Available online at www.sciencedirect.com

ScienceDirect

Theoretical Computer Science 00 (2015) 1–30

Theoretical Computer Science

Special Issue: SI CS2Bio

Unwinding Biological Systems☆

Alberto Casagrande[a], Carla Piazza[b,*]

[a]*Dept. of Mathematics and Geosciences, University of Trieste, Via Valerio 12/1, 34127 Trieste, Italy*
[b]*Dept. of Mathematics and Computer Science, University of Udine, Via delle Scienze 206, 33100 Udine, Italy*

Abstract

Unwinding conditions have been fruitfully exploited in Information Flow Security to define persistent security properties. In this paper we investigate on their meaning and possible uses in the analysis of biological systems. In particular, we elaborate on the notion of robustness and propose some instances of unwinding over the process algebra Bio-PEPA and over hybrid automata. We exploit such instances to analyse two case-studies: *Neurospora Crassa* circadian system and Influenza kinetics models.

Keywords: Systems Biology, Unwinding, Behavioural equivalences, Robustness

Introduction

Systems Biology aims at defining techniques for modelling and formally analyse the behaviours of biological entities such as genetic networks, regulatory networks, but also diseases and epidemics. Problems of interests in this context range from stability analysis to dynamics identification and parameter estimation.

Different languages have been proposed for modelling biological systems. As noticed in [1] we can distinguish two main categories: mathematical models (e.g., differential equations) and computational models (e.g., process algebras). Mathematical models rely on denotational semantics, i.e., they describe relationships among quantities. Algorithms for finding/approximating solutions of such models are not part of the model itself. On the other hand, computational models are equipped with "natural" operational semantics which describes the evolution of the system. Simulation algorithms for such models can be easily defined.

Temporal Logics and Model Checking are standard tools for the analysis of computational models (e.g., see [2]). The former provide specification languages for formulating the properties of interest. The latter brings algorithms for verifying properties on models. Two models are considered behaviourally equivalent when they are indistinguishable with respect to the temporal logic formulæ.

Behavioural equivalences can then be used to both compare models and reduce their sizes. Their use as reduction criteria is well known in Model Checking, where the state explosion problem is a major concern,

☆This work has been partially supported by Istituto Nazionale di Alta Matematica (INdAM).

*Corresponding author

Email addresses: acasagrande@units.it (Alberto Casagrande), carla.piazza@uniud.it (Carla Piazza)

and their importance for comparing models has been fruitfully exploited in other fields. In particular, in Information Flow Security behavioural equivalences are at the basis of unwinding conditions which allow one to establish whether the system would behave correctly in hostile environments (see [3, 4]).

Many authors have proposed the use of Temporal Logics and Model Checking techniques for the analysis of biological systems (e.g., see [5, 6, 7]). Unfortunately, in this field of application the high number of involved variables make the state explosion problem even more dramatic and important efforts have been made to define reduction techniques (e.g., see [8, 9]).

In this paper, we consider the problem of deciding whether a biological system is robust. Intuitively, a system is robust if its behaviour is not dramatically affected by perturbations. Biological organisms are always subject to perturbations due to changes in their living environments and robustness is the property that distinguishes between an organism that can survive to changes and another that cannot. Of course there is not just one notion of robustness, but as many possible notions as many meaning one can give to "perturbations" and "dramatically affected". We propose the use of unwinding conditions in Systems Biology to define robustness. In order to achieve this goal we develop a framework, based on the notion of labelled transition system, that exploits the unwinding conditions and we suggest how to instantiate this framework to handle different computational formalisms, different notions of perturbations, and different levels of behavioural changes. We will instantiate our framework to show that unwinding conditions allow one to generalise notions of robustness based on formulæ satisfaction which have been introduced in the literature, giving rise to robust notions of robustness where possibly infinite set of formulæ can be preserved over possibly infinite sets of perturbations.

Our framework can be instantiated on each computational language equipped with an operational semantics. In this work we instantiate it on the stochastic process algebra Bio-PEPA and on hybrid automata. On Bio-PEPA one can define discrete operational semantics and build finite labelled transition systems. Hence, Bio-PEPA can be considered in this paper as a representative of (Stochastic) Discrete Event Systems. On the other hand, the most natural operational semantics for hybrid automata is the *timed continuous-discrete semantics*. Hence, hybrid automata shift us from the classical Discrete Event Systems realm to the timed models one.

The paper is organised as follows: Section 1 introduces Labelled Transition Systems and bisimulations as standard operational semantics tools. Section 2 presents the unwinding framework in both Information Flow Security and Systems Biology, showing how it can be used to formalise a robust notion of robustness. In Section 3, we briefly recall the main features of Bio-PEPA process algebra and we instantiate our unwinding framework on Bio-PEPA systems. In Section 3.1, we test it on the Bio-PEPA model of the *Neurospora crassa* circadian network. In Section 4 we present two unwinding conditions over hybrid automata and in Section 4.1 we exploit them on kinetic models of influenza virus. Sections 4 and 4.1 are completely independent from Sections 3 and 3.1. Hence, the reader can safely skip either Bio-PEPA or hybrid automata. Conclusions are drawn in Section 5.

1. Labelled Transition Systems and Equivalences

Labelled Transition Systems (latter on referred as LTS's) are a general tool used to define the operational semantics of wide variety of models. An LTS is a directed graph with labels on edges.

Definiton 1 (Labelled Transition System). *An LTS is a tuple* (V, V_I, A, E) *where:*

- *V is a set of nodes and $V_I \subseteq V$ is a set of initial nodes;*
- *A is the set of edge labels (alphabet);*
- *$E \subseteq V \times A \times V$ is a set of edges.*

We may write $v \xrightarrow{\alpha} v'$ to indicate that (v, α, v') is an edge, i.e., $(v, \alpha, v') \in E$, and $v \to v'$ to denote that there exists some $\alpha \in A$ such that (v, α, v') is an edge. In some cases, also labels on nodes can be introduced.

Definiton 2 (*LTS*/*R*). *Let $L = (V, V_I, A, E)$ be an LTS and R an equivalence relation over V. The* quotient of L w.r.t. R*, denoted by L/R, is the LTS (V^*, V_I^*, A, E^*) such that:*

- $V^* \stackrel{def}{=} \{[v]_R \mid v \in V\}$ *where* $[v]_R$ *is the* equivalence class of v w.r.t. R*, i.e.,* $[v]_R \stackrel{def}{=} \{v' \mid v'Rv\}$;
- $V_I^* \stackrel{def}{=} \{[v]_R \mid v \in V_I\}$;
- $E^* \stackrel{def}{=} \left\{\left([v]_R, a, [v']_R\right) \mid \left(v, a, v'\right) \in E\right\}$.

Well-known classes of models which are given in terms of LTS's are *Discrete Time Markov Chains* (DTMCs) and *Continuous Time Markov Chains* (CTMCs). Both DTMCs and CTMCs are used for modelling stochastic memoryless processes in which the next state of the system only depends on the current state and on a probability distribution over its out-going edges. While in DTMCs labels on the edges represent the instantaneous crossing probability, in CTMCs they are the parameters of exponential distributions which allow one to compute the crossing probability within a time interval.

Definiton 3 (Markov Chains). *A DTMC is an LTS $D = (V^D, V_I^D, A^D, E^D)$ such that $A^D = [0,1] \subseteq \mathbb{R}$ and the sum of the labels of the edges leaving each node is equal to* 1*, i.e., $\left(\sum_{(v,\alpha,u)\in E^D} \alpha\right) = 1$ for each $v \in V^D$.*

A CTMC is an LTS $C = (V^C, V_I^C, A^C, E^C)$ such that $A^C = \mathbb{R}_{\geq 0}$.

As examples of the broad and heterogeneous use of LTS's in modelling, we will consider later in this paper two formalisms, having a quite different approach to the modelling problem, whose evolutions are naturally described by means of LTS's: the process algebra Bio-PEPA (see, e.g., [10]) and hybrid automata (see, e.g., [11]).

Once an LTS representing a system has been obtained, behavioural equivalences can be used to reduce the size of the LTS, prove properties over the system, and compare different systems. Behavioural equivalences are equivalences over LTS's that relate nodes having "similar behaviours". *Trace equivalence* and *bisimulation* are two of the most used behavioural equivalences in the literature.

Trace equivalence relates two nodes if and only if they generate the same sequences of node and edge labels (traces). It produces a drastic reduction of the model size, but its computation is expensive (PSPACE-complete) [12]. Moreover, in many applications, it equates models that are "different". In particular, it cannot distinguish models which differ because of "the time at which non determinism occurs" (see Example 1).

Definiton 4 (Trace Equivalence). *Given an LTS $T = (V, V_I, A, E)$, a* path *over T starting from $u \in V$ is a sequence ph of transitions of the form $u = u_0 \xrightarrow{\alpha_1} u_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} u_n$. The* trace *underlying the path ph is the sequence tr defined as $\alpha_1\alpha_2\ldots\alpha_n$. The set $Tr(u)$ is the set of traces underlying paths starting from u. Two nodes $u, v \in V$ are said to be* trace equivalent *if $Tr(u) = Tr(v)$.*

On the contrary, bisimulation is a finer relation which is easier to compute. In its strong version it equates models that satisfy exactly the same formulæ of modal and branching temporal logics (see, e.g., [2]). Its co-inductive characterisation is at the basis of efficient polynomial time algorithms for its computation [13, 14].

Definiton 5 (Strong Bisimulation). *Given an LTS $T = (V, V_I, A, E)$, a* strong bisimulation *over T is a relation $R \subseteq V \times V$ such that for each $(u, v) \in R$ the following conditions hold:*

- *$u \in V_I$ if and only if $v \in V_I$;*
- *if $u \xrightarrow{\alpha} u'$, then $v \xrightarrow{\alpha} v'$ and $(u', v') \in R$;*
- *if $v \xrightarrow{\alpha} v'$, then $u \xrightarrow{\alpha} u'$ and $(u', v') \in R$.*

Two nodes $u, v \in V$ are said to be strongly bisimilar*, denoted as $u \sim v$, if there exists a strong bisimulation R such that $(u, v) \in R$.*

When also labels on nodes are considered two bisimilar nodes must share the same labels. The following proposition states some well-known and appealing properties of bisimulation.

Proposition 1. *Given an LTS $T = (V, V_I, A, E)$. The relation $\sim$ is an equivalence relation, it is the largest strong bisimulation relation (i.e., it includes all the strong bisimulation relations over V), and Paige-Tarjan algorithm [14] computes it in time $O(|E| \log |V|)$. Moreover, $u \sim v$ implies $Tr(u) = Tr(v)$.*

Example 1. Let q_0 and p_0 be two nodes of an LTS T. If we are interested in deciding whether q_0 and p_0 are trace equivalent or bisimilar, it is sufficient to analyse the nodes reachable from q_0 and p_0. Let now the LTS's represented in Figure 1 be the ones that include the nodes reachable from q_0 and p_0, respectively, in which none of the nodes belong to V_I. It emerges that q_0 and p_0 are trace equivalent, since $Tr(q_0) = Tr(p_0) = \{\alpha\beta, \alpha\gamma\}$. On the other hand, q_0 is not bisimilar to p_0. As a matter of fact, $p_0 \xrightarrow{\alpha} p_1$, while $q_0 \xrightarrow{\alpha} q_1$ and $q_0 \xrightarrow{\alpha} q_2$. However, p_1 is not bisimilar to q_1, since q_1 cannot perform any γ-transition, and p_1 is not bisimilar to q_2, since q_2 cannot perform any β-transition. In this sense, bisimulation distinguishes between q_0, which immediately performs a non-deterministic choice, and p_0, which delays the non-deterministic choice after the α-transition.

Figure 1: Trace equivalence vs strong bisimulation: the two nodes p_0 and q_0 are trace equivalent but not strongly bisimilar.

There exist many variant of bisimulation (e.g., weak and stuttering), depending on the modal/temporal logic one wants to preserve. In the context of DTMCs and CTMCs bisimulations, which need to "preserve" probabilities, are usually called *lumpability relations* (see, e.g., [15]).

Notice that all the definitions introduced in this section can be applied to both finite and infinite LTS's, as in the case of LTS built from hybrid automata (see Definition 26). However, in the case of infinite LTS's symbolic techniques have to be applied and the termination of the procedures is not always guaranteed.

2. Robustness through Non-interference and Unwinding

2.1. From Biology . . .

As observed in [16], "... robustness is one of the fundamental characteristics of biological systems ... Nevertheless, a mathematical foundation that provides a unified perspective on robustness is yet to be established". In [17], robustness is defined as a property ensuring that a system maintains its functions against internal and external perturbations. A framework for analysing robustness should support the definition of both functions to be preserved and admissible perturbations.

Robustness has not to be confused with stability. A robust system can exploit instability or even evolve through new steady states in order to preserve its functionalities against perturbations.

In the following example we try to clarify these concepts.

Example 2 (Research on Pathogens). Let us consider the case of a system infected by a pathogen agent. Of course the pathogen stimulates the immune system and would probably affect some organs, i.e., the system is not stable. Hence, the immune system and all the organs directly attacked by the pathogen will exhibit a behaviour which could be very different from the standard one (at least in the acute phase). However, if the system is robust against the pathogen, behaviour of critical organs (e.g., heart, lung, kidney, brain) should not be dramatically affected.

A formalisation of "how much" a system is *robust* with respect to a pathogen is a fundamental question in medicine both in the diagnosis process (to avoid expensive/invasive exams) and in the therapy phase.

On the other hand, if a system is *stable* with respect to a pathogen, this means that the pathogen has almost no effects on the system. Hence, probably such pathogen has low medical interest: not even the patient will notice that his immune system is interacting with the pathogen.

It is important to notice that robustness is not a *local* property. The "functions to be preserved" could have no apparent relationship with the "admissible perturbations". This is better shown by the following example.

Example 3 (Drug Development). The case of different drugs for mitigating inflammations is described in [18]. Cyclooxygenase 2 (COX2) is expressed in tissues with inflammation and its inhibition reduces the inflammation process. Some drugs were designed to inhibit COX2, but since they also inhibit COX1 adverse gastrointestinal effects were observed. Hence, drugs with more selective inhibition of COX2 were considered. These do not have gastrointestinal side-effects, but at high doses the risk of cardiovascular problems increased. As Kitano pointed out, this example "highlights the fact that selectivity for a molecule in the target cells does not eliminate the risk of side effects, as the target molecule might have an important role in off-target cells. ... Drug side effects can be caused by unwanted interactions with molecules that expose the fragility of cellular or organ-level functions to specific interventions in both target cells and off-target cells" [18].

In the next section we present an unwinding framework typical of information flow security. We will take inspiration from such framework to define a general notion of robustness which tries to answer Kitano's requirements.

2.2. ... To Security ...

Information flow security deals with multilevel systems in which confidential and public data coexist. Its main goal is to ensure that no information flow from a level to a lower one. Traditionally, only two security levels are considered: *high* (H) and *low* (L). High level users have access to confidential information, while low level users can only handle public data. The interaction between high level users and the system should not influence the low level behaviours. In the case of deterministic systems, this was formalised in [19] as the notion of *non-interference*. Such notion has been generalised in different non-deterministic settings such as programming languages [20], process calculi [21, 22], probabilistic models [23], cryptographic protocols [24].

In information flow security, *unwinding conditions* have been introduced as local properties on high level actions aiming at ensuring non-interference [22, 3, 4]. We present here the notion of *unwinding* over LTS's. First we need to partition the edge labels into two sets: H, high level labels, corresponding to high level actions and L, low level labels, associated with low level actions. We dub LTS's with H and L edge labels *multilevel LTS's*. The unwinding condition is parametric with respect to two relations: a *low level behavioural equivalence*, $\sim^L$, and a *transition relation*, $\dashrightarrow$. The former establishes which nodes should be considered indistinguishable from a low level point of view. The latter identifies paths alternative to the execution of a high action.

