On the Analysis of Numerical Data Time Series in Temporal Logic

François Fages, Aurélien Rizk

firstname.lastname@inria.fr
Projet Contraintes, INRIA Rocquencourt,
BP105, 78153 Le Chesnay Cedex, France.
http://contraintes.inria.fr

Abstract. Temporal logics and model-checking techniques have proved successful to respectively express biological properties of complex biochemical systems, and automatically verify their satisfaction in both qualitative and quantitative models. In this paper, we propose a finite time horizon model-checking algorithm for the existential fragment of LTL with numerical constraints over the reals, with the ability to compute the range of values of the real variables occurring in a formula that makes it true in a model. We illustrate this approach for the analysis of biological data time series, provide a set of biologically relevant patterns of formulas, and evaluate them on models of the cell cycle control and MAPK signal transduction.

1 Introduction

Temporal logics and model-checking techniques [1] have proved useful to respectively express biological properties of complex biochemical systems and automatically verify their satisfaction in both qualitative and quantitative models, i.e. in boolean [2–4], discrete [5, 6], stochastic [7, 8] and continuous models [9, 10, 3]. This approach relies on a logical paradigm for systems biology that consists in making the following identifications [11]:

- biological model = transition system
- biological property = temporal logic formulae
- biological validation = model-checking

Having a formal language not only for describing models, i.e. transition systems by either process calculi [12–16], rules [2, 17, 18], Petri nets [19], etc..., but also for formalizing the biological properties of the system known from biological experiments under various conditions, opens a whole avenue of research for designing automated reasoning tools inspired from circuit and program verification to help the modeler [20]. However, the formalization of the biological properties as a specification in temporal logic remains a delicate task and a bottleneck of the method.

In this paper, we investigate the use of this logical paradigm to analyze numerical data, and infer temporal logic specifications from experimental data time
series. There has been work on the inference of correlations and positive and negative influences between entities from numerical data time series, especially for gene expression temporal data [21, 22]. However to our knowledge, the inference of temporal logic formulae with real valued variables from numerical data time series is new.

In this paper, we generalize the finite time horizon model-checking algorithm described in [9] and recalled in the next section, to the existential fragment of LTL with numerical constraints over the reals. This first-order setting provides the ability to compute those instantiations of a formula that are true in a model, by giving the range of values of the real valued variables occurring in the formula for which it is true. The completeness of the algorithm is shown for the considered fragment of constraint-LTL in Sec. 3.

We illustrate the relevance of this approach to the analysis of biological data time series, by providing a set of biologically relevant patterns of formulas in Sec. 4, and by evaluating them on models of cell cycle control and of signal transduction in Sec. 5. We then conclude on the results achieved so far, their generality, and their use in on-going work.

2 Preliminaries on Model-Checking in Constraint-LTL over the Reals

2.1 Constraint-LTL over the Reals

The Linear Time Logic LTL is a temporal logic [1] that extends propositional or first-order logic with modal operators for qualifying when a formula is true in a tree of timed states, called a Kripke structure. The temporal operators are $X$ (“next”, for at the next time point), $F$ (“finally”, for at some time point in the future), $G$ (“globally”, for at all time points in the future), and $U$ (“until”). These operators enjoy some simple duality properties, $\neg X \varphi = X \neg \varphi$, $\neg F \varphi = G \neg \varphi$, $\neg G \varphi = F \neg \varphi$, $\neg (\psi U \varphi) = G \neg \varphi \lor (\neg \varphi U (\neg \psi \land \neg \psi))$, and $F \varphi = \text{true} U \varphi$.

A first-order version of LTL with constraints over the reals, called constraint-LTL, is used in Biocham [9] to express temporal properties about molecular concentrations. A similar approach is used in the Darpa BioSpice project [10]. Constraint-LTL considers first-order atomic formulae with equality, inequality and arithmetic operators ranging over real values of concentrations and of their derivatives. For instance $F([A]>10)$ expresses that the concentration of $A$ eventually gets above the threshold value 10. $G([A]+[B]<[C])$ expresses that the concentration of $C$ is always greater than the sum of the concentrations of $A$ and $B$. Oscillation properties, abbreviated as oscil($\varphi$, $K$), are defined as a change of sign of the derivative of $\varphi$ at least $K$ times: $F((d[M]/dt > 0) \land F((d[M]/dt < 0) \land F((d[M]/dt > 0)\ldots)))$ The abbreviated formula oscil($\varphi$, $K$, $V$) adds the constraint that the maximum concentration of $M$ must be above the threshold $V$ in at least $K$ oscillations.
2.2 Model-Checking Algorithm

In an ODE model, and under the hypothesis that the initial state is completely defined, numerical integration methods (such as Runge-Kutta or Rosenbrock method for stiff systems) provide a discrete simulation trace. This trace constitutes a linear Kripke structure in which constraint-LTL formulae can be interpreted. Since constraints refer not only to concentrations, but also to their derivatives, we consider traces of the form

\[(< t_0, x_0, dx_0/dt, d^2x_0/dt^2 >, < t_1, x_1, dx_1/dt, d^2x_1/dt^2 >, ... )\]

where at each time point, \( t_i \), the trace associates the concentration values \( x_i \) of the variables, and the values of their first and second derivatives \( dx_i/dt \) and \( d^2x_i/dt^2 \). This choice of derivatives is justified in section 4 as a facility for expressing positive or negative influences between entities. It is worth noting that in adaptive step size integration methods, the step size \( t_{i+1} - t_i \) is not constant and is determined through an estimation of the error made by the discretization.

