Abstraction of Biochemical Reaction Systems on Polytopes ⋆

Pieter J. Collins * **Luc Habets** * **Jan H. van Schuppen** * **Ivana Černá** **
Jana Fabriková ** **David Šafránek** **

* *CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands (e-mail: Pieter.Collins,Luc.Habets,J.H.van.Schuppen@cwi.nl)*
** *Faculty of Informatics, Masaryk University, Botanicka 68 a, 602 00 Brno, Czech Republic (e-mail: xsafran1@fi.muni.cz)*

Abstract: Analysis of the dynamic behavior of large-scale biochemical reaction systems can be facilitated by abstraction followed by model checking. A biochemical reaction system can be approximated by a multi-affine system or an affine system on a rectangle. Either of these systems can be abstracted to an automaton. Model checking can then be employed to determine whether the dynamic behavior of the automaton satisfies specific properties. A relation between the system and its abstraction is proved; it is an over-approximation: any discrete state trajectory of the abstraction of the continuous state trajectory is contained in the automaton, but the automaton may contain more behavior.

Keywords: Affine system, multi-affine system; piecewise-affine hybrid system; abstraction; polytope.

1. INTRODUCTION

In systems biology, biochemical reaction systems with large numbers of variables and complicated dynamic behaviors are studied. These systems may be modeled by differential equations, but the large dimension of the state space of these systems, and uncertainty about the exact values of several system parameters, makes it difficult to study properties of solution trajectories in an analytic way. In this paper we propose the use of existing techniques of computer science and mathematics that overcome this difficulty to some extent. The approach consists of first approximating a biochemical reaction system by a (piecewise-)affine or multi-affine system on rectangles. The next step is *abstraction*: we construct a discrete automaton such that the dynamic behavior of the (piecewise) multi-affine system on rectangles is approximated by the dynamic behavior of the automaton. Specific dynamic properties of the original system can then be related to dynamic properties of the discrete automaton. Examples of such properties include the existence of a unique- or of multiple steady state(s), the existence of periodic trajectories, or the existence of a specific route through the state set. On an automaton, such properties may be verified by employment of *model checking*. The objective is to use this constructive technique from computer science to explore the qualitative dynamic behavior of biochemical reaction systems, for which the direct mathematical analysis is too complicated.

The originators of the approach are de Jong and Gouzé, see de Jong et al. [2004b, 2003, 2004a], Ropers et al. [2008]. A related approach for multi-affine systems with inputs defined on rectangles was developed by Belta and co-workers, see Belta et al. [2002], Belta and Habets [2004, 2006], Habets et al. [2006b], Batt et al. [2007, 2008]. The approach of this paper is based on the control of piecewise-affine hybrid systems on polytopes developed in Habets et al. [2006a], Habets and van Schuppen [2004], Kloetzer and Belta [2008]. It differs from that of de Jong in that another approximation is determined. It is conjectured that the approach of this paper is easier to apply than that of the references mentioned above. An alternative approach to abstraction is provided in Azuma et al. [2008]. A comparison of the approach of this paper with other approaches of abstraction is provided at the end of Section 3.

The novelty of the paper is in the theorems on the relationship between the dynamic behavior of the continuous-space system and that of its abstracted automaton. This covers the case of affine systems on polytopes and of multi-affine systems on rectangles. Liveness of the abstraction is guaranteed by verifiable conditions on the continuous dynamics; a trajectory is guaranteed to leave a polytope in finite time, provided that these conditions are satisfied.

The paper is organized as follows. In the next section the concepts of discrete automata, piecewise-affine systems on polytopes, and abstraction are introduced, and illustrated with an example. The abstraction procedure and relations between the original system and its abstraction are discussed in Section 3. Model checking of automata as practiced in computer science is presented in Section 4, while a case study is presented in Section 5. Concluding remarks are stated in Section 6.

2. DISCRETE AUTOMATA, PIECEWISE-AFFINE SYSTEMS ON POLYTOPES, AND ABSTRACTION

In this section we provide a more formal description of the class of systems considered in this paper.

Definition 1. An *automaton* is a tuple $\mathcal{A} = (Q, E, f, Q_0)$, with Q the finite *state set*, E the finite *event set*, $Q_0 \subset Q$ the set of *initial states*, and $f : Q \times E \to Q$ the *transition function*. In case the transition function f is a partial function, i.e. f is not defined on the whole set $Q \times E$, but only on a strict subset

⋆ Research is supported in part by the European Commission via the Project *Scalable modeling and analysis techniques to study emergent cell behavior - Understanding the* E. coli *stress response* (EC-MOAN, contract number 043235).

of $Q \times E$, then one calls the automaton a *generator*. A *nondeterministic automaton* is defined as an automaton except that the transition function maps $f : Q \times E \to \text{Pwrset}(Q)$. A *nondeterministic generator* is defined analogously.

Definition 2. A finite or infinite sequence $(q_i)_{i=0}^{k}$ (with $k \in \mathbb{N} \cup \{0\}$ or $k = \infty$) is a *state trajectory* of the automaton $\mathcal{A}$ if $q_0 \in Q_0$, and there exists a sequence $(e_i)_{i=1}^{k}$ of events such that $q_i \in f(q_{i-1}, e_i)$ for all $i = 1, \ldots, k$.

In this paper we want to study the evolution of continuous-time systems by abstracting them to automata, in such a way that the trajectories of these automata provide information on qualitative properties of the continuous-time systems. Before we can specify the class of systems we are mainly interested in, we first introduce some additional terminology.

Definition 3. Let $X \subset \mathbb{R}^n$ be a closed full-dimensional polytope. A partitioning $X_{\text{part}}(X) = \{X_i \mid i = 1, \ldots, m\}$ of X is called *admissible* if (1) for all $i = 1, \ldots, m$: X_i is a closed full-dimensional polytope in $\mathbb{R}^n$, (2) $\cup_{i=1}^{m} X_i = X$, and (3) for all $i, j = 1, \ldots, m$, $i \neq j$, the intersection $X_i \cap X_j$ is either empty, or a lower-dimensional common face of X_i and X_j. If both polytope X and all subpolytopes X_i, $(i = 1, \ldots, m)$ are n-dimensional rectangles, then an admissible partitioning $X_{\text{part}}(X)$ is called *rectangular*.