Definiton 6 (Unwinding $\mathcal{W}(\sim^L, \dashrightarrow)$). *Let $T = (V, V_I, H \cup L, E)$ be a multilevel LTS, $\sim^L \subseteq V \times V$ be an equivalence relation, and $\dashrightarrow \subseteq V \times V$ be a transition relation. We say that T* satisfies the unwinding condition $\mathcal{W}(\sim^L, \dashrightarrow)$ *if for each $h \in H$ and for each $u, u' \in V$ the following condition holds:*

$$u \xrightarrow{h} u' \text{ implies } \exists u''(u \dashrightarrow u'' \wedge u' \sim^L u'')$$

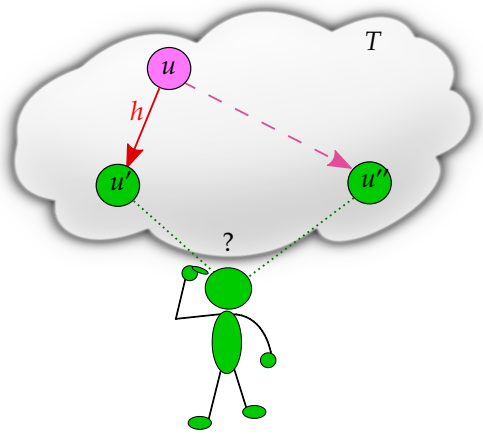

Figure 2: Intuitive representations of unwinding condition. The system T satisfies the unwinding conditions $\mathcal{W}(\sim^L, \dashrightarrow)$ if, whenever a high action can lead the state from u to u', $\dashrightarrow$ can lead T from u to u'' and u' and u'' are indistinguishable by the low user.

In Figure 2 we give an intuitive representation of a generic unwinding condition.

Whenever the information included in the nodes is also important, generalised unwinding conditions come into play. They are parametric on a further relation $=^L$ establishing which node information is low level visible or, in other terms, when two nodes are locally low level indistinguishable, i.e., indistinguishable without considering their outgoing edges.

Definiton 7 (Generalised Unwinding $\mathcal{W}(=^L, \sim^L, \dashrightarrow)$). *Let $T = (V, V_I, H \cup L, E)$ be a multilevel LTS, $=^L, \sim^L \subseteq V \times V$ be two equivalence relations, and $\dashrightarrow \subseteq V \times V$ be a transition relation. We say that T* satisfies the generalised unwinding condition $\mathcal{W}(=^L, \sim^L, \dashrightarrow)$ *if for each $h \in H$ and for each $u, u' \in V$ the following condition holds:*

$$u \xrightarrow{h} u' \text{ implies } \forall v (u =^L v \text{ implies } \exists v' (v \dashrightarrow v' \wedge u' \sim^L v'))$$

Intuitively, this corresponds to say that the low level user is not even able to tell whether the system before the high level transition was in u or in v. Generalised unwinding conditions have been used in [25] to study information flows on a basic concurrent imperative language. Unwinding conditions are nothing but generalised unwinding conditions in which the relation $=^L$ is the identity relation.

Surprisingly, in [4] it has been proved that some instances of the unwinding schema were equivalent to well-known security properties defined in terms of high level attacker models. Intuitively, if a system E satisfies the unwinding condition, then a high level malicious process Π interacting with any state E' reachable from E (denoted by $E'|\Pi$) cannot send down to the low level user private information (i.e., $E'|\Pi \sim^L E'$). In such context, the (generalised) unwinding framework is appealing for many reasons. It provides a universal quantification over an infinite set of attackers. It characterises persistent security properties, i.e., properties which hold also when the attack starts in the middle of the computation. It localises the reasoning/computation over the system. In many cases it naturally suggests refinements and correction policies.

2.3. . . . and Back: a Robust notion of Robustness

Our proposal is to use the notion of non-interference and its formalisation through unwinding conditions to define a general framework for robustness. Informally a system is robust if, when it moves to a perturbed state, the behaviours of its critical components remain almost unchanged. In other words, when a robust system performs a "high level action" which leads to a perturbed state the "low level behaviours" of its critical components are not influenced. Hence, in the biological setting high level actions play the role of perturbations and they interfere only with some specific components of the system, while low level

actions represent the functions to be preserved, where "preserved" means having a behaviour which is "equivalent" to the standard one. We try to better understand this idea on the examples presented in Section 2.1.

Example 4 (Research on Pathogens – part II). Let us consider again the system infected by a pathogen agent. The immune system and all the organs directly attacked by the pathogen are high level components, while the organs which we do not want to be dramatically affected are the low level ones (e.g., heart, lung, kidney, brain). Some changes in the immune system (high actions) could slightly modify the usual behaviour of the "low level" organs. To be acceptable such changes should be low level behaviourally equivalent to the usual evolution. In other terms, the system is robust with respect to the pathogen if the pathogen cannot *interfere* with its standard behaviour (i.e., if the system satisfies an unwinding condition).

Example 5 (Drug Development – part II). In the anti-inflammatory drugs example, COX1 and COX2 inhibition is the high level expected effect of the drugs, while gastrointestinal and cardiovascular problems are the low level undesirable changes. The "unwanted interactions" mentioned by Kitano [18] are the *interference* effects we intend to measure with low level observational equivalences. Hence, a drug can be used only if it does not interfere in the above sense with the system.

The above examples point out that when we define a notion of robustness based on non-interference, the low level actions are the ones we want to preserve and hence, they are the ones we are interested in observing. So, in the biological setting we call them *Exposed* ($\mathcal{E}xp$). On the other hand, high level actions are perturbations usually coming from the environment and, of course, actions of the system in response to such perturbations. Since, the system has limited (or possible none) control on external perturbations, we will call them *Imposed* (Imp). So we get the following definition of robustness.

Definiton 8 (Robustness through Non-Interference). *A biological system S which can both interact with the behaviour through Imposed Imp actions and perform Exposed $\mathcal{E}xp$ actions is said to be* robust through Non-Interference *if the Imposed behaviours do not influence the Exposed ones.*

As in the field of Information Flow Security, Non-Interference has been formalised through (generalised) unwinding conditions, in Systems Biology we can exploit them to characterise robustness.

Definiton 9 (Robustness through Unwinding). *Let $T = (V, V_I, Imp \cup \mathcal{E}xp, E)$ be a multilevel LTS representing a biological system, $=^{\mathcal{E}xp}, \sim^{\mathcal{E}xp} \subseteq V \times V$ be two equivalence relations, and $\dashrightarrow \subseteq V \times V$ be a transition relation. We say that T* is unwinding robust*, denoted as $T \in \mathcal{W}(=^{\mathcal{E}xp}, \sim^{\mathcal{E}xp}, \dashrightarrow)$, if for each $i \in Imp$ and for each $u, u' \in V$ the following condition holds:*

$$u \xrightarrow{i} u' \text{ implies } \forall v (u =^{\mathcal{E}xp} v \text{ implies } \exists v' (v \dashrightarrow v' \wedge u' \sim^{\mathcal{E}xp} v'))$$

Let us notice that we could have used "standard" unwinding condition from Definition 6 in place of generalised unwinding condition in our definition of robustness. However, the former is a particular case of the latter and, thus, it would have provided less instances of the framework. Moreover, as we will see in Section 3, we exploit generalised unwinding condition to reduce the state spaces of investigated models.

In order to instantiate the unwinding framework in the biological context we need to choose the exposed behavioural equivalences $\sim^{\mathcal{E}xp}$ and $=^{\mathcal{E}xp}$ and the transition relation $\dashrightarrow$. The exposed behavioural equivalences $\sim^{\mathcal{E}xp}$ and $=^{\mathcal{E}xp}$ establish which variations in the behaviour of the system are considered acceptable. In the literature comparisons between systems under different conditions are performed either observing simulations of the systems or checking that some system properties are true in all cases. These are just two possible cases of exposed behavioural equivalences. As we further explain in the rest of this section, unwinding conditions generalise these approaches. The transition relation $\dashrightarrow$ has a less intuitive meaning. We can view $\dashrightarrow$ as a possible *delay* between the behaviour of the system running in normal conditions and the system after an *imposed* change. In the case of biological systems this is coherent with a situation in which the imposed action speeds up a "reaction". Consider for instance the case of a model representing

the behaviour of a patient with a bacterial infection. The imposed action represents the decision of taking an antibiotic and immediately reaching a recovery state, while the $\dashrightarrow$ relation slowly leads to the recovery state without taking any drug. In this case $\dashrightarrow$ clearly represents a delay. On the other hand, when the imposed action is a perturbation on a system at the steady-state, the perturbed system needs time to go back to a steady-state. In the case $\dashrightarrow$ is the identity relation, i.e., no action is performed, while the "delay" needed by the perturbed system has to be absorbed by the exposed observational equivalence $\sim^{\mathcal{E}xp}$.

Going back to Kitano's requirements for a robustness framework [17], we can now say that in our general notion of robustness based on unwinding conditions, the functions of a system are mainly defined through the *exposed observational equivalences* $=^{\mathcal{E}xp}$ *and* $\sim^{\mathcal{E}xp}$, while the perturbations are modelled through *imposed actions*. Moreover, as far as the difference between stability and robustness is concerned, unwinding conditions allow us to move inside a lattice of equivalences where, travelling from finer (qualitative) relations to coarser (quantitative) ones, stability transits to robustness.

Let us explain in which sense *our notion of robustness is robust*. Recently many authors have proposed models for robustness in the biological setting (see, e.g., [26, 27, 28, 29, 30, 31, 32]). We can distinguish two main approaches: robustness as "sensitivity" and robustness as "properties preservation".

Sensitivity analysis aims at explaining the uncertainty in the output of a mathematical model by means of different sources of uncertainty in its inputs. It is based on statistical techniques and it has been successfully applied in the area of control theory to ensure robustness of engineered controllers. More recently, it has been proposed for the robustness analysis of biological systems (see e.g., [26]).

As far as properties preservation is concerned, Temporal Logics can be used as specification languages for expressing the properties of interest. Such logics have been traditionally used by the computer science community for the verification of real-time critical systems. Properties of system traces can be specified by these formalisms and Model Checking algorithms verify them on a given system (see, e.g, [2] for a general introduction). In [5] authors proposed to model biological systems through LTS's and exploit these well-established techniques in the biological setting. In such context robustness has been defined as a way to "measure the distance" (or in other terms a "degree of satisfaction") of a set of traces from a given specification (see, e.g., [27, 28, 29, 30, 31, 32]). In particular, Fainekos and Pappas defined in [27], and Brim *et al.* extended in [31], both a distance $\delta(s, s')$ between two signals (i.e., functions that associate with any time instant a tuple of values in $\mathbb{R}^n$) and a robustness degree $\mathbf{Dist}(s, \varphi)$ of the signal s on a Metric Temporal Logics (MTL) formula φ. They prove that if $\delta(s, s') \leq \mathbf{Dist}(s, \varphi)$, then the evaluations of φ on both s and s' give the same result. Hence, $\mathbf{Dist}(s, \varphi)$ represents the diameter of a maximal flow tube which includes s and over which the value of φ is constant. In other terms, it represents the minimum distance between s and signals over which φ has a different truth value. If $\mathbf{Dist}(s, \varphi) = 0$, then s is not robust on φ. More in general the following definition applies.

Definiton 10. *[Robustness as property preservation [27]] A signal s is a function $s : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ which associates with any time instant $t \in \mathbb{R}_{\geq 0}$ a tuple $s(t) = (x_1, \ldots, x_n)$ of values.*

Let s be a signal, φ a MTL formula, and S a set of perturbations S, i.e., a set of signals. We say that s is robust on φ with respect to S *if*

$$\delta(s, S) \stackrel{def}{=} \sup_{s' \in S} \delta(s, s') \leq \boldsymbol{Dist}(s, \varphi).$$

We can embed such notion of robustness in our framework as follows.

Definiton 11 (Signals LTS). *The timed LTS $LT(s)$ associated with a signal s is the tuple $(V, \{v_I\}, Imp \cup \mathcal{E}xp, E)$ where:*

- $V \stackrel{def}{=} \{(t, s(t)) \mid t \in \mathbb{R}_{\geq 0}\}$;
- $v_I \stackrel{def}{=} (0, s(0))$;
- $Imp \stackrel{def}{=} \emptyset$ *and* $\mathcal{E}xp \stackrel{def}{=} \mathbb{R}_{\geq 0}$;
- $E \stackrel{def}{=} \left\{\left(\left(t_1, s(t_1)\right), t_2 - t_1, \left(t_2, s(t_2)\right)\right) \mid t_1 \geq t_2\right\}$.

The untimed LTS is obtained from LT(s) by removing edge labels.

Example 6. Let us consider the signal s_1 defined as follows:

$$s_1(t) \stackrel{\text{def}}{=} \begin{cases} (3,-2) & \text{if } t \in [0,3] \\ (3,8.8) & \text{if } t \in (3,5] \\ (\pi,7) & \text{if } t > 5 \end{cases} .$$

The timed LTS $LT(s_1) = (V_1, \{v_{I,1}\}, \mathbb{R}_{\geq 0}, E_1)$ associated with $s_1(t)$ is such that:

- $V_1 \stackrel{\text{def}}{=} \Big\{\big(t,(3,-2)\big) \mid t \in [0,3]\Big\} \cup \Big\{\big(t,(3,8.8)\big) \mid t \in (3,5]\Big\} \cup \Big\{\big(t,(\pi,7)\big) \mid t > 5\Big\}$;
- $v_{I,1} \stackrel{\text{def}}{=} \big(0,(3,-2)\big)$;
- $E_1 \stackrel{\text{def}}{=} \Big\{\Big(\big(t_1,(3,-2)\big), t_2 - t_1, \big(t_2,(3,8.8)\big)\Big) \mid t_1 \in [0,3] \wedge t_2 \in (3,5]\Big\} \cup$
  $\Big\{\Big(\big(t_1,(3,-2)\big), t_2 - t_1, \big(t_2,(\pi,7)\big)\Big) \mid t_1 \in [0,3] \wedge t_2 > 5\Big\} \cup$
  $\Big\{\Big(\big(t_1,(3,8.8)\big), t_2 - t_1, \big(t_2,(\pi,7)\big)\Big) \mid t_1 \in (3,5] \wedge t_2 > 5\Big\}$.

Given a signal s and a set of signals S we build a model containing s and all the signals of S in which s is interpreted as the standard behaviour, while the elements of S are the perturbed behaviours which occur after an Imposed action ι.

Definiton 12 (Set Signals LTS). *Let s and S be a signal and a set of signals, respectively. Moreover, let* $(V^p, \{v_I^p\}, \mathit{Imp} \cup \mathit{Exp}^p, E^p)$ *be the LTS LT(p) for all p in* $S \cup \{s\}$. *The* set signals LTS $LT(s,S)$ *is the tuple* $(V, \{v_I\}, \mathit{Imp} \cup \mathit{Exp}, E)$ *where:*

- $V \stackrel{def}{=} \{init\} \cup (V^s \times \{s\}) \cup \bigcup_{s' \in S}(V^{s'} \times \{s'\})$;
- $v_I \stackrel{def}{=} init$;
- $\mathit{Imp} \stackrel{def}{=} \{\iota\}$ *and* $\mathit{Exp} \stackrel{def}{=} \mathbb{R}_{\geq 0} \cup \{\epsilon\}$ *where* ι *and* ϵ *are two fresh symbols;*
- $E \stackrel{def}{=} \{(init, \epsilon, (v_I^s, s))\} \cup \{(init, \iota, (v_I^{s'}, s')) \mid s' \in S\} \cup \mathit{lift}\,(E^s, s) \cup \bigcup_{s' \in S} \mathit{lift}\left(E^{s'}, s'\right)$;

and $\mathit{lift}\,(F,q) \stackrel{def}{=} \{((u,q), \alpha, (v,q)) \mid (u,\alpha,v) \in F\}$.

From an intuitive point of view, $LT(s,S)$ is the disjoint union of the LTS's of s and of each of $s' \in S$ together with a new distinguished node *init* which reaches all the nodes of the form $(0,r)$. From the node *init*, $LT(s,S)$ can reach the LTS $LT(s)$ by crossing an edge whose label is $\epsilon \in \mathit{Exp}$; on the contrary, all the edges that connect *init* to one of the LTS $LT(s')$, where $s' \in S$, are labelled by $\iota \in \mathit{Imp}$ (see Figure 3).

We define an unwinding condition which embeds the robustness notion defined in [27] in our framework.

Definiton 13. *The unwinding condition for the temporal formula* φ *is* $\mathcal{W}^\varphi \stackrel{def}{=} \mathcal{W}(id, \approx^\varphi, \xrightarrow{\epsilon})$ *where id is the identity relation and* $\approx^\varphi$ *is the smallest equivalence relation such that:*

$$\big((0,s(0)),s\big) \approx^\varphi \big((0,s'(0)),s'\big) \quad \textit{iff} \quad \delta(s,s') \leq \boldsymbol{Dist}\,(s,\varphi)$$

Latter on, with a small abuse of notation, we may write $(0,s'(0)) \approx^\varphi (0,s(0))$ in place of $((0,s'(0)),s') \approx^\varphi ((0,s(0)),s)$.