**Algorithm 1 (trace-based constraint-LTL model-checking)** [9, 10] Given an ODE model and a temporal property \( \phi \) to verify within a finite time horizon,

1. compute a finite simulation trace;
2. label each trace point by the atomic sub-formulae of \( \phi \) that are true at this point;
3. add sub-formulae of the form \( F\phi \) (resp. \( X\phi \)) to the predecessors (resp. immediate predecessor) of a point labeled with \( \phi \);
4. add sub-formulae of the form \( \phi_1 U \phi_2 \) to the points preceding a point labeled with \( \phi_2 \) as long as \( \phi_1 \) holds;
5. add sub-formulae of the form \( G\phi \) to the last state if it is labeled by \( \phi \), and to the predecessors of the points labeled by \( G\phi \) as long as \( \phi \) holds.
6. return the time points labeled by \( \phi \).

Note that being limited to finite traces, and since \( G\phi = !F(!\phi) \), we chose to label the last state by \( G\phi \) if it satisfies \( \phi \), just like if the trace was completed to the infinite by a loop on its last state. Note also that the notion of next state (operator \( X \)) refers to the state of the following time point in a discretized trace, and thus does not necessarily imply real time neighborhood. The rationale of this algorithm is that the numerical trace contains enough relevant points, and in particular those where the derivatives change abruptly, to correctly evaluate temporal logic formulae. This has been very well verified in practice with various examples of published mathematical models [9].

3 Formula Instantiation in \( \exists \)-Constraint-LTL over the Reals

3.1 The Existential Fragment \( \exists \)-Constraint-LTL

Here we consider the existential fragment of constraint-LTL, where real valued variables are allowed in the constraints, with an implicit existential quantifica-
tion. More precisely, the language of $\exists$-constraint-LTL formulae considered in this paper is defined by the following grammar:

$\text{Constraint} - \text{ltl} =$

$\text{Atom} \mid F(\text{Constraint} - \text{ltl})$

$\mid G(\text{Constraint} - \text{ltl}) \mid X(\text{Constraint} - \text{ltl})$

$\mid (\text{Constraint} - \text{ltl}) U (\text{Constraint} - \text{ltl})$

$\mid (\text{Constraint} - \text{ltl}) \text{ and } (\text{Constraint} - \text{ltl})$

$\mid (\text{Constraint} - \text{ltl}) \text{ or } (\text{Constraint} - \text{ltl})$

$\mid (\text{Constraint} - \text{ltl}) \Rightarrow (\text{Constraint} - \text{ltl})$

$\mid \text{not } (\text{Constraint} - \text{ltl})$

$\text{Atom} =$

$\text{Value < Variable} \mid \text{Value } \leq \text{Variable}$

$\mid \text{Value > Variable} \mid \text{Value } \geq \text{Variable}$

$\mid \text{Value < Value} \mid \text{Value } \leq \text{Value}$

$\mid \text{Value > Value} \mid \text{Value } \geq \text{Value}$

$\text{Value} =$

$\text{float} \mid [\text{molecule}] \mid \frac{d[\text{molecule}]}{dt} \mid \frac{d^2[\text{molecule}]}{dt^2} \mid \text{Time}$

$\mid \text{Value + Value} \mid \text{Value - Value} \mid - \text{Value} \mid \text{Value } \times \text{Value}$

$\mid \text{Value } / \text{Value} \mid \text{Value}^{\text{Value}}$

By an obvious transformation, negations and implications can be eliminated, by propagating the negations down to the atomic constraints in the formula. From now on, we will assume that all $\exists$-constraint-LTL formulae are in negation free normal form.

3.2 Formula Instantiation Algorithm

Given a trace $T$ representing a linear Kripke structure, and an $\exists$-constraint-LTL formula $\phi$ with $n$ variables, the formula instantiation problem, $\exists \nu \in \mathbb{R}^n (\phi(\nu))$, is the problem of determining the valuation $\nu$ of the variables for which the formula $\phi$ is true in $T$. In other words, we look for the domain of validity $D_\phi \subseteq \mathbb{R}^n$ such that $T \models_{\text{LTL}} \forall \nu \in D_\phi (\phi(\nu))$.

The domain of validity $D_\phi$ of $\phi$ can be computed using an algorithm similar to the model-checking algorithm of section 2.2:

**Algorithm 2 (trace-based $\exists$-constraint-LTL formula instantiation)** Given an ODE model and a temporal property $\phi$ with variables, to verify in a finite time horizon,

1. compute a finite simulation trace;
Proof. Let us prove inductively on the constraint-LTL formula structure that
domain of $v$

Theorem 1. The instantiation algorithm is correct and complete: a valuation $v$ makes $\phi$ true at time $t_i$, $T, t_i \models_{\text{LTL}} (\phi(v))$, if and only if $v$ is in the computed domain of $\phi$ at $t_i$, $v \in D_\phi(t_i)$.

