Specialized Predictor for Reaction Systems with Context Properties

Roberto Barbuti, Roberta Gori, Francesca Levi, and Paolo Milazzo

Dipartimento di Informatica, Università di 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. Reaction systems execute in an environment that provides new molecules at each step. Brijder, Ehrenfeucht and Rozemberg introduced 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. We introduced the notion of *formula based predictor*, that is a propositional logic formula that precisely characterizes environments that lead to the production of s after n steps. In this paper we revise the notion of formula based predictor by defining a *specialized version* that assumes the environment to provide molecules according to what expressed by a temporal logic formula. As an application, we use specialized formula based predictors to give theoretical grounds to previously obtained results on a model of gene regulation.

1 Introduction

In the context of Natural Computing [11, 6], *reaction systems* [10, 2] have been proposed as a model for the description of biochemical processes driven by the interaction among reactions in living cells. A *reaction system* consists of a set of objects S representing molecules and a set of rewrite rules (called *reactions*) representing chemical reactions. A reaction is a triple (R, I, P), where R, I and P are sets of objects representing reactants, inhibitors and products, respectively. A state of a reaction system is a subset of S representing available molecules. 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.

The dynamics of a reaction system is given by occurrences of reactions (i.e. rewrite rule applications) based on two opposite mechanisms: *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. The threshold supply assumption ensures that different reactions never compete for their reactants, and hence all the applicable reactions in a step are always applied. 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 the next state consists only of the products of the reactions applied in the current step.

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.

Theoretical aspects of reaction systems have been studied in [7, 3, 8, 4, 9, 13, 12]. In [3] Brijder, Ehrenfeucht and Rozemberg introduced the idea of *predictor*. Assume that one is interested in knowing whether an object $s \in S$ will be present after n steps of execution of a reaction system. Since the only source of non-determinism are the contextual elements received at each step, observing such elements can allow us to predict the production of s after n steps. In general, not all contextual elements are relevant for determining if s will be produced. A predictor is hence the subset Q of S that is actually essential to be observed among contextual elements for predicting whether s will be produced after n steps or not.

In [1] we continued the investigation on predictors by introducing the new notion of *formula based predictor*. Formula based predictors consist in a propositional logic formula to be satisfied by the sequence of (sets of) elements provided by the environment. Satisfaction of the logic formula precisely discriminates the cases in which s will be produced after n steps from those in which it will not.

Formula based predictors (as well as standard predictors) do not assume anything about the elements provided by the environment. However, it is very often the case that the sequences of sets of object provided by the environment follow specific patterns or, more generally, have specific dynamical properties. For example, it might happen that some object are never provided by the environment, or that some objects are provided only after some others.

In this paper we revise formula based predictors by introducing *specialized formula based predictors*. The revised notion is specialized with respect to a *temporal logic formula* expressing the dynamical properties of the environment. More specifically, a specialized formula based predictor is a propositional logical formula that predicts the production of an object s after n steps, by considering only the subset of the context sequences satisfying the given temporal logic formula. A specialized predictor can be substantially a simpler formula than the corresponding (non-specialized) formula based predictor.

We illustrate specialized formula based predictors on a model of the *lac* operon expression in the *E. coli* bacterium originally proposed in [5].

2 Reaction Systems

In this section we recall the basic definition of reaction systems [10, 2]. Let S be a finite set of symbols, called objects. A *reaction* is formally a triple (R, I, P) with $R, I, P \subseteq S$, composed of *reactants* R, *inhibitors* I, and *products* P. We assume reactants and inhibitors to be disjoint ($R \cap I = \emptyset$), otherwise the reaction would never be applicable. Reactants and inhibitors $R \cup I$ of a reaction are collectively called *resources* of such a reaction. The set of all possible reactions over a set S is denoted by $\mathrm{rac}(S)$. Finally, a *reaction system* is a pair $\mathcal{A} = (S, A)$, with S being the 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. 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. The application of multiple reactions at the same time occurs without any competition for the used reactants (threshold supply assumption). 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 is the assumption about the *non-permanency* of objects. Under such an assumption the objects carried over to the next step are only those produced by reactions. All the other objects vanish, even if they are not involved any reaction.

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 $\gamma = C_0, C_1, \dots, C_n$ and $\delta = D_0, D_1, \dots, D_n$ 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 $W_0, W_1, \dots, W_n$, where $W_i = C_i \cup D_i$ for all $1 \leq i \leq n$. In the following we d'say that $\gamma = C_0, C_1, \dots, C_n$ is a n-step context sequence.

