv_h f_1 \equiv_h f_2$ and neither $atom(f_1) \subseteq atom(f_2)$ nor $atom(f_2) \subseteq atom(f_1)$ holds.

Furthermore, the problem of computing the minimal forms of a formula f under an hypothesis h can be reduced to the minimal input support problem [10] on *incompletely specified boolean functions*. Given an incompletely specified boolean function, the goal is to find a completely specified formula, in sum of product form (SOP), which has a minimal numbers of different variables and which is equivalent (in the values of interest) to the incompletely specified one.

Hence, the first step is to define the *incompletely specified boolean function* representing formula f under the assumption that h holds. Let m be the cardinality of the set $\mathscr{P}$ of propositional symbols and $B = \{true, false\}$. By assuming a linear ordering on $\mathscr{P}$, we represent an interpretation I on $\mathscr{P}$ as a tuple $(x_1, \ldots, x_m)$ where x_i is the interpretation of the i-th propositional symbol of $\mathscr{P}$ according to I.

We consider the *incompletely specified boolean function* $F : B^m \to B$ composed of the following two disjoint subsets Yes_F and $DontC_F$ of B^m where:

– $Yes_F = \{(x_1, .., x_m) \mid (x_1, .., x_m)$ is a boolean assignment that satisfies $h \wedge f\}$ is the subset of B^m where F is evaluated to 1;
– $DontC_F = \{(x_1, .., x_n) \mid (x_1, .., x_n)$ is a boolean assignment that satisfies $\neg h\}$ is a subset of B^m where the value of F is not specified.

In each point of $DontC_F$, called *don't care*, F can assume the value *true* or *false*. So there are $k = 2^{|DontC_F|}$ different functions *equivalent* to F. Consider any possible extension of F, let us call them $F^1, \ldots, F^k$, obtained by considering $Yes_{Fi} = Yes_F \cup D_i$ with $D_i \subseteq DontC_F$ and $DontC_{Fi} = \emptyset$. Each F^i is now a *completely specified boolean function*. We choose, among the F^i's, the *sum of product form* (SOP) which contains a minimal number of different variables. It is worth noting that considering all possible extensions of the *incompletely specified boolean function* F as well as computing the SOP form of them with the minimal number of different variables has an exponential cost.

In [18] the author proposes *LRed* a more efficient (still exponential) algorithm that finds the optimal solution. Moreover, some very efficient heuristics (that cannot guarantee minimality) can be found in [14, 10, 16, 8, 12, 18]. Finally it is worth noting that once a minimal (w.r.t. the number of different variables) formula representing our incompletely specified boolean function is determined, this can be represented either in disjunctive normal form or,

alternatively, in a conjunctive normal form since the dependency of a boolean formula from a variable does not depend on the normal form used to represent it.

We present two examples where we compute the minimal forms of a given formula f under an hypothesis h by considering all possibile extensions of f and by selecting the one with the minimum number of variables. The same result could be obtained by applying an exact algorithm like *LRed*. In both examples, two different minimal forms f_1 and f_2 of the formula f under the hypothesis h arise. In particular, in one case we have that $f_1 \sim_h f_2$ while in the other one we have that $f_1 \sim_h f_2$ does not hold.

Example 3 Let $\mathscr{P} = \{A, B, C, D\}$ be the set of propositional symbols. We consider the propositional formula on $\mathscr{P}$ defined as follows

$$f = (A \wedge B \wedge C) \vee (A \wedge \neg B \wedge \neg C) \vee D.$$

Moreover, let us consider as an hypothesis the following propositional formula on $\mathscr{P}$,

$$h = (\neg C \vee (A \wedge B) \vee (\neg A \wedge \neg B)) \wedge \neg D.$$

Fig. 1 Part (a), Part (b) and Part (c) describe the boolean functions F, F^1 and F^2 (resp.) for the formulas of Example 3.

We apply the minimization algorithm to formula f under the assumption h to calculate the minimal forms. Figure 1 (Part (a)) describes the incompletely specified boolean function F for formula f under the assumption h using a Karnaugh's map. The quadruples of boolean values for the propositional symbols (A, B, C, D) belonging to Yes_F are the ones corresponding to the entries marked with 1 (for *true*), while the quadruples corresponding to the entries marked with $-$ belongs to $DontC_F$. The remaining entries are marked with 0 (for *false*).

In order to derive the formulas equivalent to f under the hypothesis h we have to consider all possible completely specified boolean functions that can be obtained by choosing some entries marked with $-$ and turning them into 1. For each of such functions a sum of product (SOP) form has to be calculated and the one with a smallest number of variables has to be selected. The minimal forms are the formulas having the minimum number of different propositional symbols.

Part (b) and (c) of Figure 1 show two cases of completely specified boolean functions derived from F using the Karnaugh's maps. Part (b) describes the completely specified boolean function F^1 such that

$Yes_{F^1} = Yes_F \cup \{$ $(0,0,0,1), (0,0,1,1), (0,1,0,1), (0,1,1,1), (1,1,0,1), (1,0,1,1),$
$(1,1,1,1), (1,0,0,1), (1,0,1,0)\}$

while Part (c) describes the completely specified boolean function F^2 such that

$Yes_{F^2} = Yes_F \cup DontC_F = Yes_{F^1} \cup \{(0,1,1,0)\}$.

In the case (b) a minimal SOP can be obtained by considering the three prime implicants [15] shown in the picture, which correspond to the formula

$$f_1 = (A \wedge \neg B) \vee (B \wedge C).$$

Analogously, in the case (c) a minimal SOP can be obtained by considering three prime implicants. Hence, we obtain the formula

$$f_2 = (A \wedge \neg B) \vee (A \wedge C).$$

Among all the extensions of the boolean function F the cases described in Figure 1 (b) and (c) correspond to the minimal case (note that in both case D does not appear in the simplified formula). In this case the f_1 and f_2 are minimal forms of f under the hypothesis h. Note that we have that $f \equiv_h f_1 \equiv_h f_2$ while neither $f \equiv f_1$ nor $f \equiv f_2$ holds. Moreover, f_1 and f_2 contain the same set of propositional symbols (i.e. $atom(f_1) = atom(f_2)$) and therefore $f_1 \sim_h f_2$ holds.

Example 4 Let $\mathscr{P}$ be the set of propositional symbols introduced in Example 3. In this case we consider the propositional formula on $\mathscr{P}$ defined as follows

$$f = C \wedge B \wedge \neg A.$$

Moreover, let us consider as an hypothesis the following propositional formula on $\mathscr{P}$,

$$h = \big((\neg B \wedge A) \vee (\neg A \wedge B)\big).$$

In order to calculate the minimal form of formula f under the hypothesis h we apply the minimization algorithm to formula f and to the assumption h. By applying the minimization algorithm similarly as in the case of Example 3 we derive two different formulas f_1 and f_2 which are minimal forms:

$$f_1 = C \wedge \neg A \qquad f_2 = C \wedge B.$$

Also in this case $f \equiv_h f_1 \equiv_h f_2$ holds while neither $f \equiv f_1$ nor $f \equiv f_2$ hold. However in this case, differently from the one of Example 3, we have that $f_1 \not\sim_h f_2$ since neither $atom(f_1) \subseteq atom(f_2)$ nor $atom(f_2) \subseteq atom(f_1)$ holds.

4 Formula Based Predictors for Generalized Reaction Systems

In [7] Brijder, Ehrenfeucht and Rozenberg introduced the idea of *predictor*. Assume that one is interested in knowing whether an object $s \in S$ will be present after n steps of execution of a reaction system. Since the only source of non-determinism are the contextual elements received at each step, observing such elements can allow us to predict the production of s after n steps. In general, not all contextual elements are relevant for determining if s will be produced. A predictor is hence the subset Q of S that is actually essential to be observed among contextual elements for predicting whether s will be produced after n steps or not.

In [2] we continued the investigation on predictors by introducing the new notion of *formula based predictor*. A formula based predictor consists in a propositional logic formula to be satisfied by the sequence of (sets of) elements provided by the environment. Satisfaction of the logic formula by context sequences precisely discriminates the cases in which s will be produced after n steps from those in which it will not.

In particular, we have proved that minimal formula based predictors exist for a given object s at step n and that they can be computed in a systematic way. Among all formulas that predict an object s at step n we are interested in the formulas containing the minimal numbers of propositional symbols. In fact, minimal formulas do not contain spurious causal dependencies and they can be more easily verified. In this section we revise the main notions presented in [2] adapting them to reaction systems working with generalized contexts.

4.1 Definitions and properties

We use propositional formulas to describe properties of generalized context sequences. The case when the environment supplies an object s is simply indicated by symbol s while the case when it absorbs s is indicated by $\bar{s}$. Hence, given a set of objects S, we introduce a corresponding set $\widehat{S} = S \cup \{\bar{s} \mid s \in S\}$. Moreover, labelled objects are used to relate objects with execution steps of a reaction systems. Given a set of objects S, we define a corresponding set of *labelled objects* $\widehat{S} \times \mathbb{N}$. For the sake of legibility, we denote $(x, i) \in \widehat{S} \times \mathbb{N}$ simply as x_i. Moreover, for a given $n \in \mathbb{N}$ we introduce $\widehat{S^n} = \bigcup_{i=0}^{n} \widehat{S_i}$ where $\widehat{S_i} = \{x_i \mid x \in \widehat{S}\}$.

Propositional formulas on labelled objects $\widehat{S^n}$ describe properties of n-step generalized context sequences. Intuitively, a labelled object s_i represents the presence (or the absence, if negated) of object s in the i-th element C_i of the n-step context sequence $\gamma = (C_0, \overline{C}_0), (C_1, \overline{C}_1), ..., (C_n, \overline{C}_n)$. Analogously, an object $\bar{s}_i$ represents the presence (or the absence, if negated) of object s in the i-th element $\overline{C}_i$ of the n-step context sequence. This interpretation leads to the following definition of satisfaction relation for propositional formulas on generalized context sequences (based on the satisfaction relation of Definition 3).

Definition 7 Let S be a finite set of symbols and $\gamma = (C_0, \overline{C}_0), (C_1, \overline{C}_1), ..., (C_n, \overline{C}_n)$ be a n-step generalized context sequence. We define the *interpretation* of γ, denoted by I_γ, as $I_\gamma : \widehat{S^n} \to \{\mathbf{T}, \mathbf{F}\}$ where, for $x_i \in \widehat{S^n}$:

$$I_\gamma(x_i) = \begin{cases} \mathbf{T} & \text{if } x \in C_i \text{ or } x = \bar{s}, s \in \overline{C}_i \\ \mathbf{F} & \text{otherwise} \end{cases}$$

where x denotes the object obtained by removing the information on the step i from the propositional symbol x_i.

Definition 8 Let $\gamma = (C_0, \overline{C}_0), (C_1, \overline{C}_1), ..., (C_n, \overline{C}_n)$ be a n-step generalized context sequence and $f \in F_{\widehat{S^n}}$ a propositional formula. The *satisfaction relation* $\gamma \models f$ is defined as $I_\gamma \models f$.

We say that a context sequence γ satisfies a formula f when $\gamma \models f$. As an example, let us consider the context sequence $\gamma = (C_0, \overline{C}_0), (C_1, \overline{C}_1)$, where $C_0 = \{A\}$, $\overline{C}_0 = \{B\}$, $C_1 = \{B\}$ and $\overline{C}_1 = \{\}$. We have that γ satisfies the formula $A_0 \wedge \neg B_0 \wedge \neg \overline{B}_1 \wedge \neg \overline{A}_1$ (i.e. $\gamma \models A_0 \wedge \neg B_0 \wedge \neg \overline{B}_1 \wedge \neg \overline{A}_1$) while γ does not satisfy the formula $A_0 \wedge (\neg \overline{B}_0 \vee \neg B_1)$ (i.e. $\gamma \not\models A_0 \wedge (\neg \overline{B}_0 \vee \neg B_1)$).

Definition 8 induces an equivalence relation on formulas of $F_{\widehat{S^n}}$.

Definition 9 Let $f_1, f_2 \in F_{\widehat{S^n}}$ be propositional formulas. We say that $f_1 \cong f_2$ iff for any n-step generalized context sequence γ, $\gamma \models f_1 \iff \gamma \models f_2$.

In reaction systems with generalized contexts the equivalence relation $\cong$ does not coincide with standard logical equivalence $\equiv$ (as it happens with standard contexts). This is because generalized contexts are based on the assumption that the environment can either

supply an object or absorb it but it cannot perform both actions, at the same time[1]. Hence, we have that $(s_1 \wedge \overline{s}_1) \cong \mathit{false}$ (while obviously $(s_1 \wedge \overline{s}_1) \not\equiv \mathit{false}$) since there is no generalized context sequence γ such that $\gamma \models (s_1 \wedge \overline{s}_1)$.

We can now introduce the notion of formula based predictor for reaction systems working with generalized contexts. A formula f is a predictor for an object s in $n+1$ steps if it is satisfied precisely by the generalized context sequences that will lead to the production of s in $n+1$ steps.

Definition 10 (Formula based Predictor) Let $\mathscr{A} = (S,A)$ be a reaction system, $s \in S$ an object and $f \in F_{\widehat{S}^n}$ a propositional formula. We say that f *f-predicts s in $n+1$ steps* if for any n-step generalized context sequence $\gamma = (C_0, \overline{C}_0), \ldots, (C_n, \overline{C}_n)$

$$\gamma \models f \Leftrightarrow s \in D_{n+1}$$

where $\delta = D_0, \ldots, D_n$ is the result sequence corresponding to γ and $D_{n+1} = \mathit{res}_{\mathscr{A}}(C_n \cup (D_n \backslash \overline{C}_n))$.

Note that if formula f f-predicts s in $n+1$ steps and if $f' \cong f$ then also f' f-predicts s in $n+1$ steps. The following result states that there is a $\cong$-equivalent class of formulas that f-predicts s in $n+1$ steps.

Theorem 3 *Let $\mathscr{A} = (S,A)$ be a reaction system, $s \in S$ and $f_1, f_2 \in F_{\widehat{S}^n}$ be propositional formulas. If f_1* f-predicts s in $n+1$ *steps then*

$$f_1 \cong f_2 \iff f_2 \text{ f-predicts } s \text{ in } n+1 \text{ steps.}$$

Proof Assume that $f_1 \cong f_2$, by Definition 9 we have that, for any generalized context sequence γ, $\gamma \models f_1 \iff \gamma \models f_2$. Since, by hypothesis, f_1 f-predicts s in $n+1$ steps, by Definition 10, we have that formula f_1 is such that $\gamma \models f_1 \Leftrightarrow s \in D_{n+1}$. Since $\gamma \models f_1 \iff \gamma \models f_2$, we obtain that also f_2 is a formula such that $\gamma \models f_2 \Leftrightarrow s \in D_{n+1}$. Therefore, by definition, also f_2 f-predicts s in $n+1$ steps.