Proof. Let us prove inductively on the constraint-LTL formula structure that for any time $t$, any LTL formula $\phi$ and any instantiation $v$ of the variables, if $\phi(v, t_i)$ is true then $v \in D_\phi(t_i)$ and if $v \in D_\phi(t_i)$ then $\phi(v, t_i)$ is true:

- Atomic constraint-LTL formulae considered are of the form Value $R$ Variable or Value $R$ Value where Value is an evaluable arithmetic expression and $R$ an inequality operator. For all these atomic formulae the algorithm returns the exact validity domain. For instance, formula $([A] \leq p)$ is true if and only if $p$ is greater or equal to $[A](t_i)$ and the validity domain returned is the half-space defined by $p \geq [A](t_i)$;
- $\phi_1$ and $\phi_2$. By algorithm construction $D_{\phi_1} \cap D_{\phi_2}(t_i) = D_{\phi_1}(t_i) \cap D_{\phi_2}(t_i)$ hence: $v \in D_{\phi_1}(t_i)$ and $v \in D_{\phi_2}(t_i)$ $\iff v \in D_{\phi_1}(t_i)$ and $v \in D_{\phi_2}(t_i)$ $\iff \phi_1(v, t_i)$ and $\phi_2(v, t_i)$ $\iff (\phi_1 \cap \phi_2)(v, t_i)$;
- $\phi_1$ or $\phi_2$. By algorithm construction $D_{\phi_1} \cup D_{\phi_2}(t_i) = D_{\phi_1}(t_i) \cup D_{\phi_2}(t_i)$ hence: $v \in D_{\phi_1}(t_i)$ or $v \in D_{\phi_2}(t_i)$ $\iff \phi_1(v, t_i)$ or $\phi_2(v, t_i)$ $\iff (\phi_1 \cup \phi_2)(v, t_i)$;
− $F(\phi)$. By algorithm construction $D_{F(\phi)}(t_1) = \bigcup_{j \geq 1} (D_{\phi}(t_j))$ hence: $v \in D_{F(\phi)}(t_1) \iff \exists j \geq i, v \in D_{\phi}(t_j)$;

− $G(\phi)$. By algorithm construction $D_{G(\phi)}(t_1) = \bigcap_{j \geq 1} (D_{\phi}(t_j))$ hence: $v \in D_{G(\phi)}(t_1) \iff \forall j \geq i, v \in D_{\phi}(t_j)$;

− $X(\phi)$. By algorithm construction $D_{X(\phi)}(t_1) = D_{\phi}(t_1 + 1)$ hence:

$v \in D_{X(\phi)}(t_1) \iff v \in D_{\phi}(t_1 + 1) \iff X(\phi)(v, t_1)$;

− $\phi_1 U \phi_2$. $\phi_1 U \phi_2(t_1)$ is true if and only if $\phi_2(t_1)$ is true or $\phi_1(t_1)$ is true and $(\phi_1 U \phi_2)(t_1)$ is true. Moreover by construction $D_{(\phi_1 U \phi_2)}(t_1) = D_{\phi_2}(t_1) \cup (D_{\phi_1}(t_1) \cap D_{\phi_2}(t_1))$ hence: $v \in D_{(\phi_1 U \phi_2)}(t_1) \iff v \in D_{\phi_2}(t_1) \cup (D_{\phi_1}(t_1) \cap D_{\phi_2}(t_1)) \iff (\phi_1 U \phi_2)(v, t_1)$.

Now, let us define a box of $\mathbb{R}^n$ as a finite intersection of half-spaces and the size $S(D)$ of a domain as the minimum number of boxes the domain is made of. Note that for a one dimension domain $D = D_1 \cup D_2$ (resp. $D = D_1 \cap D_2$), the maximum size is $S(D_1) + S(D_2)$ (resp. max($S(D_1), S(D_2)$)).

**Theorem 2.** In the worst case, the size of the validity domain of a LTL formula of size $k$ on a trace of length $n$ is $n^k$.

**Proof.** Let us prove inductively that the size of the projection on one variable of the validity domain (i.e., the validity domain of a single variable) of a LTL formula of size $k$, is at most $n^k$. The size of the validity domain of an atomic formula is at most 1. The maximum size of a one dimension domain of a formula of size $k$ is:

$$\max(S(D_{\phi_1}(t_1)), S(D_{\phi_2}(t_1))) \text{ for } D_{\phi_1} \text{ and } \phi_2(t_1)$$

$$S(D_{\phi_1}(t_1)) + S(D_{\phi_2}(t_1)) \text{ for } D_{\phi_1} \text{ or } \phi_2(t_1)$$

$$\sum_{j \geq 1} S(D_{\phi}(t_j)) \text{ for } D_{F(\phi)(t_1)}$$

$$\max_{j \geq 1} S(D_{\phi}(t_j)) \text{ for } D_{G(\phi)(t_1)}$$

$$S(D_{\phi}(t_1 + 1)) \text{ for } D_{X(\phi)(t_1)}$$

$$\max(S(D_{\phi_1 U \phi_2}(t_1 + 1)), S(D_{\phi_1}(t_1))) + S(D_{\phi_2}(t_1)) \text{ for } D_{\phi_1 U \phi_2}(t_1)$$

In all these cases except operators $F$ and $U$, the size of the domain is less than the sum of the domains’ size of the subformulae at one time point, which entails a size smaller than $2n^{k-1}$. The $U$ and $F$ operators make a sum on all time points which entails a size of at most $n \times n^{k-1} = n^k$. Each projection’s size of the validity domain is thus at most $n^k$. The size of the validity domain of a formula containing $v$ variables is at most $(n^k)^v \leq (n^k)^k = n^{k^2}$.

