Investigating Dynamic Causalities in Reaction Systems

Roberto Barbuti[a], Roberta Gori[a], Francesca Levi[a], Paolo Milazzo[a]

[a]*Dipartimento di Informatica, Università di Pisa*
Largo Pontecorvo 3, 56127 Pisa, Italy

Abstract

Reaction systems are a qualitative formalism for modeling systems of biochemical reactions characterized by the *non-permanency* of the elements: molecules disappear if not produced by any enabled reaction. Moreover, reaction systems execute in an environment that provides new molecules at each step. Brijder, Ehrenfeucht and Rozenberg investigated *dynamic causalities* in reaction systems by introducing the idea of *predictors*. A predictor of a molecule s, for a given n, is the set of molecules to be observed in the environment in order to determine whether s is produced or not by the system at step n.

In this paper, we continue the investigation on dynamic causalities by defining an abstract interpretation framework containing three different notions of predictor: *Formula based predictors*, that is a propositional logic formula that precisely characterizes environments that lead to the production of s after n steps; *Multi-step based predictors*, that consists of n sets of molecules to be observed in the environment, one for each step; and *Set based predictors*, that are those proposed by Brijder, Ehrenfeucht and Rozenberg, and consist of a unique set of molecules to be observed in all steps.

For each kind of predictor we define an effective operator that allows predictors to be computed for any molecule s and number of steps n. The abstract interpretation framework allows us to compare the three notions of predictor in terms of precision, to relate the three defined operators and to compute minimal predictors. We also discuss a generalization of this approach that allows predictors to be defined independently of the value of n, and a tabling approach for the practical use of predictors on reaction systems models. As an application, we use predictors, generalization and tabling to give theoretical grounds to previously obtained results on a model of gene regulation.

Keywords: Reaction systems, Dynamic causalities, Abstract interpretation.

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 [1, 2]. Many of these formalisms are based on rewriting approaches. This is due to the similarity between

Email addresses: barbuti@di.unipi.it (Roberto Barbuti), gori@di.unipi.it (Roberta Gori), levifran@di.unipi.it (Francesca Levi), milazzo@di.unipi.it (Paolo Milazzo)

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.

Among the proposed formalisms based on rewriting we mention [3, 4, 5]. Each of them deals with a different peculiar aspect of the function of living cells, that is exploited as the main computing feature. Membrane systems [3, 4], for instance, exploit *maximal parallelism* in the application of rewrite rules and the hierarchical membrane structure of living cells. On the other hand, Spiking Neural P Systems [5] exploit neuron signaling mechanisms. In addition to these, a number of variants of the membrane systems formalisms have been investigated [6, 7, 8, 9, 10, 11, 12].

Reaction systems [13, 14] have been 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 called the *threshold supply* assumption and characterizes reaction systems as a *qualitative* modeling formalism.

A reaction system evolves by means of the application of its set of 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. However, since the contextual elements that can be received at each step can be any subset of the support set S, the overall system dynamics is non deterministic.

Reaction systems have been used to model various features which are useful both for the modeling of computational devices and for the modeling of biological systems. For example, binary counters [13] form the basis for the inclusion of a notion of time [15]. In [16] an extension with duration of reaction systems is presented. Theoretical aspects of reaction systems have been

studied in [17, 18, 19, 20, 21, 22, 23]. From the biological viewpoint, a model of the *lac* operon has been presented in [24].

In [18] 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. 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 knowing 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.

Following these ideas, predictors can be profitably used to decide whether s will appear or not after n steps, without executing the reaction system, by following an approach based on *tabling*. Given a reaction system and a predictor for it, a table can be constructed that contains one line for every possible sequence of contextual sets consisting only of elements of the predictor. Each line should indicate whether for such a sequence the symbol for which the predictor has been defined is actually produced after n steps or not. Then, when an observed system has to be evaluated, this should be done by considering the sequence of objects it receives from the context restricted to the elements of the predictor. The obtained sequence can then be used to access the previously constructed table and predict whether the object of interest will be produced or not.

We believe that the concept of predictor is very interesting and therefore deserves further and deeper investigation. In this paper we push forward the previous idea of a predictor by defining three different notions of predictor for Reaction Systems. The first notion is that of *formula based predictors*, namely predictors that consist in a propositional logic formula to be satisfied by contextual elements of a given reaction system. This is the most precise predictor that can be considered: satisfaction (or not) of the logic formula will discriminate the cases in which s will be produced after n steps from those in which it will not be produced.

The second notion of predictor we define is that of *multi-step based predictors*, namely predictors that consists of sets of objects to be observed in contextual elements at each step of execution of the reaction system. Predictors of this kind are more accurate than predictors as defined in [18] since they require one to consider only those contextual objects that are relevant for that particular step of the computation. Multi-step based predictors are, however, less precise than formula based predictors since there can be two sequences of contextual elements that both lead to the production

(or non-production) of the considered object but behave differently as regards the observed objects at each step.

Finally, we define *set based predictors*, a revised version of the notion of predictor as defined in [18]. These are the least precise kind of predictor since they require the same set of objects to be observed in all steps of the computation of the reaction system.

To relate the three kinds of predictor and their properties we introduce a formal framework based on the *Abstract Interpretation* theory [25, 26]. This approach allows us to introduce the notions of predictor by successive approximations and to define abstraction functions between the corresponding domains which model the loss of information (i.e. the approximation) in a formal way. As expected, set based predictors can be obtained as abstraction of multi-step based predictors, that, in turn, can be obtained as abstraction of formula based predictors. Moreover, for any kind of predictor, we define a corresponding operator that allows it to be effectively computed. We show that minimal predictors exist for a given object s and a number of steps n and we give a way to systematically compute them.

It is worth noting that the gain in precision of the newly introduced notions of predictor has a great influence on the practical application of predictors that is to determine whether an object s will appear after n steps using a tabling approach. Indeed, while the table to be constructed in the case of the original proposal of predictors by Brijder et al. contains a very high number of rows, that is exponential in the number of steps n multiplied by the cardinality of the predictor Q, our new concepts of formula and multi-step based predictors can result in tables that are significantly smaller. As an example consider the following quite simple reaction system consisting of the following reactions:

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

According to the definition of predictor given by Brijder et al., a (set based) predictor of $\mathtt{G}$ in 2 steps is the set $Q = \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}\}$. Consequently, in order to use the tabling approach it would be necessary to construct a table with $2^{6\times 2} = 4196$ rows. As we will show, by using our new notion of multi-step predictor, in this example, the number of rows for such table is drastically reduced to $2^6 = 64$. Furthermore, by using the most precise notion of predictor we propose, namely formula based predictor, such a number of rows becomes even smaller since it is equal to 6. Such a big difference in the numbers of rows is an indicator of the difference in precision (less rows means higher precision) and practical usability of the different notions of predictor.

Finally, note that the notion of predictor has a great limitation, it is related to the production of an object $s \in S$ in a fixed number of steps n. In principle, if the number of steps of interest varies, it would be necessary to compute a different predictor for each possible value of such a parameter. We claim that this limitation can be partially overcome by our formula based predictors. Indeed, they have a great advantage, that is, in many cases they can be easily *generalized* in order to obtain a parametric formula expressing predictors for arbitrary numbers of steps.

In order to illustrate the interest of the concepts introduced in the paper we consider a rather complex biological example, the *lac* operon expression in the *E. coli* bacterium a model presented in [24]. We apply the three kinds of predictor (and the generalization technique) to give a for-

mal ground to the conclusion reached in [24], and we show how the generalization and tabling approaches can be applied to this example of biological system.

The paper is organized as follows. Section 2 contains the background material on Abstract Interpretation and Reaction Systems. Section 3 presents some formal concepts that are used in all the definitions of predictors. The new notion of formula based predictor and all the related properties can be found in Section 4. In Section 5 we propose the new notion of multi-step based predictor and we prove all its properties. Section 6 contains the notion of set based predictor (a revised version of the notion of predictor proposed in [18]). In Section 7 we discuss two aspects of the practical use of predictors on reaction systems. More specifically, Section 7.1 considers the generalization of predictors while Section 7.2 illustrates the tabling approach. Section 8 presents the application of our approach to the biological system presented in [24]. Finally, our conclusions can be found in Section 9.

For the sake of readability, all proofs have been moved to the Appendix.

2. Background

2.1. Abstract Interpretation

Abstract Interpretation is a well-known theory [25, 26] which has been proposed in order to design abstract semantics and static analysis and to formally reason about the concept of approximation.

The abstract interpretation theory has been successfully used for formally comparing different semantics of program languages, systematically define new semantics at different level of abstraction and construct hierarchy where well known semantics were related with each other by means of abstract and concretization functions.

In order to formally reason about the approximation relation between elements of different domains, the guiding idea is that of approximating properties on the exact (concrete) domain with elements of an approximate (abstract) domain. That approximation relation can be formalized by introducing a pair of functions, the abstraction α and the concretization γ, which provide a Galois connection between the concrete and the abstract domains.

Let $(C, \leq_C)$ be a partially ordered domain and $(A, \leq_A)$ be an abstract partially ordered domain, the abstraction α is a mapping associating an element $\alpha(c)$ of $(A, \leq_A)$ to each concrete element c of $(C, \leq_C)$. Since in this theory the partial order specifies the precision degree of any abstract object, we have that if $\alpha(c) \leq_A a$, then a is also a correct approximation, although in general less precise, abstract approximation of c. More formally we have the following definition.

Definition 2.1 (Galois Insertion)**.** *Let $(C, \leq_C)$ and $(A, \leq_A)$ be two posets. A* Galois Connection *is a pair of maps $\alpha : C \to A$ and $\gamma : A \to C$ such that*

1. *α and γ are monotone, that is $c_1 \leq_C c_2$ implies $\alpha(c_1) \leq_A \alpha(c_2)$, and $a_1 \leq_A a_2$ implies $\gamma(a_1) \leq_C \gamma(a_2)$,*
2. *for each $c \in C$, $c \leq_C \gamma(\alpha(c))$ and*
3. *for each $a \in A$, $\alpha(\gamma(a)) \leq_A a$.*

A Galois Insertion *is a Galois connection where, γ is injective; or equivalently for each $a \in A$, $\alpha(\gamma(a)) = a$.*

Let f be an operator on the concrete domain $(C, \leq_C)$, the corresponding abstract operator on the abstract domain $(A, \leq_A)$, can be systematically obtained using the Galois connection.

Definition 2.2. *Let (α, γ) be a Galois connection of $(C, \leq_C)$ into $(A, \leq_A)$ and f be a function $D \to C$ for a given set D. A monotone abstract function $\overline{f} : D \to A$ is a* correct approximation *of f if, for each $d \in D$,*

$$\alpha(f(d)) \leq_A \overline{f}(d).$$

Moreover, $\overline{f} : D \to A$ is an optimal (most precise) correct approximation *of f if, for each $d \in D$,*

$$\alpha(f(d)) = \overline{f}(d).$$

Moreover, abstract interpretation can be formalized in a hierarchical framework. In fact, given the Galois connections (or insertions) (α_1, γ_1) between $(C, \leq_C)$ and $(A, \leq_A)$, and (α_2, γ_2) between $(A, \leq_A)$ and $(B, \leq_B)$ it can be shown that (α, γ) is a Galois connection (or insertion) between $(C, \leq_C)$ and $(B, \leq_B)$, where $\alpha := \alpha_2 \circ \alpha_1$ and $\gamma := \gamma_1 \circ \gamma_2$. Therefore abstract interpretations can be designed by successive approximations.

2.2. Reaction Systems

In this section we recall the basic definition of reaction systems [13, 14]. Let S be a finite set of symbols, called objects (or entities). A *reaction* is formally a triple (R, I, P) with $R, I, P \subseteq S$, composed of *reactants* R, *inhibitors* I, and *products* P. We assume reactants and inhibitors to be disjoint ($R \cap I = \emptyset$), otherwise the reaction would never be applicable. The reactants and inhibitors $R \cup I$ of a reaction are collectively called the *resources* of this reaction. The set of all possible reactions over a set S is denoted by $\mathrm{rac}(S)$. Finally, a *reaction system* is a pair $\mathcal{A} = (S, A)$, with S being a finite background set, and $A \subseteq \mathrm{rac}(S)$ being its set of reactions.

The state of a reaction system is described by a set of objects. Regarding the dynamics of a reaction system, reaction systems are based on the "threshold supply" assumption which, in words, means that if an object is present in the state then it is assumed to be available in any arbitrary number of copies as needed. In fact, reaction systems can be seen as a *qualitative* model, 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 application of multiple reactions is cumulative. Formally, given a reaction system $\mathcal{A} = (S, A)$, the result of application of $\mathcal{A}$ to a set $T \subseteq S$ is defined as $\mathrm{res}_{\mathcal{A}}(T) = \mathrm{res}_A(T) = \bigcup_{a \in A} \mathrm{res}_a(T)$.

The dynamics of a reaction system is driven by the *contextual* objects, namely the objects which are supplied to the system by the external environment at each step. An important characteristic of reaction systems, which distinguishes them from most other biologically-inspired computational models, is the assumption about the *non-permanency* of objects. Under such an assumption the objects carried over to the next step are only those objects which are produced by reactions. On the other hand, all the other objects vanish, even if they are not "used" by any reaction, i.e. unused objects are never carried over to the next step.

Formally, the dynamics of a reaction system $\mathcal{A} = (S, A)$ is defined as an *interactive process* $\pi = (\gamma, \delta)$, with γ and δ being finite sequences of sets of objects called the *context sequence* and the *result sequence*, respectively. The sequences are of the form $\gamma = C_0, C_1, \ldots, C_n$ and $\delta = D_0, D_1, \ldots, D_n$ for some $n \geq 1$, with $C_i, D_i \subseteq S$, and $D_0 = \emptyset$. Each set 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 D_{i-1} and the objects C_{i-1} from the context; formally $D_i = res_{\mathcal{A}}(C_{i-1} \cup D_{i-1})$ 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 $W_i = C_i \cup D_i$ for all $1 \leq i \leq n$. In the following we say that $\gamma = C_0, C_1, \ldots, C_n$ is a n-step context sequence.

Reaction systems can be simplified by using the method proposed in [24]. *Simplification* of reaction systems is based on a simplification method for boolean expressions, and allows the number of reactions to be potentially reduced by obtaining a *functionally equivalent* [13] reaction system.

Given a reaction system, a *predictor* [18] of an object $s \in S$ after n steps is a set of objects $Q \subseteq S$ that includes all of those that are essential to determine whether s is produced or not at step n of the execution of the system. More formally, $Q \subseteq S$ is a predictor of s at step n if for any two n-steps interactive processes $\pi_1 = (\gamma_1, \delta_1)$ and $\pi_2 = (\gamma_2, \delta_2)$ the following holds: if the projections on Q of γ_1 and γ_2 are equal[1], then either s is in both in the n-th sets of δ_1 and δ_2, or in none of them. In [18] it is shown that for every reaction system, object $s \in S$ and value $n \in \mathbb{N}$, there exists a unique minimal (w.r.t. set inclusion) predictor for s in n steps.

3. Preliminaries

We start with defining some notions that will be used in all of the three definitions of predictor we will give. We adopt propositional logic formulas parametric with respect to propositional symbols.

Definition 3.1 (Logic Formulas)**.** *Let $\mathcal{P}$ be a set of propositional symbols. The set $F_{\mathcal{P}}$ of* propositional logic formulas *on $\mathcal{P}$ is inductively defined as follows:*

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

In the following $\equiv_l$ stands for the logical equivalence on propositional formulas $F_{\mathcal{P}}$. Moreover, given a formula $f \in F_{\mathcal{P}}$ we use $atom(f)$ to denote the set of propositional symbols that appear in f and $simpl(f)$ to denote the simplified version of f. The simplified version of a formula is obtained by applying the standard formula simplification procedure of propositional logic converting a formula to Conjunctive Normal Form. We recall that for any formula $f \in F_{\mathcal{P}}$ the simplified formula

[1]The projection of $\gamma = C_0, C_1, \ldots, C_n$ on a set of objects Q is defined as $\gamma' = C_0 \cap Q, C_1 \cap Q, \ldots, C_n \cap Q$.

$simpl(f)$ is equivalent to f, it is minimal with respect to the propositional symbols and unique up to commutativity. Thus, we have $f \equiv_l simpl(f)$ and $atom(simpl(f)) \subseteq atom(f)$ and there exists no formula f' such that $f' \equiv_l f$ and $atom(f') \subset atom(simpl(f))$.

In order to describe the causes of a given object 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 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 3.2. *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:*

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

Given a reaction system, we can now 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 3.3. *Let $\mathcal{A} = (S, A)$ be a r.s. and $s \in S$. The* causal predicate *of s in $\mathcal{A}$, denoted by $cause(s, \mathcal{A})$ (or $cause(s)$, when $\mathcal{A}$ is clear from the context), is defined as follows.*

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

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

We introduce simple two reaction systems as running examples.

Example 3.1. *Let $\mathcal{A}_1 = (\{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}, \mathtt{G}, \mathtt{H}\}, \{a_1, a_2, a_3, a_4, a_5\})$ be a reaction system with reaction rules*

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

The applicability predicates *of the reactions are*

$$\begin{array}{lll} ap(a_1) = \mathtt{A} & ap(a_2) = \mathtt{C} & ap(a_3) = \mathtt{E} \\ ap(a_4) = \mathtt{B} \wedge \mathtt{D} & ap(a_5) = \mathtt{F} \wedge \neg \mathtt{B}. & \end{array}$$

Thus, the causal predicates *of the objects are*

$$\begin{array}{lll} cause(\mathtt{A}) = false & cause(\mathtt{B}) = \mathtt{A} & cause(\mathtt{C}) = false \\ cause(\mathtt{D}) = \mathtt{C} & cause(\mathtt{E}) = false & cause(\mathtt{F}) = \mathtt{E} \\ cause(\mathtt{H}) = \mathtt{B} \wedge \mathtt{D} & cause(\mathtt{G}) = (\mathtt{F} \wedge \neg \mathtt{B}) \vee (\mathtt{B} \wedge \mathtt{D}). & \end{array}$$

Note that $cause(\mathtt{A}) = false$ given that $\mathtt{A}$ can not be produced by any reaction. An analogous reasoning holds for objects $\mathtt{C}$ and $\mathtt{E}$.

Example 3.2. *Let* $\mathcal{A}_2 = (\{\mathtt{I}, \mathtt{L}, \mathtt{M}, \mathtt{N}, \mathtt{O}\}, \{a_6, a_7, a_8, a_9\})$ *be a reaction system with reaction rules*

$$a_6 = (\{\mathtt{I}\}, \{\}, \{\mathtt{L}\}) \quad a_7 = (\{\mathtt{M}\}, \{\}, \{\mathtt{L}\})$$
$$a_8 = (\{\mathtt{N}\}, \{\mathtt{O}\}, \{\mathtt{I}\}) \quad a_9 = (\{\mathtt{N}\}, \{\}, \{\mathtt{M}\}).$$

In this case the applicability predicates *of the reactions are*

$$ap(a_6) = \mathtt{I} \qquad ap(a_7) = \mathtt{M}$$
$$ap(a_8) = \mathtt{N} \wedge \neg\mathtt{O} \quad ap(a_9) = \mathtt{N}.$$

Thus, the corresponding causal predicates *are*

$$cause(\mathtt{I}) = \mathtt{N} \wedge \neg\mathtt{O} \quad cause(\mathtt{L}) = \mathtt{I} \vee \mathtt{M} \quad cause(\mathtt{M}) = \mathtt{N}$$
$$cause(\mathtt{N}) = false \quad cause(\mathtt{O}) = false.$$

Given a set of objects S, we consider a corresponding set of *labelled objects* $S \times \mathbb{N}$. Labelled objects will be used to relate objects with execution steps of a reaction systems. For example, we may use the labelled object $(s, 3)$ to denote the presence of object s in the element C_3 of a context sequence (that is the context of the reaction system at the third step of execution). For the sake of legibility, we denote $(s, i) \in S \times \mathbb{N}$ simply as s_i and we introduce $S^n = \bigcup_{i=0}^{n} S_i$ where $S_i = \{s_i \mid s \in S\}$.

4. Formula Based Predictors

We introduce *formula based predictors*, namely predictors that consist in a propositional logic formula to be satisfied by context sequences of a given reaction system. We prove that minimal formula based predictors exist with respect to an *approximation order* on formulas. Finally, we define an operator to calculate in a systematic way a minimal formula based predictor for a given object s at step $n + 1$.

Labelled objects will be used in propositional logic formulas describing properties of context sequences. Specifically, a logic formula on labelled objects S^n describes the properties of n-step context sequences. 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 context sequence $\gamma = C_0, C_1, \ldots C_n$. This is formalized by the following definition that relates context sequences and propositional formulas.