Proposition 2. *Let s be a signal, S be a set of signals, and* φ *a temporal formula. It holds that s is robust on* φ *with respect to S if and only if* $LT(s,S)$ *satisfies* $\mathcal{W}^\varphi$.

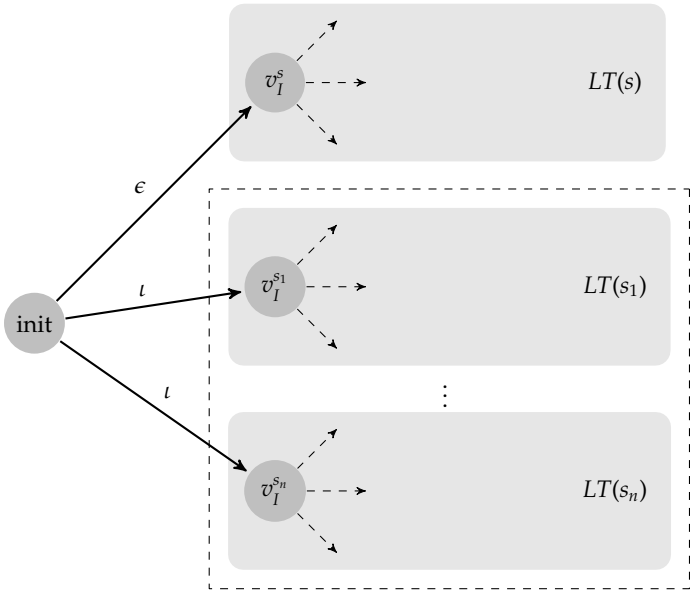

Figure 3: The set signals $LTS(s, S)$ is the disjoint union of all the LTS associated with $\bar{s} \in S \cup \{s\}$. This LTS has one single initial node *init* from which $LTS(s, S)$ can reach the initial node $v_I^{\bar{s}}$ of any $LT(\bar{s})$ by crossing an edge $e_{\bar{s}}$. If $\bar{s} = s$, then $e_{\bar{s}}$ is labelled by ϵ; in the remaining cases, $e_{\bar{s}}$ is labelled by ι. Since some of the LTS may share some states, all the nodes of the same $LT(\bar{s})$ are formally labelled by $\bar{s}$.

Proof. $\Rightarrow$) In $LT(s, S)$ if $u \xrightarrow{\iota} u'$, then $u = init$ and u' is of the form $((0, s'(0)), s')$ with $s' \in S$. Hence, $init \xrightarrow{\epsilon} ((0, s(0)), s)$. Moreover, by Definition 10, it holds that $\delta(s, s') \leq \delta(s, S) \leq \mathbf{Dist}(s, \varphi)$ and hence $((0, s'(0)), s') \approx^{\varphi} ((0, s(0)), s')$. Since *id* is the identity relation, $LT(s, S)$ satisfies $\mathcal{W}^{\varphi}$.

$\Leftarrow$) Since $LT(s, S)$ satisfies $\mathcal{W}^{\varphi}$, for each $s' \in S$ we have that $((0, s'(0)), s') \approx^{\varphi} ((0, s(0)), s)$. By definition of $\approx^{\varphi}$ this means that for each $s' \in S$ $\delta(s, s') \leq \mathbf{Dist}(s, \varphi)$. Hence, $\delta(s, S) = sup_{s' \in S} \delta(s, s') \leq \mathbf{Dist}(s, \varphi)$. □

Of course this is a trivial way to map the notion of robustness defined in [27] in our framework. However, this embedding reveals that we can immediately generalise this robustness notion and consider sets of formulæ by simply modifying $\approx^{\varphi}$. As a matter of fact, any set Φ of formulæ over a temporal language $\mathcal{L}$ can be used to define an exposed observational equivalence $\approx^{\Phi}$ as follows:

$$u \approx^{\Phi} v \qquad \text{iff} \qquad \forall \varphi \in \Phi \; u \approx^{\varphi} v.$$

In temporal and modal logics, bisimulations are the natural equivalences that establish which states are indistinguishable with respect to the logic, i.e., whenever two states are bisimilar all the formulæ of the logic have the same truth values on them. So, if we consider as exposed observational equivalence a bisimulation equivalence over $\mathcal{L}$, we are sure that the perturbed and unperturbed systems are indistinguishable with respect to any formula of $\mathcal{L}$. In this sense unwinding conditions based on bisimulation relations can be seen as *robust robustness* definitions. Such intuition can be formalised as follows.

Definiton 14. *Let $\approx$ be a bisimulation relation over $\mathcal{L}$. The* robust unwinding condition for $\mathcal{L}$ *is* $\mathcal{W}(id, \approx, \xrightarrow{\epsilon})$.

Proposition 3. *Let s be a signal and S be a set of signals. If $LT(s, S)$ satisfies the robust unwinding condition for $\mathcal{L}$, then s is robust on any formula φ over $\mathcal{L}$ with respect to S.*

A strong objection against the use of bisimulation as exposed observational equivalence for unwinding conditions characterising robustness concerns the fact that in the case of real time/space systems and even more crucially in the case of probabilistic systems (e.g., DTMCs and CTMCs), bisimulation relations could be too demanding. An answer to this objection comes from bisimulations notions which take into

consideration continuous domain, probability, errors (see, e.g., [33, 34, 35, 36, 37, 38]). In particular, [33] and [34] introduced a bisimulation distance $\delta B(u, v)$ between states that can be used to define exposed observational equivalences as $u \approx^k v$ iff $\delta B(u, v) \leq k$, where k is a constant. The algorithm proposed in [39] can be used to efficiently compute such relations on DTMC models.

Notice also that in our encoding we used LTS's in the simplest possible way. In general, using LTS's we can finitely represent any *regular* infinite set S of perturbed signals. Moreover, with a clever use of exposed actions we can exploit unwinding robustness also to compare perturbed signals and to analyse the effects of applying more than one perturbation.

3. Bisimulation Based Unwinding Conditions on Bio-PEPA

As an example of a computational language for modelling biological systems, we consider in this section the stochastic process algebra Bio-PEPA. Different process algebra for modelling biological systems have been introduced in the literature (e.g., see [40, 41, 42, 43, 44]). We decided to consider here Bio-PEPA since different semantics and behavioural equivalences have been proposed on it (e.g., see [45, 10]) and many biological systems have been modelled in Bio-PEPA and analysed through public available tools [46, 47, 48, 49].

A detailed description of Bio-PEPA is out of the scope of our work and we refer the reader to Bio-PEPA website[1], where complete bibliography, implementations, and case-studies can be found. The following definitions and notions mainly come from [10].

The main components of a Bio-PEPA system are the sequential components C and the model component P. The former describe the evolutions of each species. The latter models the interactions among species and define the initial values. Bio-PEPA also requires a context that specifies functional rates, compartments (i.e., species constraints), and parameters.

Intuitively, populations of molecules are modelled as species. The sequential components are templates of behaviours for a population and they describe all the possible reactions α involving it. The model component represents the environment in which the populations interact.

Definiton 15 (Bio-PEPA [10]). *A* well-defined Bio-PEPA sequential component C *is a recursive process of the form:*

$$C \stackrel{def}{=} (\alpha_1, \kappa_1)\mathbf{op}_1 C + \ldots (\alpha_q, \kappa_q)\mathbf{op}_q C \qquad \textit{sometime written as} \qquad C \stackrel{def}{=} \sum_{i=1}^{q} (\alpha_i, \kappa_i)\mathbf{op}_i C$$

where $\mathbf{op}_i \in \{\uparrow, \downarrow, \oplus, \ominus, \odot\}$, α_i *is an action name,* κ_i *is a stoichiometric coefficient, and* $\alpha_i \neq \alpha_j$ *for* $i \neq j$.

A well-defined Bio-PEPA model component *is a process of the form:*

$$P \stackrel{def}{=} C_1(x_1) \underset{\mathcal{L}_1}{\bowtie} \ldots \underset{\mathcal{L}_{p-1}}{\bowtie} C_p(x_p)$$

where the C_i*'s are well-defined sequential components which are pairwise different, the* x_i*'s represent initial concentrations, and* $\mathcal{L}_i$*'s are sets of actions that appear in* P.

We may interpret $\downarrow$ as a reactant, $\uparrow$ a product, $\oplus$ an activator, $\ominus$ as an inhibitor, and $\odot$ as a generic modifier. Bio-PEPA model components are used to rule the evolution of a Bio-PEPA system.

Definiton 16 (Bio-PEPA System [10]). *A* Bio-PEPA system $\mathcal{P}$ *is a 6-tuple* $(\mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}, Comp, P)$ *where* $\mathcal{V}$ *is the set of compartments,* $\mathcal{N}$ *is the set of quantities describing each species,* $\mathcal{K}$ *is a set of parameters,* $\mathcal{F}$ *is the set of functional rates, Comp is the set of well-defined sequential components, and P consists of a well-defined Bio-PEPA model over Comp.*

[1] http://homepages.inf.ed.ac.uk/jeh/Bio-PEPA/biopepa.html

$$\texttt{prefixReac}\quad \frac{}{(\alpha,\kappa)\downarrow S(l) \xrightarrow{(\alpha,[S:\downarrow(l,\kappa)])}_c S(l-\kappa)}\ N'_S+\kappa\le l\le N_S$$

$$\texttt{prefixProd}\quad \frac{}{(\alpha,\kappa)\uparrow S(l) \xrightarrow{(\alpha,[S:\uparrow(l,\kappa)])}_c S(l+\kappa)}\ N'_S\le l\le N_S-\kappa$$

$$\texttt{prefixMod}\quad \frac{}{(\alpha,\kappa)\ \ \texttt{op}\ \ S(l) \xrightarrow{(\alpha,[S:\mathsf{op}(l,\kappa)])}_c S(l)}\ \begin{array}{l} N'_S+\kappa\le l\le N_S \text{ if } \texttt{op}=\oplus \\ N'_S\le l\le N_S \text{ if } \texttt{op}=\{\ominus,\odot\}\end{array}$$

$$\texttt{choice1}\quad \frac{S_1(l) \xrightarrow{(\alpha,w)}_c S'_1(l')}{(S_1+S_2)\,(l) \xrightarrow{(\alpha,w)}_c S'_1(l')} \qquad \texttt{choice2}\quad \frac{S_2(l) \xrightarrow{(\alpha,w)}_c S'_2(l')}{(S_1+S_2)\,(l) \xrightarrow{(\alpha,w)}_c S'_2(l')}$$

$$\texttt{coop1}\quad \frac{P_1 \xrightarrow{(\alpha,w)}_c P'_1}{P_1 \underset{\mathcal{L}}{\bowtie} P_2 \xrightarrow{(\alpha,w)}_c P'_1 \underset{\mathcal{L}}{\bowtie} P_2}\ \alpha\notin\mathcal{L} \qquad \texttt{coop2}\quad \frac{P_2 \xrightarrow{(\alpha,w)}_c P'_2}{P_1 \underset{\mathcal{L}}{\bowtie} P_2 \xrightarrow{(\alpha,w)}_c P_1 \underset{\mathcal{L}}{\bowtie} P'_2}\ \alpha\notin\mathcal{L}$$

$$\texttt{coop3}\quad \frac{P_1 \xrightarrow{(\alpha,w_1)}_c P'_1 \quad P_2 \xrightarrow{(\alpha,w_2)}_c P'_2}{P_1 \underset{\mathcal{L}}{\bowtie} P_2 \xrightarrow{(\alpha,w_1::w_2)}_c P'_1 \underset{\mathcal{L}}{\bowtie} P'_2}\ \alpha\in\mathcal{L} \qquad \texttt{constant}\quad \frac{S(l) \xrightarrow{(\alpha,[S:op(l,\kappa)])}_c S'(l')}{C(l) \xrightarrow{(\alpha,[C:op(l,\kappa)])}_c S'(l')}\ C\stackrel{\text{def}}{=}S$$

Table 1: The operational semantics of Bio-PEPA model components [10]

$$\texttt{Final}\quad \frac{P \xrightarrow{(\alpha,w)}_c P'}{(\mathcal{V},\mathcal{N},\mathcal{K},\mathcal{F},Comp,P) \xrightarrow{(\alpha,r_\alpha[w,\mathcal{N},\mathcal{K}])}_s (\mathcal{V},\mathcal{N},\mathcal{K},\mathcal{F},Comp,P')}$$

$$\texttt{Qual}\quad \frac{P \xrightarrow{(\alpha,w)}_c P'}{(\mathcal{V},\mathcal{N},\mathcal{K},\mathcal{F},Comp,P) \xrightarrow{\alpha} (\mathcal{V},\mathcal{N},\mathcal{K},\mathcal{F},Comp,P')}$$

Table 2: The operational semantics of Bio-PEPA systems [10]

The operational semantics of Bio-PEPA is defined through the rules reported in Table 1 and 2.

Each species is labelled by a value, called *level*, that denotes its abundance. For instance, the level of a species A may represents the number of molecules of a substance or the number of individual of a population depending on the meaning to A itself. In a generic Bio-PEPA system $(\mathcal{V},\mathcal{N},\mathcal{K},\mathcal{F},Comp,P)$, $\mathcal{N}$ constrains the level of any species S and it ties it to an interval $[N'_S,N_S]$. N'_S and N_S are called *minimum level* and *minimum level* of S, respectively.

The rules `prefixProd`, `prefixReac`, and `prefixMod` constrain the application of an action α and, at the same time, detail their effects on a sequential component S. For instance, `prefixReac` means that, whenever S is a reactant of a reaction α (i.e., $\downarrow$) with stoichiometric coefficient κ, any occurrence of α reduces the level, l, of S to $l-\kappa$ units. In order to maintain these values in the interval $[N'_S,N_S]$, α can occurs if and only if l is included in the interval between the minimum level of S (i.e., N'_S) plus κ and the maximum level of S (i.e., N_S). Analogously, `prefixProd` rules that if S is a product of a reaction α (i.e., $\uparrow$) with stoichiometric coefficient κ, α can occur only if the level of the species S is included in the interval between the minimum

level and the maximum level plus κ. In such a case, when α occurs it increases the level of S from l to $l + \kappa$.

The rules `choice1` and `choice2` detail the behaviour of a system that chooses between two (or more) sequential components (i.e., $S_1 + S_2$): if an action α brings S_1 (S_2) with level l to S_1' (S_2') with level l', then the same action can bring $S_1 + S_2$ with level l to S_1' (S_2', respectively) with level l'.

The rules `coop1`, `coop2`, and `coop3` guide the evolution of model components: if the process P_1 (P_2) may evolve in P_1' (P_2') because of an action α, but P_1 and P_2 do not synchronise on α (i.e., $\alpha \notin \mathcal{L}$), then $P_1 \underset{\mathcal{L}}{\bowtie} P_2$ evolves in $P_1 \underset{\mathcal{L}}{\bowtie} P_2'$ ($P_1' \underset{\mathcal{L}}{\bowtie} P_2$, respectively) because of α. Otherwise, if P_1 may evolve in P_1' because of α, P_2 may evolve in P_2' because of α, and P_1 and P_2 synchronise on α (i.e., $\alpha \in \mathcal{L}$), then $P_1 \underset{\mathcal{L}}{\bowtie} P_2$ evolves in $P_1' \underset{\mathcal{L}}{\bowtie} P_2'$ because of the same action.

In [50] these rules are at the basis of the construction of a *Stochastic LTS* representing a Bio-PEPA model. In [51] such LTS's are mapped into CTMCs with levels. In [10] LTS over Bio-PEPA systems are defined by both discretising concentrations through compartments and abstracting the above mentioned rules from quantitative aspects. The authors obtain LTS's whose nodes are tuples of integers and whose edges have the form $\xrightarrow{\alpha}$, where α is a reaction name.

Example 7 (A simple Bio-PEPA model). Let us consider one of the examples presented in [10]. It represents a system with 3 species interacting through 3 reactions.

$$A = (\alpha_1, 1) \downarrow A + (\alpha_2, 1) \uparrow A + (\alpha_3, 2) \downarrow A$$
$$B = (\alpha_3, 1) \uparrow B$$
$$C = (\alpha_1, 1) \uparrow C + (\alpha_2, 1) \downarrow C$$
$$P = A(x_A) \underset{*}{\bowtie} B(x_B) \underset{*}{\bowtie} C(x_C)$$

where $\underset{*}{\bowtie}$ denotes the synchronisation on all actions.