The instantiation algorithm thus computes for each subformulae and each time point a validity domain of size at most $n^{k^2}$. 
4 Biologically Relevant Patterns of $\exists$-constraint-LTL Formulae

Temporal logic is sufficiently expressive to formalize a wide range of biological properties known from experiments under various conditions. The formula instantiation algorithm in $\exists$-constraint-LTL makes it possible to analyze concentration traces and obtain semi-quantitative information. In particular, a quantitative counterpart of the purely qualitative properties in propositional CTL studied in [3] can be expressed as follows, where variables are written using lowercase letters:

**Reachability**: $F([A] \geq p)$, what threshold $p$ species A attain in the trace?

**Checkpoints**: $\neg ([A] < p_1) \lor ([B] > p_2)$, for which thresholds $p_1$ and $p_2$ is it false that $[A]$ is lower than $p_1$ until $[B]$ is above $p_2$, i.e., for which $p_1$ and $p_2$ $[A] \geq p_1$ is a checkpoint of $[B] > p_2$?

**Stability**: $G([A] = < p_1 \land [A] \geq p_2)$, what is the range of values taken by $[A]$?

This range can be looked for in some context given by a condition like in $G(Time > 10 \rightarrow ([A] < p_1 \land [A] > p_2))$.

**Oscillation**: $F((d([A])/dt > 0 \land [A] > v_1) \land F((d([A])/dt < 0 \land [A] < v_2)))$, what amplitude $(v_1 - v_2)$ is attained in at least one oscillation? An oscillation is defined as the change of sign of the derivative. This formula can be extended for more oscillations and is abbreviated by $oscil(M, K, p)$. It states that $M$ must have amplitude $P$ in at least $K$ oscillations. By applying the algorithm for each value of $K$, beginning with 1, we can find the number of oscillations in the trace and minimal amplitude $P$ attained by $K$ oscillations for any $K$.

**Influence**: $G(d[A]/dt > p_1 \rightarrow d2[B]/dt2 = 0)$, above which threshold does the derivative of $A$ have an influence on $B$? The influence is positive if a high value of $d[A]/dt$ entails a positive second derivative of $[B]$. It is worth noticing that, as multiple species might influence $B$, this formula only indicates a correlation between the value of the derivative of $A$ and the second derivative of $B$ and gives no proof of direct influence.

5 Application to the Inference of Temporal Properties from Biological Time Series

5.1 Cell Cycle Data

In this section we present the application of the instantiation algorithm to the budding yeast cell cycle data. For the purpose of evaluation of the method, we do not use experimental data but simulation data obtained from the model of [23]. The application of the method to experimental data is discussed in section 5.3. Concentration traces are obtained by simulating the cell cycle control model in Biocham. Then, we try to recover relevant properties of the model by automatically analyzing the traces.

The reaction rules of the model are the following:
Notations \( \{ p_1 \} \) and \( \{ p_1, p_2 \} \) denote phosphorylated forms of a molecule. Figure 1 displays the obtained simulation traces for four species of this model.

\[
\begin{align*}
1. & \text{ } \Rightarrow \text{Cyclin.} \\
2. & \text{Cyclin} + \text{Cdc2}^{\{ p_1 \}} & \Rightarrow & \text{Cdc2-Cyclin}^{\{ p_1, p_2 \}} \\
3. & \text{Cdc2-Cyclin}^{\{ p_1, p_2 \}} & \Rightarrow & \text{Cdc2-Cyclin}^{\{ p_1 \}} \\
4. & \text{Cdc2-Cyclin}^{\{ p_1, p_2 \}} = [\text{Cdc2-Cyclin}^{\{ p_1 \}}] & \Rightarrow & \text{Cdc2-Cyclin}^{\{ p_1 \}} \\
5. & \text{Cdc2-Cyclin}^{\{ p_1 \}} & \Rightarrow & \text{Cyclin}^{\{ p_1 \}} + \text{Cdc2} \\
6. & \text{Cyclin}^{\{ p_1 \}} & \Rightarrow & \_ \\
7. & \text{Cdc2} & \Rightarrow & \text{Cdc2}^{\{ p_1 \}} \\
8. & \text{Cdc2}^{\{ p_1 \}} & \Rightarrow & \text{Cdc2} \\
\end{align*}
\]

Fig. 1. Budding yeast cell cycle simulation trace over 100 time units made of 94 time points.

Such traces are remarkably informative, however to automate reasoning on them, we propose to rely on constraint-LTL queries. For instance, a reachability query provides the maximum concentration attained by an entity:

\[
\text{biocham: trace_analyze}(\text{F}([\text{Cdc2-Cyclin}^{\{ p_1 \}}] \geq v)). \\
[[v=<0.194]]
\]

The result returned is a list of domains represented by lists of constraints on the variables, here a single domain is returned with a single constraint on \( v \). In formulae like \( \text{F}([\text{Cdc2-Cyclin}^{\{ p_1 \}}] \geq v) \) where the variable only appears in inequalities of the form \( \text{Value} \geq \text{Variable} \) or \( \text{Value} > \text{Variable} \), the most relevant point of the domain is the highest value of \( v \) in the domain, i.e. its boundary. Its
value is here 0.194, the maximum concentration of $\text{Cdc2-Cyclin}^{\text{p1}}$ in the trace. Table 1 gives the maximum reachable values for the four species displayed in Figure 1.