Definition 4.1. *Let* $\gamma = C_0, C_1, \ldots C_n$ *a context sequence and* $f \in F_{S^n}$ *a propositional logic formula. We say that* γ satisfies f *if and only if* $\gamma \models f$ *where the satisfaction relation* $\models$ *is inductively defined as follows:*

- $\gamma \models true$*;*
- $\gamma \models s_i$ *iff* $s \in C_i$*;*
- $\gamma \models (f')$ *iff* $\gamma \models f'$*;*
- $\gamma \models \neg f'$ *iff* $\gamma \not\models f'$ *(as usual,* $\gamma \not\models f$ *holds if it is not the case that* $\gamma \models f$*);*

- $\gamma \models f_1 \vee f_2$ *iff* $\gamma \models f_1$ *or* $\gamma \models f_2$*;*
- $\gamma \models f_1 \wedge f_2$ *iff* $\gamma \models f_1$ *and* $\gamma \models f_2$.

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

Based on the previous notion, we introduce an *equivalence relation* between context sequences.

Definition 4.2. *Let γ_1 and γ_2 be n-step context sequences and $f \in F_{S^n}$ a propositional logic formula. We say that $\gamma_1 \approx_f^n \gamma_2$ iff*

$$\gamma_1 \models f \Leftrightarrow \gamma_2 \models f.$$

The equivalence relation $\approx_f^n$ equates n-step context sequences that have the same behavior with respect to formula f (*either* both satisfy it, *or* both do not satisfy it). This equivalence relation is useful to define the notion of *formula based predictor* for reaction systems: a formula f is a predictor for object s in $n+1$ steps if two context sequence equated by $\approx_f^n$ will *either* both lead *or* both not lead to the production of object s in $n+1$ steps.

Definition 4.3 (Formula based Predictor)**.** *Let $\mathcal{A} = (S, A)$ be a reaction system, $s \in S$ and $f \in F_{S^n}$ a propositional logic formula. We say that f* f-predicts *s* in $n+1$ steps *if for all n-step context sequences $\gamma_1 = C_0^1, \ldots, C_n^1$ and $\gamma_2 = C_0^2, \ldots, C_n^2$,*

$$\gamma_1 \approx_f^n \gamma_2 \quad \Longleftrightarrow \quad \left(s \in D_{n+1}^1 \iff s \in D_{n+1}^2\right)$$

where $\delta_1 = D_0^1, \ldots, D_n^1$ and $\delta_2 = D_0^2, \ldots, D_n^2$ are the result sequences corresponding to γ_1 and γ_2 and where $D_{n+1}^1 = res_{\mathcal{A}}(C_n^1 \cup D_n^1)$ and $D_{n+1}^2 = res_{\mathcal{A}}(C_n^2 \cup D_n^2)$.

It is worth noting that the definition of formula based predictor requires not only that $\gamma_1 \approx_f^n \gamma_2$ implies $s \in D_{n+1}^1 \iff s \in D_{n+1}^2$, but also that the inverse implication holds. This is necessary to avoid that the predictor of s in the particular cases in which it is either always or never produced, is any formula. As a consequence of this definition, in order for f to be a predictor it has to partition the set of context sequences by separating those that lead to the production of s from those that do not. Hence, in the particular cases considered above, only formulas that are always equivalent either to true or to false can be predictors. Apart from these particular cases, the $\Leftarrow$ implication is a consequence of the $\Rightarrow$ implication and the definition of $\approx_f^n$.

Note that if a formula f *f-predicts object s in $n+1$ steps* the equivalence relation $\approx_f^n$ induces a partition of all context sequences in two sets: on the one hand, the set of context sequences that ensure that s will appear after $n+1$ steps; on the other hand the set of context sequences that ensure that s will not appear in $n+1$ steps. Since by Definition 4.2 it holds $\gamma_1 \approx_f^n \gamma_2$ iff $\gamma_1 \approx_{\neg f}^n \gamma_2$, if f f-predicts s in $n+1$ steps then also $\neg f$ f-predicts s in $n+1$ steps. Moreover, for the same reason any other formula f' that is equivalent either to f or to $\neg f$ f-predicts s in $n+1$ steps. These properties are formally stated by the following theorems.

Theorem 4.1. *Let $\mathcal{A} = (S, A)$ be a reaction system. If two formulas $f, f' \in F_{S^n}$* f-predict *$s \in S$ in $n+1$ steps, then either $f \equiv_l f'$ or $f \equiv_l \neg f'$.*

Theorem 4.2. *Let $\mathcal{A} = (S, A)$ be a reaction system and $f \in F_{S^n}$ be a formula such that f* f-predicts *$s \in S$ in $n+1$ steps. Then any formula $f' \in F_{S^n}$ such that either $f' \equiv_l f$ or $f' \equiv_l \neg f$* f-predict *s in $n+1$ steps.*

We introduce an *approximation order* $\sqsubseteq_f$ on formula based predictors that allows us to compare two formula based predictors in terms of precision. The approximation ordering on formulas is fundamental for reasoning on the minimality of formula based predictors and for supporting the definition of the *Abstract Interpretation* framework (that will be presented in Sections 5 and 6).

Definition 4.4 (Approximation Order)**.** *Given $f_1, f_2 \in F_{S^n}$ we say that*

1. *$f_1 \sqsubseteq_f f_2$ if and only if: (i) either $f_1 \equiv_l f_2$ or $f_1 \equiv_l \neg f_2$; and (ii) $atom(f_1) \subseteq atom(f_2)$;*
2. *$f_1 \equiv_f f_2$ if and only if $f_1 \sqsubseteq_f f_2$ and $f_2 \sqsubseteq_f f_1$.*

It is worth noting that, as a consequence of the previous definition, we have that $f_1 \sqsubset_f f_2$, namely $f_1 \sqsubseteq_f f_2$ and $f_1 \not\equiv_f f_2$, if and only if either (i) $f_1 \equiv_l f_2$ or $f_1 \equiv_l \neg f_2$ and (ii) $atom(f_1) \subset atom(f_2)$. The definition of the approximation order $\sqsubseteq_f$ on f-predictors is based on the following intuition. Condition (i) of point 1 relies on the properties of f-predictors stated by Theorems 4.1 and 4.2. By contrast, condition (ii) of point 1 is motivated by the definition of the abstraction of formula based predictors, that will be presented in Section 5. In such a context the abstraction depends on the labelled symbols that appear in the formula based predictor. As a consequence, two formula based predictors f_1 and f_2 that are logically equivalent may lead to different abstractions.

The following result shows that there exists a *unique equivalence class* of formula based predictors that is minimal with respect to the previously introduced approximation order $\sqsubseteq_f$. An equivalence class of minimal formulas with respect to $\sqsubseteq_f$ can be obtained by applying a standard formula simplification procedure. It should be clear that for each logic formula f it holds that $simpl(f) \sqsubseteq_f f$.

Theorem 4.3. *Let $\mathcal{A} = (S, A)$ be a reaction system $s \in S$ and let $f_1, f_2 \in F_{S^n}$ be two propositional logic formulas. If both f_1 and f_2 f-*predict *s in $n+1$ steps, then there exists an $f \in F_{S^n}$ such that $f \sqsubseteq_f f_1$, $f \sqsubseteq_f f_2$ and f f-*predicts *s in $n+1$ steps.*

To better illustrate the notion of formula based predictor we consider the reaction systems of Examples 3.1 and 3.2.

Example 4.1. *Let us consider the reaction system $\mathcal{A}_1 = (\{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}, \mathtt{G}, \mathtt{H}\}, \{a_1, a_2, a_3, a_4, a_5\})$ of Example 3.1 with reaction rules,*

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

We focus on the production of object G *in two steps. The following logic formulas* f-predict G in two steps

$$f_1 = ((\mathtt{F}_1 \vee \mathtt{E}_0) \wedge \neg(\mathtt{B}_1 \vee \mathtt{A}_0)) \vee ((\mathtt{B}_1 \vee \mathtt{A}_0) \wedge (\mathtt{D}_1 \vee \mathtt{C}_0)),$$

$$f_2 = ((\mathtt{F}_1 \vee \mathtt{E}_0) \wedge \neg\mathtt{B}_1 \wedge \neg\mathtt{A}_0) \vee ((\mathtt{B}_1 \vee \mathtt{A}_0) \wedge (\mathtt{D}_1 \vee \mathtt{C}_0)).$$

Note that the formulas f_1 and f_2 are logically equivalent and express the properties of the context sequences leading to the production of object $\mathtt{G}$ *after two steps. Moreover, in this case the formulas f_1 and f_2 are also equivalent with respect to $\equiv_f$ given that they contain exactly the same set of labelled symbols. Thus both formula define a minimal formula based predictor for* $\mathtt{G}$ *in two steps.*

Example 4.2. *Let us consider the reaction system $\mathcal{A}_2 = (\{\mathtt{I}, \mathtt{L}, \mathtt{M}, \mathtt{N}, \mathtt{O}\}, \{a_6, a_7, a_8, a_9\})$ of Example 3.2 with reaction rules,*

$$a_6 = (\{\mathtt{I}\}, \{\}, \{\mathtt{L}\}) \quad a_7 = (\{\mathtt{M}\}, \{\}, \{\mathtt{L}\})$$
$$a_8 = (\{\mathtt{N}\}, \{\mathtt{O}\}, \{\mathtt{I}\}) \quad a_9 = (\{\mathtt{N}\}, \{\}, \{\mathtt{M}\}).$$

We focus on the production of object $\mathtt{L}$ *in two steps. The following logic formulas* f-predict $\mathtt{L}$ in two steps

$$f_3 = ((\mathtt{I}_1 \vee (\mathtt{N}_0 \wedge \neg(\mathtt{O}_0))) \vee (\mathtt{M}_1 \vee \mathtt{N}_0)),$$

$$f_4 = (\mathtt{I}_1 \vee \mathtt{N}_0 \vee \mathtt{M}_1).$$

Note that similarly as in the previous case the formulas f_3 and f_4 are logically equivalent and express the properties of the context sequences leading to the production of object $\mathtt{L}$ *after two steps. However, in this case the formulas f_3 and f_4 are not equivalent with respect to $\equiv_f$ given that $f_4 \sqsubset_f f_3$. In particular, the formula f_4 is the simplified version of f_3 (that is $simpl(f_3) = f_4$) and is obtained removing the condition on labelled object* $\mathtt{O}_0$*. Therefore the minimal formula based predictor for* $\mathtt{L}$ *in two steps is represented by a formula which does not contain any condition on* $\mathtt{O}_0$.

We now define an operator $\mathtt{fbp}$ that allows formula based predictors to be effectively computed. The formula based predictor for an object s in $n+1$ steps is obtained computing a formula on labelled objects S^n that is satisfied by a context sequence if and only if the object s will appear in the system after $n+1$ steps.

Definition 4.5 (Formula based Predictor Operator)**.** *Let $\mathcal{A} = (S, A)$ be a reaction system and $s \in S$. We define a function $\mathtt{fbp} : S \times \mathbb{N} \to F_{S^n}$ as follows:*

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

where we adopt the auxiliary function $\mathtt{fbs} : F_S \times \mathbb{N} \to F_{S^n}$ with $\mathtt{fbs}(f, i)$ recursively defined as follows:

$$\begin{aligned}
\mathtt{fbs}(s, 0) &= s_0 \\
\mathtt{fbs}(s, i) &= s_i \vee \mathtt{fbs}(cause(s), i-1) \ \ \textit{if } i > 0 \\
\mathtt{fbs}(f_1 \wedge f_2, i) &= \mathtt{fbs}(f_1, i) \wedge \mathtt{fbs}(f_2, i) \\
\mathtt{fbs}(f_1 \vee f_2, i) &= \mathtt{fbs}(f_1, i) \vee \mathtt{fbs}(f_2, i) \\
\mathtt{fbs}(\neg f', i) &= \neg \mathtt{fbs}(f', i) \\
\mathtt{fbs}((f'), i) &= (\mathtt{fbs}(f', i)) \\
\mathtt{fbs}(true, i) &= true \\
\mathtt{fbs}(false, i) &= false
\end{aligned}$$

Note that the definition of fbs for the case (s, i) could introduce some $s_i \vee false$ sub terms in the formula when $cause(s) = false$. For the sake of readability, we will omit such disjunctions with $false$ in the examples in this paper.

The following theorem illustrates the main property of the function fbp presented in the previous definition. Any n-step context sequence satisfies the formula $\mathtt{fbp}(s, n)$ if and only if the object s will appear in the system after $n + 1$-steps.

Theorem 4.4. *Let $\mathcal{A} = (S, A)$ be a reaction system and $s \in S$. For any n-step context sequence $\gamma = C_0, C_1, \ldots, C_n$ it holds:*

$$s \in D_{n+1} \iff \gamma \models \mathtt{fbp}(s, n)$$

where $\delta = D_0, D_1, \ldots, D_n$ is the result sequence corresponding to γ and $D_{n+1} = \mathrm{res}_{\mathcal{A}}(C_n \cup D_n)$.

Lemma 4.5. *Let $\mathcal{A} = (S, A)$ be a reaction system, $s \in S$ and $f \in F_{S^n}$ be a propositional logical formula. If for all n-step context sequences $\gamma = C_0, C_1, \ldots C_n$ it holds $(s \in D_{n+1} \iff \gamma \models f)$ then f f-*predicts *s in $n + 1$ steps, where $D_{n+1} = res_{\mathcal{A}}(C_n \cup D_n)$.*

As a consequence of Theorem 4.4 and Lemma 4.5, we can conclude that the formula $\mathtt{fbp}(s, n)$ *f-predicts* s in $n + 1$ steps.

Corollary 4.6. *Let $\mathcal{A} = (S, A)$ be a reaction system. For any object $s \in S$, the formula* $\mathtt{fbp}(s, n)$ f-predicts *s in $n + 1$ steps.*

In the general case the function fbp may give a formula based predictor that it is not minimal with respect to approximation order $\sqsubseteq_f$. Therefore, the calculation of a minimal formula based predictor requires the application of a standard simplification procedure to the obtained logic formula.

Corollary 4.7. *Let $\mathcal{A} = (S, A)$ be a reaction system. For any object $s \in S$ the formula $simpl(\mathtt{fbp}(s, n))$ is a* minimal *(w.r.t. the $\sqsubseteq_f$ order on F_{S^n}) formula that* f-predicts *s in $n + 1$ steps.*

We illustrate the application of the operator fbp considering the reaction systems presented in the Examples 3.1 and 3.2.

Example 4.3. *Let us consider the reaction system $\mathcal{A}_1 = (\{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}, \mathtt{G}, \mathtt{H}\}, \{a_1, a_2, a_3, a_4, a_5\})$ of Example 3.1 with reaction rules,*

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

In order to calculate the logic formula that f-predicts G in two steps *we apply the function* fbp *as follows,*

$$\begin{aligned} \mathtt{fbp}(G, 1) &= \mathtt{fbs}((\mathtt{F} \wedge \neg \mathtt{B}) \vee (\mathtt{B} \wedge \mathtt{D}), 1) \\ &= (\mathtt{fbs}(\mathtt{F}, 1) \wedge \neg \mathtt{fbs}(\mathtt{B}, 1)) \vee (\mathtt{fbs}(\mathtt{B}, 1) \wedge \mathtt{fbs}(\mathtt{D}, 1)) \\ &= ((\mathtt{F}_1 \vee \mathtt{fbs}(\mathtt{E}, 0)) \wedge \neg(\mathtt{B}_1 \vee \mathtt{fbs}(\mathtt{A}, 0)) \vee (\mathtt{B}_1 \vee \mathtt{fbs}(\mathtt{A}, 0) \wedge (\mathtt{D}_1 \vee \mathtt{fbs}(\mathtt{C}, 0)) \\ &= ((\mathtt{F}_1 \vee \mathtt{E}_0) \wedge \neg(\mathtt{B}_1 \vee \mathtt{A}_0)) \vee (\mathtt{B}_1 \vee \mathtt{A}_0) \wedge (\mathtt{D}_1 \vee \mathtt{C}_0)) = f_1 \end{aligned}$$

According to Theorem 4.4 a context sequence satisfies the formula $\texttt{fbp}(\texttt{G}, 1)$ *if and only if the execution of the reaction system leads to the production of object* G *after two execution steps. Furthermore, in this case the obtained formula is also minimal with respect to the approximation order* $\sqsubseteq_f$*, given that it coincides with the formula* f_1 *as discussed in Example 4.1.*

Example 4.4. *Let us consider the reaction system* $\mathcal{A}_2 = (\{\texttt{I}, \texttt{L}, \texttt{M}, \texttt{N}, \texttt{O}\}, \{a_6, a_7, a_8, a_9\})$ *of Example 3.2 with reaction rules,*

$$a_6 = (\{\texttt{I}\}, \{\}, \{\texttt{L}\}) \quad a_7 = (\{\texttt{M}\}, \{\}, \{\texttt{L}\})$$
$$a_8 = (\{\texttt{N}\}, \{\texttt{O}\}, \{\texttt{I}\}) \quad a_9 = (\{\texttt{N}\}, \{\}, \{\texttt{M}\}).$$

In order to calculate the logic formula that f-predicts L in two steps *we apply the function* fbp *as follows*

$$\begin{aligned}
\texttt{fbp}(\texttt{L}, 1) &= \texttt{fbs}(\texttt{I} \vee \texttt{M}, 1) \\
&= \texttt{fbs}(\texttt{I}, 1) \vee \texttt{fbs}(\texttt{M}, 1) \\
&= ((\texttt{I}_1 \vee \texttt{fbs}(\texttt{N} \wedge \neg\texttt{O}, 0)) \vee (\texttt{M}_1 \vee \texttt{fbs}(\texttt{N}, 0))) \\
&= ((\texttt{I}_1 \vee (\texttt{N}_0 \wedge \neg(\texttt{O}_0))) \vee (\texttt{M}_1 \vee \texttt{N}_0)) = f_3
\end{aligned}$$

In this case, differently from the previous one, the formula $\texttt{fbp}(\texttt{L}, 1)$ *f-predicts* L *in two steps but it not minimal with respect to the approximation order* $\sqsubseteq_f$*. Indeed, the operator reports the formula* f_3 *discussed in Example 4.2.*

Thus, in this case, we have to apply the logical simplification to the obtained formula $\texttt{fbp}(\texttt{L}, 1)$*, obtaining*

$$simpl(\texttt{fbp}(\texttt{L}, 1)) = simpl(f_3) = (\texttt{I}_1 \vee \texttt{N}_0 \vee \texttt{M}_1) = f_4$$

The formula f_4 *is a minimal f-predictor for* L *in two steps. It is worth noting that the reaction system* $\mathcal{A}_2$ *constitutes a simplified reaction system according to definitions in [24]. Nonetheless, the computation of a minimal formula based predictor requires to apply the standard simplification procedure simpl.*

5. Multi-step Based Predictors

In this section we define *multi-step based predictors*, namely predictors based on sets of objects to be observed in the context of a reaction system at each execution step. Multi-step based predictors are less precise than formula based predictors since there can be two context sequences that both lead to the production (or non-production) of the considered object but behave differently as regards the observed objects at each step. To relate formula based predictors with multi-step based predictors we use an Abstract Interpretation approach introducing a Galois Insertion between the corresponding domains. In this framework we show that multi-step based predictors can be obtained in a systematic way as abstraction of formula based predictors. Finally, we introduce an operator mbp to effectively compute multi-step based predictors, defined as an abstract version of the operator fbp for formula based predictors (presented in Definition 4.5). We compare in terms of precision the abstract operator mbp with the abstraction of the corresponding operator fbp.

A multi-step based predictor is represented as a set of labelled objects in which labels represent execution steps. For example, the set of labelled objects $\widehat{Q} = \{\mathtt{A}_1, \mathtt{B}_1, \mathtt{C}_2\} \subseteq S^2$ describes the properties of 2-step context sequences and says that: in the initial context there is no object to be observed whereas the objects A and B have to be observed at the first execution step and the object C has to be observed in the second step.

Observing a labelled object s_i in the context sequence of a reaction system means checking whether the object s is present or not in the context sequence at the position i. Thus, two n-step context sequences are equivalent with respect to a set of labelled objects $\widehat{Q} \subseteq S^n$ if and only if every observed object in $\widehat{Q}$ is *either* present *or* absent in both the context sequences. This is formalized by the following definition of *equivalence relation* on context sequences.

Definition 5.1. *Let $\gamma_1 = C_0^1, C_1^1, \ldots C_n^1$ and $\gamma_2 = C_0^2, C_1^2, \ldots C_n^2$ be two n-step context sequence. Given a set of labelled objects $\widehat{Q} \subseteq S^n$ we say that $\gamma_1 \simeq^n_{\widehat{Q}} \gamma_2$ iff for each $i \in \{0, .., n\}$,*

$$(C_i^1 \cap \{s \mid s_i \in \widehat{Q}\}) = (C_i^2 \cap \{s \mid s_i \in \widehat{Q}\}).$$

A set of labelled objects $\widehat{Q}$ is a *multi-step based predictor* for object s in $n+1$ steps if any two context sequences that are equated by $\simeq^n_{\widehat{Q}}$ *either* both lead *or* both not lead to the production of object s in $n+1$ steps.