The possibility of associating different semantics to Bio-PEPA systems allows modeller to exploit different behavioural equivalences on them. While lumpability relations can be computed on the CTMCs associated with Bio-PEPA systems [51], a qualitative notion of bisimulation has been introduced in [10], where LTS's are first abstracted from level quantities, then compared using strong bisimulation.

We introduce here a simple instance of our framework over Bio-PEPA process algebra. First we extend the standard notation of Bio-PEPA systems. We partition the set of species into two sets *Imp* and *Exp*. As a consequence we partition the set $\mathcal{A}$ of action names into two sets Imp and Exp of imposed and exposed actions, respectively, as follows: each action which occurs in the definition of species of *Imp* is in Imp, all the remaining actions are in Exp.

Example 8. Let us consider the Bio-PEPA model specified in Example 7. If $Imp = \{C\}$ and $Exp = \{A, B\}$, then $\mathit{Imp} = \{\alpha_1, \alpha_2\}$ and $\mathit{Exp} = \{\alpha_3\}$. From now on we will use this system as illustrative example.

As observed in [10], given a system $\mathcal{P}$ we can denote the states of its LTS by using tuples of integers representing the levels of the species (see Example 9). We use the same convention, but we distinguish between species of *Imp* and *Exp*, i.e., our tuples have the form $(i_1, \dots i_m, e_1, \dots e_n)$, where $i_1, \dots, i_m$ are the levels of the species of *Imp*, while $e_1, \dots, e_n$ are the levels of the ones of *Exp*.

Definiton 17 (LTS of $\mathcal{P}$ [10]). *The* LTS $L(\mathcal{P})$ *of a Bio-PEPA system $\mathcal{P}$ is the LTS* $(\mathbb{P}, \{\mathcal{P}\}, \mathit{Imp} \cup \mathit{Exp}, \rightarrow)$ *where:*

- $\mathbb{P}$ *is the set of all the possible Bio-PEPA systems;*
- $\mathit{Imp} \cup \mathit{Exp}$ *is the set of admissible actions;*
- $\rightarrow$ *is the relation associated with the rule* `Qual` *of the Table 2.*

Whenever $\mathcal{P} \to \mathcal{P}'$, $\mathcal{P}'$ differs from $\mathcal{P}$ only in the levels of some components. Hence, the state of $L(\mathcal{P})$ can be represented by a vector of levels. Moreover, since all the species range over a finite set of levels, the LTS of a Bio-PEPA system is finite.

Example 9 (LTS of a Bio-PEPA system). Let us reconsider the Bio-PEPA system described in Example 8. When the concentration level admitted for each species ranges in $[0, 4]$ and the evolution begins from the state $x_A = 4$, $x_B = 0$, and $x_C = 0$, the LTS of it is the transition system $P(4, 0, 0[4, 4, 4])$ depicted in Fig. 4. We recall that in this LTS we have $\mathit{Imp} = \{\alpha_1, \alpha_2\}$ and $\mathit{Exp} = \{\alpha_3\}$.

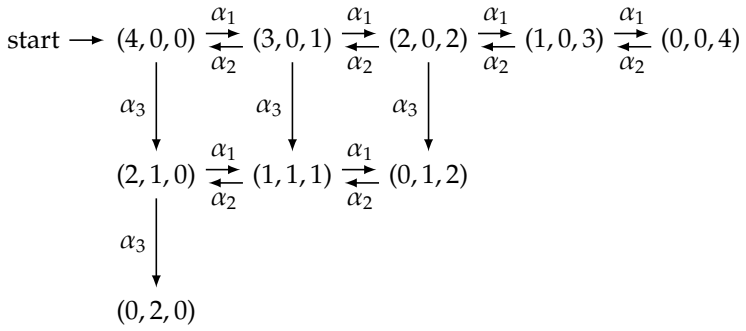

Figure 4: The transition system $P(4, 0, 0[4, 4, 4])$ of Example 7.

We define the relations which equate nodes from an exposed point of view as follows. We use the notation $u \to u'$ to denote $u \xrightarrow{\alpha} u'$ for some $\alpha \in \mathcal{A}$.

Definiton 18 ($\doteq^{Exp}$ and $\approx^{Exp}$). *Let u, v be two states of $L(\mathcal{P})$. We say $u \doteq^{Exp} v$ if and only if the exposed variables of u and v have the same values. A* strong exposed bisimulation *over $L(\mathcal{P})$ is a symmetric binary relation R over $L(\mathcal{P})$ such that for each $(u, v) \in R$:*

- *$u \doteq^{Exp} v$;*
- *if $u \to u'$, then $v \to v'$ and $(u', v') \in R$.*

Two states u and v are strongly exposed bisimilar, *denoted by $u \approx^{Exp} v$, if there exists a strong exposed bisimulation R such that $(u, v) \in R$.*

It is easy to prove that $\approx^{Exp}$ is the largest exposed bisimulation and it is an equivalence relation.
We are now ready to define our instance of the generalised unwinding framework.

Definiton 19 (Quantitative Unwinding). *$\mathcal{P}$ satisfies the* quantitative unwinding *if $L(\mathcal{P}) \in \mathcal{W}(\doteq^{Exp}, \approx^{Exp}, \to)$.*

Unwinding conditions can be introduced over any operational semantics based on LTS's. The advantages of considering unwinding conditions over process algebras such as Bio-PEPA are twofold. On the one hand, we can prove properties over the systems satisfying the unwinding conditions in terms of their interactions with other systems. In particular, we can prove that these systems are not critically affected by hostile environments (see Theorem 1, hereafter). On the other hand, we can exploit compositionality properties of the syntactic operators to both reduce the complexity of checking the unwinding condition and to suggest *corrections* when unwinding is not satisfied.

Theorem 1. *It holds that $L(\mathcal{P}) \in \mathcal{W}(\doteq^{Exp}, \approx^{Exp}, \to)$ if and only if for each $u, v \in L(\mathcal{P})$ if $u \doteq^{Exp} v$, then $u \approx^{Exp} v$.*

Proof. $\Rightarrow$) Let $\mathcal{S} = \{(u, v) \mid u, v \in L(\mathcal{P}) \text{ and } u \doteq^{Exp} v\}$. We prove that $\mathcal{S}$ is a strong exposed bisimulation up to $\approx^{Exp}$. Let $(u, v) \in \mathcal{S}$, i.e., $u \doteq^{Exp} v$. If $u \xrightarrow{\epsilon} u'$ with $\epsilon \in \mathcal{E}xp$, then since $u \doteq^{Exp} v$ and both the side-conditions and the resultant of applying an ϵ transition only depend on Exposed quantities, it holds that there exists v' such that $v \xrightarrow{\epsilon} v'$ and $u' \doteq^{Exp} v'$, i.e., $(u', v') \in \mathcal{S}$. On the other hand, if $u \xrightarrow{\iota} u'$ with $\iota \in Imp$, then, since $L(\mathcal{P}) \in \mathcal{W}(\doteq^{Exp}, \approx^{Exp}, \rightarrow)$ and $u \doteq^{Exp} v$, there exists v' such that $v \rightarrow v'$ and $u' \approx^{Exp} v'$.

$\Leftarrow$) Let $u \xrightarrow{\iota} u'$ with $\iota \in Imp$ and $u \doteq^{Exp} v$. Since $u \doteq^{Exp} v$ implies $u \approx^{Exp} v$, we get that $v \rightarrow v'$ with $u' \approx^{Exp} v'$. □

The above theorem states that changes in the imposed environment of a system satisfying the unwinding condition do not affect the exposed behaviour of the system itself. Moreover, it suggests an efficient algorithm for testing $L(\mathcal{P}) \in \mathcal{W}(\doteq^{Exp}, \approx^{Exp}, \rightarrow)$ as explicitly stated in the following corollary.

Corollary 1. *It holds that $L(\mathcal{P}) \in \mathcal{W}(\doteq^{Exp}, \approx^{Exp}, \rightarrow)$ if and only if $L(\mathcal{P})/\approx^{Exp}$ coincides with $L(\mathcal{P})/\doteq^{Exp}$.*

Proof. This is an immediate consequence of Theorem 1 and of the fact that $u \approx^{Exp} v$ implies $u \doteq^{Exp} v$. □

Hence, to test whether $L(\mathcal{P}) \in \mathcal{W}(\doteq^{Exp}, \approx^{Exp}, \rightarrow)$ one should simply start a partitioning bisimulation algorithm, such as Paige-Tarjan algorithm [14], on $L(\mathcal{P})$ to compute $\approx^{Exp}$. If the algorithm terminates without performing any split, then $L(\mathcal{P})$ satisfies the unwinding condition. On the other hand, if it performs a split, we can immediately stop the computation and return $L(\mathcal{P}) \notin \mathcal{W}(\doteq^{Exp}, \approx^{Exp}, \rightarrow)$. Since such bisimulation algorithms can work both symbolically and on-the-fly, they allow us to test the unwinding condition without fully computing $L(\mathcal{P})$ and avoid space-explosion problems.

Example 10. If we consider the system of Example 8, we get that it satisfies the quantitative unwinding, since as shown by Figure 4 in this case both $\doteq^{Exp}$ and $\approx^{Exp}$ are the identity relation.

Notice that if a system is in $\mathcal{W}(\doteq^{Exp}, \approx^{Exp}, \rightarrow)$, then imposed species cannot produce/consume exposed species, unless the same production/consumption can be also achieved exploiting only exposed species. In the case of closed chemical systems satisfying the conservation mass law this corresponds to imposing a complete separation between exposed and imposed species. However, this is not the case in biological systems where the presence of degradation laws make systems non-conservative.

Example 11. Consider a simple system in which a species $E \in Exp$ can be produced from a reaction which involves a species $I \in Imp$. Moreover, both species can degrade.

$$E = (\epsilon, 1) \downarrow E + (\iota_1, 1) \uparrow E$$
$$I = (\iota_1, 1) \downarrow I + (\iota_2, 1) \downarrow I$$

No matter which are the boundary levels defining such system, if its LTS includes one ι_1 action, then the system does not satisfy the unwinding condition. Intuitively, if I is a vaccine (or a pathogen), since it can produce a species in Exp, it interferes with the system's observable behaviour. We obtain the same result also if ι_1 would require the presence of other species of Exp.

On the other hand, let us consider a different system that contains, in addition to the reactions depicted by the above rules, a further reaction δ that produces E.

$$E = (\epsilon, 1) \downarrow E + (\iota_1, 1) \uparrow E + (\delta, 1) \uparrow E$$
$$I = (\iota_1, 1) \downarrow I + (\iota_2, 1) \downarrow I$$

Since $\approx^{\mathcal{E}xp}$ is qualitative with respect to the rates, this second system satisfies the unwinding condition for each possible values of ι_1 and δ. By imposing a more qualitative equivalence we would obtain constraints on these values. See Figure 5 to have a graphical representation of the system when $E \in [0, 3]$ and $I \in [1, 3]$. In such a case, the system satisfies the quantitative unwinding as both $\doteq^{Exp}$ and $\approx^{Exp}$ are the identity relation over E.

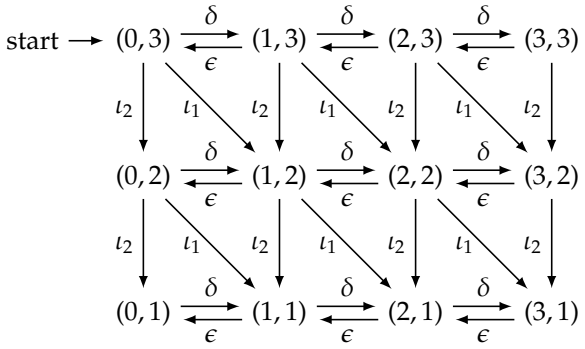

Figure 5: The transition system $P(0,3[4,3])$ of the second Bio-PEPA system of Example 11.

Let us now introduce a second unwinding condition, called *quasi-quantitative* on Bio-PEPA LTS's. At a first sight this could seem less demanding than quantitative unwinding however we will show that this is not the case.

Definiton 20 ($\doteqdot^{\mathcal{E}xp}$ and $\approxeq^{\mathcal{E}xp}$). *Let u, v be two states of $L(\mathcal{P})$. We say $u \doteqdot^{\mathcal{E}xp} v$ if and only if u and v have the same sets of $\mathcal{E}xp$ labels on outgoing edges. A* strong quasi-exposed bisimulation *over $L(\mathcal{P})$ is a symmetric binary relation R over $L(\mathcal{P})$ such that for each $(u,v) \in R$:*

- *$u \doteqdot^{\mathcal{E}xp} v$;*
- *if $u \to u'$, then $v \to v'$ and $(u', v') \in R$.*

Two states u and v are strongly quasi-exposed bisimilar, *denoted by $u \approxeq^{\mathcal{E}xp} v$, if there exists a strong exposed bisimulation R such that $(u,v) \in R$.*

Definiton 21 (Quasi-Quantitative Unwinding). *$\mathcal{P}$ satisfies the quasi-quantitative unwinding if $L(\mathcal{P}) \in \mathcal{W}(\doteqdot^{\mathcal{E}xp}, \approxeq^{\mathcal{E}xp}, \to)$.*

Even if $\doteq^{Exp}$ implies $\doteqdot^{\mathcal{E}xp}$, the following examples show that $\mathcal{W}(\doteqdot^{\mathcal{E}xp}, \approxeq^{\mathcal{E}xp}, \to) \not\supseteq \mathcal{W}(\doteq^{Exp}, \approx^{Exp}, \to)$.

Example 12. Let us consider the Bio-PEPA model defined as follows:

$$A = (\epsilon_1, 2) \downarrow A + (\epsilon_2, 3) \downarrow A$$
$$B = (\epsilon_1, 5) \uparrow B + (\epsilon_2, 6) \uparrow B + (\iota, 6) \downarrow B$$
$$C = (\iota, 1) \uparrow C$$
$$P = A(x_A) \underset{*}{\bowtie} B(x_B) \underset{*}{\bowtie} C(x_C)$$

where A and B are exposed species, while C is an imposed species.

The LTS $P(3,0,0[6,6,6])$ that is associated with the above model when the concentration level admitted for each species ranges in $[0,6]$ and the evolution begins from the state $x_A = 3$, $x_B = 0$, and $x_C = 0$ is depicted in Figure 6. It is easy to see that $P(3,0,0[6,6,6]) \in \mathcal{W}(\doteq^{Exp}, \approx^{Exp}, \to)$ because the states in $P(3,0,0[6,6,6])$ are pairwise distinct with respect to the exposed species. On the contrary, $(1,5,0)$, $(0,6,0)$, and $(0,0,1)$ have the same exposed actions and, thus, $(1,5,0) \doteqdot^{\mathcal{E}xp} (0,6,0)$. However, $(0,6,0) \xrightarrow{\iota} (0,0,1)$, while there is no transition leaving $(1,5,0)$. It follows that $P(3,0,0[6,6,6]) \notin \mathcal{W}(\doteqdot^{Exp}, \approxeq^{Exp}, \to)$.

Notice that also the system of Example 8 satisfies the quantitative unwinding (as proved in Example 10), but not the quasi-quantitative unwinding. In particular, $(0,0,4) \doteqdot^{\mathcal{E}xp} (0,2,0)$ and $(0,0,4)$ reaches $(1,0,3)$ through an imposed action, but $(0,2,0)$ does not reach any state equivalent to $(1,0,3)$, since it does not reach any state.

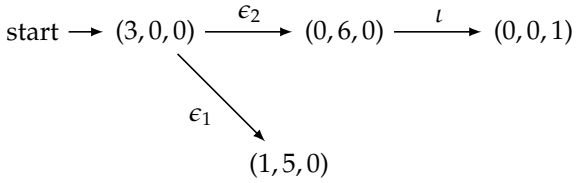

Figure 6: The LTS $P(3,0,0[6,6,6])$ of the Bio-PEPA model described in Example 12.

Given a system $\mathcal{P}$ the LTS $L(\mathcal{P})$ can be very large. Moreover, it distinguishes too precisely on the basis of quantitative information also LTS's which represent the same qualitative behaviours. The qualitative LTS of $\mathcal{P}$ is obtained as a quotient of $L(\mathcal{P})$ in which nodes having the same capabilities (outgoing actions) are identified.