<table>
<thead>
<tr>
<th>Species</th>
<th>Reachability</th>
<th>Stability</th>
<th>Amplitude of at least $n$ oscillations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$n = 1$</td>
<td>$n = 2$</td>
</tr>
<tr>
<td>$\text{Cdc2}$</td>
<td>0.500</td>
<td>(0.338,0.479)</td>
<td>0.141</td>
</tr>
<tr>
<td>$\text{Cdc2-Cyclin}^{\text{p1,p2}}$</td>
<td>0.311</td>
<td>(0.005,0.310)</td>
<td>0.306</td>
</tr>
<tr>
<td>$\text{Cdc2-Cyclin}^{\text{p1}}$</td>
<td>0.194</td>
<td>(0.002,0.194)</td>
<td>0.192</td>
</tr>
<tr>
<td>$\text{Cyclin}^{\text{p1}}$</td>
<td>0.159</td>
<td>(0.004,0.158)</td>
<td>0.155</td>
</tr>
</tbody>
</table>

**Table 1.** Results for reachability (maximum value), stability (bottom and top values in the last third part of the trace) and amplitude of at least $n$ oscillations.

For stability, let us find the range of values taken by $[\text{Cdc2}]$ in the last third part of the trace:

\[
\text{biocham: trace\_analyze}(G(\text{Time} > 66 \rightarrow ([\text{Cdc2}] \geq v_1 \& [\text{Cdc2}] \leq v_2)))
\]

$[[v_1 \geq 0.479, v_2 \leq 0.338]]$

The domain is defined by the conjunction of the two constraints $v_1 \geq 0.479$ and $v_2 \leq 0.338$. These values are the maximum and minimum values attained by $[\text{Cdc2}]$ in the last third part of the trace. The results for the other species are given in Table 1.

An oscillation query may compute several interval domains:

\[
\text{biocham: trace\_analyze}(\text{oscil}(\text{Cdc2},1)).
\]

$[[v_2 \geq 0.338, v_1 \leq 0.479], [v_2 \geq 0.341, v_1 \leq 0.479]]$

The result is the union of two boxes. In such domains, the most relevant point is not obvious. Here we look for the maximum amplitude $v_1 - v_2$. The maximum is obtained in the domain with $v_1 - v_2 = 0.479 - 0.338 = 0.141$. This result states that at least one oscillation of $\text{Cdc2}$ has an amplitude greater or equal to 0.141. The number of oscillations is then incremented until obtaining an empty validity domain. It is obtained for $\text{Cdc2}$ with the query $\text{oscil}(\text{Cdc2},3)$, stating that there are only two oscillations of $\text{Cdc2}$ in the trace.

The results for the other species are given in Table 1. Obtaining the amplitude of the oscillations is useful to distinguish between mixed amplitudes oscillations in the trace. For instance, in noisy data the amplitude can be used to count the number of oscillations regardless of small noise induced oscillations.

Whether $\text{Cdc2-Cyclin}^{\text{p1,p2}}$ acts as a checkpoint for $\text{Cdc2-Cyclin}^{\text{p1}}$ can be investigated with the following formula:

\[
\text{not}([\text{Cdc2-Cyclin}^{\text{p1,p2}}]<v_1 \cup [\text{Cdc2-Cyclin}^{\text{p1}}]>v_2)
\]

The resulting domain is a union of ten boxes. Interpreting it requires examining each box to find interesting points of the domain. Checkpoint queries are thus more delicate and perhaps not well suited for automatic analysis. In the
example, the values $v_1 = 0.311$ and $v_2 = 0.014$ are in the domain, stating that Cdc2-Cyclin$^{-\{p1,p2\}}$ is not always less than 0.311 until Cdc2-Cyclin$^{-\{p1\}}$ exceeds 0.014. In other words Cdc2-Cyclin$^{-\{p1,p2\}}$ goes beyond 0.311 before Cdc2-Cyclin$^{-\{p1\}}$ exceeds 0.014 pointing out that Cdc2-Cyclin$^{-\{p1,p2\}}$ is indeed a checkpoint.

Now, the influence of a molecule A on a molecule B is looked for with formula $G(d[A]/dt > v_1 \Rightarrow d^2[B]/dt^2 > 0)$ and $G(d[A]/dt < v_2 \Rightarrow d^2[B]/dt^2 < 0)$. The idea behind this formula is that if a species $B$ appears only in a reaction rule of the form $A \rightarrow B$ with a mass action law kinetic, the following constraint-LTL formulae are true: $G(d[A]/dt > 0 \Rightarrow d^2[B]/dt^2 > 0)$ and $G(d[A]/dt < 0 \Rightarrow d^2[B]/dt^2 < 0)$.

In a typical system each species concentration is the result of the combined effect of several other species. $\exists$-constraint-LTL formula search determines above which threshold the above formulae are true, i.e. validity domains of variables $v_1$ and $v_2$ in formulae $G(d[A]/dt > v_1 \Rightarrow d^2[B]/dt^2 > 0)$ and $G(d[A]/dt < v_2 \Rightarrow d^2[B]/dt^2 < 0)$.