Definition 5.2 (Multi-step based Predictor)**.** *Let $\mathcal{A} = (S, A)$ be a reaction system, $s \in S$ and $\widehat{Q} \subseteq S^n$. We say that $\widehat{Q}$* m-predicts *s in $n+1$* steps *if for all n-step context sequences $\gamma_1 = C_0^1, C_1^1, \ldots, C_n^1$ and $\gamma_2 = C_0^2, C_1^2, \ldots, C_n^2$ such that $\gamma_1 \simeq^n_{\widehat{Q}} \gamma_2$ we have that*

$$s \in D_{n+1}^1 \iff s \in D_{n+1}^2$$

where $\delta_1 = D_0^1, D_1^1, \ldots, D_n^1$ and $\delta_2 = D_0^2, D_1^2, \ldots, D_n^2$ are the result sequence corresponding to γ_1 and γ_2 and where $D_{n+1}^1 = res_{\mathcal{A}}(C_n^1 \cup D_n^1)$ and $D_{n+1}^2 = res_{\mathcal{A}}(C_n^2 \cup D_n^2)$.

Similarly as in the case of formula based predictors there exists a *minimal multi-step based predictor* for a given object s at step $n+1$. Given that multi step predictors are sets they can be naturally ordered by means of set inclusion.

Theorem 5.1. *Let $\mathcal{A} = (S, A)$ be a reaction system and $s \in S$. There exists exactly one minimal (w.r.t. the set inclusion order) $\widehat{Q} \subseteq S^n$ that* m-predicts *s in $n+1$ steps.*

We illustrate the notion of multi-step based predictor revisiting the reaction systems of Examples 3.1 and 3.2.

Example 5.1. *Let $\mathcal{A}_1 = (\{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}, \mathtt{G}, \mathtt{H}\}, \{a_1, a_2, a_3, a_4, a_5\})$ be the reaction system of Examples 3.1 with reaction rules*

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

Similarly as in Examples 4.1 and 4.3 we consider again the production of object $\mathtt{G}$ *in two steps. The minimal multi-step based predictor of* $\mathtt{G}$ *in two steps is given by the following set of labelled objects*

$$\widehat{Q}_1 = \{\mathtt{A}_0, \mathtt{C}_0, \mathtt{E}_0, \mathtt{B}_1, \mathtt{D}_1, \mathtt{F}_1\}.$$

It should be easy to see that we cannot find a smaller (w.r.t. set inclusion) m-predictor of $\mathtt{G}$ *in two steps. Assume, for example, that we remove* $\mathtt{B}_1$ *from* $\widehat{Q}_1$ *considering* $\widehat{Q}'_1 = \widehat{Q}_1 \setminus \{\mathtt{B}_1\}$*. Moreover, consider the context sequences* $\gamma_1 = C_0^1, C_1^1$ *with* $C_0^1 = \{\mathtt{C}\}$ *and* $C_1^1 = \{\mathtt{B}\}$*, and* $\gamma_2 = C_0^2`, C_1^2$ *with* $C_0^2 = C_0^1$ *and* $C_1^2 = \emptyset$*. We have that* γ_1 *and* γ_2 *are equivalent with respect to* $\widehat{Q}'_1$*, that is* $\gamma_1 \simeq^1_{\widehat{Q}'_1} \gamma_2$*. However, the execution of the reaction system with respect to context* γ_1 *produces* $\mathtt{G}$ *after two steps (*$\mathtt{G} \in D_2^1$*) while the execution with respect to context* γ_2 *does not produces* $\mathtt{G}$ *after two steps (*$\mathtt{G} \notin D_2^2$*). This shows that the set of labelled objects* $\widehat{Q}'_1$ *does not m-predict* $\mathtt{G}$ *in two steps. Similar arguments can be applied to show that any other labelled objects cannot be removed from* $\widehat{Q}_1$*.*

Example 5.2. *Let* $\mathcal{A}_2 = (\{\mathtt{I}, \mathtt{L}, \mathtt{M}, \mathtt{N}, \mathtt{O}\}, \{a_6, a_7, a_8, a_9\})$ *be a reaction system of Example 3.2 with reaction rules*

$$a_6 = (\{\mathtt{I}\}, \{\}, \{\mathtt{L}\}) \quad a_7 = (\{\mathtt{M}\}, \{\}, \{\mathtt{L}\})$$
$$a_8 = (\{\mathtt{N}\}, \{\mathtt{O}\}, \{\mathtt{I}\}) \quad a_9 = (\{\mathtt{N}\}, \{\}, \{\mathtt{M}\}).$$

Similarly as in Examples 4.2 and 4.4 we consider again the production of object $\mathtt{L}$ *in two steps. The minimal multi-step based predictor of* $\mathtt{L}$ *in two steps is given by the following set of labelled objects*

$$\widehat{Q}_2 = \{\mathtt{N}_0, \mathtt{I}_1, \mathtt{M}_1\}.$$

Note that it is not necessary to observe the object $\mathtt{O}$ *in the initial context. Actually, the presence of object* $\mathtt{N}$ *in the initial context guarantees that the object* $\mathtt{L}$ *will be produced after two steps. By contrast, if the object* $\mathtt{N}$ *is not offered by the initial context then the production of object* $\mathtt{L}$ *depends on the objects offered by the context sequence in the next steps.*

The two previous examples clearly show that formula based predictors (Examples 4.1 and 4.2) are more precise than multi-step based predictors. Actually, there can be two context sequences that both lead to the production (or non-production) of the considered object but behave differently with respect to the observed objects at each step. As an example, let us consider the reaction system $\mathcal{A}_1$ of Example 3.1. In this case, a minimal formula based predictor for $\mathtt{G}$ in two steps is given by

$$f_1 = ((\mathtt{F}_1 \vee \mathtt{E}_0) \wedge \neg(\mathtt{B}_1 \vee \mathtt{A}_0)) \vee ((\mathtt{B}_1 \vee \mathtt{A}_0) \wedge (\mathtt{D}_1 \vee \mathtt{C}_0))$$

while the minimal multi-step based predictor for $\mathtt{G}$ in two steps is given by

$$\widehat{Q}_1 = \{\mathtt{A}_0, \mathtt{C}_0, \mathtt{E}_0, \mathtt{B}_1, \mathtt{D}_1, \mathtt{F}_1\}.$$

Now we can consider the 1-step context sequences $\gamma_1 = C_0^1, C_1^1$ with $C_0^1 = \{\mathtt{A}, \mathtt{C}\}$ and $C_1^1 = \{\mathtt{B}\}$ and $\gamma_2 = C_0^2, C_1^2$ with $C_0^2 = C_0^1$ and $C_1^2 = C_1^1 \cup \{\mathtt{D}\}$. The two context sequences γ_1 and γ_2 are equivalent with respect to formula f_1 given that they both satisfy the formula and thus lead to the production of $\mathtt{G}$ at the second execution step. By contrast, the context sequences γ_1 and γ_2 are

not equivalent with respect to $\widehat{Q}_1$ given that object D is observed in the first step of the context sequence. Formally, we have $\gamma_1 \approx^1_{f_1} \gamma_2$ while $\gamma_1 \not\approx^1_{\widehat{Q}_1} \gamma_2$.

Therefore, the number of equivalence classes of the set of context sequences induced by multi-set based predictor is often bigger than the number of equivalence classes induced by formula based predictors. This aspect significantly increases the computational cost of the corresponding tabling approach (that will be discussed in Section 7.2).

Formula based predictors can be related to the multi-step predictors by using an Abstract Interpretation approach. First of all, we define an *abstraction function* that maps logical formulas into sets of labelled objects. The abstraction of a formula f is given by the set of labelled objects that appear in f. Let $\mathcal{F} = \bigcup_{i\in\mathbb{N}} F_{S_i}$, $\mathcal{S} = \bigcup_{i\in\mathbb{N}} S_i$ and $\wp(\mathcal{S})$ denote the power set of $\mathcal{S}$. We define $\widehat{\alpha}_1 : \mathcal{F} \to \wp(\mathcal{S})$ such that for each $f \in \mathcal{F}$ we have $\widehat{\alpha}_1(f) = atom(f)$. Note that if $f \in F_{S^n}$ then its abstraction $\widehat{\alpha}_1(f) \subseteq S^n$.

Based on the abstraction function $\widehat{\alpha}_1$ it is immediate to derive a *Galois Insertion* [25] between the corresponding domains. The *abstraction function* α_1 computes the abstraction of a set of formulas, considering the union of the abstraction of each formula contained in the set. The *concretization function* γ_1 reports the set formulas that are represented by a set of labelled objects.

Definition 5.3. *We define the* abstraction *and* concretization functions $\alpha_1 : \wp(\mathcal{F}) \to \wp(\mathcal{S})$ *and* $\gamma_1 : \wp(\mathcal{S}) \to \wp(\mathcal{F})$ *functions as follows,*

1. *for* $F \in \wp(\mathcal{F})$, $\alpha_1(F) = \bigcup_{f\in F} \widehat{\alpha}_1(f)$;
2. *for* $\widehat{Q} \in \wp(\mathcal{S})$, $\gamma_1(\widehat{Q}) = \{f \mid \widehat{\alpha}_1(f) \subseteq \widehat{Q}\}$.

Theorem 5.2 (Galois Insertion)**.** *The pair of functions* (α_1, γ_1) *in Definition 5.3 is a* Galois insertion *between* $(\wp(\mathcal{F}), \subseteq)$ *and* $(\wp(\mathcal{S}), \subseteq)$.

The abstraction function allows us to relate formula based predictors with multi-step based predictors. Not only the abstraction of a formula based predictor is a multi-step based predictor but also the abstraction of a minimal formula based predictor is a minimal multi-step based predictor. This result relies on the following auxiliary property that relates the two notions of equivalence on context sequences.

Lemma 5.3. *Let* γ_1, γ_2 *be two n-step context sequences and* $f \in F_{S^n}$ *be a propositional logic formula. We have that*

$$\gamma_1 \simeq^n_{\widehat{\alpha}_1(f)} \gamma_2 \implies \gamma_1 \approx^n_f \gamma_2.$$

Theorem 5.4. *Let* $\mathcal{A} = (S, A)$ *be a reaction system,* $s \in S$ *and* $f \in F_{S^n}$ *be a propositional logic formula. We have that*

- *if* f f-predicts s *in* $n+1$ *steps then* $\widehat{\alpha}_1(f)$ m-predicts s *in* $n+1$ *steps;*
- *if* f *is a* minimal f-predictor *of* s *in* $n+1$ *steps then* $\widehat{\alpha}_1(f)$ *is a* minimal m-predictor *of* s *in* $n+1$ *steps.*

As a consequence, the abstraction of the formula computed by the operator `fbp` (given in Definition 4.5) is a multi-step based predictor for s in $n+1$ steps.

Corollary 5.5. *Let $\mathcal{A} = (S, A)$ be a reaction system and $s \in S$. We have that*

- $\widehat{\alpha}_1(\mathtt{fbp}(s, n))$ m-predicts *s in n + 1 steps;*
- $\widehat{\alpha}_1(simpl(\mathtt{fbp}(s, n)))$ *is the* minimal m-predictor *of s in n + 1 steps.*

The previous result is very useful given that it can be applied to effectively derive a multi-step based predictor from a formula based predictor, by preserving minimality if the formula based predictor is minimal. The application of the abstraction simply requires one to collect all labelled objects appearing in the corresponding formula based predictor. To illustrate the approach we consider the reaction systems presented in the Examples 3.1 and 3.2.

Example 5.3. *Let $\mathcal{A}_1 = (\{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}, \mathtt{G}, \mathtt{H}\}, \{a_1, a_2, a_3, a_4, a_5\})$ be the reaction system of Examples 3.1 with reaction rules*

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

In order to calculate an m-predictor for object G *in two steps we apply the abstraction function to the corresponding formula based predictor calculated by the function* fbp*. Hence, using the formula shown in Example 4.3 we derive*

$$\widehat{\alpha}_1(\mathtt{fbp}(\mathtt{G}, 1)) = \{\mathtt{A}_0, \mathtt{C}_0, \mathtt{E}_0, \mathtt{B}_1, \mathtt{D}_1, \mathtt{F}_1\} = \widehat{Q}_1$$

where

$$\mathtt{fbp}(\mathtt{G}, 1) = ((\mathtt{F}_1 \vee \mathtt{E}_0) \wedge \neg(\mathtt{B}_1 \vee \mathtt{A}_0)) \vee ((\mathtt{B}_1 \vee \mathtt{A}_0) \wedge (\mathtt{D}_1 \vee \mathtt{C}_0)).$$

We recall that in this case the formula based predictor is minimal and consequently its abstraction provides the minimal multi-step based predictor $\widehat{Q}_1$, commented in Example 5.1.

Example 5.4. *Let $\mathcal{A}_2 = (\{\mathtt{I}, \mathtt{L}, \mathtt{M}, \mathtt{N}, \mathtt{O}\}, \{a_6, a_7, a_8, a_9\})$ be a reaction system of Example 3.2 with reaction rules*

$$\begin{array}{ll} a_6 = (\{\mathtt{I}\}, \{\}, \{\mathtt{L}\}) & a_7 = (\{\mathtt{M}\}, \{\}, \{\mathtt{L}\}) \\ a_8 = (\{\mathtt{N}\}, \{\mathtt{O}\}, \{\mathtt{I}\}) & a_9 = (\{\mathtt{N}\}, \{\}, \{\mathtt{M}\}). \end{array}$$

In this case differently from the previous one the function fbp *gives a formula based predictor for* L *in two steps that is not minimal. This is represented by the following formula commented in Example 4.4*

$$\mathtt{fbp}(\mathtt{L}, 1) = \big((\mathtt{I}_1 \vee (\mathtt{N}_0 \wedge \neg(\mathtt{O}_0))) \vee (\mathtt{M}_1 \vee \mathtt{N}_0)\big).$$

Therefore, by applying the abstraction function we derive the following multi-step based predictor for L *in two steps*

$$\widehat{\alpha}_1(\mathtt{fbp}(\mathtt{L}, 1)) = \{\mathtt{N}_0, \mathtt{O}_0, \mathtt{I}_1, \mathtt{M}_1\} = \widehat{Q}_3.$$

As expected the set of labelled objects $\widehat{Q}_3$ is not the minimal multi-step based predictor for L *in two steps. In this case we have to apply the simplification procedure to the corresponding formula based predictor in order to obtain the minimal multi-step based predictor $\widehat{Q}_2$ of Example 5.2. In this way, we derive*

$$\widehat{\alpha}_1(simpl(\mathtt{fbp}(\mathtt{L}, 1))) = \widehat{\alpha}_1(\mathtt{I}_1 \vee \mathtt{N}_0 \vee \mathtt{M}_1) = \{\mathtt{N}_0, \mathtt{I}_1, \mathtt{M}_1\} = \widehat{Q}_2.$$

Following an *Abstract Interpretation* approach we also introduce an operator mbp that calculates multi-step based predictors and that is obtained as the abstract variant of the operator fbp (given in Definition 4.5). The abstract operator allows us to derive a multi-step based predictor in a more efficient way without building the corresponding formula based predictor, as implied by Corollary 5.5.

Definition 5.4 (Multi-step based Predictor Operator)**.** *Let $\mathcal{A} = (S, A)$ be a reaction system. We define a function* $\mathtt{mbp} : S \times \mathbb{N} \to \mathcal{P}(S^n)$ *as follows:*

$$\mathtt{mbp}(s, n) = \mathtt{mbs}(cause(s), n)$$

where we adopt the auxiliary function $\mathtt{mbs} : F_S \times \mathbb{N} \to \mathcal{P}(S^n)$ *with* $\mathtt{mbs}(f, i)$ *recursively defined as follows:*

$$\begin{aligned}
\mathtt{mbs}(s, 0) &= \{s_0\} \\
\mathtt{mbs}(s, i) &= \{s_i\} \cup \mathtt{mbs}(cause(s), i-1) \ \ \textit{if } i > 0 \\
\mathtt{mbs}(f_1 \wedge f_2, i) &= \mathtt{mbs}(f_1, i) \cup \mathtt{mbs}(f_2, i) \\
\mathtt{mbs}(f_1 \vee f_2, i) &= \mathtt{mbs}(f_1, i) \cup \mathtt{mbs}(f_2, i) \\
\mathtt{mbs}(\neg f', i) &= \mathtt{mbs}(f', i) \\
\mathtt{mbs}((f'), i) &= \mathtt{mbs}(f', i) \\
\mathtt{mbs}(true, i) &= \emptyset \\
\mathtt{mbs}(false, i) &= \emptyset
\end{aligned}$$

We now compare the abstract operator mbp with the abstraction of the corresponding operator fbp. More in details, we prove that the abstraction of the operator fbp coincides with the result of the operator mbp.

Theorem 5.6. *Let $\mathcal{A} = (S, A)$ be a reaction system and $s \in S$. Given $n \in \mathbb{N}$ we have that*

$$\widehat{\alpha}_1(\mathtt{fbp}(s, n)) = \mathtt{mbp}(s, n).$$

As a consequence, the set of labelled objects $\mathtt{mbp}(s, n)$ is a multi-step based predictor for object s in $n + 1$ steps.

Corollary 5.7. *Let $\mathcal{A} = (S, A)$ be a reaction system and $s \in S$. We have that* $\mathtt{mbp}(s, n)$ m-predicts *s in $n + 1$ steps.*

The previous result shows the main property of abstract operator mbp. Unfortunately, in the general case the abstract operator mbp is not sufficiently precise to capture the minimal multi-step based predictor.

To illustrate the abstract operator mbp we consider the reaction systems of Examples 3.1 and 3.2.

Example 5.5. *Let* $\mathcal{A}_1 = (\{\mathtt{A},\mathtt{B},\mathtt{C},\mathtt{D},\mathtt{E},\mathtt{F},\mathtt{G},\mathtt{H}\},\{a_1,a_2,a_3,a_4,a_5\})$ *be the reaction system of Examples 3.1 with reactions*

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

In order to calculate a multi-step predictor for object G *in two steps we apply the abstract operator* mbp *and we obtain*

$$\begin{aligned}
\mathtt{mbp}(\mathtt{G},1) &= \mathtt{mbs}((\mathtt{F}\wedge\neg\mathtt{B})\vee(\mathtt{B}\wedge\mathtt{D}),1)\\
&= (\mathtt{mbs}(\mathtt{F}\wedge\neg\mathtt{B},1)\cup\mathtt{mbs}(\mathtt{B}\wedge\mathtt{D},1))\\
&= (\mathtt{mbs}(\mathtt{F},1)\cup\ \mathtt{mbs}(\neg\mathtt{B},1))\cup(\mathtt{mbs}(\mathtt{B},1)\cup\mathtt{mbs}(\mathtt{D},1))\\
&= (\mathtt{mbs}(\mathtt{F},1)\cup\ \mathtt{mbs}(\mathtt{B},1))\cup(\mathtt{mbs}(\mathtt{B},1)\cup\mathtt{mbs}(\mathtt{D},1))\\
&= (\{\mathtt{F}_1\}\cup\mathtt{mbs}(\mathtt{E},0)\cup\{\mathtt{B}_1\}\cup\mathtt{mbs}(\mathtt{A},0)\cup\{\mathtt{B}_1\}\cup\mathtt{mbs}(\mathtt{A},0)\cup\{\mathtt{D}_1\}\cup\mathtt{mbs}(\mathtt{C},0))\\
&= \{\mathtt{A}_0,\mathtt{C}_0,\mathtt{E}_0,\mathtt{B}_1,\mathtt{D}_1,\mathtt{F}_1\} = \widehat{Q}_1
\end{aligned}$$

According to Theorem 5.6 we have that $\widehat{\alpha}_1(\mathtt{fbp}(\mathtt{G},1)) = \widehat{Q}_1 = \mathtt{mbp}(\mathtt{G},1)$ *where the abstraction of operator* fbp *gives the set of labelled objects* $\widehat{Q}_1$ *commented in Example 5.3. We recall that* $\widehat{Q}_1$ *is the minimal multi-step based predictor for* G *in two steps given that the corresponding formula based predictor is also minimal.*

Example 5.6. *Let* $\mathcal{A}_2 = (\{\mathtt{I},\mathtt{L},\mathtt{M},\mathtt{N},\mathtt{O}\},\{a_6,a_7,a_8,a_9\})$ *be a reaction system of Example 3.2 with reaction rules*

$$a_6 = (\{\mathtt{I}\},\{\},\{\mathtt{L}\}) \qquad a_7 = (\{\mathtt{M}\},\{\},\{\mathtt{L}\})$$
$$a_8 = (\{\mathtt{N}\},\{\mathtt{O}\},\{\mathtt{I}\}) \quad a_9 = (\{\mathtt{N}\},\{\},\{\mathtt{M}\}).$$

In order to calculate a multi-step based predictor for object L *in two steps we apply the abstract operator* mbp *and we obtain*

$$\begin{aligned}
\mathtt{mbp}(\mathtt{L},1) &= \mathtt{mbs}(\mathtt{I}\vee\mathtt{M},1)\\
&= \mathtt{mbs}(\mathtt{I},1)\cup\mathtt{mbs}(\mathtt{M},1)\\
&= (\{\mathtt{I}_1\}\cup\mathtt{mbs}(\mathtt{N}\wedge\neg\mathtt{O},0)\cup\{\mathtt{M}_1\}\cup\mathtt{mbs}(\mathtt{N},0))\\
&= (\{\mathtt{I}_1\}\cup\mathtt{mbs}(\mathtt{N},0)\cup\mathtt{mbs}(\neg\mathtt{O},0)\cup\{\mathtt{M}_1\}\cup\{\mathtt{N}_0\})\\
&= (\{\mathtt{I}_1\}\cup\{\mathtt{N}_0\}\cup\mathtt{mbs}(\mathtt{O},0)\cup\{\mathtt{M}_1\}\cup\{\mathtt{N}_0\})\\
&= \{\mathtt{N}_0,\mathtt{O}_0,\mathtt{I}_1,\mathtt{M}_1\} = \widehat{Q}_3
\end{aligned}$$

As we have already commented in Examples 5.2 and 5.4 the set of labelled objects $\widehat{Q}_3$ *is a multi-step based predictor* L *in two steps but it is not minimal. Thus, in this case, differently from the previous one, the abstract operator* mbp *is not able to capture the minimal m-predictor. As a consequence, in order to derive the minimal multi-step based predictor, it is necessary to calculate the corresponding minimal formula based predictor and to apply the abstraction function (as illustrated in Example 5.4).*

6. Set Based Predictors

In this section we define *set based predictors* a revised version of the predictors originally presented in [18]. Set based predictors require one to observe the same set of objects in all steps of the computation of the reaction system. Consequently, they are less precise than multi-step based predictors (and hence of formula based predictors). To relate set based predictors with the other notions of predictors we use an Abstract Interpretation approach introducing a new Galois Insertion between the corresponding domains. Moreover, similarly as in Section 5 we show that set based predictors can be obtained in a systematic way as abstraction of multi-step based predictors. Finally, we introduce an operator `sbp` to compute set based predictors, defined as an abstract version of the corresponding operator `mbp` for multi-step based predictors (presented in Definition 5.4). We compare in terms of precision the abstract operator `sbp` with the abstraction of the corresponding operator `mbp`.