3 Preliminaries

In order to describe the causes of a given product, we use objects of reaction systems as propositional symbols of formulas. Formally, we introduce the set F_S of propositional formulas on S defined in the standard way: $S \cup \{true, false\} \subseteq F_S$ and $\neg f_1, f_1 \vee f_2, f_1 \wedge f_2 \in F_S$ if $f_1, f_2 \in F_S$.

The propositional formulas F_S are interpreted with respect to subsets of the objects $C \subseteq S$. Intuitively, $s \in C$ denotes the presence of element s and therefore the truth of the corresponding propositional symbol. The complete definition of the satisfaction is as follows.

Definition 1. *Let $C \subseteq S$ for a set of objects S. Given a propositional formula $f \in F_S$, the* satisfaction relation *$C \models f$ is inductively defined as follows:*

$C \models s$ *iff* $s \in C$, $\qquad C \models true$,
$C \models \neg f'$ *iff* $C \not\models f'$, $\qquad C \models f_1 \vee f_2$ *iff either* $C \models f_1$ *or* $C \models f_2$,
$C \models f_1 \wedge f_2$ *iff* $C \models f_1$ *and* $C \models f_2$.

In the following $\equiv_l$ stands for the logical equivalence on propositional formulas F_S. Moreover, given a formula $f \in F_S$ 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 applying the standard formula simplification procedure of propositional logic converting a formula to Conjunctive Normal Form. We recall that for any formula $f \in F_S$ the simplified formula $simpl(f)$ is equivalent to f, it is minimal with respect to the propositional symbols. 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 $atom(f') \subset atom(simpl(f))$.

The causes of an object in a reaction system are defined by a propositional formula on the set of objects S. 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 2. *Let $a = (R, I, P)$ be a reaction with $R, I, P \subseteq S$ for a set of objects S. The* applicability predicate *of a, denoted by $ap(a)$, is defined as follows: $ap(a) = \left(\bigwedge_{s_r \in R} s_r\right) \wedge \left(\bigwedge_{s_i \in I} \neg s_i\right)$.*

The *causal predicate* of a given object s is a propositional formula on S representing the conditions for the production of s in one step, namely that at least one reaction having s as a product has to be applicable.

Definition 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*[1]*: $cause(s, \mathcal{A}) = \bigvee_{\{(R,I,P) \in A | s \in P\}} ap\left((R, I, P)\right)$.*

We introduce a simple reaction system as running example.

Example 1. Let $\mathcal{A} = (\{A, \ldots, G\}, \{a_1, a_2, a_3\})$ be a reaction system with

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

The *applicability predicates* of the reactions are $ap(a_1) = A$, $ap(a_2) = C \wedge D$ and $ap(a_3) = G \wedge \neg B$. Thus, the *causal predicates* of the objects are

$$\begin{array}{l} cause(A) = cause(C) = cause(D) = cause(G) = false, \\ cause(B) = A,\ cause(F) = C \wedge D,\ cause(E) = (G \wedge \neg B) \vee (C \wedge D) \end{array}$$

Note that $cause(A) = false$ given that A cannot be produced by any reaction. An analogous reasoning holds for objects C, D and G.

4 Formula Based Predictors

We introduce the notion of *formula based predictors*, originally presented in [1]. A formula based predictor for an object s at step $n + 1$ is a propositional formula satisfied

[1] We assume that $cause(s) = false$ if there is no $(R, I, P) \in A$ such that $s \in P$.

exactly by the context sequences leading to the production of s at step $n+1$. Minimal formula based predictors can be calculated in an effective way.

Given a set of objects S, we consider a corresponding set of *labelled objects* $S \times \mathbb{N}$. 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 $S_i = \{s_i \mid s \in S\}$. Propositional formulas on labelled objects S^n describe properties of n-step context sequences. The set of propositional formulas on S^n, denoted by F_{S^n}, is defined analogously to the set F_S (presented in Sect. 3) by replacing S with S^n. The satisfaction relation of Def. 1 applies also to formulas in F_{S^n} on subsets of S^n.

A labelled object s_i represents the presence (or the absence, if negated) of object s in the i-th element C_i of the n-step context sequence $\gamma = C_0, C_1, \ldots C_n$. This interpretation leads to the following definition of satisfaction relation for propositional formulas on context sequences.