By comparing these domains to the range of values of $d[A]/dt$, a score $s \in [0, 1]$ is obtained indicating the influence of the derivative of $[A]$ over the second derivative of $[B]$. More precisely, if the domain of validity is $v_1 = 0$ it means that the formula is true for any positive value of $d[A]/dt$ resulting in a score 1. If the domain of validity is $v_1 \geq \max(d[A]/dt)$ it means that the formula is true for half of the positive values of $d[A]/dt$ resulting in a score 0.5. Table 2 gives influences scores computed by this method for species Cdc2 and Cdc2-Cyclin$^{-\{p1,p2\}}$.

<table>
<thead>
<tr>
<th>Species</th>
<th>Cdc2</th>
<th>Cdc2-Cyclin$^{-{p1,p2}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cdc2</td>
<td>0.00</td>
<td>0.11</td>
</tr>
<tr>
<td>Cdc2$^{-{p1}}$</td>
<td>0.01</td>
<td>0.12</td>
</tr>
<tr>
<td>Cyclin</td>
<td>0.00</td>
<td>0.34</td>
</tr>
<tr>
<td>Cdc2-Cyclin$^{-{p1,p2}}$</td>
<td>0.00</td>
<td>0.02</td>
</tr>
<tr>
<td>Cdc2-Cyclin$^{-{p1}}$</td>
<td>0.90</td>
<td>0.00</td>
</tr>
<tr>
<td>Cyclin$^{-{p1}}$</td>
<td>0.50</td>
<td>0.09</td>
</tr>
</tbody>
</table>

Table 2. Positive influence scores of all species on Cdc2 and Cdc2-Cyclin$^{-\{p1,p2\}}$. Molecules appearing in rows (resp. columns) act as molecule A (resp. B) in formulae $G(d[A]/dt > v_1 \Rightarrow d^2[B]/dt^2 > 0)$ and $G(d[A]/dt < v_2 \Rightarrow d^2[B]/dt^2 < 0)$ used to compute these scores.

According to the reaction rules, the only species having a positive influence on [Cdc2] is [Cdc2-Cyclin$^{-\{p1\}}$] (reaction (5)). The influence scores returned correctly reflect this. The score obtained by Cyclin$^{-\{p1\}}$ is due to its closeness with [Cdc2-Cyclin$^{-\{p1,p2\}}$] as it can be seen in the trace. These two species both have a concentration rise coinciding with [Cdc2] own concentration rise. Nevertheless, influence scores defined above enable to distinguish [Cdc2-Cyclin$^{-\{p1\}}$] over [Cyclin$^{-\{p1\}}$] as molecule having a positive influence on Cdc2.
According to the reaction rules, the two species having a positive influence on \( \text{Cdc2-Cyclin}^{\{p1,p2\}} \) are \([\text{Cyclin}]\) and \([\text{Cdc2}^{\{p1\}}]\) (reaction (2)). Notice that as more species influence \( \text{Cdc2-Cyclin}^{\{p1,p2\}} \) than \( \text{Cdc2} \), it is harder to find correlations between single species and \( \text{Cdc2-Cyclin}^{\{p1,p2\}} \). Therefore overall influence scores are smaller in this case. In spite of this, the two species having the highest scores are the correct ones.

### 5.2 MAPK Signal Transduction Data

The MAPK signal transduction data model is used in the same way as the cell cycle model to evaluate the analysis method. Reaction rules used to simulate concentration traces, displayed in Figure 2 are given below. All reactions rules have mass action law kinetics.

![Fig. 2. MAPK model simulation trace over 20 time units made of 50 time points.](image)

\[
\begin{align*}
(1) \text{RAF} + \text{RAFK} & \iff \text{RAF-RAFK} \\
(2) \text{RAF}^{\{p1\}} + \text{RAFPH} & \iff \text{RAF}^{\{p1\}}-\text{RAFPH} \\
(3) \text{MEK}^{\{P\}} + \text{RAF}^{\{p1\}} & \iff \text{MEK}^{\{P\}}-\text{RAF}^{\{p1\}} \\
& \text{where } p2 \text{ not in } \{P\} \\
(4) \text{MEKPH} + \text{MEK}^{\{p1\}}^{\{P\}} & \iff \text{MEK}^{\{p1\}}^{\{P\}}-\text{MEKPH} \\
(5) \text{MAPK}^{\{P\}} + \text{MEK}^{\{p1,p2\}} & \iff \text{MAPK}^{\{P\}}-\text{MEK}^{\{p1,p2\}} \\
& \text{where } p2 \text{ not in } \{P\}
\end{align*}
\]
(6) MAPKPH + MAPK−{p1}−$P$ ⇔ MAPK−{p1}−$P$−MAPKPH
(7) RAF−RAFK ⇒ RAFK + RAF−{p1}
(8) RAF−{p1}−RAFPH ⇒ RAF + RAFPH
(9) MEK−{p1}−RAF−{p1} ⇒ MEK−{p1,p2} + RAF−{p1}
(10) MEK−RAF−{p1} ⇒ MEK−{p1} + RAF−{p1}
(11) MEK−{p1}−MEKPH ⇒ MEK + MEKPH
(12) MEK−{p1,p2}−MEKPH ⇒ MEK−{p1} + MEKPH
(13) MAPK−MEK−{p1,p2} ⇒ MAPK−{p1} + MEK−{p1,p2}
(14) MAPK−{p1}−MEK−{p1,p2} ⇒ MAPK−{p1,p2} + MEK−{p1,p2}
(15) MAPK−{p1}−MAPKPH ⇒ MAPK + MAPKPH
(16) MAPK−{p1,p2}−MAPKPH ⇒ MAPK−{p1} + MAPKPH