We adapt the definition of predictor given in [18] to our Abstract Interpretation framework. A set based predictor is modeled as a set of objects representing the elements that have to be observed at each execution step of the reaction system. Thus, two n-step context sequences are equivalent with respect to a set of objects $Q \subseteq S$ if and only if every object in Q is *either* present *or* absent in both context sequences. This is formalised by the following definition of *equivalence relation* on context sequences.

Definition 6.1. *Let $\gamma_1 = C_0^1, C_1^1, \ldots C_n^1$, $\gamma_2 = C_0^2, C_1^2, \ldots C_n^2$ be two n-step context sequences. Given a set of objects $Q \subseteq S$ we say that $\gamma_1 \sim_Q^n \gamma_2$ iff for each $i = \{1, \ldots, n\}$,*

$$(C_i^1 \cap Q) = (C_i^2 \cap Q).$$

A set of objects Q is a *set based predictor* for object s in $n+1$ steps if two context sequences that are equated by $\sim_Q^n$ *either* both lead *or* both not lead to the production of object s in $n+1$ steps.

Definition 6.2 (Set based Predictor)**.** *Let $\mathcal{A} = (S, A)$ be reaction system, $s \in S$ and $Q \subseteq S$. We say that Q* s-predicts *s* in $n+1$ steps *if for all n-step context sequences $\gamma_1 = C_0^1, C_1^1, \ldots, C_n^1$ and $\gamma_2 = C_0^2, C_1^2, \ldots, C_n^2$ such that $\gamma_1 \sim_Q^n \gamma_2$ we have that*

$$s \in D_{n+1}^1 \iff s \in D_{n+1}^2$$

where $\delta_1 = D_0^1, D_1^1, \ldots, D_n^1$ and $\delta_2 = D_0^2, D_1^2, \ldots, D_n^2$ are the result sequence corresponding to γ_1 and γ_2 and where $D_{n+1}^1 = res_{\mathcal{A}}(C_n^1 \cup D_n^1)$ and $D_{n+1}^2 = res_{\mathcal{A}}(C_n^2 \cup D_n^2)$.

In [18] it is shown that there exists a unique minimal (w.r.t. set inclusion order) set based predictor for s in $n+1$ steps.

Theorem 6.1 ([18])**.** *Let $\mathcal{A} = (S, A)$ be reaction system and $s \in S$. There exists a unique minimal (w.r.t. the set inclusion order) $Q \subseteq S$ that s-predicts s in $n+1$ steps.*

We illustrate the notion of set based predictor revisiting the reaction systems of Examples 3.1 and 3.2.

Example 6.1. *Let* $\mathcal{A}_1 = (\{\mathtt{A},\mathtt{B},\mathtt{C},\mathtt{D},\mathtt{E},\mathtt{F},\mathtt{G},\mathtt{H}\},\{a_1,a_2,a_3,a_4,a_5\})$ *be the reaction system of Examples 3.1 with reaction rules*

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

Similarly as in Examples 4.1 and 5.1 we are interested in the production of object G *in two steps. The minimal set based predictor of* G *in* 2 *steps is given by the following set of objects*

$$Q_1 = \{\mathtt{A},\mathtt{C},\mathtt{E},\mathtt{B},\mathtt{D},\mathtt{F}\}.$$

Example 6.2. *Let* $\mathcal{A}_2 = (\{\mathtt{I},\mathtt{L},\mathtt{M},\mathtt{N},\mathtt{O}\},\{a_6,a_7,a_8,a_9\})$ *be a reaction system of Example 3.2 with reaction rules*

$$a_6 = (\{\mathtt{I}\},\{\},\{\mathtt{L}\}) \quad a_7 = (\{\mathtt{M}\},\{\},\{\mathtt{L}\})$$
$$a_8 = (\{\mathtt{N}\},\{\mathtt{O}\},\{\mathtt{I}\}) \quad a_9 = (\{\mathtt{N}\},\{\},\{\mathtt{M}\}).$$

Similarly as in Examples 4.2 and 5.2 we are interested in the production of object L *in two steps. The minimal set based predictor of* L *in* 2 *steps is given by the following set of objects*

$$Q_2 = \{\mathtt{N},\mathtt{I},\mathtt{M}\}.$$

The two previous examples clearly show that multi-step based predictors (Examples 5.1 and 5.2) are more precise than set based predictors. Actually, there can be two context sequences that are equivalent if a different set of objects is observed at each step, that cannot be considered as equivalent if the same set of objects is observed at all steps. As an example, let us consider the reaction system $\mathcal{A}_1$ of Example 3.1. In this case, the minimal multi-step based predictor for G in 2 steps is given by the set of labelled objects,

$$\widehat{Q}_1 = \{\mathtt{A}_0,\mathtt{C}_0,\mathtt{E}_0,\mathtt{B}_1,\mathtt{D}_1,\mathtt{F}_1\}$$

while the minimal set based predictor for G in 2 steps is given by the set of objects,

$$Q_1 = \{\mathtt{A},\mathtt{C},\mathtt{E},\mathtt{B},\mathtt{D},\mathtt{F}\}.$$

Now, we can consider the 1-step context sequences $\gamma_1 = C_0^1, C_1^1$ with $C_0^1 = \{\mathtt{A},\mathtt{C},\mathtt{E}\}$ and $C_1^1 = \{\mathtt{B},\mathtt{D},\mathtt{F}\}$ while $\gamma_2 = C_0^2, C_1^2$ with $C_0^2 = C_0^1$ and $C_1^2 = C_1^1 \cup \{\mathtt{A}\}$. Given that object A is not observed at the first execution step, we have that γ_1 and γ_2 are equivalent with respect to $\widehat{Q}_1$ while they are not equivalent with respect to Q_1. Formally, we have $\gamma_1 \simeq^1_{\widehat{Q}_1} \gamma_2$ while $\gamma_1 \not\sim^1_{Q_1} \gamma_2$.

Therefore, the number of equivalence classes on the set of context sequences in the case of set based predictors is often bigger than the number of equivalence classes in the case of multi-step based predictors. This aspect significantly increases the computational cost of the corresponding tabling approach (that will be discussed in Section 7.2).

Analogously as in Section 5 we relate set based predictors with multi-step based predictors by using an *Abstract Interpretation* approach. First of all, we define a *Galois Insertion* between the corresponding domains. The *abstraction function* α_2 computes the abstraction of a set of labelled objects, obtained by removing the indexes associated to objects. The *concretization function* γ_2 reports the set labelled objects that are represented by a set of objects.

Definition 6.3. *We define the abstraction and concretization functions* $\alpha_2 : \mathcal{P}(\mathcal{S}) \to \mathcal{P}(S)$ *and* $\gamma_2 : \mathcal{P}(S) \to \mathcal{P}(\mathcal{S})$ *functions as follows,*

1. *for* $\widehat{Q} \in \mathcal{P}(\mathcal{S})$, $\alpha_2(\widehat{Q}) = \{s \mid s_i \in \widehat{Q}, i \in \mathbb{N}\}$*;*
2. *for* $Q \in \mathcal{P}(S)$, $\gamma_2(Q) = \{s_i \mid s \in Q, i \in \{0, ..., n\}\}$.

Theorem 6.2. *The pair of functions* (α_2, γ_2) *in Definition 6.3 is a* Galois insertion *between* $(\mathcal{P}(\mathcal{S}), \subseteq)$ *and* $(\mathcal{P}(S), \subseteq)$.

The abstraction framework allows us to relate multi-step based predictors and set based predictors, analogously as in Section 5. Not only the abstraction of a multi-step based predictor is a set based predictor but also the abstraction of the minimal multi-step based predictor is the minimal set based predictor. This result relies on the following auxiliary property that relates the two notions of equivalence on context sequences (corresponding to the property formalized by Lemma 5.3).

Lemma 6.3. *Let* γ_1, γ_2 *be two n-step context sequences and* $\widehat{Q} \subseteq S^n$. *We have that*

$$\gamma_1 \sim^n_{\alpha_2(\widehat{Q})} \gamma_2 \Rightarrow \gamma_1 \simeq^n_{\widehat{Q}} \gamma_2.$$

Theorem 6.4. *Let* $\mathcal{A} = (S, A)$ *be a reaction system,* $s \in S$ *and* $\widehat{Q} \subseteq S^n$. *We have that*

- *if* $\widehat{Q}$ m-predicts s in $n + 1$ steps *then* $\alpha_2(\widehat{Q})$ s-predicts s in $n + 1$ steps*;*
- *if* $\widehat{Q}$ *is the* minimal multi-step based predictor *of* s *in* $n + 1$ *steps then* $\alpha_2(\widehat{Q})$ *is the* minimal set based predictor *of* s *in* $n + 1$ *steps.*

As a consequence, we obtain the following properties for the abstraction functions.

Corollary 6.5. *Let* $\mathcal{A} = (S, A)$ *be a reaction system and* $s \in S$. *We have that*

- $\alpha_2(\mathtt{mbp}(s, n))$ s-predicts s *in* $n + 1$ *steps;*
- $\alpha_2(\hat{\alpha}_1(simpl(\mathtt{fbp}(s, n))))$ s-predicts s *in* $n + 1$ *steps and is minimal.*

The previous main result is similar to the corresponding Corollary 5.5 for formula based and multi-step based predictors. The property guarantees that the abstraction of $\mathtt{mbp}(s, n)$ is a set based predictor for s in $n+1$ steps. However in this case the operator mbp may give a multi step predictor that is not minimal. Thus in general the minimal set based predictor has to be calculated by abstracting the corresponding minimal formula based predictor.

To illustrate the approach we consider the reaction systems presented in Examples 3.1 and 3.2.

Example 6.3. *Let* $\mathcal{A}_1 = (\{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}, \mathtt{G}, \mathtt{H}\}, \{a_1, a_2, a_3, a_4, a_5\})$ *be the reaction system of Examples 3.1 with reaction rules*

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

In order to calculate a set based predictor for object $\mathtt{G}$ *in* 2 *steps we apply the abstraction function to the corresponding multi-step based predictor (illustrated in Example 5.5). Hence we obtain*

$$\alpha_2(\mathtt{mbp}(\mathtt{G},1)) = \{\mathtt{A},\mathtt{C},\mathtt{E},\mathtt{B},\mathtt{D},\mathtt{F}\} = Q_1$$

where

$$\mathtt{mbp}(\mathtt{G},1) = \{\mathtt{A}_0,\mathtt{C}_0,\mathtt{E}_0,\mathtt{B}_1,\mathtt{D}_1,\mathtt{F}_1\}.$$

Given that in this case the corresponding multi-step based predictor is minimal, its abstraction gives the minimal set based predictor Q_1*, commented in Example 6.1.*

Example 6.4. *Let* $\mathcal{A}_2 = (\{\mathtt{I},\mathtt{L},\mathtt{M},\mathtt{N},\mathtt{O}\},\{a_6,a_7,a_8,a_9\})$ *be a reaction system of Example 3.2 with reaction rules*

$$a_6 = (\{\mathtt{I}\},\{\},\{\mathtt{L}\}) \quad a_7 = (\{\mathtt{M}\},\{\},\{\mathtt{L}\})$$
$$a_8 = (\{\mathtt{N}\},\{\mathtt{O}\},\{\mathtt{I}\}) \quad a_9 = (\{\mathtt{N}\},\{\},\{\mathtt{M}\}).$$

In this case differently from the previous one the function $\mathtt{mbp}$ *computes a multi-step based predictor for* $\mathtt{L}$ *in* 2 *steps that is not minimal, as we have illustrated in Example 5.6. Thus, by applying the abstraction function we obtain,*

$$\alpha_2(\mathtt{mbp}(\mathtt{L},1)) = \alpha_2(\{\mathtt{N}_0,\mathtt{O}_0,\mathtt{I}_1,\mathtt{M}_1\}) = \{\mathtt{N},\mathtt{O},\mathtt{I},\mathtt{M}\} = Q_3.$$

The set based predictor Q_3 *is not minimal. As commented in Example 6.2, in this case the minimal set based predictor has to be calculated by abstracting the corresponding minimal formula based predictor. More in details, we can obtain the minimal set based predictor by considering the simplified version of the formula reported by the operator* $\mathtt{fbp}$ *as follows,*

$$\alpha_2(\hat{\alpha}_1(\mathit{simpl}(\mathtt{fbp}(\mathtt{L},1)))) = \alpha_2(\{\mathtt{N}_0,\mathtt{I}_1,\mathtt{M}_1\}) = \{\mathtt{N},\mathtt{I},\mathtt{M}\} = Q_2.$$

The corresponding formula based predictor and its abstraction are shown in the Examples 4.4 and 5.4.

Following the style of Section 5 we introduce an operator $\mathtt{sbp}$ that calculates set based predictors and is obtained as the abstract variant of the operator $\mathtt{mbp}$ (given in Definition 5.4). The abstract operator allows us to derive a set based predictor in a more efficient way without building the corresponding multi-step based predictor (or analogously formula based predictor).

Definition 6.4 (Set based Predictor Operator)**.** *Let* $\mathcal{A} = (S,A)$ *be a reaction system. We define a function* $\mathtt{sbp} : S \times \mathbb{N} \to \mathcal{P}(S)$ *as follows:*

$$\mathtt{sbp}(s,n) = \mathtt{sbs}(\mathit{cause}(s),n)$$

where we adopt the auxiliary function $\mathtt{sbs} : F_S \times \mathbb{N} \to \mathcal{P}(S)$ *with* $\mathtt{sbs}(f, i)$ *recursively defined as follows:*

$$\begin{aligned}
\mathtt{sbs}(s, 0) &= \{s\} \\
\mathtt{sbs}(s, i) &= \{s\} \cup \mathtt{sbs}(\mathit{cause}(s), i-1) \ \text{ if } i > 0 \\
\mathtt{sbs}(f_1 \wedge f_2, i) &= \mathtt{sbs}(f_1, i) \cup \mathtt{sbs}(f_2, i) \\
\mathtt{sbs}(f_1 \vee f_2, i) &= \mathtt{sbs}(f_1, i) \cup \mathtt{sbs}(f_2, i) \\
\mathtt{sbs}(\neg f', i) &= \mathtt{sbs}(f', i) \\
\mathtt{sbs}((f'), i) &= \mathtt{sbs}(f', i) \\
\mathtt{sbs}(\mathit{true}, i) &= \emptyset \\
\mathtt{sbs}(\mathit{false}, i) &= \emptyset
\end{aligned}$$

We now compare the abstract operator sbp with the abstraction of the corresponding operator mbp, obtaining a result analogous to the one stated by Theorem 5.6. More in details, we prove that the abstraction of the operator mbp coincides with the result of the operator sbp.

Theorem 6.6. *Let* $\mathcal{A} = (S, A)$ *be a reaction system and* $s \in S$*. Given* $n \in \mathbb{N}$ *we have that*

$$\alpha_2(\mathtt{mbp}(s, n)) = \mathtt{sbp}(s, n).$$

As a consequence, the set of objects $\mathtt{sbp}(s, n)$ is a set based predictor for object s in $n+1$ steps.

Corollary 6.7. *Let* $\mathcal{A} = (S, A)$ *be a reaction system and* $s \in S$*. We have that* $\mathtt{sbp}(s, n)$ s-predicts *s in* $n + 1$ *steps.*

The previous main result shows that the abstract operator sbp calculates a set based predictor. Nevertheless, the abstract operator sbp is not sufficiently precise to capture the minimal set based predictor in the general case. To illustrate the abstract operator sbp we consider the reaction systems of Examples 3.1 and 3.2.

Example 6.5. *Let* $\mathcal{A}_1 = (\{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}, \mathtt{G}, \mathtt{H}\}, \{a_1, a_2, a_3, a_4, a_5\})$ *be the reaction system of Examples 3.1 with reaction rules*

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

In order to calculate a set based predictor for object G *in two steps we apply the abstract operator* sbp*. Thus, we obtain*

$$\begin{aligned}
\mathtt{sbp}(\mathtt{G}, 1) &= \mathtt{sbs}\big((\mathtt{F} \wedge \neg \mathtt{B}) \vee (\mathtt{B} \wedge \mathtt{D}), 1\big) \\
&= \big(\mathtt{sbs}(\mathtt{F} \wedge \neg \mathtt{B}, 1) \cap \mathtt{sbs}(\mathtt{B} \wedge \mathtt{D}, 1)\big) \\
&= \big(\mathtt{sbs}(\mathtt{F}, 1) \cup \ \mathtt{sbs}(\mathtt{B}, 1)\big) \cup \big(\mathtt{sbs}(\mathtt{B}, 1) \cup \mathtt{sbs}(\mathtt{D}, 1)\big) \\
&= \big(\{\mathtt{F}\} \cup \mathtt{sbs}(\mathtt{E}, 0) \cup \{\mathtt{B}\} \cup \mathtt{sbs}(\mathtt{A}, 0) \cup \{\mathtt{B}\} \cup \mathtt{sbs}(\mathtt{A}, 0) \cup \{\mathtt{D}\} \cup \mathtt{sbs}(\mathtt{C}, 0)\big) \\
&= \{\mathtt{A}, \mathtt{C}, \mathtt{E}, \mathtt{B}, \mathtt{D}, \mathtt{F}\} = Q_1
\end{aligned}$$

As expected $\mathtt{sbp}(\mathtt{G},1) = Q_1$ *is the minimal set based predictor for* G *in two steps, discussed in Example 6.1. Indeed, in this case* $\mathtt{mbp}(\mathtt{G},1)$ *gives the minimal multi-step based predictor for* G *in two steps, as it is illustrated in Example 6.3. As a consequence its abstraction that coincides with* $\mathtt{sbp}(\mathtt{G},1)$ *is also minimal, according to Theorem 6.6.*

Example 6.6. *Let* $\mathcal{A}_2 = (\{\mathtt{I},\mathtt{L},\mathtt{M},\mathtt{N},\mathtt{O}\},\{a_6,a_7,a_8,a_9\})$ *be the reaction system of Example 3.2 with reaction rules*

$$a_6 = (\{\mathtt{I}\},\{\},\{\mathtt{L}\}) \quad a_7 = (\{\mathtt{M}\},\{\},\{\mathtt{L}\})$$
$$a_8 = (\{\mathtt{N}\},\{\mathtt{O}\},\{\mathtt{I}\}) \quad a_9 = (\{\mathtt{N}\},\{\},\{\mathtt{M}\}).$$

In order to calculate a set based predictor for object L *in two steps we apply the abstract operator* sbp*. Thus, we obtain*

$$\begin{aligned}
\mathtt{sbp}(\mathtt{L},1) &= \mathtt{sbs}(\mathtt{I}\vee\mathtt{M},1)\\
&= \mathtt{sbs}(\mathtt{I},1)\cup\mathtt{sbs}(\mathtt{M},1)\\
&= (\{\mathtt{I}\}\cup\mathtt{sbs}(\mathtt{N}\wedge\neg\mathtt{O},0)\cup\{\mathtt{M}\}\cup\mathtt{sbs}(\mathtt{N},0))\\
&= (\{\mathtt{I}\}\cup\mathtt{sbs}(\mathtt{N},0)\cup\mathtt{sbs}(\neg\mathtt{O},0)\cup\{\mathtt{M}\}\cup\{\mathtt{N}\})\\
&= (\{\mathtt{I}\}\cup\{\mathtt{N}\}\cup\mathtt{sbs}(\mathtt{O},0)\cup\{\mathtt{M}\}\cup\{\mathtt{N}\})\\
&= \{\mathtt{N},\mathtt{O},\mathtt{I},\mathtt{M}\} = Q_3
\end{aligned}$$

As we have already commented in Examples 6.2 the set of objects Q_3 *s-predicts* L *in two steps but it is not minimal. Hence in this case differently from the previous one the abstract operator* sbp *is not able to compute the minimal set based predictor. As a consequence, the calculation of the minimal set based predictor requires one to apply the abstraction functions to the corresponding minimal formula based predictor, following the methodology of Example 6.4.*

7. Using predictors: generalization and tabling

In this section we discuss two aspects of the practical use of predictors on reaction systems. The first aspect is related with the limitation on the number of steps contained in the definition of predictor (of any of the three kinds). In particular, a predictor can evaluate the behavior of a reaction systems for a specific number of steps $n+1$. In principle, if the number of steps of interest for an experiment could vary, it would be necessary to compute one predictor for each possible value of such a parameter. We show, in Subsection 7.1, that in practice, in the case of formula based predictors, this is not usually the case, since one or a few generalized formula based predictors can easily be obtained for all values of $n+1$ greater than some threshold value m. Moreover, we can obtain generalized multi-step based predictors by applying the abstraction function to the corresponding formula based predictors.