On the other hand, assume that both f_1 and f_2 f-*predict s in $n+1$ steps, and that, by contradiction, $f_1 \not\cong f_2$. This implies that there exists γ such that $\gamma \models f_1$ and $\gamma \not\models f_2$. By Definition 10 we have that $\gamma \models f_1$ implies $s \in D_{n+1}$, but at the same time $\gamma \not\models f_2$ implies $s \notin D_{n+1}$, leading to a contradiction. $\qquad\qquad\square$

As a consequence of Theorem 3, there exists a *unique* $\cong$-equivalence class of formulas that are the formula based predictors for s in $n+1$ steps. It is convenient to investigate the main properties of the equivalence classes of formulas induced by the relation $\cong$.

We noted above that the equivalence relation $\cong$ does not coincide with standard logical equivalence $\equiv$. However, we can characterize $\cong$ in terms of $\equiv$. To this aim we introduce a formula on $\widehat{S}^n$ that describes the properties of n-step generalized context sequences. Let S be a finite set of objects and $n \in \mathbb{N}$, we define

$$\mathscr{H}_{S,n} = \bigwedge_{i \in \{0, \ldots, n\}} \left(\bigwedge_{s \in S} \neg(s_i \wedge \overline{s}_i) \right) \qquad (2)$$

---

[1] Note that in a generalized context sequence we have $C_i \cap \overline{C}_i = \emptyset$ for each step i.

The formula (2) guarantees that the environment at each step cannot produce and absorb the same object s for any $s \in S$. For the sake of brevity, in the following we may denote $\mathscr{H}_{S,n}$ simply with $\mathscr{H}$ (when S and n are clear from the context).

The next results establish a relevant property: the equivalence relation $\cong$ coincides with the logical equivalence relation $\equiv$ *under* the *hypothesis* $\mathscr{H}$. According to Definition 4 we use $f_1 \equiv_{\mathscr{H}} f_2$ to express that f_1 and f_2 are logically equivalent under the assumption that $\mathscr{H}$ holds (that is $\mathscr{H} \implies f_1 \equiv f_2$).

Definition 11 Let S be a finite set of symbols. An n-step *unconstrained generalized context sequence is a sequence* $\gamma_u = (C_0, \overline{C}_0), (C_1, \overline{C}_1), \ldots, (C_n, \overline{C}_n)$ *where* $C_i, \overline{C}_i \subseteq S$, *for each* $i \in \{1, \ldots, n\}$.

Intuitively, an unconstrained context sequence is a generalized context sequence for which the constraint $C_i \cap \overline{C}_i = \emptyset$ is removed, for each i. Hence, generalized context sequences (GCS), as defined in Section 2.1, form a subset of unconstrained generalized context sequences ($UGCS$). For unconstrained context sequences we assume the obvious adaptation of Definitions 7 and 8. Specifically, I_{γ_u} denotes the interpretation derived from γ_u, according to Definition 7. Note that for every $\gamma_u \in UGCS$ there is an interpretation derived from it, and for every interpretation I there is a $\gamma_u \in UGCS$ from which I is derived.

Lemma 1 *Let S be a finite set of symbols and let $\gamma_u = (C_0, \overline{C}_0), (C_1, \overline{C}_1), \ldots, (C_n, \overline{C}_n)$ be n-step unconstrained generalized context sequence. We have that $\gamma_u \models \mathscr{H}$ iff γ_u is a generalized context sequence.*

Proof We recall that $\mathscr{H} = \bigwedge_{s \in S, i=\{0,\ldots,n\}} \neg(s_i \wedge \overline{s}_i)$. Suppose $\gamma_u \not\models \mathscr{H}$, that is $I_{\gamma_u}(\mathscr{H})$ is false, then being $\mathscr{H}$ a conjunction of terms, at least one of them is false. Suppose $\neg(s_j \wedge \overline{s}_j)$ is false, then $I_{\gamma_u}(s_j)$ is true and $I_{\gamma_u}(\overline{s}_j)$ is true. From Definitions 3 and 8 we have that $\gamma_u \models s_j$ and $\gamma_u \models \overline{s}_j$, thus, for Definition 7, $s_j \in C_j$ and $s_j \in \overline{C}_j$. Then γ_u is not a generalized context sequence. Suppose $I_{\gamma_u}(\mathscr{H})$ is true, with an analogous reasoning we can conclude that γ_u is a generalized context sequence. $\square$

Theorem 4 *Let $f_1, f_2 \in F_{\widehat{S}^n}$ be propositional formulas. We have that*

$$f_1 \cong f_2 \iff f_1 \equiv_{\mathscr{H}} f_2.$$

Proof Note that by Definition 4 $f_1 \equiv_{\mathscr{H}} f_2$ iff $\mathscr{H} \implies (f_1 \equiv f_2)$. Moreover, given the definition of $\cong$ we can reformulate the conclusion of the thesis of the theorem as:

$$(\forall \gamma_u \in GCS.(\gamma_u \models f_1) \iff (\gamma_u \models f_2)) \iff (\mathscr{H} \implies f_1 \equiv f_2)$$

- Let us first prove

$$(\forall \gamma_u \in GCS.(\gamma_u \models f_1) \iff (\gamma_u \models f_2)) \implies (\mathscr{H} \implies f_1 \equiv f_2).$$

Let us consider a Generalized Context Sequence γ_u, then $(\gamma_u \models f_1) \iff (\gamma_u \models f_2)$. For Definitions 7 and 8 we have that $I_{\gamma_u}(f_1) = I_{\gamma_u}(f_2)$, and, for Lemma 1, $I_{\gamma_u}(\mathscr{H})$ is true.

- Let us prove now the implication

$$(\forall \gamma_u \in GCS.(\gamma_u \models f_1) \iff (\gamma_u \models f_2)) \impliedby (\mathscr{H} \implies f_1 \equiv f_2).$$

Consider all the interpretations I_{γ_u} such that $I_{\gamma_u}(\mathscr{H} \implies f_1 \equiv f_2)$ is true. There are two possible cases. The first one is $I_{\gamma_u}(\mathscr{H})$ is false. In this case the context sequence

γ_u from which I_{γ_u} is derived, is not a Generalized Context Sequence (Lemma 1), thus $\gamma_u \notin GCS$. Consider now the case in which $I_{\gamma_u}(\mathscr{H})$ is true. In this case $I_{\gamma_u}(f_1 \equiv f_2)$ must be true. Thus, by Lemma 1 γ_u is a Generalized Context Sequence. Because $I_{\gamma_u}(f_1 \equiv f_2)$, for Definitions 7 and 8, $\gamma_u \models f_1 \equiv f_2$, and, by definition of $\models$, $\gamma_u \models f_1$ iff $\gamma_u \models f_2$.   □

Theorem 4 establishes a fundamental property of the $\cong$-equivalence class introducing a logical formulation based on hypothesis $\mathscr{H}$. Among the $\cong$-equivalent formulas that f-predict s in $n+1$ steps we are interested in the minimal ones containing the minimal numbers of propositional symbols. Based on the property of Theorem 4 the formulas that f-predict s in $n+1$ steps can be compared using the approximation order $\sqsubseteq_{\mathscr{H}}$, given in Definition 5.

Furthermore, based on the previous results we prove another relevant property of the class of formulas that are equivalent to a given f under the hypothesis $\mathscr{H}$. In this case there exists a *unique* $\sim_{\mathscr{H}}$-class of *minimal forms* of f under the hypothesis $\mathscr{H}$ (according to Definition 6). More specifically, if there exist two different formulas f_1 and f_2 that are minimal forms of f under the hypothesis $\mathscr{H}$ it must be the case that $f_1 \sim_{\mathscr{H}} f_2$. This means that $f_1 \equiv_{\mathscr{H}} f_2$ and that f_1 and f_2 contain exactly the same set of propositional symbols (e.g. $atom(f_1) = atom(f_2)$). Next results ensure this relevant property.

Theorem 5 *Let $f, f_1, f_2 \in F_{\widehat{S}^n}$ be propositional formulas such that $f_1, f_2 \in min_{\mathscr{H}}(f)$. We have that $atom(f_1) = atom(f_2)$.*

Proof Given that f_1 and f_2 are minimal forms of formula f under the hypothesis $\mathscr{H}$, by Definitions 4 and 6 , we have that $\mathscr{H} \implies f \equiv f_1$ and $\mathscr{H} \implies f \equiv f_2$. Hence we can conclude that $\mathscr{H} \implies f_1 \equiv f_2$. Moreover, since both formulas f_1 and f_2 are minimal forms of f we have that $f_1 \sqsubseteq_{\mathscr{H}} f$ and $f_2 \sqsubseteq_{\mathscr{H}} f$. Hence, by Definition 5 we have that $atom(f_1) \subseteq atom(f)$ and $atom(f_2) \subseteq atom(f)$. Since f_2 is a minimal form w.r.t. $\sqsubseteq_{\mathscr{H}}$ order, it cannot be the case that $atom(f_1) \subset atom(f_2)$, otherwise $f_1 \sqsubset_{\mathscr{H}} f_2$ and f_2 would not be minimal. The same reasoning apply to f_1, hence it cannot be the case that $atom(f_2) \subset atom(f_1)$. We prove that $atom(f_1) \supset atom(f_1) \cap atom(f_2)$ and $atom(f_2) \supset atom(f_1) \cap atom(f_2)$ is false.

Assume by contradiction that the formula is true. Let us choose a variable $x_i \in atom(f_1) \setminus (atom(f_1) \cap atom(f_2))$ and consider an assignment v of truth values to all variables in $atom(f_1) \setminus \{x_i\}$ coherent with $\mathscr{H}$ such that the value of x_i is essential to determine the truth value of the formula f_1. Note that such assignment must exists otherwise formula f_1 would not be minimal w.r.t. the $\sqsubseteq_{\mathscr{H}}$ order and would admit further simplifications by deliting variable x_i. Without losing generality, we consider first the case that $x_i = \overline{A}_i$, for some $A \in S$. The other case is analogous. The assignment v, of course, gives a truth value to all variables in $atom(f_1) \cap atom(f_2)$. Consider now all possible assignments $v'_1, ... v'_m$, for some m, to variables in $atom(f_2)$ that coincide with v on variables in $(atom(f_1) \cap atom(f_2))$ and such that each v'_i is coherent with v under the assumption $\mathscr{H}$. That is, for each v'_i the following hold:

$$(c \in atom(f_2) \setminus \{A_i\} \wedge v'_i(c) = true) \implies (v'_i(\overline{c}) = false) \tag{3}$$

and

$$(\overline{c} \in atom(f_2) \wedge v'_i(\overline{c}) = true) \implies (v'_i(c) = false) \tag{4}$$

Note that, by construction v'_i, for all i, is an assignment of truth values to all variables in $(atom(f_1) \cup atom(f_2)) \setminus \{\overline{A}_i\}$ coherent with $\mathscr{H}$. The proof now wants to show, by absurd, that we can construct an assignment to variables in $atom(f_1) \cup atom(f_2)$ coherent under $\mathscr{H}$, that evaluates f_1 and f_2 to different truth values. This would give a contradiction, since we have assumed $f_1 \equiv_{\mathscr{H}} f_2$. Hence, we have the following cases.

- The simpler one is when $A_i \notin atom(f_2) \setminus (atom(f_1) \cap atom(f_2))$. Consider one $v'_j \in \{v'_1, \dots v'_m\}$. Assume that such assignment evaluated f_2 to *true* (the other case is analogous). Now consider the assignment v_{x_i} to variable $\overline{A}_i$ that together with v evaluates formula f_1 to *false*. Note that such an assignment exists since we have chosen v such that x_i is essential to determine the truth value of the formula f_1 evaluated under the assignment v. We can conclude that the assignment v_{x_i} together with v'_j (that is coherent with $\mathscr{H}$, by construction) evaluates f_1 to *false* while it evaluates f_2 to *true*. Hence, it cannot be the case that $f_1 \equiv_{\mathscr{H}} f_2$. This gives a contradiction.
- In this case $A_i \in atom(f_2) \setminus (atom(f_1) \cap atom(f_2))$. Now consider the assignment v_{x_i} to variable $x_i = \overline{A}_i = false$ such an assignment together with v will evaluate formula f_1 to either *true* or *false*. Assume that it evaluated f_1 to *true* (the other case is analogous). Now consider a $v'_j \in \{v'_1, \dots v'_m\}$ such that v'_j evaluate f_2 to *false*. There are two cases.
  - such v'_j exists. By construction and since v_{x_i} is such that $x_i = \overline{A}_i = false$ we can conclude that the assignment v_{x_i} together with v'_j is coherent with $\mathscr{H}$ but it evaluates f_1 to *true* while it evaluates f_2 to *false*. Hence, it cannot be the case that $f_1 \equiv_{\mathscr{H}} f_2$. This gives a contradiction.
  - such v'_j does not exists. This means that all assignments $v'_1, \dots v'_m$ evaluate formula f_2 to *true*. Now consider the assignment v_{x_i} to variable $x_i = \overline{A}_i = true$ that implies that assignment v_{x_i} together with v will evaluate formula f_1 to *false*. Note that such an assignment v_{x_i} exists since we have chosen v such that x_i is essential to determine the truth value of the formula f_1 evaluated under the assignment v. Now between all $v'_1, \dots v'_m$ choose an assignment v'_j such that if variable $A_i \in atom(f_2) \setminus (atom(f_1) \cap atom(f_2))$ it is set to *false*. It should be easy to verify that such an assignment v'_j exists since (3) above does not constraint A_i. Now we can conclude that there exists an assignment v_{x_i} together with v'_j that is coherent with $\mathscr{H}$ but such that it evaluates f_1 to *false* while it evaluates f_2 to *true*. Hence, it cannot be the case that $f_1 \equiv_{\mathscr{H}} f_2$. This gives a contradiction. □

Corollary 1 *Let $f, f_1, f_2 \in F_{\widehat{S}^n}$ be propositional formulas such that $f_1, f_2 \in min_{\mathscr{H}}(f)$. We have that $f_1 \sim_{\mathscr{H}} f_2$.*

Proof By Definitions 5 and 6 we have that $f_1 \equiv_{\mathscr{H}} f$ and $f_2 \equiv_{\mathscr{H}} f$. Therefore, we have also $f_1 \equiv_{\mathscr{H}} f_2$. Moreover, by Theorem 5 we derive $atom(f_1) = atom(f_2)$. Hence, by Definition 5 we obtain $f_1 \sim_{\mathscr{H}} f_2$. □

Corollary 1 states that, given a formula f, if two different minimal forms f_1 and f_2 of f under the hypothesis $\mathscr{H}$ exist, then $f_1 \sim_{\mathscr{H}} f_2$, that is $f_1 \equiv_{\mathscr{H}} f_2$ and f_1 and f_2 contain exactly the same set of propositional symbols (e.g. $atom(f_1) = atom(f_2)$). Moreover, as we have illustrated in Section 3, the problem of computing a minimal form of a given formula f under the hypothesis $\mathscr{H}$ can be reduced to the minimal input support problem.