A mapping $g : X \to \mathbb{R}^n$ is called *piecewise-affine* on $X_{\text{part}}(X)$ if (1a) g is continuous on X, and (2a) for all $i = 1, \ldots, m$ there exist $A_i \in \mathbb{R}^{n \times n}$ and $a_i \in \mathbb{R}^n$ such that for all $x \in X_i$: $g(x) = A_i x + a_i$, i.e. $g \mid_{X_i}$ is an affine mapping.
A mapping $g : X \to \mathbb{R}^n$ is called *multi-affine* on $X_{\text{part}}(X)$ if (1b) g is continuous on X, and (2b) for all $i = 1, \ldots, m$, $g \mid_{X_i}$ is multi-affine, i.e. $g \mid_{X_i}$ is affine w.r.t. every of its variables, while keeping all other variables constant.

Definition 4. A *piecewise-affine system on a polytope* is a tuple

$$\Sigma = (X, X_{\text{part}}(X), x_0, t_0, g),$$

where state set X is a full-dimensional polytope in $\mathbb{R}^n$, $X_{\text{part}}(X)$ is an admissible partitioning of X, $x_0 \in X$ is the initial continuous state, $t_0 \in \mathbb{R}$ is the initial time, and $g : X \to \mathbb{R}^n$ is a piecewise-affine function on $X_{\text{part}}(X)$. A trajectory $x : [t_0, t_1] \to X$ of system Σ is a solution of the differential equation

$$\dot{x}(t) = g(x(t)), \qquad x(t_0) = x_0, \tag{1}$$

where t_1 is either the time instant that the trajectory leaves state polytope X, or $t_1 = \infty$, if trajectory $x(t)$ remains in X forever.

If X is an n-dimensional rectangle, $X_{\text{part}}(X)$ is a rectangular partitioning of X, and $g : X \to \mathbb{R}^n$ is multi-affine on $X_{\text{part}}(X)$, then $\Sigma = (X, X_{\text{part}}(X), x_0, t_0, g)$ is called a *multi-affine system on rectangles*.

One may distinguish piecewise-affine systems with the same affine dynamics on all polytopes in the partitioning, and systems with different affine dynamics on each subpolytope. In the second case, the dynamics on the boundary of two polytopes is still assumed to be continuous. It is possible to drop this continuity condition, but then one needs a different solution concept, the so-called Filipov solutions. This type of systems (and solutions) is not considered in this paper.

Let Q be a finite set of the same cardinality as $X_{\text{part}}(X)$, and let $\pi : X_{\text{part}}(X) \to Q$ be an invertible mapping. Then every subpolytope X_i corresponds to exactly one discrete state $q \in Q$. Similarly, a continuous solution $x : [t_0, t_1] \to X$ of (1) corresponds to a finite or infinite sequence over Q; if x passes subsequently through the subpolytopes $X_1, \ldots, X_k$, then $(\pi(X_i))_{i=1}^{k}$ is the corresponding sequence over Q. This sequence may also be interpreted as a state trajectory of an automaton.

Definition 5. An automaton $\mathcal{A} = (Q, E, f, Q_0)$ is an *abstraction* of piecewise-affine system Σ if every solution trajectory of (1) corresponds to a state trajectory of $\mathcal{A}$.

The purpose of this paper is to present a procedure for the construction of an automaton that is an abstraction of a given piecewise-affine system. Before we describe this procedure in detail, we first illustrate the idea with a simple example.

Example 6. Consider the affine system Σ on rectangle $[0, 2] \times [0, 2]$ given by

$$\dot{x}(t) = \begin{pmatrix} -4 & 0 \\ 0 & -5 \end{pmatrix} x(t) + \begin{pmatrix} 6.8 \\ 6.5 \end{pmatrix}, \; x(t_0) = x_0.$$

Obviously, $(1.7, 1.3)^T$ is the unique steady state of this system. We partition the state set X into four squares:

$$X_{(0,0)} = [0,1] \times [0,1],\; X_{(1,0)} = [1,2] \times [0,1],$$
$$X_{(0,1)} = [0,1] \times [1,2],\; X_{(1,1)} = [1,2] \times [1,2].$$

Using the differential equation, the direction vectors of the dynamics in all vertices of the subpolytopes are easily obtained (see Figure 1). These direction vectors indicate whether it is possible to go from one rectangle to another by crossing a common facet. In particular, if in all vertices of a facet the inner product of the direction vector with the normal vector of the facet has the same sign, then this holds true in *all* points of the facet because the dynamics in each rectangle is affine. This implies that the facet can only be crossed in one direction.

To construct an abstraction, define the discrete state set of the automaton by $Q = \{q_{(0,0)}, q_{(1,0)}, q_{(0,1)}, q_{(1,1)}\}$ and the event set by $E = \{e_{(0,0,1,0)}, e_{(0,0,0,1)}, e_{(1,0,1,1)}, e_{(0,1,1,1)}, e_{(0,1,0,0)}\}$. Let $\pi : X_{\text{part}}(X) \to Q$ be defined by $\pi(X_{(i,j)}) = q_{(i,j)}$, $i, j \in \{0, 1\}$. In the automaton an event $e_{(i,j,k,l)}$ occurs if a continuous state trajectory $x(t)$ of Σ crosses the joint facet of $X_{(i,j)}$ and $X_{(k,l)}$ from $X_{(i,j)}$ to $X_{(k,l)}$, enforcing the transition $q_{(i,j)} \to q_{(k,l)}$ Note that not every event $e_{(i,j,k,l)}$, $(i, j, k, l \in \{0, 1\})$ exists because $X_{(i,j)}$ and $X_{(k,l)}$ need to have a common facet, and because most facets can only be crossed in one direction. Now we define the abstraction of system Σ by the automaton $\mathcal{A} = (Q, E, f, Q_0)$, where the transition function is obtained by inspection of the direction vectors of system Σ in the vertices of the subpolytopes (see Figure 1):

$$f(q_{(0,0)}, e_{(0,0,1,0)}) = q_{(1,0)} \;\; f(q_{(1,0)}, e_{(1,0,1,1)}) = q_{(1,1)}$$
$$f(q_{(0,0)}, e_{(0,0,0,1)}) = q_{(0,1)} \;\; f(q_{(0,1)}, e_{(0,1,1,1)}) = q_{(1,1)}$$
$$f(q_{(0,1)}, e_{(0,1,0,0)}) = q_{(0,0)}$$

Note that any state trajectory of automaton $\mathcal{A}$ that reaches $q_{(1,1)}$ terminates, because in $q_{(1,1)}$ no events can occur. This is due to the fact that a continuous state trajectory of system Σ cannot leave $X_{(1,1)}$ once it has entered $X_{(1,1)}$; indeed the direction vectors at the vertices of $X_{(1,1)}$ indicate that it is only possible to cross the facets of $X_{(1,1)}$ in the inward direction. Hence, the fact that the affine system Σ has a fixed point in atom polytope $X_{(1,1)}$ is also visible from the behavior of automaton $\mathcal{A}$.