The second aspect we consider is related with the use one could make of predictors, as suggested in [27]. Since predictors can only compare the behavior of two executions of a reaction system, their use in practice should be based on a *tabling approach*. In particular, by focusing on set based predictors, given a reaction system and a predictor for it, a table can be constructed that contains one line for every possible context sequence consisting only of elements of the predictor. Each line of the table indicates whether, for such a sequence, the symbol for which the predictor

has been defined is actually produced or not. This information is calculated by executing the reaction system with the context sequence corresponding to the row. Then, when an observed system has to be evaluated, this should be done by considering the sequence of objects it receives from the context restricted to the elements of the predictor. The obtained context sequence can then be used to access the previously constructed table and predict whether the object of interest will be produced or not.

The computational cost of such a tabling approach with set based predictors increases significantly with the size of the predictor. We show in Subsection 7.2 that such a cost can be substantially reduced by using multi-step or formula based predictors in place of set based predictors.

7.1. Discussion on generalization

In the previous sections we presented different notions of predictors of an element s in $n+1$ steps, for a given n. However, it is worth noting that formula based predictors have a great advantage, that is, in many cases they can be easily generalized in order to obtain a parametric formula expressing predictors of s in $n+1$ steps *for any n such that* $n+1 > m$*, for a certain* m. Intuitively, m should be chosen big enough so that all reactions involved in the production of object s have been applied. Even when a unique parametric formula that predicts s in step $n+1 > m$ cannot be found, it is possible to find several parametric formulas based on conditions on $n+1$.

As an example of easy generalization consider the reaction system introduced in Example 3.1 $\mathcal{A}_1 = (\{\mathtt{A},\mathtt{B},\mathtt{C},\mathtt{D},\mathtt{E},\mathtt{F},\mathtt{G},\mathtt{H}\},\{a_1,a_2,a_3,a_4,a_5\})$ with reaction rules

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

In Example 4.3 we have shown that the formula

$$\mathtt{fbp}(\mathtt{G},1) = (((\mathtt{F}_1 \vee \mathtt{E}_0) \wedge \neg(\mathtt{B}_1 \vee \mathtt{A}_0)) \vee ((\mathtt{B}_1 \vee \mathtt{A}_0) \wedge (\mathtt{D}_1 \vee \mathtt{C}_0)))$$

f-predicts G in two steps. Consider now the two formulas

$$\mathtt{fbp}(\mathtt{G},2) = (((\mathtt{F}_2 \vee \mathtt{E}_1) \wedge \neg(\mathtt{B}_2 \vee \mathtt{A}_1)) \vee ((\mathtt{B}_2 \vee \mathtt{A}_1) \wedge (\mathtt{D}_2 \vee \mathtt{C}_1)))$$

and

$$\mathtt{fbp}(\mathtt{G},3) = (((\mathtt{F}_3 \vee \mathtt{E}_2) \wedge \neg(\mathtt{B}_3 \vee \mathtt{A}_2)) \vee ((\mathtt{B}_3 \vee \mathtt{A}_2) \wedge (\mathtt{D}_3 \vee \mathtt{C}_2))) .$$

Such formulas *f-predicts* G in 3 and 4 steps, respectively. It is immediate to see that $\mathtt{fbp}(\mathtt{G},2)$ and $\mathtt{fbp}(\mathtt{G},3)$ are the same formula as $\mathtt{fbp}(\mathtt{G},1)$ with all indexes incremented by one and two, respectively. In both cases the predictor contains indexes that refer only to the two steps that precede the (possible) appearance of G. This is due to the fact that the rules of the considered reaction system produce G in exactly two steps, without any cyclic production of any intermediate object (we will show an example of this below).

The absence of a such a cyclic production of objects, and consequently the fact that the number of steps necessary to produce an object is finite, is a property that can be checked by constructing the *dependency graph* of the reaction system. Such a graph has the objects of the reaction systems as nodes, and relates by means of direct edges, each object with the reactants and the inhibitors of

the reactions producing it. (It is the same as the influence graph defined in [18], but with inverted edges.) If the subgraph of the dependency graph rooted at the object of interest does not contain any cycle (i.e. it is a direct acyclic graph), then the number of steps necessary to produce such an object is bounded, and the the bound is the depth of such a subgraph.

In such a situation, the formula based predictor can be easily generalized. For instance, we can easily generalize the formula based predictor of $\mathtt{G}$ with the parametric formula

$$\mathtt{fbp}(\mathtt{G}, n) = (((\mathtt{F}_n \vee \mathtt{E}_{n-1}) \wedge \neg(\mathtt{B}_n \vee \mathtt{A}_{n-1})) \vee ((\mathtt{B}_n \vee \mathtt{A}_{n-1}) \wedge (\mathtt{D}_n \vee \mathtt{C}_{n-1})))$$

that represents all the formula based predictors of $\mathtt{G}$ in $n+1$ steps with $n \geq 1$.

Assume now to add the following reaction to the reaction system $\mathcal{A}_1$:

$$a_6 = (\{\mathtt{C}\}, \{\}, \{\mathtt{C}\}).$$

Reaction a_6 expresses the fact that the element $\mathtt{C}$ once introduced always remains in the reaction system. This, of course, influences the formula based predictors of $\mathtt{G}$ in 3 and 4 steps given before. Indeed, now we have

$$\mathtt{fbp}(\mathtt{G}, 2) = (((\mathtt{F}_2 \vee \mathtt{E}_1) \wedge \neg(\mathtt{B}_2 \vee \mathtt{A}_1)) \vee ((\mathtt{B}_2 \vee \mathtt{A}_1) \wedge (\mathtt{D}_2 \vee \mathtt{C}_1 \vee \mathtt{C}_0))),$$

$$\mathtt{fbp}(\mathtt{G}, 3) = (((\mathtt{F}_3 \vee \mathtt{E}_2) \wedge \neg(\mathtt{B}_3 \vee \mathtt{A}_2)) \vee ((\mathtt{B}_3 \vee \mathtt{A}_2) \wedge (\mathtt{D}_3 \vee \mathtt{C}_2 \vee \mathtt{C}_1 \vee \mathtt{C}_0))).$$

Moreover, the introduction of such a reaction changes also the dependency graph discussed above by adding a cycle, since now $\mathtt{C}$ depends on itself (it is a reactant in a reaction producing itself). Nevertheless, a quite easy generalization of the formula based predictor of $\mathtt{G}$ in $n+1$ steps with $n \geq 1$ is still possible, by taking the cycle into account. The result in the example case is

$$\mathtt{fbp}(\mathtt{G}, n) = (((\mathtt{F}_n \vee \mathtt{E}_{n-1}) \wedge \neg(\mathtt{B}_n \vee \mathtt{A}_{n-1})) \vee ((\mathtt{B}_n \vee \mathtt{A}_{n-1}) \wedge (\mathtt{D}_n \vee \bigvee_{i=0}^{i \leq n-1} \mathtt{C}_i)))$$

where the cycle on $\mathtt{C}$ has required to add a parametric disjunction.

The cycle we added in the previous example is trivial: one single cyclic reaction. Cycles can be added in more complex ways, thus causing the generalized formula to become consequently more complicated. Consider, for example, the reaction system $\mathcal{A}_1$ consisting of reactions $a_1, ..., a_5$ to which we add reactions a_7 and a_8 defined as follows:

$$a_7 = (\{\mathtt{I}\}, \{\}, \{\mathtt{C}\}) \quad a_8 = \{\{\mathtt{C}\}, \{\}, \{\mathtt{I}\}.$$

In this case for the formula based predictors for $\mathtt{G}$ in 3 and 4 steps we obtain

$$\mathtt{fbp}(\mathtt{G}, 2) = (((\mathtt{F}_2 \vee \mathtt{E}_1) \wedge \neg(\mathtt{B}_2 \vee \mathtt{A}_1)) \vee ((\mathtt{B}_2 \vee \mathtt{A}_1) \wedge (\mathtt{D}_2 \vee \mathtt{C}_1 \vee \mathtt{I}_0))),$$

$$\mathtt{fbp}(\mathtt{G}, 3) = (((\mathtt{F}_3 \vee \mathtt{E}_2) \wedge \neg(\mathtt{B}_3 \vee \mathtt{A}_2)) \vee ((\mathtt{B}_3 \vee \mathtt{A}_2) \wedge (\mathtt{D}_3 \vee \mathtt{C}_2 \vee \mathtt{I}_1 \vee \mathtt{C}_0))).$$

The introduction of the reactions a_7 and a_8 changes the dependency graph that now includes a cycle between $\mathtt{C}$ and $\mathtt{I}$, since $\mathtt{C}$ depends on $\mathtt{I}$ and vice versa. By considering such a graph, a

generalization of the formula based predictor of $\mathtt{G}$ for $n+1$ steps with $n \geq 1$ is still possible, but requires the definition of two different parametric formulas[2], one for the case of n even, and the other for the case of n odd. Let us define the predicates $even(n)$ and $odd(n)$ with the obvious meaning. For n even , we obtain

$$\mathtt{fbp}(\mathtt{G}, n) = (((\mathtt{F}_n \vee \mathtt{E}_{n-1}) \wedge \neg(\mathtt{B}_n \vee \mathtt{A}_{n-1})) \vee ((\mathtt{B}_n \vee \mathtt{A}_{n-1}) \wedge (\mathtt{D}_n \vee \bigvee_{\{i \mid i \geq 1, odd(i)\}}^{i \leq n-1} \mathtt{C}_i \vee \bigvee_{\{i \mid i \geq 0, even(i)\}}^{i \leq n-1} \mathtt{I}_i)))$$

while for n odd, we have

$$\mathtt{fbp}(\mathtt{G}, n) = (((\mathtt{F}_n \vee \mathtt{E}_{n-1}) \wedge \neg(\mathtt{B}_n \vee \mathtt{A}_{n-1})) \vee ((\mathtt{B}_n \vee \mathtt{A}_{n-1}) \wedge (\mathtt{D}_n \vee \bigvee_{\{i \mid i \geq 0, even(i)\}}^{i \leq n-1} \mathtt{C}_i \vee \bigvee_{\{i \mid i \geq 1, odd(i)\}}^{i \leq n-1} \mathtt{I}_i))) .$$

Although such a generalization can be more difficult to be found than in this example, note that the finite nature of reaction systems (and consequently the finiteness of the dependency graph) should always allow us to find a generalization.

The advantages of this approach based on generalization should be clear: it allows representing formula based predictors of an element regardless the number of steps with a few parametric formulas. Moreover, starting from such generalization, we can obtain also a generalization of multi-step based predictors, simply by applying the abstraction function $\hat{\alpha}_1$ that collects the labelled objects appearing in the formula. For example, by considering again the last reaction system consisting of reactions $a_1, ..., a_5$ and a_7, a_8, we can derive the parametric multi-step based predictor of $\mathtt{G}$ for any number of steps $n+1$ with $n \geq 1$.

By applying the abstraction function for n even we obtain the set of labelled objects

$$\{\mathtt{F}_n, \mathtt{B}_n, \mathtt{D}_n, \mathtt{E}_{n-1}, \mathtt{A}_{n-1}\} \cup \bigcup_{\{i \mid i \geq 0, odd(i)\}}^{i \leq n-1} \{\mathtt{C}_i\} \cup \bigcup_{\{i \mid i \geq 1, even(i)\}}^{i \leq n-1} \{\mathtt{I}_i\} ,$$

whereas for any n odd we obtain the set of labelled objects

$$\{\mathtt{F}_n, \mathtt{B}_n, \mathtt{D}_n, \mathtt{E}_{n-1}, \mathtt{A}_{n-1}\} \cup \bigcup_{\{i \mid i \geq 0, even(i)\}}^{i \leq n-1} \{\mathtt{C}_i\} \cup \bigcup_{\{i \mid i \geq 1, odd(i)\}}^{i \leq n-1} \{\mathtt{I}_i\} .$$

7.2. Tabling

Given a reaction system and a context sequence γ, a predictor (of any of the three kinds) cannot be used by itself to predict the appearance of a given object after $n+1$ steps. It is necessary to have another context sequence γ' (equivalent to γ according to the definition of the predictor) for which it is known whether the object appears after $n+1$ steps or not.

Consequently, the way in which predictors could be used in practice (also proposed by Rozenberg in [27]) is by following a *tabling approach*. Given a predictor for object s in $n+1$ steps,

[2]The two parametric formulas could be joined in a single more complex formula. We prefer this representation into two different formulas for the sake of readability.

the idea is to construct a table that contains one row for each *equivalence class* induced by the equivalence relation associated with the predictor, and to store in each row the information about the presence or absence of s after $n + 1$ steps. In the case of set based predictors, each row of the table corresponds to a context sequence obtained by considering at each step a subset of the element of the predictor itself. Given a new context sequence γ, in order to know whether it leads to the production of s at step $n + 1$ it is enough to restrict it to the predictor objects and use the result to access the previously constructed rows.

The table to be constructed in the case of set based predictors has a very high number of rows, that is exponential in $n + 1$ multiplied by the size of the predictor. We claim that the new concepts of formula and multi-step based predictors introduced in the previous sections can result in tables that are significantly smaller than this.

In order to show how the tabling approach can be used, let us consider first the formula based predictors presented in Section 4. The equivalence classes in this case would be two, one containing all the context sequences that satisfy the formula and the other one containing all the context sequences that do not satisfy the formula. Therefore, the table could contain exactly two rows. However, in order to make the table easier to access and consistent with the tables calculated for the other predictors it is convenient to put the formula in disjunctive normal form. Then, the number of rows in the table are simply the number of different conjunctions of the disjunctive normal form of the formula obtained as predictor.

Let us consider the reaction system $\mathcal{A}_1 = (\{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}, \mathtt{G}, \mathtt{H}\}, \{a_1, a_2, a_3, a_4, a_5\})$ of Example 3.1 that consists of the following reactions

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

As shown in Example 4.3 the following formula *f-predicts* $\mathtt{G}$ in two steps,

$$\mathtt{fbp}(\mathtt{G}, 1) = (((\mathtt{F}_1 \vee \mathtt{E}_0) \wedge \neg(\mathtt{B}_1 \vee \mathtt{A}_0)) \vee ((\mathtt{B}_1 \vee \mathtt{A}_0) \wedge (\mathtt{D}_1 \vee \mathtt{C}_0))).$$

We then need to put $\mathtt{fbp}(\mathtt{G}, 1)$ in disjunctive normal form thus obtaining the following formula,

$$(\mathtt{F}_1 \wedge \neg\mathtt{B}_1 \wedge \neg\mathtt{A}_0) \vee (\mathtt{E}_0 \wedge \neg\mathtt{B}_1 \wedge \neg\mathtt{A}_0) \vee (\mathtt{B}_1 \wedge \mathtt{D}_1) \vee (\mathtt{B}_1 \wedge \mathtt{C}_0) \vee (\mathtt{A}_0 \wedge \mathtt{D}_1) \vee (\mathtt{A}_0 \wedge \mathtt{C}_0).$$

Hence, the table (reported in Table 1) consists in this case of 6 rows. Note that each row states which objects have to be present or absent at each step (irrelevant objects are not mentioned) and that only the cases in which $\mathtt{G}$ is produced are listed.

For example, we can predict that the context sequence $\gamma_1 = \{\mathtt{A}, \mathtt{C}\}, \{\mathtt{A}, \mathtt{D}\}$ will lead to the production of $\mathtt{G}$ at the second step since it satisfies the requirements of the fifth table row. On the other hand, the context sequence $\gamma_2 = \{\mathtt{B}, \mathtt{C}\}, \{\mathtt{A}, \mathtt{D}\}$ will not lead to the production of $\mathtt{G}$ since it does not satisfy any table row.

Let us now consider tabling based on multi-step based predictors, introduced in Section 5. In this case, given a multi-step based predictor $\widehat{Q} \subseteq S^n$ for $n + 1$ steps, the table has to contain one row for each possible subset of $\widehat{Q}$. Hence, the number of rows in the table is $2^{|\widehat{Q}|}$, that comes from

step 0	step 1	G at step 2?
$\neg$A	F, $\neg$B	yes
$\neg$A, E	$\neg$B	yes
	B, D	yes
C	B	yes
A	D	yes
A, C		yes

Table 1: Data table using formula based predictors

step 0	step 1	G at steps 2?
A, C, E	B, D, F	yes
$\neg$A, C, E	B, D, F	yes
A, $\neg$C, E	B, D, F	yes
A, C, $\neg$E	B, D, F	yes
$\neg$A, $\neg$C, E	B, D, F	yes
$\neg$A, C, $\neg$E	B, D, F	yes
A, $\neg$C, $\neg$E	B, D, F	yes
$\neg$A, $\neg$C, $\neg$E	B, D, F	yes
$\vdots$	$\vdots$	$\vdots$
$\neg$A, C, E	B, $\neg$D, F	yes
A, $\neg$C, E	B, $\neg$D, F	no
A, C, $\neg$E	B, $\neg$D, F	yes
$\vdots$	$\vdots$	$\vdots$

Table 2: Data table using multi-step based predictors

$2^{|\widehat{Q}\cap S_0|} \times 2^{|\widehat{Q}\cap S_1|} \times .... \times 2^{|\widehat{Q}\cap S_n|}$). We remark that although we could remove from the table the rows corresponding to cases that do not lead to the creation of the predicted object (thus obtaining a smaller table), such cases have anyway to be computed during the table construction requiring the execution of the reaction system once for each context sequence. Finally note that even if we consider only rows leading to the production on s in $n + 1$ steps, the table using the multi-step based predictor will have a bigger number of rows than the one based on the more informative formula based predictors.

Let us consider again the reaction system $\mathcal{A}_1$ of Example 3.1, previously commented. In this case, as we have illustrated in Example 5.5 the following set of labelled objects

$$\mathtt{mbp}(\mathtt{G}, 1) = \{\mathtt{A}_0, \mathtt{C}_0, \mathtt{E}_0, \mathtt{B}_1, \mathtt{D}_1, \mathtt{F}_1\}$$

m-predicts G in two steps. Hence, the table in this case has $2^6 = 64$ rows. A portion of such a table is shown in Table 2.

step 0	step 1	G at steps 2?
A, C, E, B, D, F	A, C, E, B, D, F	yes
¬A, C, E, B, D, F	A, C, E, B, D, F	yes
A, ¬C, E, B, D, F	A, C, E, B, D, F	yes
A, C, ¬E, B, D, F	A, C, E, B, D, F	yes
A, C, E, ¬B, D, F	A, C, E, B, D, F	yes
A, C, E, B, ¬D, F	A, C, E, B, D, F	yes
A, C, E, B, D, ¬F	A, C, E, B, D, F	yes
⋮	⋮	
¬A, C, E, B, D, F	A, C, E, B, ¬D, F	yes
A, ¬C, E, B, D, F	A, C, E, B, ¬D, F	no
A, C, ¬E, B, D, F	A, C, E, B, ¬D, F	yes
⋮	⋮	⋮

Table 3: Data table using set based predictors

Finally, let us now consider tabling based on set based predictors, introduced in Section 6. In this case, we observe that the table to be constructed is analogous to that of multi-step based predictors, but with the same set of objects (the predictor) to be considered for all steps.