We also present an alternative approach based on the application of a class of equivalences on formulas of $F_{\widehat{S}^n}$ that are true under the hypothesis $\mathscr{H}_{S,n}$ (for short $\mathscr{H}$). The equivalences on formulas $F_{\widehat{S}^n}$ are listed in Table 3 in which we assume that $i \in \{0, \dots, n\}$ and $s \in S$. Note that the equivalences in the second column can also be obtained from the corresponding equivalences on the first column by applying the De Morgan law. These laws can be applied to establish that two formulas f_1 and f_2 with $f_1, f_2 \in F_{\widehat{S}^n}$ are equivalent under the hypothesis $\mathscr{H}$. Moreover, the equivalence of Table 3 can also be added to the standard simplification rules of the propositional logic to derive minimal forms.

We present the main properties of the equivalences presented in Table 3. The proof relies on the application of the minimization algorithm, presented in Section 3.

$$
\begin{array}{llll}
s_i & \wedge & \overline{s}_i & \equiv_{\mathscr{H}} \; false \\
s_i & \wedge & \neg\overline{s}_i & \equiv_{\mathscr{H}} \; s_i \\
\neg s_i & \wedge & \overline{s}_i & \equiv_{\mathscr{H}} \; \overline{s}_i
\end{array}
\qquad\qquad
\begin{array}{llll}
\neg s_i & \vee & \neg\overline{s}_i & \equiv_{\mathscr{H}} \; true \\
\neg s_i & \vee & \overline{s}_i & \equiv_{\mathscr{H}} \; \neg s_i \\
s_i & \vee & \neg\overline{s}_i & \equiv_{\mathscr{H}} \; \neg\overline{s}_i
\end{array}
$$

Table 3 Equivalences where $s \in S$, $i \in \{0, \ldots, n\}$ and $\mathscr{H}$ stands for $\mathscr{H}_{S,n}$.

Theorem 6 *Let S be a finite set of objects and $n \in \mathbb{N}$. The equivalences presented in Table 3 are correct with respect to the definition of $\equiv_{\mathscr{H}_{S,n}}$. Moreover, these are the only equivalences between formulas with at most one connective that can be derived considering the equivalence under the hypothesis $\mathscr{H}_{S,n}$.*

Proof In the equivalences of Table 3 we assume that $s \in S$ and $i \in \{0, \ldots, n\}$ and $\mathscr{H}$ stands for $\mathscr{H}_{S,n}$. Moreover, we recall that $\mathscr{H} = \bigwedge_{s \in S, i = \{0, \ldots, n\}} \neg(s_i \wedge \overline{s}_i)$.

First we prove that the equivalences of Table 3 are correct. In any case the proof shows that the formula on the right is the result of the minimization algorithm applied to the formula on the left with respect to the hypothesis $\mathscr{H}$.

- Suppose that $f = s_i \wedge \overline{s}_i$. In this case $Yes_F = \{\}$, $DontC_F = \{(true, true)\}$. The minimization of formula f is obtained by considering the following completely specified boolean formula $Yes_F = \{\}$ and $DontC_F = \{\}$. This results in the minimized formula *false*.
- Suppose $f = s_i \wedge \neg\overline{s}_i$. In this case, $Yes_F = \{(true, false)\}$ while $DontC_F$ is the same as before: $DontC_F = \{(true, true)\}$. The minimization of formula f is obtained by considering the following completely specified boolean formula $Yes_F = \{(true, false), (true, true)\}$, $DontC_F = \{\}$ obtained adding the pair of $DontC_F$ to Yes_F. This results in the minimized formula s_i.
- Suppose $f = \neg s_i \wedge \overline{s}_i$. In this case, $Yes_F = \{(false, true)\}$ while $DontC_F$ is the same as before. The minimization of formula f is obtained by considering the following completely specified boolean formula $Yes_F = \{(false, true), (true, true)\}$, $DontC_F = \{\}$ obtained adding $DontC_F$ to Yes_F. This results in the minimized formula $\overline{s}_i$.

The formulas of the second column can be obtained by applying the De Morgan laws to the corresponding equivalences in the first column.

We prove that there are not other possible equivalences under the hypothesis $\mathscr{H}$ on opposite actions. The only cases to be considered are $f_1 = \neg s_i \wedge \neg\overline{s}_i$ and $f_2 = s_i \vee \overline{s}_i$. In the case of f_1 we have $Yes_F = \{(false, false)\}$ while $DontC_F$ is the same as before. The minimization of formula f_1 is obtained by considering the following completely specified boolean formula $Yes_F = \{(false, false)\}$ and $DontC_F = \{\}$. This result is still the formula $\neg s_i \wedge \neg\overline{s}_i$. The case of f_2 can be obtained by applying the De Morgan laws.

Note that we have considered all possible cases of equivalences between formulas involving an (eventually negated) object s_i and its corresponding (eventually negated) object $\overline{s}_i$ and containing at most one connective between $\wedge$ and $\vee$. Moreover, since the hypothesis $\mathscr{H}$ relates objects of the form s_i with the corresponding object $\overline{s}_i$ only, it should be clear that equivalences between formulas involving an object s_i and another unrelated object, let us say, s' or even s_j with $j \neq i$, cannot be derived under the assumption $\mathscr{H}$. $\qquad\square$

We present some examples to illustrate the equivalences presented in Table 3.

Example 5 Let $S = \{\mathtt{A}, \mathtt{B}\}$ be the set of objects. We consider the propositional formula on $\widehat{S}^0$ such that

$$
f = \left(\mathtt{A}_0 \vee \left((\neg\mathtt{B}_0 \vee \neg\overline{\mathtt{B}}_0) \wedge \neg\overline{\mathtt{A}}_0\right)\right).
$$

By applying the equivalences $\neg s_i \vee \neg \overline{s}_i \equiv_{\mathscr{H}} true$ and $s_i \vee \neg \overline{s}_i \equiv_{\mathscr{H}} \neg \overline{s}_i$ of Table 3, we obtain

$$\begin{aligned} f \quad &= \left(\mathtt{A}_0 \vee \left((\neg \mathtt{B}_0 \vee \neg \overline{\mathtt{B}}_0) \wedge \neg \overline{\mathtt{A}}_0 \right) \right) \\ &\equiv_{\mathscr{H}} \left(\mathtt{A}_0 \vee (true \wedge \neg \overline{\mathtt{A}}_0) \right) \\ &\equiv_{\mathscr{H}} \left(\mathtt{A}_0 \vee \neg \overline{\mathtt{A}}_0 \right) \\ &\equiv_{\mathscr{H}} \neg \overline{\mathtt{A}}_0 \end{aligned}$$

It is worth mentioning that in this case the formula $\neg \overline{\mathtt{A}}_0$ is exactly the minimal form of formula f under the hypothesis $\mathscr{H}$ that can be computed by applying the minimization algorithm to f and $\mathscr{H}$.

Example 6 Let $S = \{\mathtt{A}, \mathtt{B}, \mathtt{C}\}$ be the set of objects. Let us consider the propositional formula f on $\widehat{S}^2$ where

$$f = \left(\left(\mathtt{A}_2 \vee \left((\mathtt{C}_1 \vee \neg \mathtt{C}_1) \wedge \neg \overline{\mathtt{A}}_2 \right) \right) \right).$$

By applying the equivalence $s_i \vee \neg \overline{s}_i \equiv_{\mathscr{H}} \neg \overline{s}_i$ of Table 3, we obtain

$$\begin{aligned} f = \quad &\left(\left(\mathtt{A}_2 \vee \left((\mathtt{C}_1 \vee \neg \mathtt{C}_1) \wedge \neg \overline{\mathtt{A}}_2 \right) \right) \right) \\ &\equiv_{\mathscr{H}} \left(\left(\mathtt{A}_2 \vee (true \wedge \neg \overline{\mathtt{A}}_2) \right) \right) \\ &\equiv_{\mathscr{H}} \left(\mathtt{A}_2 \vee \neg \overline{\mathtt{A}}_2 \right) \\ &\equiv_{\mathscr{H}} \neg \overline{\mathtt{A}}_2 \end{aligned}$$

As in Example 5, the formula $\neg \overline{\mathtt{A}}_2$ is exactly the minimal form of formula f under the hypothesis $\mathscr{H}$ that can be computed by applying the minimization algorithm to f and $\mathscr{H}$.

4.2 Computing formula based predictors

We present a technique for computing in a systematic way formula based predictors for a given object s at step $n+1$. The definition relies on a few auxiliary notions. In order to describe the causes of an object s we use objects of reaction systems as propositional symbols.

First of all we define the *applicability predicate* of a reaction a as a propositional logic formula on S (that is a formula of F_S) describing the requirements for applicability of a, namely that all reactants have to be present and inhibitors have to be absent. This is represented by the conjunction of all atomic formulas representing reactants and the negations of all atomic formulas representing inhibitors of the considered reaction.

Definition 12 Let $a = (R, I, P)$ with $R, I, P \subseteq S$ for a finite set of objects S. The *applicability predicate* of reaction a, denoted by $ap(a)$, is the following propositional logic formula:

$$ap(a) = \left(\bigwedge_{s_r \in R} s_r \right) \wedge \left(\bigwedge_{s_i \in I} \neg s_i \right).$$

Then we define the *causal predicate* of a given object s. The causal predicate is a logic formula on S representing the conditions for the production of s in one step, that is the fact that at least one reaction having s as a product has to be applicable (as specified by the applicability predicate of such reactions).

Definition 13 Let $\mathscr{A} = (S, A)$ be a reaction system and $s \in S$. The *causal predicate* of s in $\mathscr{A}$, denoted by $cause(s, \mathscr{A})$ (or $cause(s)$, when $\mathscr{A}$ is clear from the context), is defined as follows.

$$cause(s, \mathscr{A}) = \bigvee_{\{a \in A \mid a = (R, I, P), s \in P\}} ap(a).$$

Note that if there is no reaction $(R, I, P) \in A$ such that $s \in P$ we obtain $cause(s) = false$ given that as usual the disjunction of zero clauses is false.

Example 7 Let us consider the reaction system presented in Example 1

$$\mathscr{A} = (\{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}, \mathtt{G}\}, \{a_1, a_2, a_3, a_4, a_5\})$$

with reactions

$$a_1 = (\{\mathtt{A}, \mathtt{B}\}, \{\mathtt{C}\}, \{\mathtt{D}\}) \quad a_2 = (\{\mathtt{E}\}, \{\}, \{\mathtt{B}\}) \quad a_3 = (\{\mathtt{F}\}, \{\}, \{\mathtt{B}\})$$
$$a_4 = (\{\mathtt{E}\}, \{\}, \{\mathtt{A}\}) \quad a_5 = (\{\mathtt{G}\}, \{\}, \{\mathtt{C}, \mathtt{A}\}).$$

The *applicability predicates* of the reactions are

$$ap(a_1) = \mathtt{A} \wedge \mathtt{B} \wedge \neg \mathtt{C} \quad ap(a_2) = \mathtt{E} \quad ap(a_3) = \mathtt{F}$$
$$ap(a_4) = \mathtt{E} \qquad\qquad ap(a_5) = \mathtt{G}.$$

Thus, the *causal predicates* of the objects are

$$cause(\mathtt{A}) = \mathtt{E} \vee \mathtt{G} \qquad\qquad cause(\mathtt{B}) = \mathtt{E} \vee \mathtt{F}$$
$$cause(\mathtt{C}) = \mathtt{G} \qquad\qquad cause(\mathtt{D}) = \mathtt{A} \wedge \mathtt{B} \wedge \neg \mathtt{C}$$
$$cause(\mathtt{E}) = cause(\mathtt{F}) = cause(\mathtt{G}) = false.$$

Note that $cause(\mathtt{E}) = false$ given that $\mathtt{E}$ cannot be produced by any reaction (and analogously for objects $\mathtt{F}$ and $\mathtt{G}$).

We now define an operator $\mathtt{fbp}$ that computes a formula based predictor for an object s at step $n + 1$. The predictor for s at step $n + 1$ is obtained computing a formula on labelled objects $\widehat{S^n}$ that is satisfied by a generalized context sequence iff the object s will appear in the system after $n + 1$ steps. Such a formula is constructed starting from the causal predicate of object s.

Definition 14 Let $\mathscr{A} = (S, A)$ be a reaction system. We define a function $\mathtt{fbp} : S \times \mathbb{N} \to F_{\widehat{S^n}}$ as follows:

$$\mathtt{fbp}(s, n) = \mathtt{fbs}(cause(s), n)$$

where the auxiliary function $\mathtt{fbs} : F_S \times \mathbb{N} \to F_{\widehat{S^n}}$ is recursively defined as follows:

$$\mathtt{fbs}(s, 0) = s_0 \qquad\qquad \mathtt{fbs}(s, i) = s_i \vee (\mathtt{fbs}(cause(s), i - 1) \wedge \neg \overline{s}_i) \text{ if } i > 0$$
$$\mathtt{fbs}((f), i) = (\mathtt{fbs}(f, i)) \qquad \mathtt{fbs}(f_1 \vee f_2, i) = \mathtt{fbs}(f_1, i) \vee \mathtt{fbs}(f_2, i)$$
$$\mathtt{fbs}(\neg f, i) = \neg \mathtt{fbs}(f, i) \qquad \mathtt{fbs}(f_1 \wedge f_2, i) = \mathtt{fbs}(f_1, i) \wedge \mathtt{fbs}(f_2, i)$$
$$\mathtt{fbs}(true, i) = true \qquad\qquad \mathtt{fbs}(false, i) = false$$

The next theorem shows the main property of function $\mathtt{fbp}$.