Definiton 22 (QLTS of $\mathcal{P}$). *Let $\mathcal{P}$ be a Bio-PEPA system. Given two states u, v of $L(\mathcal{P})$ we say that $u \doteqdot v$ if and only if u and v have the same sets of labels on outgoing edges.*

The qualitative LTS *$QL(\mathcal{P})$ of a Bio-PEPA system is the LTS $L(\mathcal{P})/\doteqdot$.*

Since all the states in any class of $L(\mathcal{P})/\doteqdot$ share the same admissible actions, we can denote each node of $QL(\mathcal{P})$ by using the set of these actions. The qualitative semantics defined in [10] for Bio-PEPA systems is close to our definition, with the difference that in [10] for each pair of classes $[u]_{\doteqdot}$, $[v]_{\doteqdot}$ in $QL(\mathcal{P})$ there exists at most one edge from $[u]_{\doteqdot}$ to $[v]_{\doteqdot}$ whose label is the set of actions which allow to move from $[u]_{\doteqdot}$ to $[v]_{\doteqdot}$.

Example 13 (QLTS of a Bio-PEPA system). Let us reconsider the Bio-PEPA system described by Example 7. When the concentration level admitted for each species ranges in $[0,4]$ and the evolution begins from the state $x_A = 4$, $x_B = 0$, and $x_C = 0$, the QLTS of it is the transition system depicted in Figure 7.

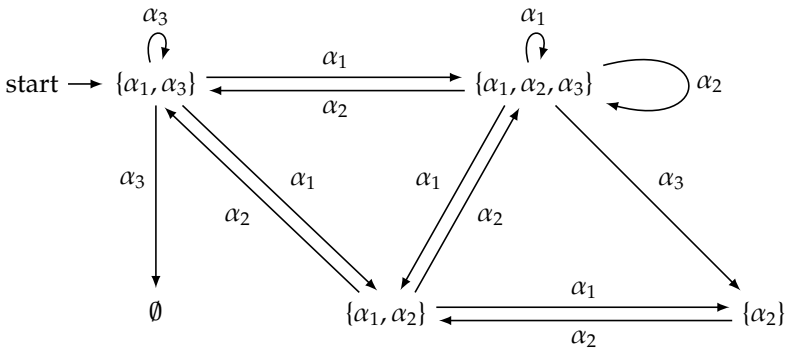

Figure 7: The qualitative transition system of the Bio-PEPA system described in Example 7.

In order to define a generalised unwinding condition over $QL(\mathcal{P})$ we introduce the following relations. With a slight abuse of notation we use the same symbols used for quasi-quantitative unwinding.

Definiton 23 ($\doteqdot^{\mathcal{E}xp}$ and $\cong^{\mathcal{E}xp}$). *Let $[u]_{\doteqdot}, [v]_{\doteqdot}$ be two states of $QL(\mathcal{P})$. We define the relation $\doteqdot^{\mathcal{E}xp}$ as follows: for each $[u]_{\doteqdot}, [v]_{\doteqdot} \in QL(\mathcal{P})$ it holds that $[u]_{\doteqdot} \doteqdot^{\mathcal{E}xp} [v]_{\doteqdot}$ if and only if $[u]_{\doteqdot}$ and $[v]_{\doteqdot}$ have the same sets of $\mathcal{E}xp$ labels on outgoing edges.*

We define the relation $\cong^{\mathcal{E}xp}$ as follows: for each $[u]_{\doteqdot}, [v]_{\doteqdot} \in QL(\mathcal{P})$ it holds that $[u]_{\doteqdot} \cong^{\mathcal{E}xp} [v]_{\doteqdot}$ if and only if there exist $u' \in [u]_{\doteqdot}$ and $v' \in [v]_{\doteqdot}$ such that $u' \cong^{\mathcal{E}xp} v'$ holds in $L(\mathcal{P})$.

Notice that $[u] \doteqdot^{\mathcal{E}xp} [v]$ in $QL(\mathcal{P})$ if and only if $u \doteqdot^{\mathcal{E}xp} v$ in $L(\mathcal{P})$. Both $\doteqdot^{\mathcal{E}xp}$ and $\cong^{\mathcal{E}xp}$ are equivalence relations on $QL(\mathcal{P})$. Exploiting such relations we obtain the following notion of qualitative unwinding.

Definiton 24 (Qualitative Unwinding). *$\mathcal{P}$ satisfies the* qualitative unwinding *if $QL(\mathcal{P}) \in \mathcal{W}(\doteqdot^{\mathcal{E}xp}, \cong^{\mathcal{E}xp}, \rightarrow)$.*

Example 14. The system of Example 8 does not satisfy the qualitative unwinding since as shown in Figure 7 $\{\alpha_1, \alpha_2\} \doteqdot^{\mathcal{E}xp} \emptyset$ and $\{\alpha_1, \alpha_2\}$ reaches other states through imposed actions, while $\emptyset$ does not reach any state.

This third instance of unwinding is an over-approximation of the previous one in the following sense.

Theorem 2. *If $L(\mathcal{P}) \in \mathcal{W}(\doteqdot^{\mathcal{E}xp}, \cong^{\mathcal{E}xp}, \rightarrow)$, then $QL(\mathcal{P}) \in \mathcal{W}(\doteqdot^{\mathcal{E}xp}, \cong^{\mathcal{E}xp}, \rightarrow)$.*

Proof. Let $[u] \xrightarrow{\iota} [u']$, with $\iota \in \mathit{Imp}$, and $[u] \doteqdot^{\mathcal{E}xp} [v]$. This means that there exists $\overline{u} \in [u]$ and $\overline{u'} \in [u']$ such that $\overline{u} \xrightarrow{\iota} \overline{u'}$ and $\overline{u} \doteqdot^{\mathcal{E}xp} u \doteqdot^{\mathcal{E}xp} v$. Hence, since $L(\mathcal{P}) \in \mathcal{W}(\doteqdot^{\mathcal{E}xp}, \cong^{\mathcal{E}xp}, \rightarrow)$ we have that $v \rightarrow v'$ and $\overline{u'} \cong^{\mathcal{E}xp} v'$. Hence, by definition of $\cong^{\mathcal{E}xp}$ over $QL(\mathcal{P})$ we have that $[u'] \cong^{\mathcal{E}xp} [v']$. □

So, since $QL(\mathcal{P})$ is usually significantly smaller than $L(\mathcal{P})$ this result can be fruitfully exploited when $L(\mathcal{P})$ is too large. We will do this in the next section on a real world case-study.

3.1. Neurospora Crassa *circadian network in Bio-PEPA*

The *Neurospora crassa* is a fungus whose circadian network has been widely investigated and almost completely brought to light (e.g., see [52]). In total darkness this organism generates spores, a process called conidiation, every 22 hours. Both light and temperature affect the rhythm of this cycle. The alternation between day and night causes spore production in the middle of the night during all the year, independently on the length of daylight. From the metabolic point of view, the conidiation period is ruled by the *white collar-1* gene and the rhythmic gene frequency *frq*. The protein associated with the *white collar-1* gene, WC-1, activates the transcription of *frq*, while the protein product of *frq* both interacts with *frq*-bound WC-1 (PW) inhibiting the expression of *frq* and positively regulates the expression of WC-1. The light promotes the binding between a flavin chromophore and WC-1 (PWL) which increases the transcription of *frq*.

Akman *et al.* presented in [46] a Bio-PEPA model of the *Neurospora crassa* circadian network accounting 22 actions and 9 species. The effects of light on the system are described by a rate change in the reaction that produces the species PWL out of PW. We call this reaction *PWtoPWL*. The authors investigated the system in two different light conditions: constant darkness (D) and light and dark 12-hour alternation (LD). We considered this model to prove the effectiveness of our framework. We focused on constant darkness (D) and constant light conditions (L) and we formally proved that $\mathit{Imp} = \{\mathit{PWtoPWL}\}$ is relevant for the system. While this result is not surprising, it is worth to notice that it was obtained by analysing exclusively the (Q)LTS's of the system and it does not rely on reaction rates.

The full LTS of the investigated Bio-PEPA model may contain up to 2^{103} nodes and, thus, it is not feasible for the analysis. However, the corresponding QLTS has at most $2^{|\mathcal{A}|}$ nodes, where $\mathcal{A}$ is the set of all the possible reactions, and we can compute it directly. We wrote a Python program to assemble the QLTS of any Bio-PEPA system avoiding the LTS construction and we applied it to the system in both D and L conditions. The program computes the set of species constraints corresponding to each node of the QLTS and reduces the reachability of the region satisfying these constraints to a set of Interger Linear Programming problems having the following form:

$$\text{Objective:} \qquad \min \sum_i n_i$$

$$\begin{pmatrix} l_0 \\ \vdots \\ l_n \end{pmatrix} - \begin{pmatrix} b_0 \\ \vdots \\ b_n \end{pmatrix} \leq \sum_i n_i \begin{pmatrix} e_{i,0} \\ \vdots \\ e_{i,n} \end{pmatrix} \leq \begin{pmatrix} u_0 \\ \vdots \\ u_n \end{pmatrix} - \begin{pmatrix} b_0 \\ \vdots \\ b_n \end{pmatrix} \tag{1}$$

where the natural variable n_i represents the number of repetitions of the action α_i, $e_{i,j}$ is the effect of the action α_i on the species s_j, and b_j and $[l_j, u_j]$ are the LTS starting value and the admissible interval for the species s_j, respectively. Whenever System (1) is not satisfiable, there is no way to reach the considered QLTS state from the starting value. On the contrary, other tests are necessary to verify that the proposed path does not cross the imposed boundaries.

By using the developed program, we built two QLTS's, QL_D and QL_L, representing the Bio-PEPA model in D and L conditions, respectively. The former has 3969 nodes and misses the *PWtoPWL* action, the latter has 11529 nodes and uses that action. We know that if the *PWtoPWL* action is not relevant for the system evolution, then, whenever a state u performs a *PWtoPWL* transition and reaches u', then each other state v such that $u \doteqdot^{\mathcal{E}xp} v$ should be able to perform an action and reaches a v' such that $u' \cong^{\mathcal{E}xp} v'$.

We projected the QL_L on *PWtoPWL* by joining all the nodes that differ exclusively for that label i.e., we merged all the pairs of nodes of QL_L, v and v', such that $v \setminus \{PWtoPWL\} = v' \setminus \{PWtoPWL\}$. The obtained QLTS, named QL'_L, has 8505 nodes and we observed that some of the nodes in QL'_L are not present in QL_D. Moreover, both QL_D and QL_L are strongly connected (i.e., all the nodes are reachable from each other) and, hence, whenever a node is reachable from the considered starting conditions, it is reachable also from any other admissible value. It follows that there exist states in QL'_L, and, as a consequence, in both QL_L and QL_{LD}, that are not reachable if we avoid the action *PWtoPWL* (i.e., the model is in D condition). Thus, the investigated system in LD conditions does not satisfy the qualitative unwinding and, by Theorem 2, does not satisfy the quasi-quantitative unwinding too. As noticed above, this means that *PWtoPWL* is relevant for the evolution of the *Neurospora crassa* model in LD conditions or in other terms *Neurospora crassa* is not robust with respect to perturbations on *PWtoPWL* action.

4. Unwinding Conditions over Hybrid Automata

Many natural systems exhibit hybrid behaviours: from one hand, they evolve in a continuous way according to a set of dynamical laws, but, from the other hand, the dynamical laws are ruled by a discrete control. Because of their double nature, these systems are called *hybrid systems*.

Hybrid systems have raised the attention of both computational and systems biology community because of three main reasons. First of all, they combine continuous models, which are traditionally used to represent timed evolutions of species concentrations, and discrete models, which can depict species interactions. Moreover, they perfectly map biological phenomena due to the interactions between continuous processes, such as gene expression, and discrete events, like the binding of a transcription factor. Finally, control theory and computer science provide established techniques and tools for the specification, the analysis, and the control of hybrid systems and computational and systems biology can benefit from them.

Hybrid automata model hybrid systems in a very natural way. In its simpler form, a hybrid automaton can be seen as a "finite-state" automaton [53] equipped of continuous variables whose values are dubbed continuous state. The discrete states are called *locations*. Each location is labelled by a dynamical law which rules the continuous state when the automaton has reached that particular location and it is also characterised by a set, called *invariant*, that define the admissible values for the continuous variables during the automaton evolution. The automaton edges are dubbed *discrete transitions* and each of them is associated to both a set of continuous values, called *activation*, and a map, called *reset*. The hybrid automaton can cross an edge only if the values of its continuous variables lay in the activation and, after a crossing, it must update the continuous state according to the reset.

Different definitions of hybrid automaton have been suggested in the literature so far [54, 55, 56, 57, 58, 59, 60]. In this paper, we rely on a definition which is close to the one given by Lynch *et al.* [11] as the unwinding framework that we are proposing fits into it in a natural way.

The definition of hybrid automaton given by Lynch *et al.* differs from the classical one because of two reasons: first of all, it omits to explicitly mention locations and the discrete transitions go from continuous states to continuous states. Nevertheless, the automata proposed by Lynch *et al.* can encode locations simply by using a new continuous variable to achieve this goal. Secondly, this definition distinguishes two kinds of continuous variables and discrete transitions. Our definition fully inherits these properties,

however, while Lynch *et al.* discriminate variables and actions in *input* ones and *output* ones, we emphasise the difference between *imposed* and *exposed* as detailed in Section 2.3.

Definiton 25 (Hybrid Automata). *A* hybrid automaton $(Imp, Exp, Q, \Theta, \mathit{Imp}, \mathcal{E}xp^{-}, D, \mathcal{T})$ *consists of:*

- *Two disjoint sets of variables: a set Imp of* imposed variables *and a set Exp of* exposed variables. $Var \stackrel{def}{=} Imp \cup Exp$;
- *A set* $Q \subseteq val(Var)$ *of states, where* $val(Var)$ *is the set of values that Var can assume;*
- *A nonempty set* $\Theta \subseteq Q$ *of start states;*
- *Two disjoint sets of actions: a set* Imp *of* imposed discrete actions *and a set* $\mathcal{E}xp^{-}$ *of* exposed discrete actions. $Act \stackrel{def}{=} \mathit{Imp} \cup \mathcal{E}xp^{-}$;
- *A set* $D \subseteq Q \times Act \times Q$ *of* discrete transitions. *The action a is* enabled *in x if there exists an x′ such that* $(x, a, x') \in D$;
- *A set* $\mathcal{T}$ *of* trajectories *for Var. Each* $\tau \in \mathcal{T}$ *is a function whose domain,* $dom(\tau)$, *is an initial subset of* $\mathbb{R}_{\geq 0}$ *and whose image set is a subset of Q, i.e.,* $\tau(t) \in Q$, *for all* $\tau \in \mathcal{T}$ *and all* $t \in dom(\tau)$. *The following axioms must hold:*

  T1 *(Prefix closure)*
  If $\tau \in \mathcal{T}$, $dom(\tau') \subseteq dom(\tau)$, *and* $\tau'(t) = \tau(t)$ *for all* $t \in dom(\tau')$, *then* $\tau' \in \mathcal{T}$;

  T2 *(Suffix closure)*
  If $\tau \in \mathcal{T}$ *and* $t' \in dom(\tau)$, *then* $\tau'(t) \stackrel{def}{=} \tau(t + t')$ *belongs to* $\mathcal{T}$;

  T3 *(Concatenation closure)*
  Let $S = \{\tau_0, \tau_1, \tau_2, \ldots\}$ *be a subset of* $\mathcal{T}$ *such that* $dom(\tau_i)$ *has the form* $[0, t_{max,i}]$ *and* $\tau_i(t_{max,i}) = \tau_{i+1}(0)$ *for all* $i + 1 < |S|$. *Then the trajectory:*

$$(\tau_0 \frown \tau_1 \frown \tau_2 \frown \ldots)(t) \stackrel{def}{=} \begin{cases} \tau_0(t) & \text{if } t \in dom(\tau_0) \\ (\tau_1 \frown \tau_2 \frown \ldots)(t - t_{max,0}) & \text{otherwise} \end{cases}$$

  belongs to $\mathcal{T}$.

The trajectories can be given in implicit form, for instance, as a differential system: the set $\mathcal{T}$ contains all the solutions of the provided differential system. In such cases, the computation of the trajectories themselves is not always trivial and may be not even computable.