As an example, consider again the reaction system $\mathcal{A}_1$ of Example 3.1, previously commented. In this case, as we have shown in Example 6.5 the following set of objects

$$\mathtt{sbp}(\mathtt{G}, 1) = \{\mathtt{A}, \mathtt{C}, \mathtt{E}, \mathtt{B}, \mathtt{D}, \mathtt{F}\}$$

s-predicts G in two steps. Hence, the table in this case has $2^{12} = 4196$ rows. We show very few of such rows in Table 3.

In the general case, given a set based predictor $Q \subseteq S$ for an object s at $n+1$ steps, the number of rows in the table is $2^{(n+1)\times|Q|}$, that is $2^{|Q|} \times 2^{|Q|} \times .... \times 2^{|Q|} = (2^{|Q|})^{n+1}$. Assume that $\widehat{Q} \subseteq S^n$ is a corresponding multi-step based predictor for the same object s and the same number of steps $n+1$. Since, for every $0 \leq i \leq n$ it holds $\alpha_1(\widehat{Q} \cap S_i) \subseteq Q$, we have that the table for set based predictors is always not smaller than that of multi-step based predictor. In practice, it is often the case that each $\widehat{Q} \cap S_i$ is significantly smaller than Q, thus giving tables for multi-step predictors that are very much smaller than those for the corresponding set based predictors.

It is worth noting that these tables can be reduced in size by pruning, for instance, all of the rows that correspond to x absent, or by grouping together rows which differ in objects that are not actually relevant. Alternatively, a table could be organized as an ordered binary decision diagram (OBDD), the size of which can be significantly reduced by using efficient heuristic methods [28]. All of these approaches may allow the representation of the table of a set based predictor to be reduced to a size similar to that obtained with the other notions of predictor. However, the construction of such a potentially compact representation would anyway pass through the computation of all the cases considered in the original table.

Finally, note that the generalization procedure presented in Section 7.1 could be combined with

the tabling approaches described in Section 7.2 in order to obtain a tabling parametric techniques based on the use of parametric generalized versions of formula based and multi-step predictors.

8. Application

In this section we introduce a more complex biological example, the *lac* operon expression in the *E. coli* bacterium. In genetics, an operon is a functional unit of genomic DNA containing a cluster of genes, which are all under the control of a specific regulatory signal.

The lactose operon in *Escherichia coli* is composed of a sequence of genes that are responsible for producing three enzymes for lactose degradation, namely the *lactose permease*, which is incorporated in the membrane of the bacterium and actively transports the sugar into the cell, the *beta galactosidase*, which splits lactose into glucose and galactose, and the *transacetylase*, whose role is marginal. The signal which regulates the lac operon functionality depends on the integration of two different control mechanisms, one mediated by lactose and the other by glucose. Since gene expression is a energy consuming process, *Escherichia coli* synthesizes the proteins involved in the metabolism of lactose when this nutrient is both present in the environment and the environment does not provides glucose, which is a more readily available source of energy.

We borrow the formalization as a reaction system of this biological system from [24]. Let $\mathcal{A} = (S, \{a_1, \ldots, a_{10}\})$ be the reaction system where the background set is

$$S = \{lac,\ lacI,\ I,\ I\text{-}OP,\ cya,\ cAMP,\ crp,\ CAP,\ cAMP\text{-}CAP,\ lactose,\ glucose,\ Z,\ Y,\ A\}$$

and the reaction rules are defined as follows

$$\begin{aligned}
a_1 &= (\{lac\}, \{\}, \{lac\}) && (lac \text{ operon duplication})\\
a_2 &= (\{lacI\}, \{\}, \{lacI\}) && (\text{repressor gene duplication})\\
a_3 &= (\{lacI\}, \{\}, \{I\}) && (\text{repressor gene expression})\\
a_4 &= (\{I\}, \{lactose\}, \{I\text{-}OP\}) && (\text{regulation mediated by lactose})\\
a_5 &= (\{cya\}, \{\}, \{cya\}) && (cya \text{ duplication})\\
a_6 &= (\{cya\}, \{\}, \{cAMP\}) && (cya \text{ expression})\\
a_7 &= (\{crp\}, \{\}, \{crp\}) && (crp \text{ duplication})\\
a_8 &= (\{crp\}, \{\}, \{CAP\}) && (crp \text{ expression})\\
a_9 &= (\{cAMP, CAP\}, \{glucose\}, \{cAMP\text{-}CAP\}) && (\text{regulation mediated by glucose})\\
a_{10} &= (\{lac, cAMP\text{-}CAP\}, \{I\text{-}OP\}, \{Z, Y, A\}) && (lac \text{ operon expression})
\end{aligned}$$

A schematic representation of the reaction system model is depicted in Figure 1. The three genes of the operon *LacI*, *PR* and *OP* regulate the production of the enzymes, represented by *Z*, *Y* and *A*. The genes for the three enzymes are represented, for short, by *lac*. The regulation process is as follows: gene *LacI* encodes the lac repressor *I*, which, in the absence of lactose, binds to gene *OP* (the operator). Transcription of structural genes into mRNA is performed by the RNA polymerase enzyme, which usually binds to gene *PR* (the promoter) and scans the operon from left to right by transcribing the three structural genes represented by *lac* into a single mRNA fragment.

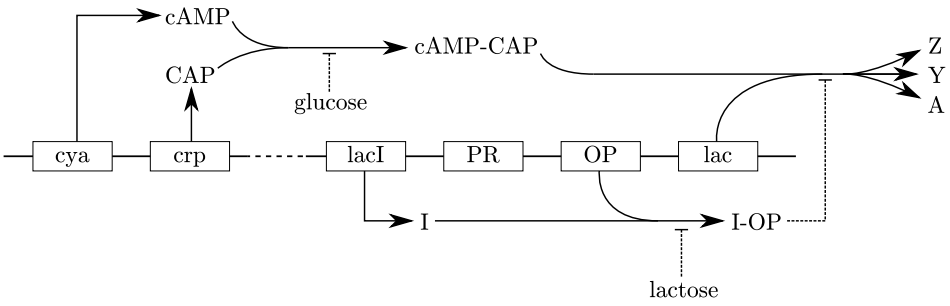

Figure 1: The *lac* operon.

When the lac repressor I is bound to gene OP (that is, the complex I-OP is present) it becomes an obstacle for the RNA polymerase, and transcription of the structural genes is not performed. On the other hand, when lactose is present inside the bacterium, it binds to the repressor thus inhibiting the binding of I to OP. This inhibition allows the transcription of genes represented by *lac* by the RNA polymerase.

Two more genes encode for the production of two particular proteins: $cAMP$ and CAP. These genes are called, respectively, *cya* and *crp*, and they are indirectly involved in the regulation of the lac operon expression. When glucose is not present, $cAMP$ and CAP proteins can produce the complex $cAMP$-CAP which can increase significantly the expression of *lac* genes. Of course, also in presence of the $cAMP$-CAP complex, the expression of the *lac* genes is inhibited by I-OP.

Note that the reactions $\{a_1, a_2, a_5, a_7\}$ are needed to ensure the permanency of the genes in the system while the reactions $\{a_4, a_9, a_{10}\}$ can only be enabled when the current state of the system does not include the inhibitors specified in each reaction. Reaction a_4 can be applied only in the absence of lactose, reaction a_9 only in the absence of glucose and reaction a_{10} only when the repressor I is not bound to the operator OP.

In [24] the authors investigate the effects on the production of enzymes Z, Y and A when the context provides both glucose and lactose, only glucose, only lactose, or none of them. The genomic elements *lac*, *lacI*, *cya* and *crp* together with the proteins I, $cAMP$ and CAP, that are normally present in the bacterium, are supplied to the systems by the starting context C_0. Then, an example context sequence $\gamma = C_0, \ldots, C_{40}$ is considered, in which every element C_i with $1 < i <= 40$ is a subset of $\{glucose, lactose\}$. Such a context sequence represents an environment in which the supply of glucose and lactose varies over time. By observing the result states $D_1, \ldots, D_{40}$ obtained by executing the reaction system, the authors conclude that the enzymes Z, Y and A are produced in a step i only if *lactose* was the only element provided to the system two steps before. Formally, this can be expressed as follows:

$$Z, Y, A \in D_i \text{ iff } C_{i-2} = \{lactose\}, \text{ with } i > 3 .$$

This conclusion has been reached empirically, by observing a single execution of the system with respect to an example context sequence. The conditions for the production of Z, Y and A can instead be studied by applying the notions of predictor we defined in this paper. Consequently, in what follows we use all three notions of predictors for (i) giving a formal ground to the conclusion reached [24], and (ii) applying a tabling approach to this biological system.

8.1. Formula based Predictor

Since the effects of all reactions can be observed after four steps we consider the formula based predictors for the enzymes Z, Y and A in 4 steps. We obtain the following formula [3]

$$\begin{aligned}\texttt{fbp}(Z,3) =&\texttt{fbp}(Y,3) = \texttt{fbp}(A,3) = \\ &((lac_3 \vee lac_2 \vee lac_1 \vee lac_0) \\ &\wedge (cAMPCAP_3 \vee ((cAMP_2 \vee cya_1 \vee cya_0) \wedge (CAP_2 \vee crp_1 \vee crp_0) \wedge \neg glucose_2)) \\ &\wedge ((\neg IOP_3 \wedge \neg I_2 \wedge \neg lacI_1 \wedge \neg lacI_0) \vee lactose_2)).\end{aligned}$$

The obtained formula is minimal with respect to the approximation order $\sqsubseteq_f$ and therefore does not require the application of simplification techniques. Moreover, the formula clearly describes the properties to be satisfied by the context sequences leading to the production of Z, Y and A after 4 steps.

A quite simple generalization can be applied in order to derive a parametric formula following the methodology described in Section 7.1. The formula based predictor of enzymes Z, Y and A in $n+1$ steps with $n \geq 3$ is given by

$$\begin{aligned}\texttt{fbp}(Z,n) =&\texttt{fbp}(Y,n) = \texttt{fbp}(A,n) = \\ &(\bigvee_{i=0}^{i\leq n} lac_i) \\ &\wedge (cAMPCAP_n \vee ((cAMP_{n-1} \vee \bigvee_{i=0}^{i\leq n-2} cya_i) \wedge (CAP_{n-1} \vee \bigvee_{i=0}^{i\leq n-2} crp_i) \wedge \neg glucose_{n-1})) \\ &\wedge ((\neg IOP_n \wedge \neg I_{n-1} \wedge \bigwedge_{i=0}^{i\leq n-2} \neg lacI_i) \vee lactose_{n-1})).\end{aligned}$$

It is worth noting that from the generalized formula based predictors for Z, Y and A we derive a much complete information with respect to the execution of the reaction system considered in [24]. Indeed, assuming the initial context $C_0 = \{lac, lacI, cya, crp, I, cAMP, CAP\}$ the formula can be simplified into $(cAMPCAP_n \vee \neg glucose_{n-1}) \wedge lactose_{n-1}$. This formula is logically equivalent to $(cAMPCAP_n \wedge lactose_{n-1}) \vee (\neg glucose_{n-1} \wedge lactose_{n-1})$. Therefore, if the environment provides only *glucose* and *lactose* as in the experiment of [24] the formula ensures exactly that the enzymes Z, Y and A are produced at step $n+1$ only if *lactose* was the only element provided to the system two steps before. Moreover, the formula tells us that the production of enzymes Z, Y and A in $n+1$ steps is possible also if the environment provides *lactose* two steps before and *cAMPCAP* one step before.

We also investigate the application of the tabling approach for formula based predictors, discussed in Section 7.2. For formula based predictors the rows correspond exactly to the number of

[3] All formula based predictors of this section were automatically computed by using the tool available here: [29]

disjunctions of the disjunctive normal form of the formula based predictor. In order to build the table we have to put the formula $\mathtt{fbp}(Z,3) = \mathtt{fbp}(Y,3) = \mathtt{fbp}(A,3)$ in disjunctive normal form obtaining

$$\begin{aligned}
\bigvee_{i=0}^{i\leq 3} (\quad & (lac_i \wedge cAMPCAP_3 \wedge \neg IOP_3 \wedge \neg I_2 \wedge \neg lacI_1 \wedge \neg lacI_0) \\
& \vee (lac_i \wedge cAMPCAP_3 \wedge lactose_2) \\
& \vee (lac_i \wedge cAMP_2 \wedge CAP_2 \wedge \neg glucose_2 \wedge \neg IOP_3 \wedge \neg I_2 \wedge \neg lacI_1 \wedge \neg lacI_0) \\
& \vee (lac_i \wedge cAMP_2 \wedge CAP_2 \wedge \neg glucose_2 \wedge lactose_2) \\
& \vee (lac_i \wedge cAMP_2 \wedge crp_1 \wedge \neg glucose_2 \wedge \neg IOP_3 \wedge \neg I_2 \wedge \neg lacI_1 \wedge \neg lacI_0) \\
& \vee (lac_i \wedge cAMP_2 \wedge crp_1 \wedge \neg glucose_2 \wedge lactose_2) \\
& \vee (lac_i \wedge cAMP_2 \wedge crp_0 \wedge \neg glucose_2 \wedge \neg IOP_3 \wedge \neg I_2 \wedge \neg lacI_1 \wedge \neg lacI_0) \\
& \vee (lac_i \wedge cAMP_2 \wedge crp_0 \wedge \neg glucose_2 \wedge lactose_2) \\
& \vee (lac_i \wedge cya_1 \wedge CAP_2 \wedge \neg glucose_2 \wedge \neg IOP_3 \wedge \neg I_2 \wedge \neg lacI_1 \wedge \neg lacI_0) \\
& \vee (lac_i \wedge cya_1 \wedge CAP_2 \wedge \neg glucose_2 \wedge lactose_2) \\
& \vee (lac_i \wedge cya_1 \wedge crp_1 \wedge \neg glucose_2 \wedge \neg IOP_3 \wedge \neg I_2 \wedge \neg lacI_1 \wedge \neg lacI_0) \\
& \vee (lac_i \wedge cya_1 \wedge crp_1 \wedge \neg glucose_2 \wedge lactose_2) \\
& \vee (lac_i \wedge cya_1 \wedge crp_0 \wedge \neg glucose_2 \wedge \neg IOP_3 \wedge \neg I_2 \wedge \neg lacI_1 \wedge \neg lacI_0) \\
& \vee (lac_i \wedge cya_1 \wedge crp_0 \wedge \neg glucose_2 \wedge lactose_2) \\
& \vee (lac_i \wedge cya_0 \wedge CAP_2 \wedge \neg glucose_2 \wedge \neg IOP_3 \wedge \neg I_2 \wedge \neg lacI_1 \wedge \neg lacI_0) \\
& \vee (lac_i \wedge cya_0 \wedge CAP_2 \wedge \neg glucose_2 \wedge lactose_2) \\
& \vee (lac_i \wedge cya_0 \wedge crp_1 \wedge \neg glucose_2 \wedge \neg IOP_3 \wedge \neg I_2 \wedge \neg lacI_1 \wedge \neg lacI_0) \\
& \vee (lac_i \wedge cya_0 \wedge crp_1 \wedge \neg glucose_2 \wedge lactose_2) \\
& \vee (lac_i \wedge cya_0 \wedge crp_0 \wedge \neg glucose_2 \wedge \neg IOP_3 \wedge \neg I_2 \wedge \neg lacI_1 \wedge \neg lacI_0) \\
& \vee (lac_i \wedge cya_0 \wedge crp_0 \wedge \neg glucose_2 \wedge lactose_2) \qquad)
\end{aligned}$$

As a consequence, the data table that describes all the context sequences leading to the production of enzymes Z, Y and A after 4 steps has 80 horizontal lines.

8.2. Multi-step based Predictor

In the case of multi-step based predictor for the enzymes Z, Y and A we obtain the following set of labelled objects (that turns out to be minimal)

$$\begin{aligned}
\mathtt{mbp}(Z,3) = & \mathtt{mbp}(Y,3) = \mathtt{mbp}(A,3) = \\
& \{lac_0, cya_0, crp_0, lacI_0, lac_1, cya_1, crp_1, lacI_1, lac_2, cAMP_2, CAP_2, glucose_2, I_2, lactose_2, \\
& lac_3, cAMPCAP_3, IOP_3\}.
\end{aligned}$$

Moreover, since we found a quite simple generalization of the formula based predictor of enzymes Z, Y and A we can derive a corresponding generalized multi-step based predictor. To this aim we have to apply the abstraction function to the parametric formula describing the formula based predictor of Z, Y and A. Hence, we obtain

$$\begin{aligned}\widehat{\alpha}_1(\texttt{fbp}(Z,n) =& \widehat{\alpha}_1(\texttt{fbp}(Y,n) = \widehat{\alpha}_1(\texttt{fbp}(A,n)) = \\ & \{cAMP_{n-1}, CAP_{n-1}, glucose_{n-1}, I_{n-1}, lactose_{n-1}, cAMPCAP_n, IOP_n\} \cup \\ & \bigcup_{i=0}^{i\leq n-2} \{cya_i, crp_i, lacI_i\} \cup \bigcup_{i=0}^{i\leq n} \{lac_i\}\end{aligned}$$

The previous set of labelled objects defines the minimal multi-step based predictor for Z, Y and A in $n+1$ steps with $n \geq 3$. Multi-step predictors are less precise than formula based predictors. However, they provide relevant information for understanding the dynamic behavior of the system. For example, we can deduce that both *glucose* and *lactose* has to be observed at $n-1$ step only. Therefore, their presence in the environment is not relevant in any other step.

As formula based predictors also multi-step based predictors can be used for recording in a data table information on context sequences that lead (or not lead) to the production of X, Y and A in 4 steps. In this case, the table has 2^{17} rows given that it contains one row for each possible subset of the predictor.

8.3. Set based Predictor

In the case of set based predictor for the enzymes Z, Y and A we obtain the following set of object (that turns out to be minimal)

$$\begin{aligned}\texttt{sbp}(Z,3) =& \texttt{sbp}(Y,3) = \texttt{sbp}(A,3) = \\ & \{lac, cya, crp, lacI, cAMP, CAP, glucose, I, lactose, cAMPCAP, IOP\}.\end{aligned}$$

Set based predictors are less precise than multi-step based predictors. Actually, the information given by the set based predictors says all the objects have to observed apart from Z, Y and A.

As in the case of multi-step based predictors also set based predictors can be used for recording in a data table information on context sequences that lead (or not lead) to the production of X, Y and A in 4 steps. In this case, the table has $(2^{11})^4$ rows given that it contains one row for each possible combination of four subsets of the predictor.

9. Conclusions

In this paper we have presented three different notions of predictor using an abstract interpretation framework: (i) the formula based predictor, which precisely models the necessary conditions for an object s to be produced in n steps, (ii) the multi-step based predictor, which models the objects to be observed at each step for determining the production of s in n steps, and (iii) the set based predictor, that is a revised version of the predictor proposed in [18]. Each notion of predictor comes equipped with an effective operator able to compute a predictor of an object s in n steps.

The relation between the different operators is formalized in the abstract interpretation framework in terms of abstraction functions. Moreover, we have proved that each class of predictors has a minimal element. In the case of formula based predictors such a minimum can be computed by applying a simplification procedure to the boolean formula obtained by the corresponding operator. In the cases of multi-step and set based predictors, in general it is not possible to compute the minimal element from the results of the corresponding operators. A minimal multi-step or set based predictor can be obtained from the minimal formula based predictor by applying abstraction functions. The tabling procedures based on the previous new notions of predictor have been illustrated and compared in terms of space occupied by the constructed tables. We have also proposed a generalization of the concept of predictor by considering parametric formula based and multi-step based predictors that enable prediction of the production of an object $s \in S$ in a *arbitrary number of steps*.

As future work we plan to define predictors that allow constraints on the context sequences to be considered. For example, constraints could represent the constant absence of some objects from the context sequence, or the appearance of some objects in the context after a given number of steps, or the periodic appearance and disappearance of some objects from the context. Such constraints will be expressed by means of a suitable temporal logic to be defined. Moreover, new operators for the computation of these new kinds of predictor will be investigated.