In Example 6 the differential equation is easy to analyze, so abstraction is not really needed. The motivation for this study is the need to analyze very large sets of differential equations, for example the dynamic behavior of genetic, signal, and metabolic

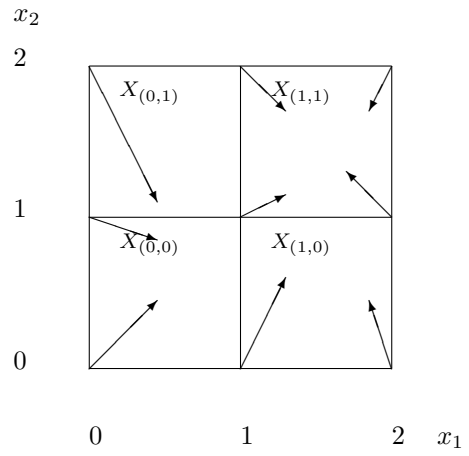

Fig. 1. Partitioned state set and vectors of affine system.

networks in systems biology. The differential equations of these networks are often polynomial or rational, and may be described or approximated by piecewise-affine or multi-affine systems. Furthermore, the parameter values of these systems are often not known exactly, but may range over a polytope. An exact analysis of this type of systems seems impossible, but still one wants to answer questions on existence and possible locations of steady states, existence of periodic solutions, or the solvability of specific reach-avoid problems. For this purpose, abstraction to an automaton may be helpful. If the system property one is interested in is also reflected in the discrete behavior of the abstraction, one may try to verify whether the abstraction satisfies the required condition. For this purpose *model checking* may be applied: the area in computer science that determines properties of automata by computation.

3. THE ABSTRACTION PROCEDURE

In this section the intuitive ideas on abstraction of a piecewise-affine system Σ to an automaton $\mathcal{A}$, as illustrated in Example 6, are formalized. The following questions will be discussed: (1) How to determine the so-called exit facets, i.e. the facets of a subpolytope that are crossed by a continuous state trajectory of system Σ? (2) To which other subpolytope does the state trajectory proceed after exiting? (3) Does the state trajectory leave the subpolytope in finite time? (4) How to formulate the relation between the dynamic behavior of the system and that of the automaton?

3.1 Preliminaries of the procedure

How to determine exit facets? Let $\Sigma = (X, X_{\text{part}}(X), x_0, t_0, g)$ be a piecewise-affine system on a polytope X. A facet F of subpolytope X_i is called an *exit facet* if there exists a trajectory of system Σ, starting in X_i, that attempts to leave X_i in finite time by crossing facet F. Let n_F denote the normal vector of F, pointing out of subpolytope X_i, and let the affine dynamics on X_i be described by $\dot{x} = A_i x + a_i$. Then F is an exit facet if and only if there exists $\hat{x} \in F$ such that

$$n_F^T(A_i\hat{x} + a_i) > 0. \tag{2}$$

Since the dynamics $\dot{x} = A_i x + a_i$ is affine, it suffices to check condition (2) on $\mathcal{V}(F)$, i.e. on the set of all vertices of facet F.

Lemma 7. Facet F of subpolytope X_i is an exit facet w.r.t. the dynamics $\dot{x} = A_i x + a_i$ on X_i if and only if

$$\exists v \in \mathcal{V}(F) : \; n_F^T(A_i v + a_i) > 0.$$

To which subpolytope does the state trajectory proceed after exiting? If a state trajectory leaves subpolytope X_i through a point in the relative interior of an exit facet, but stays in polytope X, then the trajectory proceeds to an adjacent subpolytope X_j. The situation is more complicated if the exit point is in a lower dimensional face of X_i, because this face may belong to more than one adjacent subpolytope. Keeping track of all possibilities makes the abstraction procedure extremely complex. Instead we only allow transitions to adjacent subpolytopes, that are also reachable via the relative interior of the joint facet. Exit through a lower dimensional face is then modeled by the subsequent execution of several transitions at the same time instant. E.g. in Figure 1, the continuous state trajectory that leaves subpolytope $X_{(0,0)}$ through the north-east vertex, and enters $X_{(1,1)}$, is modeled by one of the transition sequences

$$X_{(0,0)} \longrightarrow X_{(1,0)} \longrightarrow X_{(1,1)},$$
$$X_{(0,0)} \longrightarrow X_{(0,1)} \longrightarrow X_{(1,1)}.$$

Does the state trajectory leave a subpolytope in finite time? In the abstraction, an event will occur if the trajectory of the corresponding continuous-time system crosses the joint facet of two subpolytopes. So, in order to guarantee that a new event occurs after finite time, one has to verify whether the continuous state leaves a subpolytope in finite time.

Theorem 8. [Habets et al., 2006a, Th. 3.1] Consider an affine system $\dot{x}(t) = A_i x(t) + a_i$ on a closed full-dimensional subpolytope $X_i \subset \mathbb{R}^n$. There exists an initial state $x_0 \in X_i$ such that for all times $t \in T = [t_0, \infty)$ the state trajectory belongs to the subpolytope, i.e. $x(t; t_0, x_0) \in X_i$ if and only if there exists a fixed state in subpolytope X_i.

The existence of a fixed state can be checked by computing it numerically. Alternatively, one may use the direction vectors at the vertices of the polytope:

Theorem 9. Consider an affine system $\dot{x}(t) = A_i x(t) + a_i$ on a closed full-dimensional subpolytope $X_i \subset \mathbb{R}^n$. There exists an $\hat{x} \in X_i$ such that $A_i\hat{x} + a_i = 0$ if and only if

$$0 \in \text{ConvexHull}(\{A_i v + a_i \mid v \in \mathcal{V}(X_i)\}), \tag{3}$$

i.e. if and only if the zero vector is a convex combination of the direction vectors at the vertices. The latter condition can be verified by checking the solvability of a finite set of linear equations.