Definition 4. *Let $\gamma = C_0, C_1, \ldots C_n$ be a n-step context sequence and $f \in F_{S^n}$ a propositional formula. The* satisfaction relation *$\gamma \models f$ is defined as*

$$\{s_i \mid s \in C_i, 0 \leq i \leq n\} \models f\,.$$

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

The latter notion of satisfaction allows us to define formula based predictor.

Definition 5 (Formula based Predictor). *Let $\mathcal{A} = (S, A)$ be a r.s. $s \in S$ and $f \in F_{S^n}$ a propositional formula. We say that f* f-predicts s in $n+1$ steps *if for any n-step context sequence $\gamma = C_0, \ldots, C_n$*

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

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

Note that if formula f f-predicts s in $n+1$ steps and if $f' \equiv_l f$ then also f' f-predicts s in $n+1$. More specifically, we are interested in the formulas that f-predict s in $n+1$ and contain the minimal numbers of propositional symbols, so that their satisfiability can easily be verified. This is formalised by the following approximation order on F_{S^n}.

Definition 6 (Approximation Order). *Given $f_1, f_2 \in F_{S^n}$ we say that $f_1 \sqsubseteq_f f_2$ if and only if $f_1 \equiv_l f_2$ and $atom(f_1) \subseteq atom(f_2)$.*

It can be shown that there exists a *unique equivalence class* of formulas based predictors for s in $n+1$ steps that is minimal w.r.t. the order $\sqsubseteq_f$.

We now define an operator `fbp` that allows formula based predictors to be effectively computed.

Definition 7. *Let $\mathcal{A} = (S, A)$ be a r.s. and $s \in S$. We define a function* $\texttt{fbp} : S \times \mathbb{N} \rightarrow F_{S^n}$ *as follows:* $\texttt{fbp}(s, n) = \texttt{fbs}(\mathit{cause}(s), n)$*, where the auxiliary function* $\texttt{fbs} : F_S \times \mathbb{N} \rightarrow F_{S^n}$ *is recursively defined as follows:*

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

The function $\texttt{fbp}$ gives a formula based predictor that, in general, may not be minimal w.r.t. to $\sqsubseteq_f$. Therefore, the calculation of a minimal formula based predictor requires the application of a standard simplification procedure to the obtained logic formula.

Theorem 1. *Let $\mathcal{A} = (S, A)$ be a r.s.. For any object $s \in S$,*

- $\texttt{fbp}(s, n)$ f-predicts *s in $n+1$ steps;*
- *$\mathit{simpl}(\texttt{fbp}(s, n))$* f-predicts *$s$ in $n+1$ steps and is* minimal *w.r.t. $\sqsubseteq_f$.*

Example 2. Let us consider again the reaction system of Ex. 1. We are interested in the production of E after 4 steps. Hence, we calculate the logic formula that *f-predicts E in 4 steps* applying the function $\texttt{fbp}$:

$$\begin{aligned}
\texttt{fbp}(E, 3) &= \texttt{fbs}\big((G \wedge \neg B) \vee (C \wedge D), 3\big) \\
&= \big(\texttt{fbs}(G, 3) \wedge \neg\texttt{fbs}(B, 3)\big) \vee \big(\texttt{fbs}(C, 3) \wedge \texttt{fbs}(D, 3)\big) \\
&= \big((G_3) \wedge \neg(B_3 \vee \texttt{fbs}(A, 2))\big) \vee (C_3 \wedge D_3) \\
&= \big(G_3 \wedge \neg B_3 \wedge \neg A_2\big) \vee (C_3 \wedge D_3)\big)
\end{aligned}$$

A context sequence satisfies $\texttt{fbp}(E, 3)$ iff the execution of the reaction system leads to the production of object E after 4 steps. Furthermore, in this case the obtained formula is also minimal given that $\mathit{simpl}(\texttt{fbp}(E, 3)) = \texttt{fbp}(E, 3)$.

5 The Temporal Logic for Context Sequences

We introduce a linear temporal logic for the description of properties of finite context sequences. In the logic, propositional formulas describe the properties of single contexts (i.e. the symbols that can/cannot appear in an element of a context sequence). Hence, such formulas play the role of state formulas in traditional temporal logics. Temporal properties are expressed by variants of the usual *next* and *until* operators, and by derived *eventually* and *globally* operators.