Alongside with imposed and exposed variables, imposed and exposed actions, a set of initial continuous state Θ, and an invariant set Q, above definition of hybrid automata requires a set of discrete transitions, labelled by actions, that map continuous values in continuous values. Moreover, it specifies the properties required to the automaton continuous laws, called *trajectories*, through the axioms **T1**, **T2**, and **T3**. Axioms **T1** and **T2** ensure that any trajectory τ can be broken into two trajectories whose overall behaviour is equivalent to τ itself. On the contrary, axiom **T3** guarantees that any sequence of trajectories, whose boundaries are pairwise the same, can be composed in a trajectory that is equivalent to the overall behaviour of all the trajectories.

The semantics of hybrid automata are usually given in terms of LTS by associating a hybrid automaton with an infinite LTS whose nodes are the states of the automaton and whose edges have the form $\xrightarrow{t}_C$ or $\xrightarrow{e}_D$. As the *continuous transition relation* $\xrightarrow{t}_C$ concerns, $q \xrightarrow{t}_C q'$ holds if and only if there exists a $\tau \in \mathcal{T}$ such that $dom(\tau) = [0, t_{max}]$, $\tau(0) = q$ and $\tau(t_{max}) = q'$. Regarding the *discrete transition relation* $\xrightarrow{t}_D$, $q \xrightarrow{a}_D q'$ holds if and only if $(q, a, q') \in D$.

Definiton 26 (Hybrid Automata - Semantics). *Given a hybrid automaton* $H = (Imp, Exp, Q, \Theta, Imp, \mathcal{E}xp^-, D, \mathcal{T})$, *the LTS associated with H is the tuple* $L(H) = (Q, \Theta, Imp \cup \mathcal{E}xp, R)$, *where* $\mathcal{E}xp = \mathcal{E}xp^- \cup \mathbb{R}_{\geq 0}$ *and* $R \stackrel{def}{=} \{(q, \alpha, q') \mid q \stackrel{\alpha}{\rightarrow}_C q' \wedge \alpha \in \mathbb{R}_{\geq 0}$ *or* $q \stackrel{\alpha}{\rightarrow}_D q' \wedge \alpha \in Imp \cup \mathcal{E}xp^-\}$.

In the above definition we consider all the actions associated with the continuous transitions, i.e., the positive real numbers labelling the transition relation $\rightarrow_C$, as exposed actions. This is an arbitrary choice. Intuitively we interpret all the continuous evolutions as internal transitions not influenced by the environment. The imposed interactions with the environment are represented only through some discrete transition labels. For those who are more familiar with the definition of hybrid automata given in [54, 55], this corresponds to say that the imposed interactions cause a change of location in the automaton and hence a possible change in the differential laws regulating the continuous evolution.

Example 15 (A Simple Thermostat Model). Let us model a simple thermostat by using a hybrid automaton. The discrete variable *mode* represents the state of the heater (i.e., $mode = 1$ means "heater on" and $mode = 0$ "heater off"), while the variable x_T is associated with the temperature. Whenever the temperature reaches 15 °C, the thermostat activates the heater (exposed action *switchOn*), while, if the temperature rises up to 20 °C, the heater is turned off (exposed action *switchOff*). The users can switch on and off the heater independently from the thermostat status by using the imposed actions *forceOn* and *forceOff*, respectively. The two constants k_r and k_h are the dispersion and heating coefficients, respectively, while the variable X represents the room temperature. Initially, the heater is switched off and the temperature is 17 °C.

Meaning	**Actions**
Turn off the heater	*switchOff*
Turn on the heater	*switchOn*
Force off the heater	*forceOff*
Force on the heater	*forceOn*

(a) Actions

$Imp \setminus \mathcal{E}xp$	**Preconditions**	**Action**	**Effects**
$\mathcal{E}xp$	$(mode = 1) \wedge (x_T \geq 20)$	*switchOff*	$mode \leftarrow 0$
$\mathcal{E}xp$	$(mode = 0) \wedge (x_T \leq 15)$	*switchOn*	$mode \leftarrow 1$
Imp	$mode = 1$	*forceOff*	$mode \leftarrow 0$
Imp	$mode = 0$	*forceOn*	$mode \leftarrow 1$

(b) Transitions

Trajectories
$\dot{x_T} = k_h * mode - k_r * x_T$
$\dot{mode} = 0$

(c) Trajectories

$Imp \setminus \mathcal{E}xp$	**Meaning**	**Variables**
$\mathcal{E}xp$	Temperature	$x_T : \mathbb{R} \leftarrow 17$,
$\mathcal{E}xp$	Heater state	$mode : \{0, 1\} \leftarrow 0$

(d) Variables

Table 3: A formal specification of the hybrid automaton representing a simple thermostat.

Example 16 (Delta-Notch). Delta and Notch are transmembrane proteins [61]. They are at the basis of the differentiation and signal mechanisms between neighbouring cells. Delta production in a cell is triggered by low Notch concentrations in the cell itself, while Notch production is due to high Delta levels in neighbouring cells. Delta concentration is directly connected to differentiation. A a matter of fact, high Delta concentration produces differentiated cells, while low Delta levels cause undifferentiated ones. The Delta-Notch mechanism is the core of biological pattern formation and because of that it has garnered the attention of the scientific community. A mathematical model for Delta-Notch signalling was presented in [61] and described in terms of hybrid automata in [62, 63].

We can model the one cell system in our framework by using a hybrid automaton having five continuous variables, p_D, p_N, x_D, x_N, and x'_D (see Table 4). The variables p_D and p_N represent Delta and Notch production flags for the considered cell: $p_D = 1$ ($p_N = 1$) means that Delta (Notch) is in production in the cell, while $p_D = 0$ ($p_N = 0$) means Delta (Notch) is not in production. The variables x_D and x_N characterise Delta and Notch concentrations in the considered cell, respectively. Finally, x'_D quantifies the level of Delta in the neighbouring cells.

The hybrid automaton has four discrete transitions. Each of them models a status change in the production of either Delta and Notch. Whenever Delta is not in production (i.e., $p_D = 0$) and the level of

Notch in the modelled cell x_N decreases under h_N, the production of Delta is triggered (i.e., $p_D \leftarrow 1$). At the same time, if the level of Delta in neighbouring cells x'_D exceeds a bound h_D and Notch is not in production, then a transition may activates the expression of Notch (i.e., $p_N \leftarrow 1$).

The trajectories of the hybrid automaton depends on four parameters: R_D, R_N, δ_D, and δ_N that model the production rates of Delta and Notch and the degradation rates of Delta and Notch, respectively. Both Delta and Notch are constantly degraded during system evolution ($-\delta_D * x_D$ and $-\delta_N * x_N$ terms of the trajectories). However, if Delta production has been triggered (i.e., $p_D = 1$), Delta level rises at rate R_D ($p_D * R_D$ term). Analogously, Notch increases at rate R_N if and only if it is in production (i.e., $p_N = 1$).

The values of the parameters R_D, R_N, δ_D, δ_N, h_D, and h_N were estimated, for instance, in [63].

Meaning	**Actions**
Turn off Delta	*DeltaOff*
Turn on Delta	*DeltaOn*
Turn off Notch	*NotchOff*
Turn on Notch	*NotchOn*

(a) Actions

$Imp \setminus Exp$	**Preconditions**	**Action**	**Effects**
Exp	$p_D = 0 \wedge x_N < h_N$	*DeltaOn*	$p_D \leftarrow 1$
Exp	$p_D = 1 \wedge x_N \geq h_N$	*DeltaOff*	$p_D \leftarrow 0$
Exp	$p_N = 0 \wedge x'_D > h_D$	*NotchOn*	$p_N \leftarrow 1$
Exp	$p_D = 1 \wedge x'_D \leq h_D$	*NotchOff*	$p_N \leftarrow 0$

(b) Transitions

Trajectories
$\dot{x_D} = p_D * R_D - \lambda_D * x_D$
$\dot{x_N} = p_N * R_N - \lambda_N * x_N$
$\dot{p_D} = 0$
$\dot{p_N} = 0$

(c) Trajectories

$Imp \setminus Exp$	**Meaning**	**Variables**
Exp	Delta in production	$p_D : \{0, 1\} \leftarrow 0,$
Exp	Notch in production	$p_N : \{0, 1\} \leftarrow 0,$
Exp	Delta	$x_D : \mathbb{R} \leftarrow 1,$
Exp	Notch	$x_N : \mathbb{R} \leftarrow 0,$
Imp	Neighbouring Delta	$x'_D : \mathbb{R} \leftarrow 1$

(d) Variables

Table 4: A formal specification of the hybrid automaton representing the Delta-Notch signalling mechanism.

In the context of hybrid automata, (bi)simulation reductions [64, 65], series of abstractions [66], piecewise linear approximations [67, 68] have been proposed in the literature to abstract the infinite LTS representing the semantics of a hybrid automaton into finite ones.

In the remaining part of this section, we introduce an unwinding condition over hybrid automata. Let us observe that the unwinding conditions defined over LTS's generated from Bio-PEPA systems can be used also on LTS's generated from hybrid automata. As a matter of fact, one of the advantages of unwinding conditions is that they are defined on LTS and hence they do not depend on the modelling language from which the LTS has been inferred. However, we prefer to introduce here a further unwinding condition to give to the reader an idea of the flexibility of the framework. In order to achieve this goal, we first need to define a new equivalence relation (i.e., $\asymp^{Exp}$) and two transition relations (i.e., $\rightarrow^{Exp}$ and $\rightarrow^{>t}$) over continuous states.

As done in the quantitative unwinding in the previous section, we consider two nodes of the LTS to be "equal" if the have the same values on the exposed variables. We consider them to be "equivalent" if using only exposed transitions they generate the same sequences of exposed variables values. Finally, the transition relation $\dashrightarrow$ is a generic exposed transition.

Definiton 27 ($\doteq^{Exp}$, $\asymp^{Exp}$, $\rightarrow^{Exp}$ and $\rightarrow^{>t}$). *Let H be a hybrid automaton and $L(H) = (Q, \Theta, Imp \cup Exp, R)$ be the LTS associated with H. The relation $\doteq^{Exp} \subseteq Q \times Q$ is defined as follows: $u \doteq^{Exp} v$ if and only if the exposed variables have the same values on u and v. A* strong bi-exposed bisimulation over $L(H)$ *is a symmetric binary relation R over $L(H)$ such that for each $(u, v) \in R$:*

- $u \doteq^{Exp} v$;
- *if $u \xrightarrow{e} u'$ with $e \in Exp$, then $v \xrightarrow{e} v'$ and $(u', v') \in R$.*

Two states u and v are strongly bi-exposed bisimilar, *denoted by $u \asymp^{Exp} v$ if there exists a strong exposed bisimulation R such that $(u, v) \in R$.*

The relations $\rightarrow^{\mathcal{E}xp} \subseteq Q \times Q$ and $\rightarrow^{>t} \subseteq Q \times Q$, where $t \in \mathbb{R}_{\geq 0}$, are defined as follows: $u \rightarrow^{\mathcal{E}xp} v$ if and only if there exists $e \in \mathcal{E}xp$ such that $u \xrightarrow{e} v$ and $u \rightarrow^{>t} v$ if and only if there exists $t' > t$ such that $u \xrightarrow{t'} v$.

Notice that $\doteq^{Exp}$ is the same relation introduced on LTS's of Bio-PEPA systems. We are now ready to introduce two unwinding conditions. The first one is called bi-exposed since exposed actions are involved in the definition of both the equivalence relation $\asymp^{\mathcal{E}xp}$ and the transition relation $\rightarrow^{\mathcal{E}xp}$. Once the system has moved from u to u' through an imposed action if $u \doteq^{Exp} v$, then v has to reach v' through an exposed action and u' and v' are compared only on the basis of exposed values and exposed transitions. This unwinding condition abstracts transitions on exposed actions by reducing all of them to $\rightarrow^{\mathcal{E}xp}$ and can be used to highlight quantitative differences between states or systems on the reachable values of exposed variables.

The second unwinding condition is called delayed unwinding since the transition relation $\dashrightarrow$ is instantiated as $\rightarrow^{>t}$ which represents a delay of time greater than t. In this second instance u' and v' are compared on the basis of all transitions through the relation $\approx^{Exp}$ introduced in the previous section. This condition requires that any transitions spend a time greater than t causing a delay. It can reveal behaviours that are equivalent aside from delays.

Definiton 28 (Bi-Exposed Unwinding and Delayed Unwinding). *H satisfies the bi-exposed unwinding if $L(H) \in \mathcal{W}(\doteq^{Exp}, \asymp^{\mathcal{E}xp}, \rightarrow^{\mathcal{E}xp})$. H satisfies the t-delayed unwinding if $L(H) \in \mathcal{W}(\doteq^{Exp}, \approx^{\mathcal{E}xp}, \rightarrow^{>t})$.*

In the next section we apply bi-exposed unwinding in the analysis of influenza models, proving that, as one could expect, influenza is not robust with respect to antiviral and interferons treatments. Moreover, t-delayed unwinding allows to show that also if we refer to a single type of treatment (e.g., antiviral) the result strongly depends on the time at which the treatment is imposed.

4.1. Influenza Kinetics Analysis through hybrid automata

Influenza is an infectious disease caused by a family of RNA viruses known as influenza viruses. Its symptoms include fever, weakness, and coughing and, in the most acute form, it can bring a severe threat to the respiratory system. Worldwide, 250,000 to 500,000 deaths per years are ascribed to the complications of the seasonal influenza virus [69] and the infamous *Spanish flu* of the 1918 infected 500 million people leading to the death of 20 to 100 million of them [70, 71].

Understanding the dynamics of infection plays a crucial role in avoiding or, at least, controlling possible influenza pandemics. Many models have been suggested so far to achieve this goal [72, 73, 74, 75, 76, 77].

Handel *et al.* deal with the effects of the most effective drugs against influenza, the *Neuraminidase inhibitors* (NI), and take into account the rise of virus strains resistant to this antiviral [74]. The model distinguishes the load of viruses that are NI-sensitive (V_s) from that of viruses that are NI-resistant (V_r). Uninfected cells (U) are infected by either NI-resistant virus or NI-sensitive virus at a rate proportional to the correspondent virus load and became NI-resistant infected (I_r) or NI-sensitive infected (I_s), respectively. Both NI-resistant and NI-sensitive infected cells increase virus load of the respective strain. Moreover, due to natural mutations, a fraction of the viruses released by NI-sensitive infected cells belongs to the NI-resistant strain. Whatever is the strain of the viruses produced by NI-sensitive infected cells, NI represses their release at a rate that is proportional to the efficiency of the antiviral itself. A natural immune response (X) restrains the increase in viral load in all the strains too. As it occurs also to the virus strains, both the kinds of infected cells share the same decay rate.

On the contrary, Saenz *et al.* consider the interactions between viral agents and immune system and describe the antiviral response modulated by the type I interferon (IFN-α/β) which is triggered by infection [75]. In their model, uninfected cells (U) are infected at a rate proportional to the virus load (V). Newly infected cells spend some time in an eclipse phase (E_1) and then they move to a state (I), having a limited span life, in which they increase the virus load. Infective cells produce IFN (F) which is able to bring uninfected cells into a prerefractory state (W) and, possibly, in a refractory state (R) at a rate proportional to

Meaning	Variables
Uninfected cells	$U : \mathbb{R}_{\geq 0} \leftarrow 4.0 * 10^8$
Cells infected by virus sensitive to NI in eclipse phase	$E_{1,s} : \mathbb{R}_{\geq 0} \leftarrow 0.0$
Cells infected by virus resistant to NI in eclipse phase	$E_{1,r} : \mathbb{R}_{\geq 0} \leftarrow 0.0$
Prerefractory cells infected by virus sensitive to NI in eclipse phase	$E_{2,s} : \mathbb{R}_{\geq 0} \leftarrow 0.0$
Prerefractory cells infected by virus resistant to NI in eclipse phase	$E_{2,r} : \mathbb{R}_{\geq 0} \leftarrow 0.0$
Cells in prerefractory state	$W : \mathbb{R}_{\geq 0} \leftarrow 0.0$
Refractory cells	$R : \mathbb{R}_{\geq 0} \leftarrow 0.0$
Cells infected by virus sensitive to NI	$I_s : \mathbb{R}_{\geq 0} \leftarrow 0.0$
Cells infected by virus resistant to NI	$I_r : \mathbb{R}_{\geq 0} \leftarrow 0.0$
Virus sensitive to NI	$V_s : \mathbb{R}_{\geq 0} \leftarrow 7.7 * 10^{-3}$
Virus resistant to NI	$V_r : \mathbb{R}_{\geq 0} \leftarrow 0.0$
Interferons (IFN-α/β)	$F : \mathbb{R}_{\geq 0} \leftarrow 0.0$
Immune response	$X : \mathbb{R}_{\geq 0} \leftarrow 3.4 * 10^{-1}$
Elapsed time	$T : \mathbb{R}_{\geq 0} \leftarrow 0.0$
Antiviral injection mode ($A = 1$ injecting NI, $A = 0$ not injecting NI)	$A : \{0, 1\} \leftarrow 0$
Model mode ($M = 0$ treatment to be selected, $M = 1$ NI-based treatment, $M = 2$ IFN-based treatment, $M = 3$ treatment concluded)	$M : [0, 3] \leftarrow 0$

Table 5: Variables of the hybrid automaton modelling the influenza kinetics.