Theorem 9 is based on the observation that if the dynamics are affine, then $A_i X_i + a_i = \text{ConvexHull}(\{A_i v + a_i \mid v \in \mathcal{V}(X_i)\})$. Condition (3) has the important advantage that it is also applicable if the dynamics are not known exactly, because some system parameters may range over a set of values in a polytope.

One may now distinguish three different cases:

(1) Subpolytope X_i contains a fixed point, and at all vertices of X_i, the direction vector of the differential equation is pointing inward. In this case all trajectories that enter subpolytope X_i will remain in X_i forever. In the abstraction no real switching event will occur.
(2) Subpolytope X_i does not contain a fixed point. Then all trajectories that enter X_i leave X_i in finite time (negation of Theorem 8).
(3) Subpolytope X_i contains a fixed point, and there exists a vertex of X_i where the direction vector of the differential equation is pointing out of X_i. In this case there exist trajectories that start in X_i and remain in X_i forever, and

other trajectories that start in X_i but leave X_i in finite time.

In the abstraction, case (1) corresponds to a discrete state in which no events can occur, and case (2) to a discrete state in which an event must occur in finite time. The direction vectors at the vertices determine which events are possible. Case (3) is a combination of (1) and (2): either no event occurs, because the continuous state stays in the subpolytope forever, or an admissible event occurs, because the continuous state leaves the polytope in finite time. Since the behaviors in cases (2) and (3) are conceptually different, the abstraction automaton has to reflect the distinction between case (2) and case (3). For this purpose, we define for every subpolytope X_i that contains a fixed point, two corresponding discrete states: the ordinary state $q_i = \pi(X_i)$ and the terminal state $q_{i,\tau}$. We also add an empty event ϵ to the event set E. Finally, transition function f is extended with transition $f(q_i, \epsilon) = q_{i,\tau}$ and self-transition $f(q_{i,\tau}, \epsilon) = q_{i,\tau}$.

With this extension, the cases (1), (2), and (3) can still be recognized from the abstraction. In case (1) the state $q_i = \pi(X_i)$ allows for only the empty event ϵ, which transfers the state to $q_{i,\tau}$. In this state only self-transitions are possible. In case (2) there are no changes with respect to the original situation as described in Example 6. In case (3) the state q_i allows for more than one event. One of these events is ϵ and transfers the state to $q_{i,\tau}$, where again only self-transitions occur. Note that it remains visible from the event whether a transition is a real transition from one polytope to the other, or an artificial (self)-transition (empty event ϵ).

3.2 Procedure of abstraction

After this modification of the abstraction, all discrete state trajectories that correspond to continuous state trajectories that remain in X forever, are infinite sequences. Finite discrete state trajectories only occur if after a finite number of transitions the continuous state leaves the global polytope X through one of its exit facets. Since the model checking techniques that will be described in Section 4 are only applicable if all discrete state trajectories of the abstraction automaton are infinite, another modification of the abstraction is needed. First, an additional discrete state s is introduced, and transition function f is extended with the self-transition $f(s, \epsilon) = s$. Discrete state s represents the case that the continuous state trajectory is only defined on a finite time interval. For every subpolytope X_i with an exit facet F that is contained in a facet of global polytope X, we add a new event η_i to the event set, representing that a transition to s is possible. Simultaneously, transition function f is extended with the transition $f(q_i, \eta_i) = s$. This leads to the following algorithm for the construction of an abstraction:

Algorithm 1. Let $\Sigma = (X, X_{\text{part}}(X), x_0, t_0, g)$ be a piecewise-affine system on $X_{\text{part}}(X)$, and denote the number of subpolytopes in $X_{\text{part}}(X)$ by N. Construct the generator $\mathcal{A} = (Q, E, f, Q_0)$ in the following way.

(1) Define $Q := \{q_1, \ldots, q_N\} \cup \{s\}$ and $\pi : X_{\text{part}}(X) \to Q$ by $\pi(X_i) = q_i$, $(i = 1, \ldots, N)$. Set $E := \{\epsilon\}$.
(2) For all $i = 1, \ldots, N$: Check whether there exists $\hat{x} \in X_i$ such that $g(\hat{x}) = 0$. Use Theorem 8 or 9 for this. If X_i contains a zero of g, then $Q := Q \cup \{q_{i,\tau}\}$.
(3) For all $i \in \{1, \ldots, N\}$: let $\mathcal{F}(X_i)$ be the set of facets of X_i. For every $F \in \mathcal{F}(X_i)$, the normal vector of facet F pointing out of X_i is denoted by n_F.
For all $F \in \mathcal{F}(X_i)$, for which there exists a vertex $v \in \mathcal{V}(F)$ such that $n_F^T g(v) > 0$ do:
- if F is the common facet of subpolytopes X_i and X_j, then $E := E \cup \{e_{i,j}\}$,
- if F is not the common facet of two subpolytopes, then $E := E \cup \{\eta_i\}$.

(4) For state set Q as constructed in (1) and (2), and event set E as constructed in (1) and (3), define the transition function $f : Q \times E \longrightarrow \text{Pwrset}(Q)$ in the following way. For $i = 1, \ldots, N$ and $e \in E$:

$$f(q_i, e) := \begin{cases} \{q_k\} \text{ if } e = e_{i,k} \text{ for some } k \neq i, \\ \varnothing \text{ if } e = \epsilon \text{ and } q_{i,\tau} \notin Q, \\ \{q_{i,\tau}\} \text{ if } e = \epsilon \text{ and } q_{i,\tau} \in Q, \\ \{s\} \text{ if } e = \eta_i, \\ \varnothing \text{ if } e \in E \backslash (\{e_{i,k} \mid k \neq i\} \cup \{\epsilon, \eta_i\}) \end{cases}$$

For $i = 1, \ldots, N$ such that $q_{i,\tau} \in Q$ and $e \in E$:

$$f(q_{i,\tau}, e) := \begin{cases} \varnothing \text{ if } e \neq \epsilon, \\ \{q_{i,\tau}\} \text{ if } e = \epsilon. \end{cases}$$

Finally, for all $e \in E$:

$$f(s, e) := \begin{cases} \varnothing \text{ if } e \neq \epsilon, \\ \{s\} \text{ if } e = \epsilon. \end{cases}$$

(5) Set $Q_0 = \{\pi(X_i)\}$ where X_i is a subpolytope such that $x_0 \in X_i$ and $x(t; t_0, x_0) \in X_i$ for all $t \in (t_0, t_0 + \varepsilon)$, for some $\varepsilon > 0$. If x_0 is not fixed, Q_0 consists of all discrete states that correspond to a subpolytope, in which a continuous trajectory starts.
(6) Output: the generator $\mathcal{A} = (Q, E, f, Q_0)$.