Definition 8 (Temporal Formulas). *Let S be a set of objects. The syntax of* temporal logic formulas *on S is defined by the following grammar:*

$$\psi \ ::= \ f \ \ \big| \ \ \psi \vee \psi \ \ \big| \ \ \psi \wedge \psi \ \ \big| \ \ \mathbf{X}\psi \ \ \big| \ \ \psi \mathbf{U}_k \psi \ \ \big| \ \ \mathbf{F}_k \psi \ \ \big| \ \ \mathbf{G}_k \psi$$

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

Given a context sequence $\gamma = C_0, \ldots, C_n$, the simple temporal formula f states that f is satisfied by C_0, according to the definition of satisfiability of the propositional logic. Formula $\mathbf{X}\psi$ states that ψ is satisfied by the context sequence after one step, namely by $C_1, \ldots, C_n$. Formula $\psi_1 \mathbf{U}_k \psi_2$ states that there exists $k' \leq k$ such that $C_0, \ldots, C_{k'-1}$ satisfies ψ_1 and $C_{k'}$ satisfies ψ_2. Formula $\mathbf{F}_k\psi$ states that there exists $k' \leq k$ such that $C_{k'}$ satisfies ψ. Formula $\mathbf{G}_k\psi$ states that for all $i \leq k$ context C_i satisfies ψ. Finally, operators $\vee$ and $\wedge$ are as usual.

Note that the value of n has a significant role in the satisfiability of temporal formulas. In particular, if $n < k$ we have that $\psi_1 \mathbf{U}_k \psi_2$ is satisfied also if ψ_1 is satisfied by all C_i with $i \leq n$ (never satisfying ψ_2). Moreover, if $n < k$ we have that $\mathbf{F}_k\psi$ is always satisfied since, intuitively, ψ may be satisfied by a context C_j, with $n < j \leq k$, that is not contained in γ. An analogous reasoning holds for $\mathbf{X}\psi$ when $n = 0$ (or $\mathbf{XX}\psi$ when $n = 1$, etc...). As regards $\mathbf{G}_k\psi$, when $n < k$ it is equivalent to $\mathbf{G}_n\psi$.

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

The formal definition of satisfiability of temporal logic formulas on finite n-step context sequences is as follows.

Definition 9. *Let $\gamma = C_0, C_1, \ldots, C_n$ be a n-step context sequence with $n \geq 0$. Given a temporal logic formula $\psi \in TL_S$, the* satisfaction relation *$\gamma \vdash \psi$ is inductively defined as follows:*

$$\gamma \vdash f \text{ iff } C_0 \models f \qquad \gamma \vdash \mathbf{X}\psi \text{ iff either } n = 0 \text{ or } \gamma' \vdash \psi$$

$$\gamma \vdash \psi_1 \wedge \psi_2 \text{ iff } \gamma \vdash \psi_1 \text{ and } \gamma \vdash \psi_2$$

$$\gamma \vdash \psi_1 \vee \psi_2 \text{ iff either } \gamma \vdash \psi_1 \text{ or } \gamma \vdash \psi_2$$

$$\begin{aligned}\gamma \vdash \psi_1 \mathbf{U}_\infty \psi_2 \text{ iff either } & \gamma \vdash \psi_2 \\ & \text{or } \gamma \vdash \psi_1 \text{ and if } n > 0 \text{ then } \gamma' \vdash \psi_1 \mathbf{U}_\infty \psi_2\end{aligned}$$

where $\gamma' = C'_0, \ldots, C'_{n-1}$ with $C'_i = C_{i+1}$ for $0 \leq i \leq n-1$.

Finally, we present an encoding of temporal formulas on objects S into propositional formulas on labelled objects S^n. The encoding depends on the parameter $n \in \mathbb{N}$ reporting the length of the context sequences that we want to model.

Definition 10. *Let S be a set of objects and $n \in \mathbb{N}$. The* encoding *of a temporal logic formula $\psi \in TL_S$ into a propositional logic formula on S^n is given by $[\![\psi]\!]^n_0$ where the function $[\![\_]\!]^n_\_ : TL_S \times \mathbb{N} \to F_{S^n}$ is defined as follows:*

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

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

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

where $\lfloor\_\rfloor_i : F_S \rightarrow F_{S_i}$ is a function that replaces, in a given propositional logic formula f, every $s \in S$ with the corresponding labelled object $s_i \in S_i$.

The following theorem states the main property of the encoding of temporal formulas: the result of the encoding of a temporal formula is equivalent.