Reachability, stability and oscillations queries results are given in Table 3. Stability queries return very small ranges of values in the last third part of the trace for most species indicating stable behaviors. For instance, RAF is inside the [0.310, 0.311] interval in the last third part of the trace. There are no oscillations of the species except a very small one for RAFK.

<table>
<thead>
<tr>
<th>Species</th>
<th>Reachability</th>
<th>Stability</th>
<th>Amplitude of at least $n$ oscillations</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAFK</td>
<td>1</td>
<td>(0.865, 0.866)</td>
<td>0.001</td>
</tr>
<tr>
<td>RAF</td>
<td>1</td>
<td>(0.310, 0.311)</td>
<td>-</td>
</tr>
<tr>
<td>MEK</td>
<td>1</td>
<td>(0.519, 0.537)</td>
<td>-</td>
</tr>
<tr>
<td>MAPK−{p1}</td>
<td>1</td>
<td>(0.891, 0.916)</td>
<td>-</td>
</tr>
<tr>
<td>RAF−{p1}</td>
<td>0.311</td>
<td>(0.311, 0.311)</td>
<td>-</td>
</tr>
<tr>
<td>MEK−{p1}</td>
<td>0.178</td>
<td>(0.174, 0.178)</td>
<td>-</td>
</tr>
<tr>
<td>MEK−{p1,p2}</td>
<td>0.060</td>
<td>(0.052, 0.060)</td>
<td>-</td>
</tr>
<tr>
<td>MAPK−{p1}</td>
<td>0.052</td>
<td>(0.039, 0.052)</td>
<td>-</td>
</tr>
<tr>
<td>MAPK−{p1,p2}</td>
<td>0.003</td>
<td>(0.001, 0.003)</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3. Results for Reachability (maximum value) and Stability (bottom and top values in the last third part of the trace)

This model is made of a cascade of phosphorylation reactions. According to the reaction rules, RAFK acts as a kinase on RAF (reactions 1 and 7), RAF acts as a kinase on MEK (reactions 3, 9 and 10) and MEK acts as a kinase on MAPK (reactions 5,13 and 14).

We looked for positive influence of any species an all phosphorylated forms of RAF, MEK and MAPK. The highest score for RAF−{p1} is 0.96 and is attained by species [RAF−RAFK] while [RAFK] ’s score is 0. This is consistent with the way phosphorylation reactions are written in the model, that is a complexation reaction and then a decomplexation-phosphorylation rule. Highest influence score for the phosphorylated form of MEK is correctly obtained for [MEK−RAF−{p1}], while in the case of [MAPK−{p1}] the correct complex [MAPK−MEK−{p1,p2}] only gets the second highest score, and the situation is even more confused for [MAPK−{p1,p2}].
Notice that lots of other species have relatively high influence score, which is no surprising given the similar shape of all curves in the trace. Nevertheless retaining only species having the highest scores as having positive influence, gives an overall good indication of the direct influences between species.

<table>
<thead>
<tr>
<th>Species</th>
<th>RAF {p1}</th>
<th>MEK {p1}</th>
<th>MEK {p1,p2}</th>
<th>MAPK {p1}</th>
<th>MAPK {p1,p2}</th>
</tr>
</thead>
<tbody>
<tr>
<td>[RAFK]</td>
<td>0.00</td>
<td>0.11</td>
<td>0.46</td>
<td>0.77</td>
<td>0.50</td>
</tr>
<tr>
<td>[RAF]</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.20</td>
<td>0.02</td>
</tr>
<tr>
<td>[MEK]</td>
<td>0.26</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>[MAPK]</td>
<td>0.50</td>
<td>0.47</td>
<td>0.11</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>[MAPKPH]</td>
<td>0.50</td>
<td>0.49</td>
<td>0.20</td>
<td>0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>[MEKPH]</td>
<td>0.48</td>
<td>0.06</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>[RAFPH]</td>
<td>0.14</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>[RAF-RAFK]</td>
<td><strong>0.96</strong></td>
<td>0.50</td>
<td>0.50</td>
<td>0.00</td>
<td>0.50</td>
</tr>
<tr>
<td>[RAFPH-RAF{p1}]</td>
<td>0.00</td>
<td>0.22</td>
<td>0.47</td>
<td>0.42</td>
<td><strong>0.50</strong></td>
</tr>
<tr>
<td>[MEK-RAF{p1}]</td>
<td>0.50</td>
<td><strong>0.79</strong></td>
<td><strong>0.66</strong></td>
<td>0.42</td>
<td><strong>0.50</strong></td>
</tr>
<tr>
<td>[MEK{p1}-RAF{p1}]</td>
<td>0.00</td>
<td>0.00</td>
<td>0.27</td>
<td>0.41</td>
<td>0.48</td>
</tr>
<tr>
<td>[MEKPH-MEK{p1}]</td>
<td>0.00</td>
<td>0.00</td>
<td>0.34</td>
<td>0.45</td>
<td>0.49</td>
</tr>
<tr>
<td>[MAPK-MEK{p1}]</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.60</td>
<td>0.34</td>
</tr>
<tr>
<td>[MAPK-MEK{p1,p2}]</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.62</td>
<td>0.37</td>
</tr>
<tr>
<td>[MAPKPH-MEK{p1}]</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.09</td>
</tr>
<tr>
<td>[MAPKPH-MAPK{p1}]</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.19</td>
</tr>
<tr>
<td>[MAPKPH-MAPK{p1,p2}]</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 4. Positive influence scores of all species on phosphorylated forms of RAF, MEK and MAPK.