Remark 10. In the definition of transition function f in (4) it is required that $e \in E$. The definition is correct because $e_{i,j} \in E$ if and only if there exists a continuous trajectory that moves from subpolytope X_i to X_j by crossing their joint facet.

Remark 11. For general convex polytopes X and arbitrary partitions $X_{\text{part}}(X)$ the execution of Algorithm 1 will be difficult in general, because of the geometric complexity of the partition, that needs to be unraveled in steps (2) and (3). This problem does not occur if X is a multi-dimensional rectangle, that is partitioned in smaller rectangles of the same dimension. Using a suitable encoding of the atom rectangles, the search for vertices of facets and adjacent rectangles becomes trivial. Therefore the results in this paper are in particular of interest for the class of piecewise-affine systems on multi-dimensional *rectangles*.

As a consequence of the construction of abstraction $\mathcal{A}$ in Algorithm 1, the definition of the discrete state trajectory, corresponding to a given continuous state trajectory of a piecewise-affine system, has to be modified.

Definition 12. Let $\Sigma = (X, X_{\text{part}}(X), x_0, t_0, g)$ be a piecewise affine system on $X_{\text{part}}(X)$, and let $x : [t_0, t_1] \to X$ be a solution of (1) with t_1 either finite or $t_1 = \infty$. Suppose that x subsequently passes through subpolytopes $X_1, \ldots, X_k \in X_{\text{part}}(X)$, with $k \in \mathbb{N} \cup \{0\}$ or $k = \infty$, where we use the definition of proceeding to a new subpolytope as explained in Subsection 3.1. Then the discrete state trajectory $\rho(x) \in Q^\omega$ corresponding to x is the infinite sequence defined by:
If $k = \infty$ then

$$\rho(x) = (\pi(X_j))_{j=1}^{\infty}.$$

If k is finite, and $q_{k,\tau}$ denotes the terminal state corresponding to $\pi(X_k)$ then the sequence $\rho(x)$ is given by

$$(\rho(x))_j = \begin{cases} \pi(X_j) \text{ if } j \in \{1, \ldots, k\}, \\ q_{k,\tau} \text{ if } j > k \text{ and } t_1 = \infty, \\ s \text{ if } j > k \text{ and } t_1 < \infty. \end{cases}$$

3.3 The relation between a system and its abstraction

Definition 13. Let $\Sigma = (X, X_{\text{part}}(X), x_0, t_0, g)$ be a piecewise affine system on $X_{\text{part}}(X)$, and consider the associated automaton abstraction $\mathcal{A} = (Q, E, f, Q_0)$ as defined in Algorithm 1.

(a) *Global sufficiency* of the relation between the piecewise-affine system Σ and the associated automaton $\mathcal{A}$ is said to hold if for any initial state $x_0 \in X$ the continuous trajectory $x = x(t; t_0, x_0)$ of Σ corresponds to a discrete state trajectory of automaton $\mathcal{A}$, i.e. $\rho(x)$, as defined in Definition 12, is a state trajectory of $\mathcal{A}$

(b) *Global necessity* of the relation between the piecewise-affine system Σ and the associated automaton $\mathcal{A}$ is said to hold if for every discrete state trajectory d of automaton $\mathcal{A}$ there exists an initial state $x_0 \in X$ such that the continuous trajectory $x = x(t; t_0, x_0)$ of Σ corresponds to discrete state trajectory d, i.e. $\rho(x) = d$.

Theorem 14. Let $\Sigma = (X, X_{\text{part}}(X), x_0, t_0, g)$ be a piecewise-affine system on $X_{\text{part}}(X)$, and let $\mathcal{A} = (Q, E, f, Q_0)$ be its associated automaton abstraction as defined in Algorithm 1.

(a) The relation between the piecewise-affine system Σ and the corresponding automaton $\mathcal{A}$ is globally sufficient.

(b) There exist piecewise-affine systems Σ for which the relation with its associated abstracted automaton $\mathcal{A}$ is *not* globally necessary.

The abstraction from the piecewise-affine system to the automaton is an approximation in that the automaton may have trajectories for which there does not exist a state trajectory of the piecewise-affine system. This is described above as that global necessity of the abstraction does not hold.

For several analysis problems, such as reachability with uncertain initial conditions, abstraction is more effective and efficient than numerical simulation or mathematical analysis. The complexity of the abstraction procedure remains to be investigated.

Novel aspects of this paper include the abstraction procedure with terminal discrete states representing the case that a continuous trajectory does not leave a subpolytope in finite time. Theorems 8 and 9 provide necessary and sufficient conditions for this, which considerably reduces the over-approximation. This is an advantage in comparison with earlier approaches. Other contributions are the observation of the particular role of rectangular partitions to avoid computational complexity, and the exact mathematical formulation of the results.

3.4 Extension to multi-affine systems

The abstraction procedure described in Algorithm 1 may be extended to multi-affine systems (cp. Belta et al. [2002], Belta and Habets [2006]). For biochemical systems this is of particular interest, because several models of these systems, e.g. of some genetic networks, are multi-affine systems. However, the dynamic behavior of the abstraction automaton is in general larger than in the piecewise-affine case. Therefore, model checking of these automata leads to more conservative results. The reason for this decrease in performance is the fact that Theorems 8 and 9 do -to the knowledge of the authors- not extend to the multi-affine case. Instead one may prove the following:

Theorem 15. Let $\dot{x}(t) = g(x(t))$ be a multi-affine system on an n-dimensional rectangle $R_i \subset \mathbb{R}^n$. If there exists a vector $w \in \mathbb{R}^n$ such that for all vertices $v \in \mathcal{V}(R_i)$ we have $w^T g(v) > 0$, then all state trajectories of this system leave rectangle R_i in finite time.

Note that the sufficient condition in Theorem 15 can be verified by solving a set of linear inequalities in the unknown vector w. However, if the condition is not satisfied, it is unclear whether all solutions leave rectangle R_i in finite time, and one has to take the possibility of a trajectory staying in R_i forever into account. In the abstraction, this may be the source of unrealistic transitions to terminal states.