Theorem 7 *Let $\mathscr{A} = (S, A)$ be a reaction system. For any object $s \in S$, $\mathtt{fbp}(s, n)$ f-predicts s in $n + 1$ steps.*

Proof By definition of formula based predictor and of fbp what we have to prove is that for every n-step generalized context sequence γ it holds $s \in D_{n+1} \iff \gamma \models \texttt{fbs}(cause(s),n)$, where $D_{n+1} = \text{res}_{\mathscr{A}}(C_n \cup (D_n \setminus \overline{C}_n))$ with $\delta = D_0, D_1, \ldots, D_n$ the result sequence corresponding to $\gamma = (C_0, \overline{C}_0), (C_1, \overline{C}_1), \ldots, (C_n, \overline{C}_n)$.

The proof is done by induction on n.

Base case: $n = 0$. We start by proving the $\Rightarrow$ implication $s \in D_1 \Rightarrow \gamma \models \texttt{fbs}(cause(s),0)$. From $s \in D_1$ it follows that there must exist in A at least a reaction $a_v = (R_v, I_v, P_v)$ with $s \in P_v$ such that $s \in \text{res}_{a_v}(C_0 \cup (D_0 \setminus \overline{C}_0))$, that is a_v has been applied in the first execution step of the reaction system. Since, by definition, $D_0 = \emptyset$ this implies that $R_v \subseteq C_0$ and $I_v \cap C_0 = \emptyset$. Hence, we are sure that

$$\gamma \models (\bigwedge_{s' \in R_v} s'_0) \wedge (\bigwedge_{s'' \in I_v} \neg s''_0).$$

By definition of ap and of fbs, we have that :

$$\gamma \models \texttt{fbs}(ap(a_v),0).$$

As a consequence,

$$\gamma \models \texttt{fbs}(ap(a_1),0) \vee \ldots \vee \texttt{fbs}(ap(a_k),0)$$

where $a_1, \ldots, a_k$ are all the reactions having s_1 as a product. By definition of *cause* and of fbs we can conclude that

$$\gamma \models \texttt{fbs}(cause(s),0) .$$

Now we prove the $\Leftarrow$ implication, namely $s_1 \in D_1 \Leftarrow \gamma \models \texttt{fbs}(cause(s),0)$. By definition of *cause* and of fbs we have:

$$\gamma \models \texttt{fbs}(ap(a_1),0) \vee \ldots \vee \texttt{fbs}(ap(a_k),0)$$

where $a_1, \ldots, a_k$ are the reactions having s_1 as a product. Now, there exists $v \in 1, \ldots, k$ such that

$$\gamma \models \texttt{fbs}(ap(a_v),0) .$$

By assuming $a_v = (R_v, I_v, P_v)$ we obtain, by definition of ap and of fbs, the following result:

$$\gamma \models (\bigwedge_{s' \in R_v} s'_0) \wedge (\bigwedge_{s'' \in I_v} \neg s''_0)$$

By definition of $\models$ (see Definition 8) we have that $I_\gamma \models (\bigwedge_{s' \in R_v} s'_0) \wedge (\bigwedge_{s'' \in I_v} \neg s''_0)$. Hence, by definition of I_γ it must be the case that $R_v \subseteq C_0$ and $I_v \cap C_0 = \emptyset$. Hence, reaction a_v can be applied at the first step of the execution of the reaction system giving $s_1 \in D_1$.

Induction case: $n > 0$. We start by proving the $\Rightarrow$ implication, namely $s \in D_{n+1} \Rightarrow \gamma \models \texttt{fbs}(cause(s),n)$. From $s \in D_{n+1}$ it follows that there must exist in A a reaction $a = (R,I,P)$ with $s \in P$ that has been applied in the $n+1$-th execution step of the reaction system. This implies that $R \subseteq C_n \cup (D_n \setminus \overline{C}_n)$ and $I \cap (C_n \cup (D_n \setminus \overline{C}_n)) = \emptyset$. As a consequence, there exist two disjoint sets R_D and R_C such that $R = R_D \cup R_C$, $R_D \subseteq (D_n \setminus \overline{C}_n)$, $R_C \subseteq C_n$ and $R_D \cap C_n = \emptyset$. (This means that all common elements of C_n and $D_n \setminus \overline{C}_n$ are in R_C.) Note that by definition of R_D we also have $R_D \cap \overline{C}_n = \emptyset$.

Now, under this hypothesis we have to prove that $\texttt{fbs}(cause(s),n)$ is satisfied on γ. By definition of *cause* and fbs we have

$$\texttt{fbs}(cause(s),n) = \bigvee_{a_k \in \{(R_k,I_k,P_k) \in A \mid s \in P_k\}} \texttt{fbs}(ap(a_k),n) .$$

Since there exist a value of k such that $a_k = a$, the proof of $\gamma \models \mathtt{fbs}(cause(s),n)$ can be reduced to the proof of $\gamma \models \mathtt{fbs}(ap(a),n)$.

By definition of ap we have

$$\mathtt{fbs}(ap(a),n) = \mathtt{fbs}((\bigwedge_{s' \in R} s') \wedge (\bigwedge_{s'' \in I} \neg s''), n)$$

that is equivalent to

$$\mathtt{fbs}((\bigwedge_{s^d \in R_D} s^d) \wedge (\bigwedge_{s^c \in R_C} s^c) \wedge (\bigwedge_{s'' \in I} \neg s''), n)$$

that, in turn, by definition of $\mathtt{fbs}$, is equivalent to

$$\bigwedge_{s^d \in R_D} (s_n^d \vee (\mathtt{fbs}(cause(s^d),n-1) \wedge \neg \bar{s}_n^d)) \wedge \bigwedge_{s^c \in R_C} (s_n^c \vee (\mathtt{fbs}(cause(s^c),n-1) \wedge \neg \bar{s}_n^c))$$
$$\wedge \bigwedge_{s'' \in I} (\neg(s_n'' \vee (\mathtt{fbs}(cause(s''),n-1) \wedge \neg \bar{s}_n''))).$$

Since $R_D \cap C_n = \emptyset$, $R_D \cap \overline{C}_n = \emptyset$, $R_C \subseteq C_n$ and $I \cap C_n = \emptyset$, this formula can be simplified as:

$$\bigwedge_{s^d \in R_D} (\mathtt{fbs}(cause(s^d),n-1)) \wedge \bigwedge_{s'' \in I} (\neg(\mathtt{fbs}(cause(s''),n-1) \wedge \neg \bar{s}_n'')).$$

Now, by induction hypothesis we have $s \in D_n \iff \gamma \models \mathtt{fbs}(cause(s),n-1)$. Since $R_D \subseteq D_n$ it follows that the formula can be further simplified as follows:

$$\bigwedge_{s'' \in I} (\neg(\mathtt{fbs}(cause(s''),n-1) \wedge \neg \bar{s}_n'')).$$

In order to continue we need to partition I into two subsets: $I_D = I \cap D_n \backslash \overline{C}_n$ and $I_{\overline{C}} = I \backslash I_D \subseteq \overline{C}_n$. Our formula can now be rewritten as follows:

$$\bigwedge_{s'' \in I_D} (\neg(\mathtt{fbs}(cause(s''),n-1) \wedge \neg \bar{s}_n'')) \wedge \bigwedge_{s'' \in I_{\overline{C}}} (\neg(\mathtt{fbs}(cause(s''),n-1) \wedge \neg \bar{s}_n''))$$

that, by definition of I_D and $I_{\overline{C}}$, turns out to be equivalent to

$$\bigwedge_{s'' \in I_D} (\neg(\mathtt{fbs}(cause(s''),n-1))).$$

The obtained formula is true by induction hypothesis.

Now we prove the $\Leftarrow$ implication, namely $s \in D_{n+1} \Leftarrow \gamma \models \mathtt{fbs}(cause(s),n)$. By definition of $\mathtt{fbs}$ we can rewrite the right-hand side of the implication as follows:

$$\gamma \models \bigvee_{a_k \in \{(R_k,I_k,P_k) \in A \mid s \in P_k\}} \mathtt{fbs}(ap(a_k),n).$$

In order for the disjunction to hold, there must be at least one disjoint that is modeled by γ. By definition, this implies that there exists at least one reaction $a \in \{(R_k,I_k,P_k) \in A \mid s \in P_k\}$ such that $\mathtt{fbs}(ap(a),n)$ is modeled by γ. Let us assume $a = (R,I,P)$, we have

$$\gamma \models \mathtt{fbs}(ap(a),n).$$

Now, the proof is reduced to proving that reaction a was applicable at step n of the execution of the reaction system, causing s to be present in D_{n+1}. By definition of ap we have

$$\mathtt{fbs}(ap(a),n) = \mathtt{fbs}((\bigwedge_{s' \in R} s') \wedge (\bigwedge_{s'' \in I} \neg s''),n)\,.$$

Let us consider the following partition of R, namely $R_C = R \cap C_n$ and $R_D = R \setminus R_C$ (that implies $R_D \cap C_n = \emptyset$).

We can now apply to $\mathtt{fbs}(ap(a),n)$ the same simplification as in the previous induction case, namely we can rewrite $\mathtt{fbs}(ap(a),n)$ as follows:

$$\mathtt{fbs}((\bigwedge_{s^d \in R_D} s^d) \wedge (\bigwedge_{s^c \in R_C} s^c) \wedge (\bigwedge_{s'' \in I} \neg s''),n)$$

that, by definition of $\mathtt{fbs}$, is equivalent to

$$\bigwedge_{s^d \in R_D} (s_n^d \vee (\mathtt{fbs}(cause(s^d),n-1) \wedge \neg \overline{s}_n^d)) \wedge \bigwedge_{s^c \in R_C} (s_n^c \vee (\mathtt{fbs}(cause(s^c),n-1) \wedge \neg \overline{s}_n^c))$$

$$\wedge \bigwedge_{s'' \in I} (\neg(s_n'' \vee (\mathtt{fbs}(cause(s''),n-1) \wedge \neg \overline{s}_n'')))\,.$$

Since γ models this formula, it follows $I \cap C_n = \emptyset$ (each s_n'' has to be false in the formula). Conditions $I \cap C_n = \emptyset$, $R_C \subseteq C_n$ and $R_D \cap C_n = \emptyset$ allow us to simplify the formula as follows:

$$\gamma \models \bigwedge_{s^d \in R_D} (\mathtt{fbs}(cause(s^d),n-1) \wedge \neg \overline{s}_n^d) \wedge \bigwedge_{s'' \in I} (\neg \mathtt{fbs}(cause(s''),n-1) \vee \overline{s}_n'')\,.$$

Consequently, we have that for each $s^d \in R_D$ it holds $\gamma \models \mathtt{fbs}(cause(s^d),n-1)$ and $s^d \notin \overline{C}_n$, whereas for each $s'' \in I$ it holds either $\gamma \not\models \mathtt{fbs}(cause(s''),n-1)$ or $s'' \in \overline{C}_n$. From these, by induction hypothesis we obtain $R_D \subseteq D_n \setminus \overline{C}_n$ and $I \cap (D_n \setminus \overline{C}_n) = \emptyset$. These conditions, together with conditions $R_C \subseteq C_n$ and $I \cap C_n = \emptyset$, allow us to conclude that $R = R_C \cup R_D \subseteq C_n \cup (D_n \setminus \overline{C}_n)$ and $I \cap (C_n \cup (D_n \setminus \overline{C}_n)) = \emptyset$. As a consequence, $s \in res_a(C_n \cup (D_n \setminus \overline{C}_n))$, that is $s \in D_{n+1}$. $\square$

Theorem 7 guarantees that the formula $\mathtt{fbp}(s,n)$ is a formula based predictor for s at $n+1$ steps but it does not ensure minimality. Hence, in general the formula computed by operator $\mathtt{fbp}$ is not minimal with respect to the order $\sqsubseteq_{\mathscr{H}}$ (introduced in Sections 3 and 4.1). However, as we have shown, there exists a *unique*-class of minimal forms of the formula $\mathtt{fbp}(s,n)$ under the hypothesis $\mathscr{H}$. Therefore, given two minimal forms f_1 and f_2 of the formula $\mathtt{fbp}(s,n)$ under the hypothesis $\mathscr{H}$ we have that $f_1 \sim_{\mathscr{H}} f_2$. As a consequence, all the minimal forms of the formula based predictor $\mathtt{fbp}(s,n)$ not only are equivalent under the hypothesis $\mathscr{H}$, but also contain exactly the same set of propositional symbols.

Corollary 2 *Let $\mathscr{A} = (S,A)$ be a reaction system and let $s \in S$ be an object. For any $f \in min_{\mathscr{H}}(\mathtt{fbp}(s,n))$, f f-predicts the object s in $n+1$ steps.*

Proof By Theorem 7 the formula $\mathtt{fbp}(s,n)$ f-predicts s in $n+1$ steps. Moreover, by definition of minimal form (Definition 6) if $f \in min_{\mathscr{H}}(\mathtt{fbp}(s,n))$ then $f \sqsubseteq_{\mathscr{H}} \mathtt{fbp}(s,n)$ and therefore (by Definition 5) we have also that $f \equiv_{\mathscr{H}} \mathtt{fbp}(s,n)$. Hence, by Theorems 3 and 4 we obtain that also f f-*predicts* s in $n+1$ steps. $\square$

Corollary 3 *Let $\mathscr{A} = (S,A)$ be a reaction system and let $s \in S$ be an object. For any $f_1, f_2 \in min_{\mathscr{H}}(\mathtt{fbp}(s,n))$, we have that $f_1 \sim_{\mathscr{H}} f_2$.*

Proof It follows immediately by Corollary 1. □

Furthermore, previous results suggest a methodology for computing minimal formula based predictors for an object s in $n + 1$ steps. In fact, given the formula $\mathtt{fbp}(s, n)$ a minimal form can be derived by applying the methodologies discussed in Section 3.

We illustrate the application of the operator $\mathtt{fbp}$ considering some examples of reaction systems.