Theorem 2. *Let $\gamma = C_0, \ldots, C_n$ be a n-step context sequence. For any temporal formula $\psi \in TL_S$ it holds that $\gamma \vdash \psi \iff \gamma \models [\![\psi]\!]_0^n$.*

6 Specialized Formula Based Predictors

We specialize the notion of formula based predictor (given in Def. 5) w.r.t. a subset of context sequences characterized by a temporal logic formula ψ.

Definition 11 (Specialized Formula based Predictor). *Let $\mathcal{A} = (S, A)$ be a r.s. $s \in S$, $f \in F_{S^n}$ and $\psi \in TL_S$. We say that f* f-predicts s in $n+1$ steps with respect to ψ *iff for any n-step context sequence $\gamma = C_0, \ldots, C_n$ such that $\gamma \vdash \psi$ we have that*

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

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

It should be clear that any formula f that f-predicts s in $n+1$ steps also f-predicts s in $n+1$ steps with respect to any logical formula ψ. In particular, the formula $\texttt{fbp}(s, n)$ (and its simplified version $simpl(\texttt{fbp}(s, n))$ is a specialized predictor for s in $n+1$ steps for any ψ. However, the formula $simpl(\texttt{fbp}(s, n))$ typically is too general and therefore is not a minimal (w.r.t $\sqsubseteq_f$) formula based predictor specialized w.r.t. ψ.

Thus, we introduce a methodology that allows us to calculate minimal formula based predictors for an object s at $n+1$ step, specialized w.r.t. ψ. The idea is to use the encoding of the temporal formula ψ computed with respect to the lenght of the context sequences n. The encoding $[\![\psi]\!]_0^n$ is a propositional formula on labelled objects S^n that models the n-step context sequences satisfying ψ. A formula based predictor specialized w.r.t. ψ can be derived by exploiting the encoding $[\![\psi]\!]_0^n$ to simplify a corresponding formula based predictor.

More formally, suppose that f *f-predicts* s in $n+1$ steps. A formula f' *f-predicts* s in $n+1$ steps w.r.t a temporal formula ψ whenever f' is logically equivalent to f considering the context sequences satisfying ψ, that is assuming that the conditions formalized by the formula $[\![\psi]\!]_0^n$ holds. The following theorem establishes this fundamental property of specialized formula based predictors.

Theorem 3. *Let $\mathcal{A} = (S, A)$ be a r.s., $s \in S$, and $f \in F_{S^n}$ such that f* f-predicts s in $n+1$ steps. *Given $\psi \in TL_S$ and $f' \in F_{S^n}$ such that*

$$[\![\psi]\!]_0^n \Rightarrow (f \equiv_l f')$$

we have that f' f-predicts s in $n+1$ steps with respect to ψ.

We introduce a minimization algorithm that allows us to calculate a minimal formula f' that satisfies the property of Theorem 3, given a predictor f and a temporal logic formula ψ.

Let $B = \{true, false\}$ and consider the *incompletely specified booean function* $F : B^n \rightarrow B$ composed of the following two disjoint subsets $Yes, DontC$ of B^n.
$Yes_F = \{x_1, .., x_n \mid x_1, .., x_n$ is a boolean assignment that satisfies $[\![\psi]\!]_0^n \wedge f\}$ is the subset of B^n where F is evaluated to 1 and
$DontC_F = \{x_1, .., x_n \mid x_1, .., x_n$ is a boolean assignment that satisfies $\neg[\![\psi]\!]_0^n\}$ is a subset of B^n where the value of F is not specified. In each point of $DontC_F$, called *don't care*, F can assume the value 1 or 0. So there are $m = 2^{|DontC_F|}$ different functions *equivalent* to F that may take the place of F according to particular needs. Consider any possible extention of F, let us call them $F^1, ..., F^m$, obtained by considering $Yes_{F^i} = Yes_F \cup D_i$ with $D_i \subseteq DontC_F$ and $DontC_{F^i} = \emptyset$. Each F^i is now a *completely specified boolean function* and it can be shown that in this case its *minimal sum of product form* (SOP) also contains a minimal number of different variables. Hence, any algorithm that computes the minimal SOP form of F^i (i.e. any well known algorithm looking for the prime implicants of F^i) can be applied in order to minimize the number of variables of F^i. Then consider the f' corresponding to the minimal SOP form between all the minimal SOP form of $F^1, ..., F^m$, we are guaranteed that such f' has the minimal number of variables.