In the multi-affine case, Theorem 14 remains valid. The abstraction is still globally sufficient, but not globally necessary. The larger behavior of the abstraction will lead to more discrete trajectories in the automaton, that do not correspond to trajectories of the real multi-affine system.

3.5 Comparison with the literature

The discrete abstractions computed in this paper are for piecewise-affine systems on a given polyhedral partition or subdivision. By using techniques based on interval analysis, the underlying method can, in principle, be extended to *nonlinear systems*. Abstractions of nonlinear systems based on interval analysis have been computed in Stursberg et al. [1997] and Ratschan and She [2007]. Although they are guaranteed to give over-approximations to the discrete dynamics of the continuous system (sufficiency), the quality of the approximation can be (and typically is) rather poor. Even refining the partition may not improve the quality of the approximation. Only when the system dynamics is related to the structure of the state-space decomposition, as in de Jong et al. [2003], good results may be expected. In practice, in order to obtain usable abstractions, more advanced methods must be used.

Closest to the approach presented here is that in Boczko et al. [2007]. Rather than use a fixed polyhedral subdivision, a simplicial subdivision is computed based on the system itself, and the resulting *flow induced* multivalued map provides a good abstraction of the continuous system. The method can be used for both nonlinear and piecewise-affine systems. However, a good polyhedral subdivision itself is extremely difficult to compute, and the method is likely to be useful only in low dimensions.

An alternative approach is to use a fixed polyhedral subdivision (typically rectangular), and to integrate the continuous dynamics for a fixed time step h (see e.g. Asarin et al. [2000], Grüne [2002] and the tool Ariadne (Balluchi et al. [2006])). This method is very good for smooth nonlinear systems, but may have problems handling piecewise-smooth systems. A hybrid strategy, comprised of piecing together continuous trajectories to obtain a so-called sub-shift of finite type for the discrete abstraction, may be more appropriate for piecewise-smooth systems.

4. MODEL CHECKING

In the field of formal verification of SW/HW systems, model checking refers to the problem of automatically verifying whether a simplified model of a system meets a given specification. A specification is stated by means of a temporal logic formula. Models are considered as finite-state automata (or generators) on infinite strings – the so-called Kripke structures.

For a biochemical reaction system modeled as a system of ODEs, temporal logics provide a formalism which can suffi-

ciently express dynamic properties of reaction kinetics, such as all kinds of stability (including non-hyperbolic equilibria as well as multi-stability), oscillations, temporal ordering of certain concentration levels, causality, etc. Rizk et al. [2008], Monteiro et al. [2008]. We employ linear temporal logic (LTL) interpreted on infinite paths in automata resulting from the abstraction procedure of Section 3. In this setting, model checking can be used in two basic ways:

(1) to automatically detect presence of particular dynamics phenomena in the system
(2) to verify correctness of the model (i.e., checking whether some undesired property is exactly avoided)

4.1 Turning an abstraction automaton into a Kripke structure

In order to reason about temporal properties, individual states of the abstraction automaton have to be assigned local (static) properties in terms of propositions. As these propositions make the atoms for temporal formulae construction, they are called atomic propositions. Since any discrete state represents a polytope or a rectangle in $\mathbb{R}^n$, it is required that each atomic proposition has the same validity for all points in the respective set. The *set of atomic propositions* is denoted AP.

Definition 16. Let $A = (Q, E, f, Q_0)$ be a (non-deterministic) automaton, as constructed in Alg. 1. The *Kripke structure of* A is a tuple $\mathcal{K}(A) = (Q, f', Q_0, L)$, with $f' : Q \to 2^Q$ given by

$$f'(q) = \bigcup_{e \in E} f(q, e),$$

and $L : Q \to 2^{AP}$ a labeling function, that assigns to every $q \in Q$ the set $L(q)$ of atomic propositions valid in q.
A path π^{q_0} of $\mathcal{K}(A)$ is defined as an infinite sequence $\pi^{q_0} = q_0 q_1 \cdots$ such that $q_0 \in Q_0$ and $q_{i+1} \in f'(q_i)$ for all $i \in \mathbb{N}$.

Validity of an LTL property φ is traditionally defined on infinite paths in terms of the satisfaction relation $\pi^{q_0} \models \varphi$ (formal definition of basic LTL syntax and semantics is available e.g. in Clarke et al. [2000]). LTL logic is interpreted universally on automata provided that a formula φ is satisfied by the automaton A, written $A \models \varphi$, only if for all $q \in Q_0$ and all paths π^q in $\mathcal{K}(A)$ it holds that $\pi^q \models \varphi$. The general *model checking problem* is then formulated as follows:

Problem 17. For a given abstraction automaton A with Q_0 the set of initial states and a given LTL formula φ decide whether $A \models \varphi$. In the negative case, return some path π^q in $\mathcal{K}(A)$ where $q \in Q_0$ such that $\pi^q \not\models \varphi$ – a so-called counterexample.

4.2 Interpreting model checking results on original systems

LTL can be directly interpreted on trajectories of dynamic systems (see e.g. Rizk et al. [2008] for definition of the semantics). Given a dynamic system S with a particular initial state, we can say that S satisfies a formula φ, written $S \models \varphi$, only if the trajectory starting at the initial state satisfies φ. The following theorem characterizes the relation between validity of φ in the abstraction and in the original system.

Theorem 18. Consider a dynamic system S of any class and the associated abstraction automaton A.

(1) If global necessity holds then $S \models \varphi$ implies $A \models \varphi$.
(2) If global sufficiency holds then $A \models \varphi$ implies $S \models \varphi$.

If for affine systems on polytopes and multi-affine systems on rectangles model checking of a property on the corresponding automaton yields true, then the original system also satisfies this property. However, when the result is negative, the returned counterexample may have no corresponding trajectory in the original system. The main point is the characterization of necessary and sufficient conditions for leaving a polytope in finite time. For affine systems the results of this paper enable sufficient model checking of any LTL property on abstracted dynamic systems, including liveness properties which rely on time progress of the system. However, for multi-affine systems the framework is practically limited since conditions for finite-time leaving of a rectangle are only necessary thus imposing another source of over-approximation. Safety properties expressing non-reachability of a convex subset of the state space from the given initial region can be still sufficiently checked. Problems of liveness checking are studied in Batt et al. [2007].