Example 8 We consider the reaction system $\mathscr{A} = (\{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}, \mathtt{G}\}, \{a_1, a_2, a_3, a_4, a_5\})$ introduced in Example 1 (and discussed in Example 7) with reactions

$$a_1 = (\{\mathtt{A}, \mathtt{B}\}, \{\mathtt{C}\}, \{\mathtt{D}\}) \quad a_2 = (\{\mathtt{E}\}, \{\}, \{\mathtt{B}\}) \quad a_3 = (\{\mathtt{F}\}, \{\}, \{\mathtt{B}\}$$
$$a_4 = (\{\mathtt{E}\}, \{\}, \{\mathtt{A}\}) \quad a_5 = (\{\mathtt{G}\}, \{\}, \{\mathtt{C}, \mathtt{A}\}).$$

In order to observe the production of $\mathtt{D}$ after 3 steps we calculate the logic formula that f-*predicts* $\mathtt{D}$ *in 3 steps*. By applying the function $\mathtt{fbp}$ we obtain $\mathtt{fbp}(\mathtt{D}, 2) = f$ where:

$$
\begin{aligned}
f \quad &= \mathtt{fbs}\big((\mathtt{A} \wedge \mathtt{B} \wedge \neg \mathtt{C}), 2\big) \\
&= \big(\mathtt{fbs}(\mathtt{A}, 2) \wedge \mathtt{fbs}(\mathtt{B}, 2) \wedge \neg \mathtt{fbs}(\mathtt{C}, 2)\big) \\
&= \big((\mathtt{A}_2 \vee (\mathtt{fbs}(\mathtt{E} \vee \mathtt{G}, 1) \wedge \neg \overline{\mathtt{A}}_2)) \wedge (\mathtt{B}_2 \vee (\mathtt{fbs}(\mathtt{E} \vee \mathtt{F}, 1) \wedge \neg \overline{\mathtt{B}}_2)) \wedge \neg (\mathtt{C}_2 \vee (\mathtt{fbs}(\mathtt{G}, 1) \wedge \neg \overline{\mathtt{C}}_2))\big) \\
&= \big((\mathtt{A}_2 \vee (\mathtt{fbs}(\mathtt{E}, 1) \vee \mathtt{fbs}(\mathtt{G}, 1) \wedge \neg \overline{\mathtt{A}}_2)) \wedge (\mathtt{B}_2 \vee (\mathtt{fbs}(\mathtt{E}, 1) \vee \mathtt{fbs}(\mathtt{F}, 1) \wedge \neg \overline{\mathtt{B}}_2)) \wedge \neg (\mathtt{C}_2 \vee (\mathtt{G}_1 \wedge \neg \overline{\mathtt{C}}_2))\big) \\
&= \big((\mathtt{A}_2 \vee ((\mathtt{E}_1 \vee \mathtt{G}_1) \wedge \neg \overline{\mathtt{A}}_2)) \wedge (\mathtt{B}_2 \vee ((\mathtt{E}_1 \vee \mathtt{F}_1) \wedge \neg \overline{\mathtt{B}}_2)) \wedge \neg (\mathtt{C}_2 \vee (\mathtt{G}_1 \wedge \neg \overline{\mathtt{C}}_2))\big) \\
&= \big((\mathtt{A}_2 \vee ((\mathtt{E}_1 \vee \mathtt{G}_1) \wedge \neg \overline{\mathtt{A}}_2)) \wedge (\mathtt{B}_2 \vee ((\mathtt{E}_1 \vee \mathtt{F}_1) \wedge \neg \overline{\mathtt{B}}_2)) \wedge \neg \mathtt{C}_2 \wedge (\neg \mathtt{G}_1 \vee \overline{\mathtt{C}}_2)\big)
\end{aligned}
$$

In this case the obtained formula f is a minimal form (with respect to order $\sqsubseteq_{\mathscr{H}}$) that is $f \in min_{\mathscr{H}}(f)$. Note that also the formula f' is such that $f' \in min_{\mathscr{H}}(f)$ where

$$f' = (\mathtt{A}_2 \vee \mathtt{E}_1 \vee \mathtt{G}_1) \wedge \neg \overline{\mathtt{A}}_2 \wedge (\mathtt{B}_2 \vee \mathtt{E}_1 \vee \mathtt{F}_1) \wedge \neg \overline{\mathtt{B}}_2 \wedge \neg \mathtt{C}_2 \wedge (\neg \mathtt{G}_1 \vee \overline{\mathtt{C}}_2).$$

It is clear that $f' \sqsubseteq_{\mathscr{H}} f$ holds. Moreover, in this case we have also that $f \sim_{\mathscr{H}} f'$ given that the simplified version f' contains exactly the same set of propositional symbols of formula f (that is $atom(f') = atom(f)$).

Example 9 Let us consider the reaction system $\mathscr{A}_1 = (\{\mathtt{A}, \mathtt{B}, \mathtt{C}\}, \{a_1, a_2, a_3\})$ with the following reaction rules:

$$a_1 = (\{\mathtt{A}\}, \{\}, \{\mathtt{B}\}) \qquad a_2 = (\{\mathtt{C}\}, \{\}, \{\mathtt{A}\}) \qquad a_3 = (\{\}, \{\mathtt{C}\}, \{\mathtt{A}\}).$$

We are interested in the production of $\mathtt{B}$ after 3 steps by computing the logic formula that f-*predicts* $\mathtt{B}$ *in 3 steps*. By applying the function $\mathtt{fbp}$ we obtain $\mathtt{fbp}(\mathtt{B}, 2) = f$ with

$$
\begin{aligned}
f \quad &= \mathtt{fbs}(\mathtt{A}, 2) \\
&= \big(\mathtt{A}_2 \vee (\mathtt{fbs}(\mathtt{C} \vee \neg \mathtt{C}, 1) \wedge \neg \overline{\mathtt{A}}_2)\big) \\
&= \big((\mathtt{A}_2 \vee ((\mathtt{fbs}(\mathtt{C}, 1) \vee \mathtt{fbs}(\neg \mathtt{C}, 1)) \wedge \neg \overline{\mathtt{A}}_2))\big) \\
&= \big((\mathtt{A}_2 \vee ((\mathtt{C}_1 \vee \neg \mathtt{fbs}(\mathtt{C}, 1)) \wedge \neg \overline{\mathtt{A}}_2))\big) \\
&= \big((\mathtt{A}_2 \vee ((\mathtt{C}_1 \vee \neg \mathtt{C}_1) \wedge \neg \overline{\mathtt{A}}_2))\big)
\end{aligned}
$$

In this case, differently from the previous one, the obtained formula f is not minimal with respect to $\sqsubseteq_{\mathscr{H}}$. Actually, as we have already seen in Example 6, the formula f can be simplified under the hypotesis $\mathscr{H}$. In particular the formula $f' = \neg \overline{\mathtt{A}}_2$ is a minimal form of f under the hypothesis $\mathscr{H}$.

Note that $atom(f') \subset atom(f)$ and therefore $f \sim_{\mathscr{H}} f'$ does not hold (differently from the previous case).

5 Specialized Predictors for Generalized Reaction Systems

Formula based predictors do not assume any knowledge about the behaviour of the environment. However, in many cases the environment acts following specific patterns or, more generally, the context sequences representing the behaviour of the environment have specific dynamic properties. For example, it might happen that some kind of objects are never provided by the environment, or that some objects are provided only after some others. Moreover, an environment may absorb only one kind of object.

Formula based predictors can be quite complex because they characterize *all possible environments* that lead to the production of an object in the given number of steps. However, if we have some knowledge on the class of environments we are interested in, the formula based predictor can be simplified using the property we know on the environment.

In a previous paper [1] we revised formula based predictors by introducing *specialized formula based predictors* for standard reaction systems. A specialized formula based predictor is a propositional logical formula that predicts the production of an object after a given number of steps, by considering only the subset of the context sequences that already satisfy the properties we know to hold for the environments of interest. The properties on the behaviour of the environment can be expressed by temporal logic formulas on context sequences.

In this section we adapt the definition of specialized formula based predictor to the model of reaction systems with generalized contexts.

5.1 The temporal logic for generalized context sequences

We introduce a linear temporal logic for describing the properties of generalized context sequences. In the logic, propositional formulas on symbols $\widehat{S}$ describe the properties of single elements of a generalized context sequences. Hence, such formulas play the role of state formulas in traditional temporal logics. Temporal properties are expressed by variants of the usual next and until operators, and by derived eventually and globally operators.

The main difference with respect to the definition of the logic given in [1] is that formulas on the set of objects S are replaced by formulas on symbols $\widehat{S}$. Such formulas describe the properties of a single element (i.e. a pair $(C, \overline{C})$) of a generalized context sequence. In this case, a formula $f \in F_{\widehat{S}}$ expresses the properties of both sets of objects C and $\overline{C}$ and describes the objects that can/cannot appear in C as well as the objects that can/cannot appear in $\overline{C}$. More formally, we say that an element $(C, \overline{C})$ satisfies the formula f if and only if $I_{(C,\overline{C})} \models f$ (according to Definition 3) for the interpretation $I_{(C,\overline{C})} : \widehat{S} \to \{\mathbf{T}, \mathbf{F}\}$ where, for $x \in \widehat{S}$:

$$I_{(C,\overline{C})}(x) = \begin{cases} \mathbf{T} & \text{if } x \in C \text{ or } x = \overline{s}, s \in \overline{C} \\ \mathbf{F} & \text{otherwise} \end{cases}$$

Definition 15 (Temporal Formulas) Let S be a finite set of objects. The syntax of *temporal logic formulas* on $\widehat{S}$ is defined by the following grammar:

$$\psi ::= f \quad | \quad \psi \lor \psi \quad | \quad \psi \land \psi \quad | \quad \mathbf{X}\psi \quad | \quad \psi \mathbf{U}_k \psi \quad | \quad \mathbf{F}_k \psi \quad | \quad \mathbf{G}_k \psi$$

where $f \in F_{\widehat{S}}$ and $k \in \mathbb{N} \cup \{\infty\}$. We denote with $TL_{\widehat{S}}$ the set of all temporal logic formulas on $\widehat{S}$.

Given an n-step generalized context sequence $\gamma = (C_0, \overline{C}_0), \ldots, (C_n, \overline{C}_n)$, the simple temporal formula $f \in F_{\widehat{S}}$ states that f is satisfied by the first element of the context sequence $(C_0, \overline{C}_0)$ (formally $I_{(C_0, \overline{C}_0)} \models f$).

Formula $\mathbf{X}\psi$ (called *next*) states that ψ is satisfied by the generalized context sequence after one step, namely by $(C_1, \overline{C}_1), \ldots, (C_n, \overline{C}_n)$. Formula $\psi_1 \mathbf{U}_k \psi_2$ (called *until*) states that there exists $k' \leq k$ such that $(C_0, \overline{C}_0), \ldots, (C_{k'-1} \overline{C}_{k'-1})$ satisfies ψ_1 and $(C_{k'}, \overline{C}_{k'})$ satisfies ψ_2. Formula $\mathbf{F}_k \psi$ (called *future*) states that there exists $k' \leq k$ such that $(C_{k'}, \overline{C}_{k'})$ satisfies ψ. Formula $\mathbf{G}_k \psi$ (called *globally*) states that for all $i \leq k$ the element $(C_i, \overline{C}_i)$ satisfies ψ. Finally, operators $\vee$ and $\wedge$ are as usual.

As usual in temporal logics, formulas $\mathbf{F}_k \psi$ and $\mathbf{G}_k \psi$ are actually syntactic sugar for $true\mathbf{U}_k \psi$ and $\psi\mathbf{U}_k false$, respectively. Moreover, if $k \in \mathbb{N}$, we have that also $\psi_1 \mathbf{U}_k \psi_2$ can be rewritten either into $\psi_2 \vee (\psi_1 \wedge \mathbf{X}(\psi_1 \mathbf{U}_{k-1} \psi_2))$ when $k > 0$, or into ψ_2, when $k = 0$. Consequently, in the semantics of the temporal logic, we can omit derived operators $\mathbf{F}_k$ and $\mathbf{G}_k$, with $k \in \mathbb{N} \cup \{\infty\}$, and $\mathbf{U}_k$ with $k \in \mathbb{N}$.

We now present the formal definition of satisfiability of temporal logic formulas on n-step generalized context sequences. The satisfiability of a temporal logic formula ψ on a finite n-step generalized context sequence γ is defined in terms of the satisfiability relation $\models$, presented in Definition 8. To this aim we introduce an *encoding* of temporal formulas $TL_{\widehat{S}}$ into propositional formulas on labelled objects $\widehat{S}_n$. The encoding depends on the parameter $n \in \mathbb{N}$ reporting the length of the generalized context sequences that we want to model.

Definition 16 Let S be a finite set of objects and $n \in \mathbb{N}$. The *encoding* of a temporal logic formula $\psi \in TL_{\widehat{S}}$ into a propositional logic formula in $F_{\widehat{S}^n}$ is given by $[\![\psi]\!]_0^n$ where the function $[\![\_]\!]_-^n : TL_{\widehat{S}} \times \mathbb{N} \to F_{\widehat{S}^n}$ is defined as follows:

$$[\![f]\!]_i^n = \lfloor f \rfloor_i \qquad [\![\mathbf{X}\psi]\!]_i^n = \begin{cases} [\![\psi]\!]_{i+1}^n & \text{if } i < n \\ true & \text{if } i = n \end{cases}$$

$$[\![\psi_1 \vee \psi_2]\!]_i^n = [\![\psi_1]\!]_i^n \vee [\![\psi_2]\!]_i^n \qquad [\![\psi_1 \wedge \psi_2]\!]_i^n = [\![\psi_1]\!]_i^n \wedge [\![\psi_2]\!]_i^n$$

$$[\![\psi_1 \mathbf{U}_\infty \psi_2]\!]_i^n = \begin{cases} [\![\psi_2]\!]_i^n \vee ([\![\psi_1]\!]_i^n \wedge [\![\psi_1 \mathbf{U}_\infty \psi_2]\!]_{i+1}^n) & \text{if } i < n \\ [\![\psi_2]\!]_i^n \vee [\![\psi_1]\!]_i^n & \text{if } i = n \end{cases}$$

where $\lfloor \_ \rfloor_i : F_{\widehat{S}} \to F_{\widehat{S}^i}$ is a function that replaces, in a given propositional logic formula $f \in F_{\widehat{S}}$, every $x \in \widehat{S}$ with the corresponding labelled symbol $x_i \in \widehat{S}^i$.

Based on the encoding we define the satisfaction relation for n-step generalized context sequences and temporal formulas as follows.