It is worth noting that considering all possible extensions of the *incompletely specified* boolean function F *as well computing the SOP form of a* completely specified *booean function* F^i has an exponential cost. However, some heuristic that, in general, cannot guarantee minimality can be designed in order to improve the efficiency.

Example 3. Let us consider the r.s. of Ex. 1 with the following reactions:

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

We introduce the following temporal formulas in order to describe the properties of the context sequences:

$$\psi_1 = \mathbf{G}_\infty\big(\neg B \wedge (\neg C \vee D)\big), \;\; \psi_2 = \mathbf{G}_\infty B, \;\; \psi_3 = \mathbf{F}_3 B$$

Formula ψ_1 describes an invariant property which holds at any step of the context sequence. The property models the situation in which object B is not a synthesizable product, i.e., it can not be found in the environment but it has to be produced by the reactions. Moreover, if the environment supplies C it also supplies D at the same time. Similarly, formula ψ_2 shows that the environment always supply the object B. By contrast, formula ψ_3 says that the environment will eventually supply the object B within 3 steps.

We show the *specialized formula based predictors* w.r.t. the previous temporal formulas considering again the production of E in 4-steps. We recall that the corresponding minimal formula based predictor (presented in Ex. 2) is

$$\mathit{simpl}(\texttt{fbp}(E, 3)) = \texttt{fbp}(E, 3) = \big(G_3 \wedge \neg B_3 \wedge \neg A_2\big) \vee (C_3 \wedge D_3).$$

In order to calculate the specialized versions of predictor we compute the encoding of the temporal formulas obtaining:

$$[\![\psi_1]\!]_0^3 = (\neg B_0 \wedge \neg B_1 \wedge \neg B_2 \wedge \neg B_3) \wedge (\neg C_0 \vee D_0) \wedge (\neg C_1 \vee D_1) \wedge (\neg C_2 \vee D_2) \wedge (\neg C_3 \vee D_3)\,,$$

$$[\![\psi_2]\!]_0^3 = B_0 \wedge B_1 \wedge B_2 \wedge B_3\,, \qquad [\![\psi_3]\!]_0^3 = B_0 \vee B_1 \vee B_2 \vee B_3\,.$$

By applying the minimization algorithm to $\texttt{fbp}(E, 3)$ and to ψ_1, ψ_2 and ψ_3, we obtain the correponding specialized formulas f_1, f_2 and f_3, respectively:

$$f_1 = (G_3 \wedge \neg A_2) \vee C_3 \qquad f_2 = (C_3 \wedge D_3), \qquad f_3 = \texttt{fbp}(E, 3).$$

In the case of formulas ψ_1 and ψ_2, the specialized formulas f_1 and f_2 are substantially reduced w.r.t. the formula based predictor, while formula ψ_3 does not lead to a reduced specialized predictor.

7 Application

In this section we introduce a more complex biological example: the *lac* operon expression in the *E. coli* bacterium. The lactose operon is a sequence of genes that are responsible for producing enzymes for lactose degradation. We borrow the formalization of this biological system as a reaction system from [5]. Let $\mathcal{A} = (S, \{a_1, \ldots, a_{10}\})$ be the reaction system where

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

and the reaction rules are defined as follows

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

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 transcribes the three structural genes represented by lac into a single mRNA fragment. When the lac repressor I is bound to gene OP (that is, the complex $I\text{-}OP$ 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. 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.

In [5] the authors investigate the effects on the production of enzymes Z, Y and A when the environment 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 system 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.

For sake of simplicity we consider the formula based predictors for the enzymes Z, Y and A in 4 steps, noting that the effects of all reactions can be observed after four steps. We obtain the following formula

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

The obtained formula is minimal w.r.t. $\sqsubseteq_f$ and therefore does not require the application of simplification techniques. Since we are interested in studying context sequences that supply only $glucose$ and $lactose$ it is useful to specialize the previous formula based predictor for context sequences described by the following temporal formula:

$$\psi = f_{initial} \wedge \mathbf{G}_\infty \neg f_{internal} \wedge \mathbf{XG}_\infty \neg f'_{initial}$$

where $f_{initial} = lac \wedge lacI \wedge cya \wedge crp \wedge I \wedge cAMP \wedge CAP$, $f'_{initial} = lac \vee lacI \vee cya \vee crp \vee I \vee cAMP \vee CAP$ and $f_{internal} = I \vee IOP \vee cAMPCAP \vee Z \vee Y \vee A$.