By employing the results discussed in this paper, we have developed a prototype tool BioDiVinE (Barnat, Brim and Šafránek [2010]) that implements parallel LTL model checking algorithms for rectangular abstractions of dynamic systems. The tool currently supports affine and multi-affine dynamic systems on rectangles.

$\mathtt{AmtB} + \mathtt{NH_4ex}^* \overset{k_2}{\underset{k_1}{\leftrightarrows}} \mathtt{AmtB\text{-}NH_4}$	$k_1 = 5 \cdot 10^8, k_2 = 5 \cdot 10^3$
$\mathtt{AmtB\text{-}NH_4} \overset{k_3}{\to} \mathtt{AmtB\text{-}NH_3} + \mathtt{H}^*_{\mathtt{ex}}$	$k_3 = 50$
$\mathtt{AmtB\text{-}NH_3} \overset{k_4}{\to} \mathtt{AmtB} + \mathtt{NH_3in}$	$k_4 = 50$
$\mathtt{NH_4in} \overset{k_5}{\to}$	$k_5 = 80$
$\mathtt{NH_3in} + \mathtt{H}^*_{\mathtt{in}} \overset{k_7}{\underset{k_6}{\leftrightarrows}} \mathtt{NH_4in}$	$k_6 = 1 \cdot 10^{15}, k_7 = 5.62 \cdot 10^5$
$\mathtt{NH_3ex}^* \overset{k_8}{\underset{k_9}{\leftrightarrows}} \mathtt{NH_3in}$	$k_8 = k_9 = 1.4 \cdot 10^4$

Table 1. $\mathtt{NH_3}$ transport biochemical model

$$
\begin{aligned}
d\mathtt{AmtB}/dt &= -k_1 \cdot \mathtt{AmtB} \cdot \mathtt{NH_4ex} + k_2 \cdot \mathtt{AmtB\text{-}NH_4} + k_4 \cdot \mathtt{AmtB\text{-}NH_3} \\
d\mathtt{AmtB\text{-}NH_3}/dt &= k_3 \cdot \mathtt{AmtB\text{-}NH_4} - k_4 \cdot \mathtt{AmtB\text{-}NH_3} \\
d\mathtt{AmtB\text{-}NH_4}/dt &= k_1 \cdot \mathtt{AmtB} \cdot \mathtt{NH_4ex} - k_2 \cdot \mathtt{AmtB\text{-}NH_4} - k_3 \cdot \mathtt{AmtB\text{-}NH_4} \\
d\mathtt{NH_3in}/dt &= k_4 \cdot \mathtt{AmtB\text{-}NH_3} - k_7 \cdot \mathtt{NH_3in} + k_6 \cdot \mathtt{NH_4in} \\
d\mathtt{NH_4in}/dt &= k_5 \cdot \mathtt{NH_4in} + k_7 \cdot \mathtt{NH_3in} \cdot \mathtt{H_{in}} - k_6 \cdot \mathtt{NH_4in}
\end{aligned}
$$

Table 2. $\mathtt{NH_3}$ transport mathematical model

5. CASE STUDY

We have conducted model checking analysis of a biochemical system representing a module of the *E. coli* metabolic network defined in Ma et al. [2009]. In particular, we considered reactions involved in ammonium transport into the cell (Table 1). Due to reactions with two substrates, the resulting model is a multi-affine system (Table 2).

The model represents a pathway providing two parallel channels for ammonium exchange – direct diffusion through the membrane (reactions k_8, k_9) and an ion-channel mechanism (reactions $k_1, ..., k_7$) employed under low external ammonium conditions. Species marked by asterisk were considered constant, thus the continuous state set is chosen to be in $\mathbb{R}^5$. We analyzed several reachability properties by testing the maximal achievable level of internal ammonium forms w.r.t. a given (biologically correct) range of external ammonium concentrations. Details on the model setting and individual analysis tasks are provided in Barnat et al. [2009].

Based on expected concentration values, first a threshold is set, bounding the domain of each variable. The internal thresholds defining the abstraction partition are initially set to a single intermediate concentration level, to be chosen freely. Additional thresholds are identified by atomic propositions for particular

reachability properties. Finally, a single iteration of nullcline-driven partition refinement (Kloetzer and Belta [2010]) is employed, resulting into a partition with $3.9 \cdot 10^4$ rectangles. We considered properties of the form $\mathbf{G}(\mathtt{NH_4in} < \alpha)$ and $\mathbf{G}(\mathtt{NH_3in} < \alpha)$ to ensure that levels above α are not reachable. By a sequence of model checking tasks we identified an upper approximation for maximal reachable level α. However, due to over-approximation the result was quite far from data provided by biologists. Our main result was the proof that an increase in $\mathtt{NH_3in}$ is, up-to certain concentrations, not affected by the levels of external ammonia. This confirms the advantage of the approach presented in this paper, because such a global property cannot be tested sufficiently by simulation.

Parallel LTL model checking algorithms employed for the analysis allowed us to obtain results in several seconds on 36 cores. The problem is the over-conservative character of the abstraction. As discussed in Kloetzer and Belta [2010], a potential solution is nullcline-driven refinement. However, this iterative procedure may increase the size of the state set exponentially with the model dimension, even in a single iteration. In our case, a single threshold refinement iteration has been applied.

In Barnat et al. [2010], we have successfully employed the abstraction described in this paper for property-driven estimation of kinetic parameters in the ammonium transport model. For a review of other experiments the reader is referred to Barnat, Brim and Šafránek [2010].

6. CONCLUDING REMARKS

The paper presents an abstraction procedure for affine systems on polytopes and multi-affine systems on rectangles. Arbitrary biochemical reaction system are often rational positive systems. Abstraction for such systems has not yet been treated mathematically. Also a more detailed analysis of the relation between an arbitrary biochemical reaction system and its approximation is needed. These problems remain to be investigated.

REFERENCES

E. Asarin, O. Bournez, T. Dang, and O. Maler. Approximate reachability analysis of piecewise-linear dynamical systems. In *Hybrid Systems: Computation and Control*, vol. 1790 of *Lecture Notes in Computer Science*, pages 20–31. Springer-Verlag, Berlin, 2000.

S. Azuma, E. Yanagisawa, and J. Imura. Controllability analysis of biosystems based on piecewise-affine systems approach. *IEEE Transactions on Automatic Control and IEEE Transactions on Circuits and Systems*, pages 139–152, 2008.