F itself. Whenever cells in the prerefractory state are infected, they move to an eclipse phase (E_2), in which they release IFN, and, eventually, become infective (I).

IFNs have been long used as a treatment for various autoimmune, viral, and tumour diseases [78, 79, 80, 81, 82, 83]. So, one may wonder if IFNs can be used as antiviral drugs and, in particular, whether IFNs and NIs are equivalent or not with respect to the virus load, i.e., if IFNs can be used in place of NIs, and vice versa, in influenza treatment. Notice that, whatever the answer is, the NI-based therapies will be still preferable to the IFN-based ones in normal condition since the latter exhibit many serious side effects in humans [84, 85, 86] and they are less cost-effective than the former. However, above questions maintain some relevance in the case of a pandemic produced by a viral strain that is resistant to the antivirals.

We developed a model that takes into account the effects of both IFN and NI on the virus load. As done in [74], we admit the existence of a NI-resistant strain and, analogously to [75], we represent the antiviral response due to IFN. The trajectories, the actions, and the transitions of our model are reported in Table 7. The values of their parameters are dependent on the virus strain and on the host species; we focused on human hosts infected with influenza A/Texas/91 (H1N1) and we fit our model on the data produced by the IR kinetic model described in [74] by minimising the cost function:

$$SSE \stackrel{\text{def}}{=} \sum_i \left(\frac{\log V_s(i) - \log \hat{V}_s(i)}{\log \max_i \hat{V}_s(i)} \right)^2 + \sum_i \left(\frac{\log V_r(i) - \log \hat{V}_r(i)}{\log \max_i \hat{V}_r(i)} \right)^2 + (D - \hat{D})^2, \tag{2}$$

where V_s, V_r, and D are the number of NI-sensitive virus, the number of NI-resistant virus, the total dead cells estimated by our model, respectively, while $\hat{V}_s$, $\hat{V}_r$, and $\hat{D}$ are the same quantities evaluated by the IR model suggested in [74]. In order to achieve this goal we used the open-source software Octave [87] and, in particular, its built-in function `fminunc`. The resulting values of the parameters and their sources are reported in Table 6.

Notice that the IR model does not take into account IFN and, thus, the value estimated for the parameters that is directly connected with IFN, i.e., q and ψ, may need some scaling to match the real kinetics of interferons in humans infected by H1N1. Nevertheless, our model exhibits the same IFN peek as the one proposed by Saenz *et al.* in the case of an A/equine/Kildare/89 (H3N8) infection [75].

Symbol	Meaning	Value	Source
k_i	Eclipse phase period	2 Days^{-1}	[88, 75]
a	Prefactory period	4 Days^{-1}	[89, 75]
μ	Mutation rate	10^{-5}	[74]
σ	Fitness cost of resistance	0.1	[74]
n	IFN-reduced production	1	[75]
m	IFN-reduced infectivity	1	[75]
β	Virus infectivity	$1.18368 * 10^{-1}$	Fitted
p	Virus production per cell	$5.96623 * 10^{-4}$	Fitted
q	IFN induction per cell	$5.18653 * 10^{-11}$	Fitted
ψ	IFN efficiency	3.68109	Fitted
d_I	Infected cell death rate	$1.23661 * 10^{-1}$	Fitted
d_V	Virus death rate	$0.80710 * 10^{-1}$	Fitted
d_F	IFN clearance rate	1.80413	Fitted
r	Immune response growth rate	1.14835	Fitted
ϵ_{NI}	Antiviral efficacy	0.97647	Fitted
f	IFN released per injection	6	Arbitrary

Table 6: Trajectory parameters of the hybrid automaton modelling the influenza kinetics.

Our hybrid model chooses one treatment between the IFN-based and the NI-based one by using either the action *treatIFN* or the action *treatNI*, respectively. In the former case, the effect of the antiviral is assumed to be constant along all the treatment; in the latter case, the host is injected with an arbitrary dose f of IFN every t_i days from the beginning of the treatment until the end of it. Figure 8 represents the evolution of the two variables V_s (plain line) and V_r (dotted line) under both the considered treatments.

First of all, we may want to decide whether the effectiveness of two treatments depends or not on the timing. We notice that the LTS associated with our hybrid model does not satisfy the 0.5-delayed unwinding $\mathcal{W}(\doteq^{Exp}, \approx^{Exp}, \rightarrow^{>0.5})$. As a matter of fact, the peek of the V_r load observable by starting the NI-based treatment at the time of the infection can be obtained neither with different timing nor by using the IFN-based treatment (see Figure 8). This proves that the effectiveness of the treatments are time-dependent.

Another question that deserves attention is whether IFN and NI are equivalent or not with respect to the virus level. We consider *treatNI* being an imposed action and verify that the LTS associated with the proposed influenza model does not satisfies the bi-exposed unwinding $\mathcal{W}(\doteq^{Exp}, \asymp^{Exp}, \rightarrow^{Exp})$, i.e., it is not possible to obtain the same virus load produced by the NI-treatment, whenever it has began, in the same time.

More details about this example are given in [90].

5. Conclusions

We proposed a framework for the analysis of biological systems based on unwinding conditions. The framework can be instantiated on all modelling languages which rely on operational semantics defined in terms of LTS's.

The finer and richer is the operational semantics that defines the LTS, the wider is the range of unwinding conditions that one can define.

Within such a large variety we can find both fine unwinding conditions characterising very restrictive notions of robustness and coarser ones allowing for more "fluctuations" around standard behaviours.

Not surprisingly, fine unwinding conditions are usually computationally harder to test (this is not always the case, e.g., see bisimulation vs trace equivalence), but more amenable coarser ones can be exploited to test them through some kind of necessary conditions.

The unwinding conditions over the process algebra Bio-PEPA, introduced in Section 3, fall in the schema described above: quantitative unwinding is finer than the qualitative one, but computationally

Trajectories
$\dot{U} = -\beta(V_r + V_s)U - \phi FU$
$\dot{E}_{1,j} = \beta V_j U - k_1 E_{1,j}$
$\dot{E}_{2,j} = m\beta V_j W - k_2 E_{2,j}$
$\dot{W} = \phi FU - m\beta(V_r + V_s)W - aW$
$\dot{R} = aW$
$\dot{I}_j = k_1 E_{1,j} + k_2 E_{2,j} - d_I I_j$
$\dot{V}_s = (1 - A * \epsilon_{NI})(1 - \mu)pI_s - d_V V_s - XV_s$
$\dot{V}_r = (1 - A * \epsilon_{NI})\mu pI_s + (1 - \sigma)pI_r - d_V V_r - XV_r$
$\dot{F} = nq(E_{2,r} + E_{2,s}) + q(I_r + I_s) - d_F F$
$\dot{X} = rX$
$\dot{T} = 1$
$\dot{A} = 0$
$\dot{M} = 0$

(a) Trajectories

Meaning	Actions
Begin NI-based treatment	*treatNI*
Begin IFN-based treatment	*treatIFN*
End the treatment	*conclude*
Inject IFN	*injectIFN*

(b) Actions

Preconditions	Action	Effects
$M = 0$	*treatNI*	$A \leftarrow 1 \wedge M \leftarrow 1$
$M = 0$	*treatIFN*	$M \leftarrow 2$
$M = 1$	*conclude*	$A \leftarrow 0 \wedge M \leftarrow 3$
$M = 2$	*conclude*	$M \leftarrow 3$
$M = 2 \wedge t_i \mid T$	*injectIFN*	$F \leftarrow F + f$

(c) Transitions

Table 7: Trajectories, actions, and transitions of the hybrid automaton modelling the influenza kinetics.

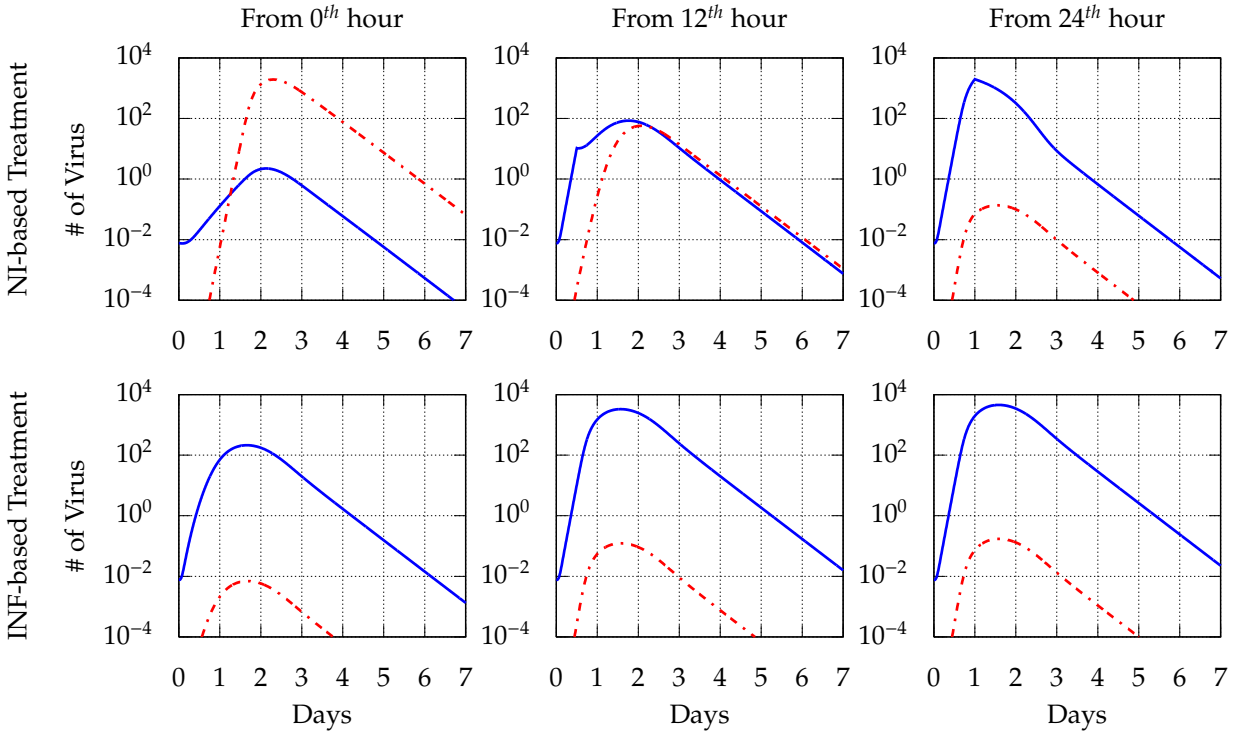

Figure 8: A comparison between the kinetics of the proposed model under NI-based treatment (first row) and IFN-based treatment (second row). The plain and the dotted lines represent the number of NI-sensible virus (V_s) and the number of NI-resistant virus (V_r), respectively. The first, the second, and the third columns depict the evolutions when the treatment, whatever it is, begins 0, 12, and 24 hours after the infection, respectively. All the treatments continue until the 7$^{\text{th}}$ day. Early inoculation of NI leads to a proliferation of the NI-resistant strain, while interferons can be used in early stage of the infection to control the proliferation of both the virus strains. On the contrary, NI maintains effectiveness throughout the course of the disease as opposed to IFN which seems to be useless whenever the treatment begins after the peek in the number of virus, i.e., about 1.5 day after the infection.

more demanding.

While operational semantics and behavioural equivalences have been largely used in the literature to investigate computational languages such as process algebras, hybrid automata mix computational and mathematical domains.

In Section 4, we showed that, despite of their double nature, unwinding conditions can be defined and fruitfully exploited also over hybrid automata. In this case, we simply introduced two orthogonal notions of unwinding and proved their effectiveness in the analysis of the influenza case of study.

As observed in [91] robustness in biology is strongly related with redundancy. This is exactly in the spirit of our unwinding framework, where for each imposed transition we have to find an alternative path leading to an equivalent situation.

From the modelling point of view, unwinding conditions have the advantage of clarifying which questions one has to answer in order to choose the right formalisation: (1) which are the exposed (observable) actions/species and which are the imposed ones? (2) which are the relevant exposed observational equivalences? (3) which are the alternative transitions ($\dashrightarrow$)?

Many aspects of unwinding conditions on systems biology still have to be investigated.

In this paper, we proposed the use of "approximated" behavioural equivalences (e.g., see [33, 10]) to obtain flexible instantiations. As future work we intend to explore also the use of downgrading like techniques (see [25]) to this aim.

The relationships with some communication notions in the context of hybrid automata has been considered in [90]. A deeper investigation of this aspect on different modelling languages is still lacking.

References

[1] J. Fisher, T. A. Henzinger, Executable Cell Biology, Nature Biotechnology 25 (2007) 1239–1249.
[2] E. M. Clarke, O. Grumberg, D. A. Peled, Model Checking, MIT Press, 1999.
[3] P. Y. A. Ryan, S. A. Schneider, Process algebra and non-interference, Journal of Computer Security 9 (1/2) (2001) 75–103.
[4] A. Bossi, R. Focardi, C. Piazza, S. Rossi, Bisimulation and unwinding for verifying possibilistic security properties, in: Proc. of 4th Int. Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI 2003), Vol. 2575 of LNCS, Springer, 2003, pp. 223–237.
[5] C. Piazza, M. Antoniotti, V. Mysore, A. Policriti, F. Winkler, B. Mishra, Algorithmic algebraic model checking I: Challenges from systems biology, in: Computer Aided Verification, 17th Int. Conference, CAV 2005, Edinburgh, Scotland, UK, July 6-10, 2005, Proceedings, Vol. 3576 of LNCS, Springer, 2005, pp. 5–19.
[6] M. Kwiatkowska, G. Norman, D. Parker, Using probabilistic model checking in systems biology, ACM SIGMETRICS Performance Evaluation Review 35 (4) (2008) 14–21.
[7] G. Bernot, J.-P. Comet, On the use of temporal formal logic to model gene regulatory networks, in: Computational Intelligence Methods for Bioinformatics and Biostatistics, Springer, 2010, pp. 112–138.
[8] A. Naldi, E. Remy, D. Thieffry, C. Chaouiya, Dynamically consistent reduction of logical regulatory graphs, Theoretical Computer Science 412 (21) (2011) 2207–2218.
[9] J. Anderson, Y.-C. Chang, A. Papachristodoulou, Model decomposition and reduction tools for large-scale networks in systems biology, Automatica 47 (6) (2011) 1165 – 1174, special Issue on Systems Biology.
[10] V. Galpin, J. Hillston, A semantic equivalence for Bio-PEPA based on discretisation of continuous values, Theor. Comput. Sci. 412 (21) (2011) 2142–2161.
[11] N. A. Lynch, R. Segala, F. W. Vaandrager, Hybrid I/O automata, Information and Computation 185 (1) (2003) 105–157.
[12] A. R. Meyer, L. J. Stockmeyer, The equivalence problem for regular expressions with squaring requires exponential space, in: Switching and Automata Theory, 1972., IEEE Conference Record of 13th Annual Symposium on, IEEE, 1972, pp. 125–129.
[13] P. C. Kanellakis, S. A. Smolka, Ccs expressions, finite state processes, and three problems of equivalence, Information and Computation 86 (1) (1990) 43–68.
[14] R. Paige, R. E. Tarjan, Three partition refinement algorithms, SIAM Journal on Computing 16 (6) (1987) 973–989.
[15] Baarir, S. and Beccuti, M. and Dutheillet, C. and Franceschinis, G. and Haddad, S., Lumping Partially Symmetrical Stochastic Models, Performance Evaluation 68 (1) (2011) 21–44.
[16] H. Kitano, Towards a theory of biological robustness, Molecular Systems Biology 3 (137).
[17] H. Kitano, Biological robustness, Nature Reviews Genetics 5 (11) (2004) 826–837.
[18] H. Kitano, A robustness-based approach to systems-oriented drug design, Nature Reviews Drug Discovery 5 (2007) 202–210.
[19] J. A. Goguen, J. Meseguer, Security policies and security models, in: IEEE Symposium on Security and Privacy, IEEE Computer Society Press, 1982, pp. 11–20.
[20] A. Sabelfeld, A. C. Myers, Language-based information-flow security, IEEE Journal on Selected Areas in Communications 21 (1) (2003) 5–19.
[21] R. Focardi, R. Gorrieri, A taxonomy of security properties for process algebras, Journal of Computer Security 3 (1) (1995) 5–34.