Definition 17 Let γ be a n-step generalized context sequence. For any temporal logic formula $\psi \in TL_{\widehat{S}}$, we say that γ *satisfies* ψ, denoted by $\gamma \vdash \psi$, if and only if $\gamma \models [\![\psi]\!]_0^n$.

It is worth noting that in [1] we have given a standard inductive definition for $\gamma \vdash \psi$ for reaction systems. In this case we have used a more compact definition relying on the encoding of the temporal formulas. The definition presented in [1] can be adapted to the case of reaction system with generalized contexts in the obvious way.

5.2 Specialized formula based predictors

We now revise the notion of formula based predictor (presented in Definition 10) in order to take into account the knowledge we may have on the behaviour of the environment. We use a temporal logic formula ψ on generalized context sequence to express the properties of the environment.

Definition 18 (Specialized Formula based Predictor) Let $\mathscr{A} = (S,A)$ be a reaction system, $s \in S$ an object and $f \in F_{\widehat{S}^n}$ a propositional formula. Given a temporal logic formula $\psi \in TL_{\widehat{S}}$, we say that f *f-predicts s in $n+1$ steps with respect to* ψ iff for any n-step generalized context sequence $\gamma = (C_0, \overline{C}_0) \ldots, (C_n, \overline{C}_n)$ such that $\gamma \vdash \psi$ we have that

$$\gamma \models f \Leftrightarrow s \in D_{n+1}$$

where $\delta = D_0, \ldots, D_n$ is the result sequence corresponding to γ and $D_{n+1} = res_{\mathscr{A}}(C_n \cup (D_n \backslash \overline{C}_n))$.

It should be clear that any formula f that f-predicts s in $n+1$ steps also f-predicts s in $n+1$ steps with respect to any possible temporal formula ψ. In particular, the formula $\texttt{fbp}(s,n)$ (and analogously its minimal forms) is also a specialized predictor for s in $n+1$ steps for any ψ. However, even the minimal forms of the formula $\texttt{fbp}(s,n)$ are typically too general, because they do not use the knowledge on the properties that the context sequences have to satisfy. Therefore, in general, formula based predictors specialized with respect to a temporal formula ψ have a smaller number of propositional symbols.

In this case similarly as for formula based predictors there exists a *equivalence class* of formulas that are the specialized formula based predictors for s in $n+1$ steps with respect to a given temporal formula ψ. The following theorem establishes this fundamental property of the set of formulas that are specialized formula based predictors, based on the definition of $\mathscr{H}$ given in Equation 2 of Section 4.

Theorem 8 *Let $\mathscr{A} = (S,A)$ be a reaction system, $s \in S$, $f_1, f_2 \in F_{\widehat{S}^n}$ propositional formulas and $\psi \in TL_{\widehat{S}}$ a temporal formula. If f_1 f-predicts s in $n+1$ steps with respect to ψ then*

$$f_1 \equiv_{\mathscr{H} \wedge [\![\psi]\!]_0^n} f_2 \iff f_2 \text{ f-predicts } s \text{ in } n+1 \text{ steps with respect to } \psi.$$

Proof   – We prove the $\Longrightarrow$ implication. We assume that $f_1 \equiv_{\mathscr{H} \wedge [\![\psi]\!]_0^n} f_2$. By Definition 4, $f_1 \equiv_{\mathscr{H} \wedge [\![\psi]\!]_0^n} f_2$ iff $\mathscr{H} \wedge [\![\psi]\!]_0^n \implies f_1 \equiv f_2$. Note that $\mathscr{H} \wedge [\![\psi]\!]_0^n \implies (f_1 \equiv f_2)$ is logically equivalent to $\mathscr{H} \implies ([\![\psi]\!]_0^n \implies (f_1 \equiv f_2))$ that is, in turn, equivalent to $\mathscr{H} \implies (([\![\psi]\!]_0^n \implies f_1) \equiv ([\![\psi]\!]_0^n \implies f_2))$. Hence, by definition we have $([\![\psi]\!]_0^n \implies f_1) \equiv_{\mathscr{H}} ([\![\psi]\!]_0^n \implies f_2)$. By applying Theorem 4 we obtain that $([\![\psi]\!]_0^n \implies f_1) \cong ([\![\psi]\!]_0^n \implies f_2)$. By definition of $\cong$ we obtain that for any n-step generalized context sequence γ, $\gamma \models ([\![\psi]\!]_0^n \implies f_1) \iff \gamma \models ([\![\psi]\!]_0^n \implies f_2)$. Now since we have assumed that f_1 f-predicts s in $n+1$ steps with respect to ψ, we have that for any n-step generalized context sequence γ such that $\gamma \vdash \psi$ we have that $\gamma \models f_1 \Leftrightarrow s \in D_{n+1}$. Since, by definition $\gamma \vdash \psi \iff \gamma \models [\![\psi]\!]_0^n$, we can conclude that for any n-step generalized context sequence γ such that $\gamma \models [\![\psi]\!]_0^n$ we have that $\gamma \models f_1 \Leftrightarrow \gamma \models f_2 \Leftrightarrow s \in D_{n+1}$. Hence, by Definition 18, also f_2 f-predicts s in $n+1$ steps with respect to ψ.

– We prove the $\Longleftarrow$ implication. We assume that both formulas f_1 and f_2 f-predicts s in $n+1$ steps with respect to ψ. Hence, by Definition 18, for any generalized context sequence $\gamma = (C_0, \overline{C}_0) \ldots, (C_n, \overline{C}_n)$ such that $\gamma \vdash \psi$ we have that:

$$\gamma \models f_1 \Leftrightarrow s \in D_{n+1}, \ \gamma \models f_2 \Leftrightarrow s \in D_{n+1}.$$

It should be clear that similarly as for standard formula based predictors for any context sequence γ such that $\gamma \vdash \psi$ we have that $(\gamma \models f_1 \Leftrightarrow \gamma \models f_2)$.

Let us assume by contradiction that the claim does not hold. Thus, let us assume that the formulas f_1 and f_2 are not equivalent under the hypothesis $\mathcal{H} \wedge [\![\psi]\!]_0^n$. By Definition 4 this implies that $\mathcal{H} \wedge [\![\psi]\!]_0^n \implies f_1 \equiv f_2$ is not true. Therefore, there exists an interpretation I on symbols $\widehat{S^n}$ such that $I(\mathcal{H} \wedge [\![\psi]\!]_0^n) = \mathbf{T}$ and $I(f_1 \equiv f_2) = \mathbf{F}$. Note that if $I(f_1 \equiv f_2) = \mathbf{F}$ then there are two possible cases: either $I(f_1) = \mathbf{T}$ and $I(f_2) = \mathbf{F}$ or $I(f_1) = \mathbf{F}$ and $I(f_2) = \mathbf{T}$. We suppose that $I(f_1) = \mathbf{T}$ and $I(f_2) = \mathbf{F}$. The other case is analogous.

We now note that there exists an unconstrained context sequence $\gamma_u \in UGCS$ (defined according to Definition 11), corresponding to the interpretation I. This means that I coincides with the interpretation I_{γ_u} derived from γ_u. We recall that any possible interpretation on symbols $\widehat{S^n}$ corresponds to an unconstrained context sequences and vice-versa. There are two possible cases:

- $\gamma_u \notin GCS$. In this case γ_u is not a generalized context sequence and therefore by Lemma 1 we have $I_{\gamma_u}(\mathcal{H}) = \mathbf{F}$. As a consequence we have also $I_{\gamma_u}(\mathcal{H})(\mathcal{H} \wedge [\![\psi]\!]_0^n) = \mathbf{F}$. This is a contradiction since by hypothesis we have $I_{\gamma_u}(\mathcal{H} \wedge [\![\psi]\!]_0^n) = \mathbf{T}$.
- $\gamma_u \in GCS$. In this case γ_u is a generalized context sequence and therefore by Lemma 1 we have $I_{\gamma_u}(\mathcal{H}) = \mathbf{T}$. There are two possible cases: either $I_{\gamma_u}([\![\psi]\!]_0^n) = \mathbf{T}$ or $I_{\gamma_u}([\![\psi]\!]_0^n) = \mathbf{F}$.
    - Suppose that $I_{\gamma_u}([\![\psi]\!]_0^n) = \mathbf{F}$. Similarly as before we have also $I_{\gamma_u}(\mathcal{H})(\mathcal{H} \wedge [\![\psi]\!]_0^n) = \mathbf{F}$. This is a contradiction given that by hypothesis we have $I_{\gamma_u}(\mathcal{H} \wedge [\![\psi]\!]_0^n) = \mathbf{T}$.
    - Suppose that $I_{\gamma_u}([\![\psi]\!]_0^n) = \mathbf{T}$. In this case by definition we have $I_{\gamma_u} \models [\![\psi]\!]_0^n$ and therefore by Definition 17 we derive that $\gamma_u \vdash \psi$. As a consequence, given that f_1 and f_2 are predictors we have that $(\gamma_u \models f_1 \Leftrightarrow \gamma_u \models f_2)$. This is a contradiction given that by hypothesis we have $I_{\gamma_u}(f_1) = \mathbf{T}$ and $I_{\gamma_u}(f_2) = \mathbf{F}$. $\qquad\square$

As a consequence, there exists a *unique* $\mathcal{H} \wedge [\![\psi]\!]_0^n$-equivalence class of formulas that are the formula based predictors for s in $n+1$ steps with respect to ψ. Analogously as in the case of formula based predictors we are interested in minimal formulas having the smallest number of propositional symbols. Due to Theorem 8, two specialized predictors can be compared in terms of precision using the approximation order $\sqsubseteq_{\mathcal{H} \wedge [\![\psi]\!]_0^n}$ (presented in Definition 5). As expected in the case of specialized formula based predictors the approximation order depends also on the temporal logic formula ψ describing the set of the context sequences.

Analogously as for formula based predictors, we discuss the properties of the minimal forms of a formula f that f-*predicts* the object s in $n+1$ steps with respect to ψ (under the hypothesis $\mathcal{H} \wedge [\![\psi]\!]_0^n$). It is worth noting that given a formula f *it does not exists* a unique class of minimal forms, as it was the case for formula based predictors. Actually, *there might exist two different formulas f_1 and f_2 that are minimal forms of f under the hypotesis $\mathcal{H} \wedge [\![\psi]\!]_0^n$ such that $f_1 \not\sqsubset_{\mathcal{H} \wedge [\![\psi]\!]_0^n} f_2$*. This means that $f \equiv_{\mathcal{H} \wedge [\![\psi]\!]_0^n} f_1 \equiv_{\mathcal{H} \wedge [\![\psi]\!]_0^n} f_2$ holds but neither $atom(f_1) \subseteq atom(f_2)$ nor $atom(f_2) \subseteq atom(f_1)$ holds.

Moreover, previous results suggest a methodology to compute a formula based predictor for an object s in $n+1$ steps specialized with respect to a temporal formula ψ from a corresponding formula based predictor for s in $n+1$ steps. The *specialized versions* with respect to ψ can be obtained by computing the minimal forms of the formula $\mathtt{fbp}(s,n)$ under the hypothesis $\mathcal{H} \wedge [\![\psi]\!]_0^n$. Such minimal forms can be calculated in a systematic way by applying the *minimization algorithms* (discussed in Section 3) to formula $\mathtt{fbp}(s,n)$ and to the assumption $\mathcal{H} \wedge [\![\psi]\!]_0^n$.

The next examples of reaction systems illustrate the methodology to compute specialized formula based predictors from a corresponding formula based predictor.

Example 10 We consider the reaction system $\mathscr{A}'' = (\{\texttt{A},\texttt{B},\texttt{C},\texttt{D},\texttt{E}\},\{a_1,a_2\})$ with reactions

$$a_1 = (\{\texttt{B},\texttt{C},\texttt{E}\},\{\texttt{A}\},\{\texttt{D}\}) \qquad a_2 = (\{\texttt{B},\texttt{E}\},\{\texttt{C},\texttt{A}\},\{\texttt{D}\}).$$

By considering the production of object formula based predictor of D in 1-step we compute a minimal form of the corresponding formula based predictor (that is $\texttt{fbp}(\texttt{D},1)$) obtaining:

$$f = \texttt{E}_0 \wedge \texttt{B}_0 \wedge \neg \texttt{A}_0.$$

Let us assume now that the environment will supply *always* exactly one between object A and object B and that it will *never* absorb any object. The previous behaviour of the environment can be formalized by the following temporal logic formula:

$$\psi = \mathbf{G}_\infty\big((\neg \texttt{B} \wedge \texttt{A}) \vee (\neg \texttt{A} \wedge \texttt{B})\big) \wedge \mathbf{G}_\infty\big(\neg \overline{\texttt{A}} \wedge \neg \overline{\texttt{B}} \wedge \neg \overline{\texttt{C}} \wedge \neg \overline{\texttt{D}} \wedge \neg \overline{\texttt{E}}\big).$$

In order to calculate the specialized version of predictor of D in 1-step with respect to ψ we have to apply the minimization algorithm to formula f and to the hypothesis $\mathscr{H} \wedge [\![\psi]\!]_0^0$. In this case the encoding of the temporal formula ψ is given by

$$[\![\psi]\!]_0^0 = \big((\neg \texttt{B}_0 \wedge \texttt{A}_0) \vee (\neg \texttt{A}_0 \wedge \texttt{B}_0)\big) \wedge \neg \overline{\texttt{A}}_0 \wedge \neg \overline{\texttt{B}}_0 \wedge \neg \overline{\texttt{C}}_0 \wedge \neg \overline{\texttt{D}}_0 \wedge \neg \overline{\texttt{E}}_0.$$

Hence, by applying the minimization algorithm we obtain two formulas that *f-predict* D in 1-step with respect to ψ:

$$f_1 = \texttt{E}_0 \wedge \neg \texttt{A}_0 \qquad f_2 = \texttt{E}_0 \wedge \texttt{B}_0.$$

Note that both formulas f_1 and f_2 are minimal forms of f (that is $f_1 \sqsubseteq_{\mathscr{H} \wedge [\![\psi]\!]_0^0} f$ and $f_2 \sqsubseteq_{\mathscr{H} \wedge [\![\psi]\!]_0^0} f$). However, the formulas f_1 and f_2 contain different propositional symbols and neither $atom(f_1) \subseteq atom(f_2)$ holds nor $atom(f_2) \subseteq atom(f_1)$ holds. Thus, we have that $f_1 \not\sim_{\mathscr{H} \wedge [\![\psi]\!]_0^n} f_2$.