A. Balluchi, A. Casagrande, P. Collins, A. Ferrari, T. Villa, and A.L. Sangiovanni-Vincentelli. Ariadne: a framework for reachability analysis of hybrid automata. In *Proc. 17th Int. Symposium on Mathematical Theory of Networks and Systems (MTNS 2006)*, pages 1259–1267, Kyoto, 2006.

J. Barnat, L. Brim, I. Černá, S. Dražan, J. Fabriková, J. Láník, D. Šafránek, and H. Ma. BioDiVinE: A Framework for Parallel Analysis of Biological Models. In *Proc. 2nd Int. Workshop on Computational Models for Cell Processes (COMPMOD 2009)*, pages 31–45, EPTCS 6, 2009.

J. Barnat, L. Brim and D. Šafránek. High-Performance Analysis of Biological Systems Dynamics with the DiVinE Model Checker. *Briefings in Bioinformatics* 11(3):301–312, 2010.

J. Barnat, L. Brim, D. Šafránek, and M. Vejnar. Parameter Scanning by Parallel Model Checking with Applications in Systems Biology. In *Proc. PDMC/HIBI 2010*, pages 95–104, IEEE Computer Society, 2010.

G. Batt, B. Yordanov, R. Weiss, and C. Belta. Robustness analysis and tuning of synthetic gene networks. *Bioinformatics* 23:2415–2422, 2007.

G. Batt, C. Belta, and R. Weiss. Temporal logic analysis of gene networks under parameter uncertainty. *IEEE Transactions on Automatic Control and IEEE Transactions on Circuits and Systems*, pages 215–229, 2008.

C. Belta and L.C.G.J.M. Habets. Constructing decidable hybrid systems with velocity bounds. In *Proc. 43th IEEE Conf. on Decision and Control*, pages 467–472, New York, 2004. IEEE Press.

C. Belta and L.C.G.J.M. Habets. Controlling a class of nonlinear systems on rectangles. *IEEE Trans. Automatic Control*, 51:1749–1759, 2006.

C. Belta, L.C.G.J.M. Habets, and V. Kumar. Control of multi-affine systems on rectangles with applications to hybrid biomolecular networks. In *Proc. 41th IEEE Conf. on Decision and Control*, pages 534–539, New York, 2002. IEEE Press.

E. Boczko, W.D. Kalies, and K. Mischaikow. Polygonal approximation of flows. *Topology Appl.*, 154:2501–2520, 2007.

L. Calzone, F. Fages, and S. Soliman. BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22:1805–1807, 2006.

E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking*. MIT Press, 2000.

L. Grüne. *Asymptotic behavior of dynamical and control systems under perturbation and discretization*. *Lecture Notes in Mathematics*, vol. 1783, Springer-Verlag, Berlin, 2002.

L.C.G.J.M. Habets and J.H. van Schuppen. A control problem for affine dynamical systems on a full-dimensional polytope. *Automatica*, 40:21–35, 2004.

L.C.G.J.M. Habets, P.J. Collins, and J.H. van Schuppen. Reachability and control synthesis for piecewise-affine hybrid systems on simplices. *IEEE Trans. Automatic Control*, 51:938–948, 2006a.

L.C.G.J.M. Habets, M. Kloetzer, and C. Belta. Control of rectangular multi-affine hybrid systems. In *Proc. 45th IEEE Conf. on Decision and Control*, pages 2619–2624, New York, 2006b. IEEE Press.

S.K. Jha, E.M. Clarke, C.J. Langmead, A. Legay, A. Platzer, and A.P. Zuliani. A Bayesian approach to model checking biological systems. In *Computational Methods in Systems Biology (CMSB 2009)*, volume 5688 of *Lecture Notes in Computer Science*, pages 218–234. Springer-Verlag, Berlin, 2009.

H. de Jong, J. Geiselmann, C. Hernandez, and M. Page. Genetic network analyzer: Qualitative simulation of genetic regulatory networks. *Bioinformatics*, 19:336–344, 2003.

H. de Jong, J. Geiselmann, G. Batt, C. Hernandez, and M. Page. Qualitative simulation of the initiation of sporulation in *bacillus subtilis*. *Bull. Math. Biology*, 66:261–300, 2004a.

H. de Jong, J.-L. Gouzé, C. Hernandez, M. Page, T. Sari, and J. Geiselman. Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bull. Math. Biology*, 66:301–340, 2004b.

M. Kloetzer and C. Belta. A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Trans. Automatic Control*, 53:287–297, 2008.

M. Kloetzer and C. Belta. Reachability analysis of multi-affine systems. *Transactions of the Institute of Measurement and Control* 32:445–467, 2010.

X. Liu, J. Jiang, O. Ajayi, X. Gu, D. Gilbert, and R. Sinnott. BioNessie. *Global Healthgrid: e-Science Meets Biomedical Informatics - Proceedings of HealthGrid 2008*, (138):147–157, 2008.

H. Ma, F. Boogerd and I. Goryanin. Modelling nitrogen assimilation of Escherichia coli at low ammonium concentration. *Journal of Biotechnology*, 144:175–183, 2009.

P.T. Monteiro, D. Ropers, R. Mateescu, A.T. Freitas, and H. de Jong. Temporal logic patterns for querying dynamic models of cellular interaction networks. *Bioinformatics*, 24:227–233, 2008.

S. Ratschan and Z. She. Safety verification of hybrid systems by constraint propagation based abstraction refinement. *ACM Trans. Embedded Computing Systems*, 6(1), 2007.

A. Rizk, G. Batt, F. Fages and S. Soliman. On a continuous degree of satisfaction of temporal logic formulae with applications to systems biology. In *Proc. CMSB 2008*, volume 5307 of *Lecture Notes in Computer Science*, pages 251–268. Springer-Verlag, Berlin, 2008.

D. Ropers, V. Baldazzi, and H. de Jong. Model reduction and the dynamics of the global regulatory network controlling the carbon starvation response in *escherichia coli*. Report, INRIA Rhône-Alpes, Mont Bonnot, 2008.

O. Stursberg, S. Kowalewski, I. Hoffmann, and J. Preußig. Comparing timed and hybrid automata as approximations of continuous systems. In *Hybrid Systems IV*, volume 1273 of *Lecture Notes in Computer Science*, pages 361-377. Springer-Verlag, Berlin, 1997.