5.3 Experimental Data

Experimental data for measuring the evolution over time of gene expression levels or of protein concentrations, typically involve between 6 and 50 time points taken at regular intervals. Furthermore, experimental data are noisy, and it is not one trace but several ones that have to be analyzed in order to extract their significant features. The strategy here is thus to analyze the traces separately and retain the intersection set of their properties, or the most frequent ones only.

In order to evaluate the instantiation algorithm on similar experimental-like concentration traces, we extracted eleven equally spaced time points from the cell cycle simulation trace. The obtained trace is displayed in Figure 3.

We applied on this trace the same queries than on the original simulated one, results are given in Tables 5 and 6. Oscillations properties are still obtained but with smaller amplitudes, because the peaks are missed in the sampling. For instance, Cdc2-Cyclin\{p1\} has one oscillation of size 0.192 but two oscillations of size only greater than 0.012. This is a limit inherent to a low number of time points as the first peak of Cdc2-Cyclin\{p1\} almost disappeared in this trace. Having a small number of time points also tends to give high self positive...


Fig. 3. Curve of concentrations every 10 units of time extracted from the cell cycle simulation trace.

<table>
<thead>
<tr>
<th>Species</th>
<th>Reachability</th>
<th>Stability</th>
<th>Amplitude of at least $n$ oscillations $\scriptstyle n = 1$</th>
<th>Amplitude of at least $n$ oscillations $\scriptstyle n = 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cdc2</td>
<td>0.500</td>
<td>(0.341,0.441)</td>
<td>0.125</td>
<td>0.089</td>
</tr>
<tr>
<td>Cdc2-Cyclin$^{{p1,p2}}$</td>
<td>0.311</td>
<td>(0.031,0.308)</td>
<td>0.279</td>
<td>0.222</td>
</tr>
<tr>
<td>Cdc2-Cyclin$^{{p1}}$</td>
<td>0.194</td>
<td>(0.002,0.194)</td>
<td>0.192</td>
<td>0.012</td>
</tr>
<tr>
<td>Cyclin$^{{p1}}$</td>
<td>0.100</td>
<td>(0.005,0.100)</td>
<td>0.095</td>
<td>0.018</td>
</tr>
</tbody>
</table>

Table 5. Results for reachability, stability and oscillation queries in experimental-like data.

influence scores but considering only highest scores except self influence still correctly determines the influence between species.

6 Conclusion

Considering the bottleneck of specifying in temporal logic with numerical constraints the biological properties of a system known from experiments, we have proposed an algorithm for inferring constraint-LTL formulae from numerical data time series. To this end, the finite time horizon model-checking algorithm described in [9] has been generalized to an instantiation algorithm in the existential fragment of LTL with numerical constraints over the reals. A strong completeness theorem stating that the ranges of real valued variables computed for a formula describe exactly the solution space, has been shown, together with
Table 6. Positive influence scores of all species on Cdc2 and Cdc2-Cyclin\(^{-p1,p2}\).

<table>
<thead>
<tr>
<th>Species</th>
<th>Cdc2</th>
<th>Cdc2-Cyclin(^{-p1,p2})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cdc2</td>
<td>0.59</td>
<td>0.00</td>
</tr>
<tr>
<td>Cdc2(^{-p1})</td>
<td>0.59</td>
<td>0.00</td>
</tr>
<tr>
<td>Cyclin</td>
<td>0.00</td>
<td>0.73</td>
</tr>
<tr>
<td>Cdc2-Cyclin(^{-p1,p2})</td>
<td>0.00</td>
<td>0.59</td>
</tr>
<tr>
<td>Cdc2-Cyclin(^{-p1})</td>
<td>0.49</td>
<td>0.00</td>
</tr>
<tr>
<td>Cyclin(^{-p1})</td>
<td>0.48</td>
<td>0.00</td>
</tr>
</tbody>
</table>

a complexity bound in \(n^k^2\) on the representation of the domain, where \(n\) is the number of time points and \(k\) the size of the formula.

For the purpose of evaluating the method, we worked with time series generated from models by simulation, and considered one experimental-like time series extracted from the simulation trace in few time points taken at regular intervals of time. In the near future, we plan to apply the method to the analysis of experimental temporal data of FSH signaling proteins for designing a model of FSH signal transduction together with its temporal specification, and proceed similarly with cell cycle and circadian cycle data for cancer chronotherapies in the framework of the EU project Tempo\(^1\).

References


\(^1\) http://www.chrono-tempo.org/