Let us focus on the production of A, noting that, since $\texttt{fbp}(Z,3) = \texttt{fbp}(Y,3) = \texttt{fbp}(A,3)$, the same reasoning apply to enzymes Z and Y. By applying the minimization procedure described in Sect. 6 to $\texttt{fbp}(A,3)$ and ψ, we obtain the following reduced predictor of A in 4 steps specialized w.r.t. ψ

$$f' = \neg glucose_2 \wedge lactose_2.$$

Formula f' clearly shows that the enzymes A is produced at step 4 only if $lactose$ was actually the only element provided to the system two steps before.

The specialized predictor f' we obtained agrees with the conclusion reached in [5] by reasoning on an example of context sequence.

Note that the same conclusion was reached in [1] by simplifying predictor $\texttt{fbp}(A, 3)$ on the basis of informal reasoning on the properties of context sequences. In this paper, we have been able to obtain f' in a fully formalized and automatic way, on the basis of $\texttt{fbp}(A, 3)$ and ψ.

8 Conclusions

We have presented a revised notion of formula based predictor [1] in which the predictor is specialized with respect to a temporal logic formula ψ expressing the properties of the context sequences. More specifically, the formula based predictor models the necessary conditions for an object s to be produced in n steps, assuming that the reaction system is executed with respect to a context sequence satisfying ψ. The advantage of the specialized version of predictor is that the resulting propositional formula can be substantially reduced with respect to the corresponding formula based predictor. As a consequence, this approach is very convenient whenever we are interested to observe whether an element s will be produced after n steps or not for a certain class of context sequences rather than for any possible context sequences.

As future work we plan to consider the application of the tabling and generalization procedures described in [1] to specialized predictors.

References

1. Barbuti, R., Gori, R., Levi, F., Milazzo, P.: Investigating dynamic causalities in reaction systems. Submitted for pubblication (2015)
2. Brijder, R., Ehrenfeucht, A., Main, M.G., Rozenberg, G.: A Tour of reaction Systems. Int. J. Found. Comput. Sci. 22(7), 1499–1517 (2011)
3. Brijder, R., Ehrenfeucht, A., Rozenberg, G.: A Note on Causalities in Reaction Systems. ECEASST 30 (2010)
4. Brijder, R., Ehrenfeucht, A., Rozenberg, G.: Representing Reaction Systems by Trees. In: Computation, Physics and Beyond. LNCS, vol. 7160, pp. 330–342 (2012)
5. Corolli, L., Maj, C., Marini, F., Besozzi, D., Mauri, G.: An excursion in reaction systems: From computer science to biology. Theor. Comput. Sci. 454, 95–108 (2012)
6. De Castro, L.N.: Fundamentals of natural computing: basic concepts, algorithms, and applications. CRC Press (2006)
7. Ehrenfeucht, A., Rozenberg, G.: Events and modules in reaction systems. Theor. Comput. Sci. 376(1-2), 3–16 (2007)
8. Ehrenfeucht, A., Main, M.G., Rozenberg, G.: Functions Defined by Reaction Systems. Int. J. Found. Comput. Sci. 22(1), 167–178 (2011)

9. Ehrenfeucht, A., Main, M.G., Rozenberg, G., Brown, A.T.: Stability and Chaos in reaction Systems. Int. J. Found. Comput. Sci. 23(5), 1173–1184 (2012)
10. Ehrenfeucht, A., Rozenberg, G.: Reaction Systems. Fundam. Inform. 75(1-4), 263–280 (2007)
11. Kari, L., Rozenberg, G.: The many facets of natural computing. Communications of the ACM 51(10), 72–83 (2008)
12. Pardini, G., Barbuti, R., Maggiolo-Schettini, A., Milazzo, P., Tini, S.: A compositional semantics of reaction systems with restriction. In: Bonizzoni, P., Brattka, V., Löwe, B. (eds.) The Nature of Computation. Logic, Algorithms, Applications, Lecture Notes in Computer Science, vol. 7921, pp. 330–339. Springer Berlin Heidelberg (2013)
13. Pardini, G., Barbuti, R., Maggiolo-Schettini, A., Milazzo, P., Tini, S.: Compositional semantics and behavioural equivalences for reaction systems with restriction. Theor. Comput. Sci. 551, 1–21 (2014)