Appendix A. Proofs of Theorems in Section 4

Theorem 4.1. *Let $\mathcal{A} = (S, A)$ be a reaction system. If two formulas $f, f' \in F_{S^n}$* f-predict *$s \in S$ in $n+1$ steps, then either $f \equiv_l f'$ or $f \equiv_l \neg f'$.*

Proof. Note that $\approx^n_f$ determines a partition in two sets of all context sequences, since either a context sequence γ models f or it does not model f. Each context sequence in the same subset of the partition share the same behaviour w.r.t. the fact that $s \in D_{n+1}$ or $s \notin D_{n+1}$. It should be clear that $\approx^n_f$ and $\approx^n_{\neg f}$ induce exactly the same partition on context sequences. Assume by contradiction that $f \not\equiv_l f'$ and $f \not\equiv_l \neg f'$. Hence, let us assume γ_1 and γ_2 such that γ_1 satisfies both formulas f and f', while γ_2 satisfies only one of the two. Note that since $f \not\equiv_l f'$ and $f \not\equiv_l \neg f'$, such γ_1 and γ_2 exist. We have that $\gamma_1 \approx^n_f \gamma_2$ but $\gamma_1 \not\approx^n_{f'} \gamma_2$. Assume, without loosing generality, that $s \in D^1_{n+1}$. Since *f f-predicts s in $n+1$ steps*, we can conclude that $s \in D^2_{n+1}$. On the other hand since $\gamma_1 \not\approx^n_{f'} \gamma_2$ and *f' f-predicts s in $n+1$ steps*, we can conclude that $s \notin D^2_{n+1}$. This gives a contradiction. □

Theorem 4.2. *Let $\mathcal{A} = (S, A)$ be a reaction system and $f \in F_{S^n}$ be a formula such that f* f-predicts *$s \in S$ in $n+1$ steps. Then any formula $f' \in F_{S^n}$ such that either $f' \equiv_l f$ or $f' \equiv_l \neg f$* f-predict *s in $n+1$ steps.*

Proof. It follows from the observation that the equivalence relations $\approx_{f'}$ and $\approx_f$, for $f' \equiv_l f$ or $f' \equiv_l \neg f$, induce the same partition on all context sequences. □

Theorem 4.3. *Let $\mathcal{A} = (S, A)$ be a reaction system $s \in S$ and let $f_1, f_2 \in F_{S^n}$ be two propositional logic formula. If both f_1 and f_2 f-*predict *s in $n+1$ steps, then there exist $f \in F_{S^n}$ such that $f \sqsubseteq_f f_1$, $f \sqsubseteq_f f_2$ and f f-*predicts *s in $n+1$ steps.*

Proof. Let us assume first $f_1 \equiv_l f_2$. In this case we can consider $f = simpl(f_1) = simpl(f_2)$. If instead $f_1 \equiv_l \neg f_2$, let $f_2' = \neg f_2$. By the previous case we have $f = simpl(f_1) = simpl(f_2') \sqsubseteq_f f_2'$. By definition of $\sqsubseteq_f$ this implies $f \sqsubseteq_f f_2$. As a consequence of Theorem 4.2 in both the cases f f-predicts s in $n+1$ steps. □

Theorem 4.4. *Let $\mathcal{A} = (S, A)$ be a reaction system and $s \in S$. For any n-step context sequence $\gamma = C_0, C_1, \ldots, C_n$ it holds:*

$$s \in D_{n+1} \iff \gamma \models \texttt{fbp}(s, n)$$

where $\delta = D_0, D_1, \ldots, D_n$ is the result sequence corresponding to γ and $D_{n+1} = \text{res}_{\mathcal{A}}(C_n \cup D_n)$.

Proof. By definition of `fbp` what we have to prove is actually $s \in D_{n+1} \iff \gamma \models \texttt{fbs}(cause(s), n)$. The proof is done by induction on n.

Base case: $n = 0$. We start by proving the $\Rightarrow$ implication, namely $s \in D_1 \Rightarrow \gamma \models \texttt{fbs}(cause(s), 0)$.

From $s \in D_1$ it follows that there must exist in $\mathcal{A}$ at least a reaction $a_v = (R_v, I_v, P_v)$ with $s \in P_v$ such that $s \in \text{res}_{a_v}(C_0 \cup D_0)$, 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

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

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 $\models$ we have $R_v \subseteq C_0$ 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 $\mathcal{A}$ a reaction $a = (R, I, P)$ with $s \in P$ that has been applied in the $n+1$-th execution step of the reaction systems. This implies that $R \subseteq C_n \cup D_n$ and $I \cap (C_n \cup D_n) = \emptyset$. As a consequence, there exist two disjoint sets R_D and R_C such that $R = R_D \cup R_C$, $R_D \subseteq D_n, R_C \subseteq C_n$ and $R_D \cap C_n = \emptyset$.

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 $\gamma \models \texttt{fbs}(cause(s), n)$ can be reduced to the proof of $\gamma \models \texttt{fbs}(ap(a), n)$.

By definition of ap we have

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

that is equivalent to

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

that, in turn, by definition of fbs, is equivalent to

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

Since $R_D \cap C_n = \emptyset$, $R_C \subseteq C_n$ and $I \cap C_n = \emptyset$, this formula can be simplified as follows:

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

Now, by induction hypothesis we have $s \in D_n \iff \gamma \models \texttt{fbs}(cause(s), n-1)$. Since $R_D \subseteq D_n$ and $I \cup D_n = \emptyset$, it follows that our simplified formula holds.

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

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

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 $\texttt{fbs}(ap(a), n)$ is modeled by γ. Let us assume $a = (R, I, P)$, we have

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

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

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

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

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

that, by definition of fbs, is equivalent to

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

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:

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

Consequently, we have that for each $s^d \in R_D$ it holds $\gamma \models \texttt{fbs}(cause(s^d), n-1)$, whereas for each $s'' \in I$ it holds $\gamma \not\models \texttt{fbs}(cause(s''), n-1)$. From this, by induction hypothesis we obtain $R_D \subseteq D_n$ and $I \cap D_n = \emptyset$. These conditions, together with conditions $R_C \subseteq C_n$ and $I \cap C_n = \emptyset$, allow us to conclude that $R = R_C \cup R_D \subseteq C_n \cup D_n$ and $I \cap (C_n \cup D_n) = \emptyset$. As a consequence, $s \in \mathrm{res}_a(C_n \cup D_n)$, that is $s \in D_{n+1}$. □

Lemma 4.5. *Let $\mathcal{A} = (S, A)$ be a reaction system, $s \in S$ and $f \in F_{S^n}$ be a propositional logical formula. If for all n-step context sequences $\gamma = C_0, C_1, \ldots C_n$ it holds $(s \in D_{n+1} \iff \gamma \models f)$ then f f-*predicts *s in $n+1$ steps, where $D_{n+1} = res_{\mathcal{A}}(C_n \cup D_n)$.*

Proof. Let us consider two n-step context sequences γ_1 and γ_2 such that $\gamma_1 \approx^n_f \gamma_2$. By definition of the equivalence relation this means that $\gamma_1 \models f \Leftrightarrow \gamma_2 \models f$. Since, by hypothesis, for formula f $(s \in D^1_{n+1} \iff \gamma_1 \models f)$ and $(s \in D^2_{n+1} \iff \gamma_2 \models f)$ hold, we can conclude that $s \in D^1_{n+1} \iff s \in D^2_{n+1}$. Conversely, if $(s \in D^1_{n+1} \iff \gamma_1 \models f)$ and $(s \in D^2_{n+1} \iff \gamma_2 \models f)$ we can deduce that both $\gamma_1 \models f$ and $\gamma_2 \models f$. Hence, $\gamma_1 \models f \Leftrightarrow \gamma_2 \models f$, which implies $\gamma_1 \approx^n_f \gamma_2$, by definition of the equivalence relation. □

Appendix B. Proofs of Theorems in Section 5

We first define the following well-founded order on $\mathbb{N} \times F_S$ that will be used in some of the following proofs.

Definition B.1. *Let the pairs* $(i, f), (j, f') \in \mathbb{N} \times F_S$. *We say that* $(i, f) \leq_c (j, f')$ *iff either* $i < j$ *or* $i = j$ *and* f *is a sub-formula of* f' *(namely a fragment of formula* f*).*

Theorem 5.1. *Let* $\mathcal{A} = (S, A)$ *be a reaction system and* $s \in S$. *There exists exactly one minimal (w.r.t. the set inclusion order)* $\widehat{Q} \subseteq S^n$ *that* m-predicts s *in* $n + 1$ *steps.*

The following proof follows the lines of the proof of Theorem 1 of [18].

Proof. Assume, by contradiction, that there exists two distinct subsets $\widehat{Q}_1$ and $\widehat{Q}_2$ of S^n that m-predicts s in $n + 1$ steps and are minimal. Then consider $\widehat{Q}_3 = \widehat{Q}_1 \cap \widehat{Q}_2$. Since $\widehat{Q}_3 \subseteq \widehat{Q}_1$ and $\widehat{Q}_3 \subseteq \widehat{Q}_2$ and $\widehat{Q}_1$ and $\widehat{Q}_2$ are minimal, then $\widehat{Q}_3$ does not m-predicts s in $n + 1$ steps.

Hence, by definition of m-predictor, there exist two n-step context sequences $\gamma_1 = C_0^1, C_1^1, \dots C_n^1$ and $\gamma_2 = C_0^2, C_1^2, \dots C_n^2$ such that $\gamma_1 \simeq^n_{\widehat{Q}_3} \gamma_2$ but $s \in D^1_{n+1}$ and $s \notin D^2_{n+1}$. Let us consider the n-step context sequence $\gamma_3 = C_0^3, C_1^3, \dots C_n^3$ such that, for $i \in \{0, \dots, n\}$, $C_i^3 = (C_i^1 \cap \{s \mid s_i \in \widehat{Q}_1\}) \cup (C_i^2 \cap \{s \mid s_i \in \widehat{Q}_2\})$. Now we want to show that $\gamma_3 \simeq^n_{\widehat{Q}_1} \gamma_1$ and $\gamma_3 \simeq^n_{\widehat{Q}_2} \gamma_2$.

In particular we show the case of $\gamma_3 \simeq^n_{\widehat{Q}_1} \gamma_1$ (the case of $\gamma_3 \simeq^n_{\widehat{Q}_2} \gamma_2$ is analogous). Note that, for $i \in \{0, \dots, n\}$, we have

$$C_i^3 \cap \{s \mid s_i \in \widehat{Q}_1\} = (C_i^1 \cap \{s \mid s_i \in \widehat{Q}_1\}) \cup (C_i^2 \cap \{s \mid s_i \in \widehat{Q}_2 \cap \{s \mid s_i \in \widehat{Q}_1\}).$$

Now, for $i \in \{0, \dots, n\}$ we have

$$(C_i^2 \cap \{s \mid s_i \in \widehat{Q}_2\} \cap \{s \mid s_i \in \widehat{Q}_1\}) = (C_i^2 \cap \{s \mid s_i \in (\widehat{Q}_1 \cap \widehat{Q}_2)\}) = (C_i^2 \cap \{s \mid s_i \in \widehat{Q}_3\}).$$

Moreover, since by hypothesis $\gamma_1 \simeq^n_{\widehat{Q}_3} \gamma_2$ we have also for $i \in \{0, \dots, n\}$,

$$(C_i^2 \cap \{s \mid s_i \in \widehat{Q}_3\} = (C_i^1 \cap \{s \mid s_i \in \widehat{Q}_3\}.$$

Now remember that $\widehat{Q}_3 \subseteq \widehat{Q}_1$ so that $i \in \{0, \dots, n\}$, we have $(C_i^1 \cap \{s \mid s_i \in \widehat{Q}_3\} \subseteq (C_i^1 \cap \{s \mid s_i \in \widehat{Q}_1\})$ and $C_i^3 \cap \{s \mid s_i \in \widehat{Q}_1\} = (C_i^1 \cap \{s \mid s_i \in \widehat{Q}_1\})$. Hence we can conclude that $\gamma_3 \simeq^n_{\widehat{Q}_1} \gamma_1$.

On one hand, $\widehat{Q}_1$ m-predicts s in $n + 1$ steps so that $s \in D^3_{n+1} \Leftrightarrow s \in D^1_{n+1}$. On the other hand, $\widehat{Q}_2$ m-predicts s in $n + 1$ steps so that $s \in D^3_{n+1} \Leftrightarrow s \in D^2_{n+1}$. Thus, we obtain a contradiction with the assumption that $s \in D^1_{n+1}$ and $s \notin D^2_{n+1}$. □

Theorem 5.2 (Galois Insertion)**.** *The pair of functions* (α_1, γ_1) *in Definition 5.3 is a* Galois insertion *between* $(\wp(\mathcal{F}), \subseteq)$ *and* $(\wp(\mathcal{S}), \subseteq)$.

Proof. We show that the pair of functions (α_1, γ_1) is a Galois insertion. To this aim we need to prove that :

- We have to prove that $\alpha_1 : \wp(\mathcal{F}) \to \wp(\mathcal{S})$ and $\gamma_1 : \wp(\mathcal{S}) \to \wp(\mathcal{F})$ are monotone. Given two sets of formulas $F_1, F_2 \in \wp(\mathcal{F})$, such that $F_1 \subseteq F_2$ we have $\alpha_1(F_1) = \bigcup_{f \in F_1} \widehat{\alpha}_1(f) = \bigcup_{f \in F_1} atom(f)$ and $\alpha_1(F_2) = \bigcup_{f' \in F_2} \widehat{\alpha}_1(f') = \bigcup_{f' \in F_2} atom(f')$. Since $F_1 \subseteq F_2$ we can conclude that $\alpha_1(F_1) \subseteq \alpha_1(F_2)$.

  Conversely, given two sets of labelled objects $\widehat{Q}_1, \widehat{Q}_2 \in \wp(\mathcal{S})$ such that $\widehat{Q}_1 \subseteq \widehat{Q}_2$ we obtain $\gamma_1(\widehat{Q}_1) = \{f \mid \widehat{\alpha}_1(f) \subseteq \widehat{Q}_1\} = \{f \mid atom(f) \subseteq \widehat{Q}_1\}$ and $\gamma_1(\widehat{Q}_2) = \{f' \mid \widehat{\alpha}_1(f') \subseteq \widehat{Q}_2\} = \{f' \mid atom(f') \subseteq \widehat{Q}_2\}$. Since $\widehat{Q}_1 \subseteq \widehat{Q}_2$ also in this case we can conclude that $\gamma_1(\widehat{Q}_1) \subseteq \gamma_1(\widehat{Q}_2)$.

- We have to prove that, for each $F \in \wp(\mathcal{F})$, $F \subseteq \gamma_1(\alpha_1(F))$. We have that $\alpha_1(F) = \bigcup_{f\in F} \widehat{\alpha}_1(f) = \bigcup_{f\in F} atom(f)$ and therefore

  $$\gamma_1(\alpha_1(F)) = \gamma_1(\bigcup_{f\in F} atom(f)) = \{f' \mid \widehat{\alpha}_1(f') \subseteq (\bigcup_{f\in F} atom(f)) = \{f' \mid atom(f') \subseteq (\bigcup_{f\in F} atom(f))\}.$$

  We can conclude that $F \subseteq \gamma_1(\alpha_1(F))$ given that the set of all possible formulas on the symbols appearing in the formulas of F obviously contains the set of formulas F.

- We have to prove that, for each $\widehat{Q} \in \wp(\mathcal{S})$, $\alpha_1(\gamma_1(\widehat{Q})) = \widehat{Q}$. We have that $\gamma_1(\widehat{Q}) = \{f \mid \widehat{\alpha}_1(f) \subseteq \widehat{Q}\}$ and therefore

  $$\gamma_1(\widehat{Q}) = \{f \mid \widehat{\alpha}_1(f) \subseteq \widehat{Q}\} = \{f \mid atom(f) \subseteq \widehat{Q}\} = F.$$

  Moreover, we have that

  $$\alpha_1(F) = \bigcup_{f\in F} \widehat{\alpha}_1(f) = \bigcup_{f\in F} atom(f) = \bigcup_{f\in\{f'\mid atom(f')\subseteq\widehat{Q}\}} atom(f) = \widehat{Q}.$$

□

Lemma 5.3. *Let γ_1, γ_2 be two n-step context sequences and $f \in F_{S^n}$ be a propositional logic formula. We have that*

$$\gamma_1 \simeq^n_{\widehat{\alpha}_1(f)} \gamma_2 \implies \gamma_1 \approx^n_f \gamma_2.$$

Proof. Let γ_1, γ_2 be two n-step context sequences such that $\gamma_1 \simeq^n_{\widehat{\alpha}_1(f)} \gamma_2$ where $\widehat{\alpha}_1(f) = atom(f)$. By definition of the equivalence relation $\simeq^n_{atom(f)}$ we have that each labelled object $s_i \in atom(f)$ is either present or absent in both the i-th element of the context sequence γ_1 and in the i-th element of the context sequence γ_2. This implies that f must either hold or not hold on both context sequences, namely $\gamma_1 \models f \iff \gamma_2 \models f$. By definition of the equivalence relation $\approx^n_f$ we derive that $\gamma_1 \approx^n_f \gamma_2$. □

Theorem 5.4. *Let $\mathcal{A} = (S, A)$ be a reaction system, $s \in S$ and $f \in F_{S^n}$ be a propositional logic formula. We have that*

- *if f* f-predicts *s in $n+1$ steps then $\widehat{\alpha}_1(f)$* m-predicts *s in $n+1$ steps;*
- *if f is a* minimal f-predictor *of s in $n+1$ steps then $\widehat{\alpha}_1(f)$ is a* minimal m-predictor *of s in $n+1$ steps.*

Proof.

- By definition of f-predictor, for any two n-step context sequences γ_1 and γ_2 we have that $\gamma_1 \approx^n_f \gamma_2$ implies $s \in D^1_{n+1} \iff s \in D^2_{n+1}$. By Lemma 5.3 we also have that $\gamma_1 \simeq^n_{\widehat{\alpha}_1(f)} \gamma_2$ implies $\gamma_1 \approx^n_f \gamma_2$. As a consequence, if $\gamma_1 \simeq^n_{\widehat{\alpha}_1(f)s} \gamma_2$ then $s \in D^1_{n+1} \iff s \in D^2_{n+1}$. Thus, $\widehat{\alpha}_1(f)$ m-predicts s in $n+1$ steps.

- Let us assume, by contradiction, that there exists a set $\widehat{Q} \subseteq S^n$ that m-predicts s in $n+1$ steps and such that $\widehat{Q} \subset \widehat{\alpha}_1(f)$ where $\widehat{\alpha}_1(f) = atom(f)$. Let $\gamma_1 = C_0^1 \dots C_n^1, \dots, \gamma_w = C_0^w \dots C_n^w$ be all of the possible n-step context sequences on objects S. Furthermore, let $D_{n+1}^1, \dots, D_{n+1}^w$ be the corresponding results at step $n+1$, produced by the execution of the reaction system for each $1 \in \{1, \dots, w\}$.

  For each $i \in \{1, \dots, w\}$ we consider the set of labelled symbols $\widehat{Q}_i = C_0^i \cup \dots \cup C_n^i$ and we define the following formula

$$f' = \bigvee_{\{i \in \{1,\dots,w\} | s \in D_{n+1}^i\}} \left(\bigwedge_{s_j \in (\widehat{Q}_i \cap \widehat{Q})} s_j \wedge \bigwedge_{s_j \in (\widehat{Q} \setminus \widehat{Q}_i)} \neg s_j \right).$$

  It is easy to see that the formula f' f-predicts s in $n+1$ steps. Moreover, we have that $atom(f') \subset atom(f)$ since $\widehat{Q} \subset atom(f)$. This contradicts the assumption that the formula f f-predicts s in $n+1$ steps and is minimal.

□

Corollary 5.5. *Let $\mathcal{A} = (S, A)$ be a reaction system and $s \in S$. We have that*

- $\widehat{\alpha}_1(\mathtt{fbp}(s, n))$ m-predicts *s in $n+1$ steps;*
- $\widehat{\alpha}_1(simpl(\mathtt{fbp}(s, n)))$ *is the* minimal m-predictor *of s in $n+1$ steps.*

Proof.

- Follows from Corollary 4.6 and Theorem 5.4.
- Follows from Corollary 4.7 and Theorem 5.4.

□

Theorem 5.6. *Let $\mathcal{A} = (S, A)$ be a reaction system and $s \in S$. Given $n \in \mathbb{N}$ we have that*

$$\widehat{\alpha}_1(\mathtt{fbp}(s, n)) = \mathtt{mbp}(s, n).$$

Proof. By definition we have that $\mathtt{fbp}(s, n) = \mathtt{fbs}(cause(s), n)$ and analogously $\mathtt{mbp}(s, n) = \mathtt{mbs}(cause(s), n)$. Therefore it is enough to prove that $\widehat{\alpha}_1(\mathtt{fbs}(f, i)) = \mathtt{mbs}(f, i)$ for each formula $f \in F_S$ and for each $i \in \mathbb{N}$.