As we have commented in Section 2, reaction systems with ordinary contexts can be simulated by reaction systems working with generalized contexts, in which generalized context sequences have the form $\gamma = (C_0, \emptyset) \dots, (C_n, \emptyset)$. Moreover, we can use the notion of specialized formula based predictors to model formula based predictors for standard reaction systems. This behaviour can be captured by defining a temporal formula ensuring that the environment at each step does not have the capability to absorb objects.

Example 11 We consider the reaction system $\mathscr{A} = (\{\texttt{A},\texttt{B},\texttt{C},\texttt{D},\texttt{E},\texttt{F},\texttt{G}\},\{a_1,a_2,a_3,a_4,a_5\})$ introduced in Example 1 (and discussed in Example 8) with reactions

$$\begin{aligned} a_1 &= (\{\texttt{A},\texttt{B}\},\{\texttt{C}\},\{\texttt{D}\}) & a_2 &= (\{\texttt{E}\},\{\},\{\texttt{B}\}) & a_3 &= (\{\texttt{F}\},\{\},\{\texttt{B}\} \\ a_4 &= (\{\texttt{E}\},\{\},\{\texttt{A}\}) & a_5 &= (\{\texttt{G}\},\{\},\{\texttt{C},\texttt{A}\}). \end{aligned}$$

We show *formula based predictors* again for the production of object D in 3-steps *specialized* w.r.t. the given temporal formulas. We recall that the corresponding formula based predictor (that is a minimal form of $\texttt{fbp}(\texttt{D},2)$ presented in Example 8) is given by

$$f' = (\texttt{A}_2 \vee \texttt{E}_1 \vee \texttt{G}_1) \wedge \neg \overline{\texttt{A}}_2 \wedge (\texttt{B}_2 \vee \texttt{E}_1 \vee \texttt{F}_1) \wedge \neg \overline{\texttt{B}}_2 \wedge \neg \texttt{C}_2 \wedge (\neg \texttt{G}_1 \vee \overline{\texttt{C}}_2).$$

Let us assume first that the environment does not have the ability to absorb objects, as it is the case in standard reaction systems. This behaviour is formalized by the following temporal formula:

$$\psi_1 = \mathbf{G}_\infty(\neg \overline{A} \wedge \neg \overline{B} \wedge \neg \overline{C} \wedge \neg \overline{D} \wedge \neg \overline{E} \wedge \neg \overline{F} \wedge \neg \overline{G}).$$

Formula ψ_1 describes an invariant property which holds at any step of the generalized context sequence.

In order to calculate the specialized versions of predictor of object D in 3-steps we compute the encoding of the temporal formula ψ_1 obtaining:

$$\begin{aligned}
[\![\psi_1]\!]_0^2 = \quad & \neg \overline{A}_0 \wedge \neg \overline{B}_0 \wedge \neg \overline{C}_0 \wedge \neg \overline{D}_0 \wedge \neg \overline{E}_0 \wedge \neg \overline{F}_0 \wedge \neg \overline{G}_0 \\
\wedge \quad & \neg \overline{A}_1 \wedge \neg \overline{B}_1 \wedge \neg \overline{C}_1 \wedge \neg \overline{D}_1 \wedge \neg \overline{E}_1 \wedge \neg \overline{F}_1 \wedge \neg \overline{G}_1 \\
\wedge \quad & \neg \overline{A}_2 \wedge \neg \overline{B}_2 \wedge \neg \overline{C}_2 \wedge \neg \overline{D}_2 \wedge \neg \overline{E}_2 \wedge \neg \overline{F}_2 \wedge \neg \overline{G}_2.
\end{aligned}$$

We can now apply the minimization algorithm to formula f' with respect to the hypothesis $\mathcal{H} \wedge [\![\psi_1]\!]_0^2$ in order to derive corresponding specialized formulas with respect to ψ_1. In this case we obtain the following specialized version:

$$f'' = (A_2 \vee E_1 \vee G_1) \wedge (B_2 \vee E_1 \vee F_1) \wedge \neg C_2 \wedge \neg G_1.$$

It is worth noting that the specialized formula f'' is substantially reduced with respect to the corresponding formula based predictor f' (even in minimal form). Moreover, the specialized formula f'' is exactly the same formula based predictor that we would have obtained in case of standard reaction systems (see [2]).

Let us assume now that the environment satisfies additional properties. For example the following temporal formula describes again an invariant property

$$\psi_2 = \mathbf{G}_\infty(\neg E \wedge (\neg B \vee F)).$$

In this case the formula ψ_2 says that at each step: (a) the object E is not supplied by the environment but it has to be produced by the reactions; and (b) if the environment supplies B it also supplies F (namely, B implies F that is equivalent to $\neg B \vee F$).

We compute the *specialized formula based predictors* for the production of D in 3-steps with respect to the temporal formula $\psi_1 \wedge \psi_2$. To this aim we compute the encoding of the temporal formula ψ_2 obtaining:

$$\begin{aligned}
[\![\psi_2]\!]_0^2 = \quad & (\neg E_0 \wedge (\neg B_0 \vee F_0)) \ \wedge \\
& (\neg E_1 \wedge (\neg B_1 \vee F_1)) \ \wedge \\
& (\neg E_2 \wedge (\neg B_2 \vee F_2)).
\end{aligned}$$

Note that by definition of the encoding $[\![\psi_1 \wedge \psi_2]\!]_0^2 = [\![\psi_1]\!]_0^2 \wedge [\![\psi_2]\!]_0^2$. In this case, by applying the minimization algorithm to formula f' with respect to the hypothesis $\mathcal{H} \wedge [\![\psi_1 \wedge \psi_2]\!]_0^2$ we obtain the following specialized formula with respect to $\psi_1 \wedge \psi_2$:

$$f''' = (A_2 \vee G_1) \wedge B_2 \wedge \neg C_2 \wedge \neg G_1.$$

It is worth noting that the specialized formula f''' is substantially reduced with respect to the formula f'' specialized with respect to ψ_1 and consequently also with respect to the corresponding formula based predictor f'.

6 Applications

6.1 Interactions between pathways

This paper presents reaction systems working with environments that can not only supply objects, but also absorb them. This new mechanism can be very useful when, modelling a biological system, we are not interested in describing all possible reactions that compose it, but we want to focus on particular behaviours. Assume, for example, that we are interested in modelling just one particular pathway between many different ones that interact with it. In this case we might want to describe the reactions of the chosen pathway in details abstracting away from the description of all the other pathways that may interact with it. If this is case, the role of the environment becomes really crucial, because it has to model all the possible interactions the pathway of interest can have with the other non described reactions.

Following this approach, consider, for example, a very simple reaction system with the following reaction rules:

$$a_1 = (\{A\}, \{E\}, \{B\}) \qquad a_2 = (\{B\}, \{\}, \{C\}).$$

The fact that object A or B can also be produced by other reactions and therefore can be available for producing C, can be modelled by assuming an environment that does supply A or B at some steps. However, other reactions (that we are not interested in modelling in detail) may interfere with our pathway, for example by using object B in such a way that B becomes no longer available for producing C. Analogously, other reactions may cause E to be produced or removed by interfering with the production of B. If we are not interested in introducing in our model all the possibily complex interferences of our reactions with others, we can exploit the possibility of context sequences to provide or absorb objects to describe in an abstract way the production or removal of B and of E by other pathways at some given steps.

In order to show an application of this approch to the abstract representation of interference between pathways, let us consider the well-known *Epidermial Growth Factor (EGF)* pathway. Such a pathway aims at regulating cell growth and reproduction in response to an external stimulus. Such a stimulus (or signal) is the presence, in the environment, of a relevant concentration of EGF proteins, produced by other cells when tissue they constitute requires new cells.

The EGF signal is received by a cell through a transmembrane protein called *EGF receptor* EGFR. Such a protein is located on the external cell membrane (plasma membrane) and binds EGF proteins from the environment. When the EGFR is bound to an EGF protein, it starts interacting with proteins inside the cell in order to activate growth and reproduction mechanisms.

The EGF pathway is one of the pathways that is most frequently involved in cancer development. This is due to the fact that cancer cells usually aim at reproducing as much as possibile, and exploit mutations of the EGFR or of other proteins involved in the EGF pathway in order to activate the growth and reproduction mechanisms even in the absence of EGF signals. As a consequence, the EGF pathway is the target of several cancer therapies. For example, in the case of lung cancer, in which a mutated version of EGFR is often present, drugs such as monoclonal antibodies tend to bind to and inhibit the EGFR protein.

Drugs are molecules that interfere with the reactions of a target pathway. Drugs are usually also involved in other reactions, in which they are decomposed and become inactive. In order to model the EGF pathway and the effect of drugs (such as monoclonal antibodies)

on it we can use suitable context sequences to model the reactions of drug decomposition in an abstract way.

The following is a reaction system modelling the first steps of the EGF pathway:

$$\mathscr{A}_{EGF} = (\{\texttt{EGF}, \texttt{EGFR}, \texttt{actEGFR}, \texttt{S}, \texttt{P}\}, \{egf_1, egf_2, egf_3, egf_4\})$$

where $\texttt{EGF}$, $\texttt{EGFR}$ represent the EGF and the EGFR proteins, respectively, $\texttt{actEGFR}$ represents the activate form of EGFR (after the binding with EGF), and $\texttt{S}$ and $\texttt{P}$ represent the substrate and the product of the reaction that becomes enabled after the activation of EGFR. Moreover, reactions of the model are defined as follows:

$$egf_1 = (\{\texttt{EGFR}\}, \{\}, \{\texttt{EGFR}\})$$
$$egf_2 = (\{\texttt{EGF EGFR}\}, \{\}, \{\texttt{actEGFR}\})$$
$$egf_3 = (\{\texttt{actEGFR S}\}, \{\}, \{\texttt{actEGFR P}\})$$
$$egf_4 = (\{\texttt{S}\}, \{\}, \{\texttt{S}\})$$

Reaction egf_1 models the permanency of receptors on the cell membrane. Reacton egf_2 models the activation of EGFR by EGF. Finally, reaction egf_3 models the transformation of S into P performed by the activated EGFR, and reaction egf_4 models the permanency of substrate S.

The effect of the drug and the interference between the drug decomposition pathway and the EGF pathway can be modelled by context sequences that absorb $actEGFR$ as long as the drug is present. For instance, if the system starts with both EGF and $EGFR$ present and we want to describe the administration of the drug at the second step, with a duration of the decomposition pathway that goes on for the following 3 steps, we could consider a context sequence as in the following example.

	0	1	2	3	4	5	6	7	8
γ	EGF EGFR S	EGF	EGF $\overline{actEGFR}$	EGF $\overline{actEGFR}$	EGF $\overline{actEGFR}$	EGF $\overline{actEGFR}$	EGF	EGF	EGF
δ		EGFR actEGFR S	EGFR actEGFR S P	EGFR actEGFR S	EGFR actEGFR S	EGFR actEGFR S	EGFR actEGFR S	EGFR actEGFR S P	EGFR actEGFR S P

We can also describe the effects of the drug administration and the drug decomposition pathway, as described above, using a temporal formula on object $\overline{actEGFR}$. For instance, as before, our environment has to absorb $actEGFR$ starting exactly at the second step until the fifth step. This can be described by the following formula:

$$\psi_{drug} = \left(\mathbf{G}_1 \neg \overline{actEGFR}\right) \wedge \left(\mathbf{F}_2 \mathbf{G}_5 \overline{actEGFR}\right) \wedge \left(\mathbf{F}_6 \mathbf{G}_\infty \neg \overline{actEGFR}\right).$$

Reasonable requirements on the other molecules are that $EGFR$ and S are available from the beginning and after that the environment can only provide EGF molecules. These requirements can be formalized as follows:

$$\psi_{egf} = EGFR \wedge S \wedge \mathbf{G}_\infty \left(\neg actEGFR \wedge \neg P\right) \wedge \mathbf{X}\mathbf{G}_\infty \left(\neg EGFR \wedge \neg S\right).$$

Let us now exploit ψ_{drug} and ψ_{egf} to evaluate the interference of the drug pathway with the EGFR pathway. Since we are interested in the production of P in, let us say, 4 steps, we compute the corresponding formula based predictor:

$$\texttt{fbp}(P, 3) = \left(S_3 \vee \left(\left(S_2 \vee \left(\left(S_1 \vee \left(S_0 \wedge \neg \overline{S_1}\right)\right) \wedge \neg \overline{S_2}\right)\right) \wedge \neg \overline{S_3}\right)\right)$$
$$\wedge \left(actEGFR_3 \vee \left(\texttt{fbs}(cause(actEGFR), 2) \wedge \neg \overline{actEGFR_3}\right)\right)$$

where

$$\texttt{fbs}(cause(actEGFR),2) = \big(EGF_2 \wedge \big(EGFR_2 \vee \big((EGFR_1 \vee (EGFR_0 \wedge \neg\overline{EGFR_1})) \wedge \neg\overline{EGFR_2}\big)\big)\big)$$
$$\vee \big((S_2 \vee ((S_1 \vee (S_0 \wedge \neg\overline{S_1})) \wedge \neg\overline{S_2}))$$
$$\wedge \big(actEGFR_2 \vee (\texttt{fbs}(cause(actEGFR),1) \wedge \neg\overline{actEGFR_2})\big)\big)$$

and

$$\texttt{fbs}(cause(actEGFR),1) = \big(EGF_1 \wedge \big(EGFR_1 \vee (EGFR_0 \wedge \neg\overline{EGFR_1})\big)\big)$$
$$\vee \big((S_1 \vee (S_0 \wedge \neg\overline{S_1}))$$
$$\wedge \big(actEGFR_1 \vee ((EGF_0 \wedge EGFR_0) \vee (S_0 \wedge actEGFR_0)) \wedge \neg\overline{actEGFR_1}\big)\big)$$

The resulting formula is quite complex and expresses all the conditions that have to be met in order for object P to be produced at the fourth step. Now, if we consider only the requirements expressed by ψ_{egf}, we obtain the corresponding specialized predictor that is simplified as follows:

$$f_{\psi_{egf}} = \neg\overline{S_1} \wedge \neg\overline{S_2} \wedge \neg\overline{S_3}$$
$$\wedge \big(\big(\big((EGF_0 \wedge \neg\overline{actEGFR_1}) \vee (EGF_1 \wedge \neg\overline{EGFR_1})\big) \wedge \neg\overline{actEGFR_2}\big) \vee \big(EGF_2 \wedge \neg\overline{EGFR_1} \wedge \neg\overline{EGFR_2}\big)\big)$$
$$\wedge \neg\overline{actEGFR_3}$$

Such formula tells us that in order to produce P at the fourth step, we need: (a) S available at the third step. Since our environments will supply it just at the begining, this implies that S is never absorbed in the steps 1 to 3; (b) $actEGFR$ available at the third step. Since our environments do not supply $actEGFR$, it is necessary that $actEGFR$ is produced in one of the first three steps (once produced, reaction egf_3 ensures it is preserved) and never absorbed in the following.