[22] P. Y. A. Ryan, A CSP formulation of non-interference and unwinding, Cipher: IEEE Comput. Soc. Tech. Comm. Newsl. Secur. Priv. (1991) 19–30.
[23] A. Aldini, M. Bravetti, R. Gorrieri, A process-algebraic approach for the analysis of probabilistic noninterference, Journal of Computer Security 12 (2) (2004) 191–245.
[24] C. Bodei, P. Degano, F. Nielson, H. Nielson, Static analysis for secrecy and non-interference in networks of processes, in: Parallel Computing Technologies, Vol. 2127 of LNCS, Springer, 2001, pp. 27–41.
[25] A. Bossi, C. Piazza, S. Rossi, Compositional information flow security for concurrent programs, Journal of Computer Security 15 (3) (2007) 373–416.
[26] M. Komorowski, M. J. Costa, D. A. Rand, M. P. Stumpf, Sensitivity, robustness, and identifiability in stochastic chemical kinetics models., Proc. of the National Academy of Sciences of the United States of America 108 (21) (2011) 8645–8650.
[27] G. E. Fainekos, G. J. Pappas, Robustness of temporal logic specifications for continuous-time signals, Theor. Comput. Sci. 410 (42) (2009) 4262–4291.
[28] A. Donzé, O. Maler, Robust satisfaction of temporal logic over real-valued signals, in: Formal Modeling and Analysis of Timed Systems, Vol. 6246 of LNCS, Springer, 2010, pp. 92–106.
[29] A. Rizk, G. Batt, F. Fages, S. Soliman, Continuous valuations of temporal logic specifications with applications to parameter optimization and robustness measures, tcs 412 (26) (2011) 2827–2839.
[30] A. Donzé, T. Ferrère, O. Maler, Efficient robust monitoring for STL, in: Proc. of 25th Int. Conference on Computer Aided Verification (CAV 2013), Vol. 8044 of LNCS, Springer, 2013, pp. 264–279.
[31] L. Brim, T. Vejpustek, D. Safránek, J. Fabriková, Robustness analysis for value-freezing signal temporal logic, in: Proc. of 2nd Int. Workshop on Hybrid Systems and Biology (HSB 2013), Vol. 125 of EPTCS, 2013, pp. 20–36.
[32] E. Bartocci, L. Bortolussi, L. Nenzi, G. Sanguinetti, On the robustness of temporal properties for stochastic models, in: Proc. of 2nd Int. Workshop on Hybrid Systems and Biology (HSB 2013), Vol. 125 of EPTCS, 2013, pp. 3–19.
[33] F. van Breugel, J. Worrell, Approximating and computing behavioural distances in probabilistic transition systems, Theor. Comput. Sci. 360 (1–3) (2006) 373 – 385.
[34] M. Tracol, J. Desharnais, A. Zhioua, Computing distances between probabilistic automata, in: Proc. of the 9th Workshop on Quantitative Aspects of Programming Languages (QAPL 2011), Vol. 52 of EPTCS, 2011, pp. 148–162.
[35] J. Sproston, A. Troina, Simulation and bisimulation for probabilistic timed automata, in: Proc. of 8th Int. Conference on Formal Modeling and Analysis of Timed Systems (FORMATS 2010), Vol. 6246 of LNCS, Springer, 2010, pp. 213–227.
[36] J. Feret, T. A. Henzinger, H. Koeppl, T. Petrov, Lumpability abstractions of rule-based systems, Theor. Comput. Sci. 431 (2012) 137–164.
[37] A. Sharma, J.-P. Katoen, Weighted lumpability on markov chains, in: Perspectives of Systems Informatics, Vol. 7162 of LNCS, Springer, 2012, pp. 322–339.
[38] D. Gebler, S. Tini, Compositionality of approximate bisimulation for probabilistic systems, in: Proc. Combined 20th Int. Workshop on Expressiveness in Concurrency and 10th Workshop on Structural Operational Semantics (EXPRESS/SOS 2013), Vol. 120 of EPTCS, 2013, pp. 32–46.
[39] G. Bacci, G. Bacci, K. G. Larsen, R. Mardare, On-the-fly exact computation of bisimilarity distances, in: TACAS, Vol. 7795 of LNCS, Springer, 2013, pp. 1–15.
[40] C. Priami, P. Quaglia, Beta binders for biological interactions, in: Proc. of the 20th Int. Conference on Computational Methods in Systems Biology (CMSB 2004), CMSB'04, Springer, 2005, pp. 20–33.
[41] A. Regev, E. M. Panina, W. Silverman, L. Cardelli, E. Shapiro, Bioambients: an abstraction for biological compartments, Theor. Comput. Sci. 325 (1) (2004) 141 – 167.
[42] V. Danos, C. Laneve, Formal molecular biology, Theor. Comput. Sci. 325 (1) (2004) 69 – 110.
[43] L. Cardelli, Brane calculi, in: V. Danos, V. Schachter (Eds.), Computational Methods in Systems Biology, Vol. 3082 of LNCS, Springer, 2005, pp. 257–278.
[44] L. Cardelli, Bitonal membrane systems: Interactions of biological membranes, Theor. Comput. Sci. 404 (1–2) (2008) 5 – 18.
[45] V. Galpin, J. Hillston, Equivalence and discretisation in bio-pepa, in: Proc. of the 7th Int. Conference Computational Methods in Systems Biology (CMSB 2009), LNCS, Springer, 2009, pp. 189–204.
[46] O. E. Akman, F. Ciocchetta, A. Degasperi, M. L. Guerriero, Modelling biological clocks with Bio-PEPA: Stochasticity and robustness for the *neurospora crassa* circadian network, in: Proc. of the 7th Int. Conference Computational Methods in Systems Biology (CMSB 2009), Vol. 5688 of LNCS, Springer, 2009, pp. 52–67.
[47] F. Ciocchetta, A. Duguid, M. L. Guerriero, A compartmental model of the camp/pka/mapk pathway in bio-pepa, in: Proc. 3rd Workshop on Membrane Computing and Biologically Inspired Process Calculi (MeCBIC 2009), Vol. 11 of EPTCS, 2009, pp. 71–90.
[48] F. Ciocchetta, M. L. Guerriero, J. Hillston, Investigating modularity in the analysis of process algebra models of biochemical systems, in: Proc. 3rd Workshop From Biology To Concurrency and back (FBTC 2010), Vol. 19 of EPTCS, 2010, pp. 55–69.
[49] O. E. Akman, M. L. Guerriero, L. Loewe, C. Troein, Complementary approaches to understanding the plant circadian clock, in: Proc. 3rd Workshop From Biology To Concurrency and back (FBTC 2010), Vol. 19 of EPTCS, 2010, pp. 1–19.
[50] F. Ciocchetta, J. Hillston, Bio-PEPA: A framework for the modelling and analysis of biological systems, Theor. Comput. Sci. 410 (33-34) (2009) 3065–3084.
[51] F. Ciocchetta, A. Degasperi, J. Hillston, M. Calder, Some investigations concerning the CTMC and the ODE model derived from Bio-PEPA, Electronic Notes in Theoretical Computer Science 229 (1) (2009) 145–163.
[52] J. Loros, J. Dunlap, Genetic and molecular analysis of circadian rhythms in *Neurospora*, Annu Rev Physiol 63 (2001) 757–94.
[53] J. E. Hopcroft, J. D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, 1979.
[54] R. Alur, C. Courcoubetis, T. A. Henzinger, P.-H. Ho, Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems, in: Hybrid Systems, Springer-Verlag, London, UK, UK, 1993, pp. 209–229.
[55] T. A. Henzinger, The Theory of Hybrid Automata, in: Logic in Computer Science, 1996. LICS '96. Proceedings., Eleventh Annual

IEEE Symposium on, IEEE Computer Society Press, 1996, pp. 278–292.
[56] O. Maler, Z. Manna, A. Pnueli, From Timed to Hybrid Systems, in: Proceedings of the Real-Time: Theory in Practice, REX Workshop, Springer, London, UK, 1992, pp. 447–484.
[57] A. Puri, P. Varaiya, Decidability of hybrid systems with rectangular differential inclusions, in: D. L. Dill (Ed.), Proceedings of Int. Conference on Computer Aided Verification (CAV'94), Vol. 818 of LNCS, Springer, 1994, pp. 95–104.
[58] T. A. Henzinger, P. W. Kopke, State Equivalences for Rectangular Hybrid Automata, in: U. Montanari, V. Sassone (Eds.), Proc. of Int. Conference on Concurrency Theory (Concur'96), Vol. 1119 of LNCS, Springer, 1996, pp. 530–545.
[59] A. Chutinan, B. Krogh, Verification of Polyhedral-Invariant Hybrid Automata Using Polygonal Flow Pipe Approximations, in: F. W. Vaandrager, J. H. van Schuppen (Eds.), Proc. of Hybrid Systems: Computation and Control (HSCC'99), Vol. 1569 of LNCS, Springer, 1999, pp. 76–90.
[60] A. Casagrande, C. Piazza, A. Policriti, B. Mishra, Inclusion dynamics hybrid automata, Inform. and Comput. 206 (12) (2008) 1394–1424.
[61] J. R. Collier, N. A. M. Monk, P. K. Maini, J. H. Lewis, Pattern Formation by Lateral Inhibition with Feedback: a Mathematical Model of Delta-Notch Intercellular Signalling, Journal of Theoretical Biology 183 (1996) 429–446.
[62] R. Ghosh, C. Tomlin, Lateral Inhibition through Delta-Notch signaling: A Piecewise Affine Hybrid Model, in: M. D. D. Benedetto, A. Sangiovanni-Vincentelli (Eds.), Proc. of Hybrid Systems: Computation and Control (HSCC'01), Vol. 2034 of LNCS, Springer, 2001, pp. 232–246.
[63] R. Ghosh, A. Tiwari, C. Tomlin, Automated Symbolic Reachability Analysis; with Application to Delta-Notch Signaling Automata, in: Proc. of Hybrid Systems: Computation and Control (HSCC'03), Vol. 2623 of LNCS, Springer, 2003, pp. 233–248.
[64] R. Alur, D. L. Dill, A theory of timed automata, Theor. Comput. Sci. 126 (2) (1994) 183–235.
[65] G. Lafferriere, G. J. Pappas, S. Sastry, O-minimal hybrid systems, Mathematics of Control, Signals, and Systems 13 (2000) 1–21.
[66] A. Tiwari, G. Khanna, Series of Abstraction for Hybrid Automata, in: C. J. Tomlin, M. Greenstreet (Eds.), Proc. of Hybrid Systems: Computation and Control (HSCC'02), Vol. 2289 of LNCS, Springer, 2002, pp. 465–478.
[67] R. Ghosh, A. Tiwari, C. Tomlin, Automated symbolic reachability analysis; with application to delta-notch signaling automata, in: Hybrid Systems: Computation and Control, Springer, 2003, pp. 233–248.
[68] R. Ghosh, C. Tomlin, Symbolic reachable set computation of piecewise affine hybrid automata and its application to biological modelling: Delta-notch protein signalling, Systems Biology 1 (1) (2004) 170–183.
[69] W. H. Organization, Influenza fact sheet no. 211 (Mar. 2014).
URL `http://www.who.int/mediacentre/factsheets/fs211/en/`
[70] N. P. A. S. Johnson, J. Mueller, Updating the accounts: global mortality of the 1918-1920 "Spanish" influenza pandemic, Bull Hist Med 76 (1) (2002) 105–15.
[71] J. K. Taubenberger, D. M. Morens, 1918 Influenza: the mother of all pandemics, Emerging Infectious Diseases 12 (1) (2006) 15–22, PMID: 16494711.
[72] E. W. Larson, J. W. Dominik, A. H. Rowberg, G. A. Higbee, Influenza virus population dynamics in the respiratory tract of experimentally infected mice, Infect Immun 13 (2) (1976) 438–47.
[73] P. Baccam, C. Beauchemin, C. A. Macken, F. G. Hayden, A. S. Perelson, Kinetics of influenza a virus infection in humans, Journal of virology 80 (15) (2006) 7590–7599.
[74] A. Handel, I. M. Longini, Jr, R. Antia, Neuraminidase inhibitor resistance in influenza: assessing the danger of its generation and spread, PLoS Comput Biol 3 (12) (2007) e240.
[75] R. A. Saenz, M. Quinlivan, D. Elton, S. MacRae, A. S. Blunden, J. A. Mumford, J. M. Daly, P. Digard, A. Cullinane, B. T. Grenfell, et al., Dynamics of influenza virus infection and pathology, Journal of virology 84 (8) (2010) 3974–3983.
[76] K. A. Pawelek, G. T. Huynh, M. Quinlivan, A. Cullinane, L. Rong, A. S. Perelson, Modeling within-host dynamics of influenza virus infection including immune responses, PLoS Comput Biol 8 (6) (2012) e1002588.
[77] A. Madrahimov, T. Helikar, B. Kowal, G. Lu, J. Rogers, Dynamics of influenza virus and human host interactions during infection and replication cycle, Bull Math Biol 75 (6) (2013) 988–1011.
[78] J. R. Quesada, J. Reuben, J. T. Manning, E. M. Hersh, J. U. Gutterman, Alpha interferon for induction of remission in hairy-cell leukemia, New England Journal of Medicine 310 (1) (1984) 15–18.
[79] D. K. Wong, A. M. Cheung, K. O'Rourke, C. D. Naylor, A. S. Detsky, J. Heathcote, Effect of alpha-interferon treatment in patients with hepatitis B e antigen-positive chronic hepatitis BA meta-analysis, Annals of internal medicine 119 (4) (1993) 312–323.
[80] H. C. Lane, Interferons in HIV and related diseases, AIDS 8 (1994) S19–S24.
[81] C. Goujon, O. Moncorgé, H. Bauby, T. Doyle, C. C. Ward, T. Schaller, S. Hué, W. S. Barclay, R. Schulz, M. H. Malim, Human mx2 is an interferon-induced post-entry inhibitor of hiv-1 infection, Nature 502 (7472) (2013) 559–62.
[82] S. Pol, Hepatitis C virus: 25 years-old, the end?, Med Sci (Paris) 29 (11) (2013) 998–1003.
[83] L. D. Jacobs, D. L. Cookfair, R. A. Rudick, R. M. Herndon, J. R. Richert, A. M. Salazar, J. S. Fischer, D. E. Goodkin, C. V. Granger, J. H. Simon, et al., Intramuscular interferon beta-1a for disease progression in relapsing multiple sclerosis, Annals of neurology 39 (3) (1996) 285–294.
[84] A. D. Valentine, C. A. Meyers, M. A. Kling, E. Richelson, P. Hauser, Mood and cognitive side effects of interferon-alpha therapy., in: Seminars in oncology, Vol. 25 (1 Suppl 1), 1998, pp. 39–47.
[85] E. Walther, R. Hohlfeld, Multiple sclerosis side effects of interferon beta therapy and their management, Neurology 53 (8) (1999) 1622–1622.
[86] M. W. Fried, Side effects of therapy of hepatitis c and their management, Hepatology 36 (S1) (2002) S237–S244.
[87] Octave community, GNU Octave 3.8.0 (2014).
URL `www.gnu.org/software/octave/`
[88] I. Julkunen, K. Melén, M. Nyqvist, J. Pirhonen, T. Sareneva, S. Matikainen, Inflammatory responses in influenza a virus infection, Vaccine 19 (2000) S32–S37.

[89] F. Dianzani, I. Viano, M. Santiano, M. Zucca, P. Gullino, S. Baron, Tissue culture models of in vivo interferon production and action, in: Human Interferon, Springer, 1978, pp. 119–131.
[90] A. Casagrande, C. Piazza, External interactions on hybrid models of biological systems, in: Formal Methods in Macro-Biology, Vol. 8738 of LNCS, Springer, 2014, pp. 63–81.
[91] J. M. Whitacre, Biological robustness: paradigms, mechanisms, and systems principles, Frontiers in Genetics 3.