The proof proceeds by induction on the well-founded order $\leq_c$ of Definition B.1.

The base case is when $i = 0$ and either $f = s$ where s a positive literal or $f = true$ or $f = false$. Assume first that $f = s$. Then, it is immediate to prove that $\widehat{\alpha}_1(\mathtt{fbs}(s, 0)) = \widehat{\alpha}_1(s_0) = \{s_0\} = \mathtt{mbs}(s, 0)$. For $f = true$, we have that $\widehat{\alpha}_1(\mathtt{fbs}(true, i)) = \widehat{\alpha}_1(true) = \emptyset = \mathtt{mbs}(true, i)$ for any $i \in \mathbb{N}$, and, analogously for the case $f = false$.

Then, we have to prove that $\mathtt{fbs}(f, i) = \widehat{\alpha}_1(\mathtt{fbs}(f, i))$ holds, knowing that $\mathtt{fbs}(f', j) = \widehat{\alpha}_1(\mathtt{fbs}(f', j))$ for all $f \in F_S$ and $j \in \mathbb{N}$ such that $(j, f') <_c (i, f)$.

We have the following different cases:

- $i > 0$ and $f = s$ where s is a positive literal. Then we have

$$\widehat{\alpha}_1(\texttt{fbs}(s,i)) = \widehat{\alpha}_1(s_i \vee \texttt{fbs}(cause(s), i-1)) = \{s_i\} \cup \widehat{\alpha}_1(\texttt{fbs}(cause(s), i-1)).$$

  On the other hand we have $\texttt{mbs}(s,i) = \{s_i\} \cup \texttt{mbs}(cause(s), i-1)$. We can conclude that $\widehat{\alpha}_1(\texttt{fbs}(s,i)) = \texttt{mbs}(s,i)$ given that by induction hypothesis we have $\widehat{\alpha}_1(\texttt{fbs}(cause(s), i-1) = \texttt{mbs}(cause(s), i-1)$.

- $f = f_1 \wedge f_2$. Then, by definition, we have $\texttt{fbs}(f_1 \wedge f_2, i) = \texttt{fbs}(f_1, i) \wedge \texttt{fbs}(f_2, i)$. Thus, by applying the abstraction function we have,

$$\widehat{\alpha}_1(\texttt{fbs}(f_1 \wedge f_2, i)) = \widehat{\alpha}_1((\texttt{fbs}(f_1, i) \wedge (\texttt{fbs}(f_2, i))) = \widehat{\alpha}_1(\texttt{fbs}(f_1, i)) \cup \widehat{\alpha}_1(\texttt{fbs}(f_2, i)).$$

  On the other hand, we have $\texttt{mbs}(f_1 \wedge f_2, i) = \texttt{mbs}(f_1, i) \cup \texttt{mbs}(f_2, i)$. We can conclude that $\texttt{mbs}(f_1 \wedge f_2, i) = \widehat{\alpha}_1(\texttt{fbs}(f_1 \wedge f_2, i))$ given that by induction hypothesis we have $\widehat{\alpha}_1(\texttt{fbs}(f_1, i)) = \texttt{mbs}(f_1, i)$ and $\widehat{\alpha}_1(\texttt{fbs}(f_2, i)) = \texttt{mbs}(f_2, i)$.

- $f = f_1 \vee f_2$. Then, by definition, we have $\texttt{fbs}(f_1 \vee f_2, i) = \texttt{fbs}(f_1, i) \vee \texttt{fbs}(f_2, i)$. Thus, by applying the abstraction function we have,

$$\widehat{\alpha}_1(\texttt{fbs}(f_1 \vee f_2, i)) = \widehat{\alpha}_1((\texttt{fbs}(f_1, i) \vee (\texttt{fbs}(f_2, i))) = \widehat{\alpha}_1(\texttt{fbs}(f_1, i)) \cup \widehat{\alpha}_1(\texttt{fbs}(f_2, i)).$$

  On the other hand, we have $\texttt{mbs}(f_1 \vee f_2, i) = \texttt{mbs}(f_1, i) \cup \texttt{mbs}(f_2, i)$. We can conclude that $\texttt{mbs}(f_1 \vee f_2, i) = \widehat{\alpha}_1(\texttt{fbs}(f_1 \vee f_2, i))$ given that by induction hypothesis we have $\widehat{\alpha}_1(\texttt{fbs}(f_1, i)) = \texttt{mbs}(f_1, i)$ and $\widehat{\alpha}_1(\texttt{fbs}(f_2, i)) = \texttt{mbs}(f_2, i)$.

- $f = \neg f'$. Then then by definition we have $\texttt{fbs}(\neg f', i) = \neg\texttt{fbs}(f', i)$. Thus, by applying the abstraction function we have,

$$\widehat{\alpha}_1(\texttt{fbs}(\neg f', i)) = \widehat{\alpha}_1(\neg(\texttt{fbs}(f', i))) = \widehat{\alpha}_1(\texttt{fbs}(f', i)).$$

  On the other hand, we have $\texttt{mbs}(\neg f', i) = \texttt{mbs}(f', i)$. We can conclude that $\texttt{mbs}(\neg f', i) = \widehat{\alpha}_1(\texttt{fbs}(\neg f', i))$ given that by induction hypothesis we have $\widehat{\alpha}_1(\texttt{fbs}(f', i)) = \texttt{mbs}(f', i)$.

- $f = (f')$ or $f = true$ or $f = false$ the proof is immediate reasoning analogously as in the previous cases.

□

Corollary 5.7. *Let $\mathcal{A} = (S, A)$ be a reaction system and $s \in S$. We have that* $\texttt{mbp}(s, n)$ m-predicts *s in $n + 1$ steps.*

Proof. Follows from Theorem 5.6 and Corollary 5.5. □

Appendix C. Proofs of Theorems in Section 6

Theorem 6.1. *Let $\mathcal{A} = (S, A)$ be reaction system and $s \in S$. There exists exactly a unique minimal (w.r.t. the set inclusion order) $Q \subseteq S$ that s-predicts s in $n+1$ steps.*

Proof. The proof of this theorem is in [18] □

Theorem 6.2. *The pair of functions (α_2, γ_2) in Definition 6.3 is a* Galois insertion *between $(\mathcal{P}(\mathcal{S}), \subseteq)$ and $(\mathcal{P}(S), \subseteq)$.*

Proof. Analogous to the proof of the Theorem 5.2. □

Lemma 6.3. *Let γ_1, γ_2 be two n-step context sequences and $\widehat{Q} \subseteq S^n$. We have that*

$$\gamma_1 \sim^n_{\alpha_2(\widehat{Q})} \gamma_2 \Rightarrow \gamma_1 \simeq^n_{\widehat{Q}} \gamma_2.$$

Proof. Let $\gamma_1 = C^1_0, \ldots, C^1_n$ and $\gamma_2 = C^2_0, \ldots, C^2_n$ be two n-step context sequences such that $\gamma_1 \sim^n_{\alpha_2(\widehat{Q})} \gamma_2$. By definition of $\sim^n_{\alpha_2(\widehat{Q})}$ it follows that $i \in \{0, \ldots, n\}$, $(C^1_i \cap \alpha_2(\widehat{Q})) = (C^2_i \cap \alpha_2(\widehat{Q}))$.

We now apply the definition of the abstraction function, obtaining that $\alpha_2(\widehat{Q}) = \{s \mid \exists s_j \in \widehat{Q})\}$. As a consequence, we have that for each $i \in \{0, \ldots, n\}$, $(C^1_i \cap \{s \mid \exists s_j \in \widehat{Q})\}) = (C^2_i \cap \{s \mid \exists s_j \in \widehat{Q})\})$. Note that for each $\widehat{Q}$ it holds that $\{s \mid s_i \in \widehat{Q}\} \subseteq \{s \mid \exists j,\ s_j \in \widehat{Q}\}$. As a consequence we have also that we can conclude that for each $i \in \{0, \ldots, n\}$, $(C^1_i \cap \{s \mid \exists s_i \in \widehat{Q})\}) = (C^2_i \cap \{s \mid \exists s_i \in \widehat{Q})\})$. Therefore, by applying the definition of the equivalence $\simeq^n_{\widehat{Q}}$ we conclude that $\gamma_1 \simeq^n_{\widehat{Q}} \gamma_2$.

□

Theorem 6.4. *Let $\mathcal{A} = (S, A)$ be a reaction system, $s \in S$ and $\widehat{Q} \subseteq S^n$. We have that*

- *if* $\widehat{Q}$ m-predicts s in $n+1$ steps *then* $\alpha_2(\widehat{Q})$ s-predicts s in $n+1$ steps*;*
- *if $\widehat{Q}$ is the* minimal multi-step based predictor *of s in $n+1$ steps then $\alpha_2(\widehat{Q})$ is the* minimal set based predictor *of s in $n+1$ steps.*

Proof.

- By definition of m-predictor, for any two n-steps context sequences γ_1 and γ_2 we have that $\gamma_1 \simeq^n_{\widehat{Q}} \gamma_2$ implies $s \in D^1_{n+1} \iff s \in D^2_{n+1}$. Moreover by Lemma 6.3 we also have that $\gamma_1 \sim^n_{\alpha_2(\widehat{Q})} \gamma_2$ implies $\gamma_1 \simeq^n_{\widehat{Q}} \gamma_2$. As a consequence, $\gamma_1 \sim^n_{\alpha_2(\widehat{Q})} \gamma_2$ implies $s \in D^1_{n+1} \iff s \in D^2_{n+1}$. Thus, $\alpha_2(\widehat{Q})$ s-predicts s in $n+1$ steps.

- Assume, by contradiction, that $\alpha_2(\widehat{Q})$ s-predicts s in $n+1$ steps but it is not minimal. This mean that there exists $Q_1 \subseteq S$ that s-predicts s in $n+1$ steps and such that $Q_1 \subset \alpha_2(\widehat{Q})$.

  Consider now the set of labelled objects $\widehat{Q_1} = \{s_i \mid s \in Q_1\}$. Given that Q_1 s-predicts s in $n+1$ steps it is easy to see that also $\widehat{Q_1}$ m-predicts s in $n+1$ steps. Moreover, by applying the abstraction function we have that $\alpha_2(\widehat{Q_1}) = \{s \mid \exists j\ s_j \in \widehat{Q_1}\} = Q_1$.

  By hypothesis $\widehat{Q}$ is a minimal m-predictor that is $\widehat{Q} \subseteq \widehat{Q_1}$. Moreover, by applying the abstraction function we have $\alpha_2(\widehat{Q}) = \{s \mid \exists j\ s_j \in \widehat{Q}\}$. As a consequence using the monotonicity of the abstraction function we have $\alpha_2(\widehat{Q}) \subseteq \alpha_2(\widehat{Q_1}) = Q_1$. This gives a contradiction with the assumption that $Q_1 \subset \alpha_2(\widehat{Q})$.

□

Theorem 6.6. *Let $\mathcal{A} = (S, A)$ be a reaction system and $s \in S$. Given $n \in \mathbb{N}$ we have that*

$$\alpha_2(\mathtt{mbp}(s, n)) = \mathtt{sbp}(s, n).$$

Proof. The proof is similar to that of Theorem 5.6. In this case we have that $\mathtt{sbp}(s, n) = \mathtt{sbs}(cause(s), n)$ and analogously $\mathtt{mbp}(s, n) = \mathtt{mbs}(cause(s), n)$. Therefore it is enough to prove that $\alpha_2(\mathtt{mbs}(f, i)) = \mathtt{sbs}(f, i)$ for each formula $f \in F_S$ and for each $i \in \mathbb{N}$.

The proof proceeds by induction on the well-founded order $\leq_c$ of Definition B.1.

The base case is when $i = 0$ and either $f = s$ where s a positive literal or $f = true$ or $f = false$. Assume first that $f = s$. Then, it is immediate to prove that $\alpha_2(\mathtt{mbs}(s, 0)) = \alpha_2(\{s_0\}) = \{s\} = \mathtt{sbs}(s, 0)$.
. For $f = true$, we have that $\alpha_2(\mathtt{mbs}(true, i)) = \alpha_2(\emptyset) = \mathtt{sbs}(true, i)$ for any $i \in \mathbb{N}$, and, analogously for the case $f = false$.

Then, we have to prove that $\mathtt{sbs}(f, i) = \alpha_2(\mathtt{mbs}(f, i))$ holds, knowing that $\mathtt{sbs}(f', j) = \alpha_2(\mathtt{mbs}(f', j))$ for all $f \in F_S$ and $j \in \mathbb{N}$ such that $(j, f') <_c (i, f)$.

We have the following different cases:

- $i > 0$ and $f = s$ where s is a positive literal. Then we have

  $$\alpha_2(\mathtt{mbs}(s, i)) = \alpha_2(\{s_i\} \cup \mathtt{mbs}(cause(s), i - 1)) = \{s\} \cup \alpha_2(\mathtt{mbs}(cause(s), i - 1)).$$

  On the other hand we have $\mathtt{sbs}(s, i) = \{s\} \cup \mathtt{sbs}(cause(s), i - 1)$. We can conclude that $\alpha_2(\mathtt{mbs}(s, i)) = \mathtt{sbs}(s, i)$ given that by induction hypothesis we have $\alpha_2(\mathtt{mbs}(cause(s), i - 1)) = \mathtt{sbs}(cause(s), i - 1)$.

- $f = f_1 \wedge f_2$. Then by definition we have $\mathtt{mbs}(f_1 \wedge f_2, i) = \mathtt{mbs}(f_1, i) \cup \mathtt{mbs}(f_2, i)$. Thus, by applying the abstraction function we have,

  $$\alpha_2(\mathtt{mbs}(f_1 \wedge f_2, i)) = \alpha_2((\mathtt{mbs}(f_1, i) \cup (\mathtt{mbs}(f_2, i))) = \alpha_2(\mathtt{mbs}(f_1, i)) \cup \alpha_2(\mathtt{mbs}(f_2, i)).$$

  On the other hand, we have $\mathtt{sbs}(f_1 \wedge f_2, i) = \mathtt{sbs}(f_1, i) \cup \mathtt{sbs}(f_2, i)$. We can conclude that $\mathtt{sbs}(f_1 \wedge f_2, i) = \alpha_2(\mathtt{mbs}(f_1 \wedge f_2, i))$ given that by induction hypothesis we have $\alpha_2(\mathtt{mbs}(f_1, i)) = \mathtt{sbs}(f_1, i)$ and $\alpha_2(\mathtt{mbs}(f_2, i)) = \mathtt{sbs}(f_2, i)$.

- $f = f_1 \vee f_2$. Then the proof is analogous to the previous case.

- $f = \neg f'$. Then then by definition we have $\mathtt{mbs}(\neg f', i) = \mathtt{mbs}(f', i)$. Thus, by applying the abstraction function we have,

  $$\alpha_2(\mathtt{mbs}(\neg f', i)) = \alpha_2((\mathtt{mbs}(f', i))).$$

  On the other hand, we have $\mathtt{sbs}(\neg f', i) = \mathtt{sbs}(f', i)$. We can conclude that $\mathtt{sbs}(\neg f', i) = \alpha_2(\mathtt{mbs}(\neg f', i))$ given that by induction hypothesis we have $\alpha_2(\mathtt{mbs}(f', i)) = \mathtt{sbs}(f', i)$.

- $f = (f')$ or $f = true$ or $f = false$. Then the proof is immediate reasoning analogously as in the previous cases.

□

Corollary 6.7. *Let $\mathcal{A} = (S, A)$ be a reaction system and $s \in S$. We have that* $\mathtt{sbp}(s, n)$ s-predicts *s in $n + 1$ steps.*

Proof. It follows directly from Corollary 6.5 and Theorem 6.6. □

References

[1] L. Kari, G. Rozenberg, The many facets of natural computing, Communications of the ACM 51 (10) (2008) 72–83.

[2] L. N. De Castro, Fundamentals of natural computing: basic concepts, algorithms, and applications, CRC Press, 2006.

[3] G. Păun, Computing with membranes, Journal of Computer and System Sciences 61 (1) (2000) 108–143.

[4] G. Păun, Membrane Computing: An Introduction, Natural Computing Series, Springer-Verlag GmbH, 2002.

[5] M. Ionescu, G. Păun, T. Yokomori, Spiking neural P systems, Fundamenta informaticae 71 (2) (2006) 279–308.

[6] A. Paun, G. Paun, The power of communication: P systems with symport/antiport, New Generation Computing 20 (3) (2002) 295–305.

[7] R. Barbuti, A. Maggiolo-Schettini, P. Milazzo, S. Tini, P systems with transport and diffusion membrane channels, Fundamenta Informaticae 93 (1) (2009) 17–31.

[8] G. Ciobanu, L. Pan, G. Păun, M. J. Pérez-Jiménez, P systems with minimal parallelism, Theoretical Computer Science 378 (1) (2007) 117–130.

[9] F. Bernardini, F. J. Romero-Campero, M. Gheorghe, M. J. Pérez-Jiménez, A modeling approach based on P systems with bounded parallelism, in: Membrane Computing, Springer, 2006, pp. 49–65.

[10] R. Barbuti, A. Maggiolo-Schettini, P. Milazzo, G. Pardini, L. Tesei, Spatial P systems, Natural Computing 10 (1) (2011) 3–16.

[11] R. Barbuti, A. Maggiolo-Schettini, P. Milazzo, G. Pardini, Simulation of spatial P system models, Theoretical Computer Science 529 (2014) 11–45.

[12] R. Barbuti, A. Maggiolo-Schettini, P. Milazzo, S. Tini, An overview on operational semantics in membrane computing, International Journal of Foundations of Computer Science 22 (01) (2011) 119–131.

[13] A. Ehrenfeucht, G. Rozenberg, Reaction systems, Fundam. Inform. 75 (1-4) (2007) 263–280.

[14] R. Brijder, A. Ehrenfeucht, M. G. Main, G. Rozenberg, A tour of reaction systems, Int. J. Found. Comput. Sci. 22 (7) (2011) 1499–1517.

[15] A. Ehrenfeucht, G. Rozenberg, Introducing time in reaction systems, Theor. Comput. Sci. 410 (4-5) (2009) 310–322.

[16] R. Brijder, A. Ehrenfeucht, G. Rozenberg, Reaction systems with duration, in: Computation, Cooperation, and Life, Vol. 6610 of LNCS, 2011, pp. 191–202.

[17] A. Ehrenfeucht, G. Rozenberg, Events and modules in reaction systems, Theor. Comput. Sci. 376 (1-2) (2007) 3–16.

[18] R. Brijder, A. Ehrenfeucht, G. Rozenberg, A Note on causalities in reaction systems, ECEASST 30.

[19] A. Ehrenfeucht, M. G. Main, G. Rozenberg, Functions defined by reaction systems, Int. J. Found. Comput. Sci. 22 (1) (2011) 167–178.

[20] R. Brijder, A. Ehrenfeucht, G. Rozenberg, Representing reaction systems by trees, in: Computation, Physics and Beyond, Vol. 7160 of LNCS, 2012, pp. 330–342.

[21] A. Ehrenfeucht, M. G. Main, G. Rozenberg, A. T. Brown, Stability and chaos in reaction systems, Int. J. Found. Comput. Sci. 23 (5) (2012) 1173–1184.

[22] G. Pardini, R. Barbuti, A. Maggiolo-Schettini, P. Milazzo, S. Tini, Compositional semantics and behavioural equivalences for reaction systems with restriction, Theor. Comput. Sci. 551 (2014) 1–21.

[23] G. Pardini, R. Barbuti, A. Maggiolo-Schettini, P. Milazzo, S. Tini, A compositional semantics of reaction systems with restriction, in: P. Bonizzoni, V. Brattka, B. Lwe (Eds.), The Nature of Computation. Logic, Algorithms, Applications, Vol. 7921 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 330–339.
[24] L. Corolli, C. Maj, F. Marini, D. Besozzi, G. Mauri, An excursion in reaction systems: From computer science to biology, Theor. Comput. Sci. 454 (2012) 95–108.
[25] P. Cousot, R. Cousot, Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints, in: Conference Record of the Fourth ACM Symposium on Principles of Programming Languages, Los Angeles, California, USA, January 1977, 1977, pp. 238–252.
[26] P. Cousot, R. Cousot, Systematic design of program analysis frameworks, in: Conference Record of the Sixth Annual ACM Symposium on Principles of Programming Languages, San Antonio, Texas, USA, January 1979, 1979, pp. 269–282.
[27] G. Rozenberg, A framework for exploring biology, PhD course held in Pisa on June 2014.
[28] I. Wegener, Branching programs and binary decision diagrams: theory and applications, Vol. 4, SIAM, 2000.
[29] Generator of predictors for reaction systems.