At this step, if we include also the interference with the drug pathway described by ψ_{drug} we obtain

$$f_{\psi_{egf+drug}} = false$$

that proves that the interaction of the EGF patwhay with the drug pathway causes P not to be produced in the fourth step.

6.2 Cell metabolism and respiration

Glycolysis is a cascade of reactions that reduce glucose to pyruvate. A side product of such reactions is ATP, that is a source of energy for the cell. Pyruvate is then involved in cellular respiration: in the presence of oxygen it is decomposed into simpler molecules, and during such a decomposition further ATP molecules (much more than those produced during glycolysis) are released. Pyruvate can also be transformed into lactate, and such a transformation is reversible.

Glucose can enter the cell through the GLUT channel. GLUT is a transmembrane protein that is specialized into transportation of glucose molecules from the external environment into the cell. Similarly, lactate can enter and exit the cell through specific channels (proteins MCT1 and MCT4, respectively). Oxygen, instead, do not need any specific channel to enter the cell. A schematic representation of the mentioned pathways and of the channels is depicted in Figure 2.

The figure clearly shows that a cell can produce energy (ATP) in different ways, depending on which molecules are available in the environment. In particular, if glucose is present,
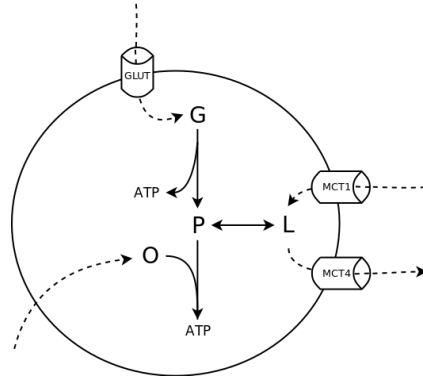
Fig. 2 A schematic representation of cell metabolism and respiration. G stands for glucose, P for pyruvate, L for lactate and O for oxygen.

it can be used to produce ATP during transformation into pyruvate (glycolysis). Pyruvate can then either be used to produce further ATP (renspiration) if oxygen is available, or be transformed into lactate to be released in the environment. The latter case is, for instance, followed by muscles during exertion leading to release of lactic acid when oxygen availablity is not enough for the needs of the muscle cells. When glucose is not available, the cell can follow another way to produce energy, that assumes lactate and oxygen to be available. In this case, lactate enters the cell through the MCT1 channel, it is transformed into pyruvate that, in turn, is used together with oxigen to produce ATP through the respiration pathway.

The possibility for the cell to produce ATP by exploiting different environmental resources allows cells to survive in different contexts. Moreover, in some situations it allows different cells to perform a synergistic use of resources. This is for instance the case of many kinds of tumour. Sometimes, when a tumour constitutes a mass of a relevant size, it encounters the problem that its central cells do not receive enough oxygen to perform respiration. In these cases, what can happen is that the central cells produce ATP from glucose through glycolysis, and release lactate. Cells in the perifery of the tumour mass organize themselves in order to produce ATP by exploiting lactate released by the central cells, and do not consume any glucose (that is left to the central cells). Such a synergy in the use of resources allows even a big tumour mass to survive. Such a role in tumour survival makes metabolic and respiration pathways of particular interest for drug development [21].

The following reaction system is a high level model of cell metabolism and respiration:

$$\mathscr{A}_{metabolism} = (\{\texttt{G},\texttt{P},\texttt{L},\texttt{O},\texttt{ATP}\}, \{gl, p2l, l2p, resp\})$$

where $\texttt{G}$ represents glucose, $\texttt{P}$ represents pyruvate, $\texttt{L}$ represents lactate, $\texttt{O}$ represents oxygen and $\texttt{ATP}$ represents ATP. Moreover, reactions of the model are defined as follows:

$$gl = (\{\texttt{G}\}, \{\}, \{\texttt{P},\texttt{ATP}\})$$
$$p2l = (\{\texttt{P}\}, \{\}, \{\texttt{L}\})$$
$$l2p = (\{\texttt{L}\}, \{\}, \{\texttt{P}\})$$
$$resp = (\{\texttt{O},\texttt{P}\}, \{\}, \{\texttt{ATP}\})$$

Reaction *gl* represents the glycolysis pathway, that transforms glucose received from the environment into pyruvate. A side product of glycolysis is ATP, thus this pathway provides

some energy to the cell. Reactions $p2l$ and $l2p$ represent the transformation of pyruvate into lactate and vice-versa. Lactate could be provided or absorbed by the cell environment due to the activity of the MCT1 and MCT4 transporters. Finally, reaction *resp* represents cell respiration consuming oxygen (provided by the environment) and pyruvate to obtain ATP.

The enviroment of the cell can only provide glucose, lactate and oxygen. The other substances (pyruvate and ATP) have to be produced by performing reactions. Moreover, lactate can be absorbed by the enviroment. As a consequence, the temporal logic formula that describes acceptable context sequences is the following:

$$\psi_{cell} = \mathbf{G}_\infty \left((G \vee L \vee O \vee \overline{L}) \wedge \neg P \wedge \neg ATP \wedge \neg \overline{G} \wedge \neg \overline{P} \wedge \neg O \wedge \neg \overline{ATP} \right).$$

The experiments pictured in the following tables show what happens when the following sets of substances are, respectively, constantly provided by the environment:

$$\{G,O\}, \{G\}, \{G,O,\overline{L}\}, \{G,\overline{L}\}, \{L,O\}, \{L\}.$$

Other cases are not shown since $\{O\}$ and $\{\overline{L}\}$ do not allow for any reaction, $\{G,L\}$ is analogous to $\{G\}$ and $\{G,O,L\}$ is analogous to $\{G,O\}$.

	0	1	2	3	4	5
γ	G O	G O	G O	G O	G O	G O
δ		P ATP	P ATP L	P ATP L	P ATP L	P ATP L

	0	1	2	3	4	5
γ	G	G	G	G	G	G
δ		P ATP	P ATP L	P ATP L	P ATP L	P ATP L

	0	1	2	3	4	5
γ	G O $\overline{L}$	G O $\overline{L}$	G O $\overline{L}$	G O $\overline{L}$	G O $\overline{L}$	G O $\overline{L}$
δ		P ATP	P ATP L	P ATP L	P ATP L	P ATP L

	0	1	2	3	4	5
γ	G $\overline{L}$	G $\overline{L}$	G $\overline{L}$	G $\overline{L}$	G $\overline{L}$	G $\overline{L}$
δ		P ATP	P ATP L	P ATP L	P ATP L	P ATP L

	0	1	2	3	4	5
γ	L O	L O	L O	L O	L O	L O
δ		P	P ATP L	P ATP L	P ATP L	P ATP L

	0	1	2	3	4	5
γ	L	L	L	L	L	L
δ		P	P L	P L	P L	P L

The experiments suggest that the cell can produce ATP (and hence survive) only when either glucose or both lactate and oxygen are provided by the environment. Let us see what happens if, for instance, the environment does not provide lactose and also stops providing glucose after a few steps. This situation is described by the following tables.

	0	1	2	3	4	5
γ	G O	G O	G O	O	O	O
δ		P ATP	P ATP L	P ATP L	P ATP L	P ATP L

	0	1	2	3	4	5
γ	G	G	G			
δ		P ATP	P ATP L	P ATP L	P L	P L

	0	1	2	3	4	5
γ	G O $\overline{L}$	G O $\overline{L}$	G O $\overline{L}$	O $\overline{L}$	O $\overline{L}$	O $\overline{L}$
δ		P ATP	P ATP L	P ATP L	L	

	0	1	2	3	4	5
γ	G $\overline{L}$	G $\overline{L}$	G $\overline{L}$	$\overline{L}$	$\overline{L}$	$\overline{L}$
δ		P ATP	P ATP L	P ATP L	L	

In three of the four considered cases, the interruption in the supply of glucose causes also the production of ATP to stop. This does not happen when oxygen is constantly available (first table), where the permanence of P (and of L) is guaranteed by reactions $p2l$ and $l2p$.

An exhaustive and more detailed description of the situations under which the cell produces ATP can be obtained by computing the predictor for ATP specialized for context sequences satisfying ψ_{cell}. Such a predictor (after simplification) is as follows:

$$\mathtt{fbp}(ATP,4) = G_4 \vee (O_4 \wedge (G_3 \vee L_3 \vee (G_1 \vee (L_1 \wedge \neg \overline{L_3})))).$$

The predictor proves what suggested by the experiments in the tables above. Moreover, it allows some dependencies between different execution steps to be observed. For instance, we can see that there is a correlation between the presence of lactate at the first and at the third steps. This is due to reactions $p2l$ and $l2p$, that can allow P and L to alternatively appear and disappear. As a consequence, we have that ATP is produced after step 4 if oxygen is present at step 4 and lactate is present either at step 3 or at step 1, as in the following experiments.

	0	1	2	3	4	5
γ				L		
					O	
δ					P	
						ATP

	0	1	2	3	4	5
γ		L				
					O	
δ			P		P	
				L		ATP

7 Conclusions

In this paper we extend the theory reaction systems by introducing generalized context sequences. Such sequences are used to model environments that not only supply objects at certain, fixed steps but that also have the capability to absorb objects. We have shown, by several realistic examples, that such capability can be very useful to more faithfully and easily model the behaviour of environment in biological systems.

On one hand the introduction of generalized context sequences has a quite simple and natural semantics, on the other hand it has many effects on dynamical causal dependences. In [7] Brijder, Ehrenfeucht and Rozenberg initiate an investigation of causalities in reaction systems, i.e. the ways that entities of a reaction system influence each other, introducing the idea of predictor. In [2] we enhanced the original idea of predictor by introducing the new notion of formula based predictor and in [1,3] we specialized formula based predictors by considering a temporal formula expressing properties of context sequences.

Both notion of predictor and specilized predictor needed to be revised for reaction systems with generalized contexts. This is because the introduction of generalized context sequences complicates the computation of predictors changing the theoretical setting. In this paper we revised all these concepts adapting them to reaction systems with generalized contexts and deeply investigating their properties such as the existence of minimal predictors.

As a future work, we plan to consider the application of the tabling and generalization procedures described in [2] to predictors of reaction systems with generalized contexts.

References

1. Barbuti, R., Gori, R., Levi, F., Milazzo, P.: Specialized predictor for reaction systems with context properties. In: Proc. of the 24th Int. Workshop on Concurrency, Specification and Programming, CS&P 2015, pp. 31–43 (2015)

2. Barbuti, R., Gori, R., Levi, F., Milazzo, P.: Investigating dynamic causalities in reaction systems. Theoretical Computer Science **623**, 114–145 (2016)
3. Barbuti, R., Gori, R., Levi, F., Milazzo, P.: Specialized predictor for reaction systems with context properties. Fundamenta Informaticae **147**(2-3), 173–191 (2016)
4. Bodei, C., Gori, R., Levi, F.: An analysis for causal properties of membrane interactions. Electr. Notes Theor. Comput. Sci. **299**, 15–31 (2013)
5. Bodei, C., Gori, R., Levi, F.: Causal static analysis for brane calculi. Theor. Comput. Sci. **587**, 73–103 (2015)
6. Brijder, R., Ehrenfeucht, A., Main, M.G., Rozenberg, G.: A tour of reaction systems. Int. J. Found. Comput. Sci. **22**(7), 1499–1517 (2011)
7. Brijder, R., Ehrenfeucht, A., Rozenberg, G.: A Note on causalities in reaction systems. ECEASST **30** (2010)
8. Brown, F.M.: Boolean reasoning - the logic of boolean equations. Kluwer (1990)
9. De Castro, L.N.: Fundamentals of natural computing: basic concepts, algorithms, and applications. CRC Press (2006)
10. Dietmeyer, D.: Logic Design of Digital Systems. Allyn and Bacon Inc., Boston (1978)
11. Ehrenfeucht, A., Rozenberg, G.: Reaction systems. Fundamenta informaticae **75**(1-4), 263–280 (2007)
12. Fujita, M., Matsunaga, Y.: Multi-level logic minimization based on minimal support and its application to the minimization of look-up table type FPGAS. In: 1991 IEEE/ACM International Conference on Computer-Aided Design, ICCAD 1993, Santa Clara, CA, USA, November 11-14, 1991. Digest of Technical Papers, pp. 560–563 (1991)
13. Gori, R., Levi, F.: Abstract interpretation based verification of temporal properties for bioambients. Inf. Comput. **208**(8), 869–921 (2010)
14. Halatsis, C., Gaitanis, N.: Irredundant normal forms and minimal dependece sets of a boolean function. IEEE Trans. Computers **27**(11), 1064–1068 (1978)
15. Hassoun, S., Sasao, T. (eds.): Logic Synthesis and Verification. Kluwer Academic Publishers (2002)
16. Kambayashi, Y.: Logic design of programmable logic arrays. IEEE Trans. Computers **28**(9), 609–617 (1979)
17. Kari, L., Rozenberg, G.: The many facets of natural computing. Communications of the ACM **51**(10), 72–83 (2008)
18. Konieczny, P., Jóźwiak, L.: Minimal input support problem and algorithms to solve it. Eindhoven University of Thechnology Reaserch Report pp. 1–55 (1995)
19. Păun, G.: Computing with membranes. Journal of Computer and System Sciences **61**(1), 108–143 (2000)
20. Păun, G.: Membrane Computing: An Introduction. Natural Computing Series. Springer-Verlag GmbH (2002)
21. Sonveaux, P., Végran, F., Schroeder, T., Wergin, M.C., Verrax, J., Rabbani, Z.N., De Saedeleer, C.J., Kennedy, K.M., Diepart, C., Jordan, B.F., et al.: Targeting lactate-fueled respiration selectively kills hypoxic tumor cells in mice. The Journal of clinical investigation **118**(12), 3930–3942 (2008)