Algorithmic Algebraic Model Checking II: Decidability of Semi-Algebraic Model Checking and its Applications to Systems Biology *

V. Mysore[1], C. Piazza[2], and B. Mishra[1,3]

[1] Courant Institute, New York University, New York, NY, U.S.A.
[2] Dept. of Mathematics and Computer Science, University of Udiné, Udine, Italy
[3] NYU School of Medicine, New York University, New York, NY, U.S.A.
vm40@nyu.edu, piazza@dimi.uniud.it, mishra@nyu.edu

Abstract. Motivated by applications to systems biology, and the emergence of *semi-algebraic hybrid systems* as a natural framework for modeling biochemical networks, we continue exploring the decidability problem for model-checking with TCTL (Timed Computation Tree Logic) over this broad class of *semi-algebraic hybrid systems*. Previously, we had introduced these models, demonstrated the close connection to the goals of systems biology. However, we had only developed the techniques for bounded reachability, arguing for the adequacy of such an approach in a majority of the biological applications. Here, we present a semi-decidable symbolic algebraic dense-time TCTL model checking algorithm, which satisfies two desirable properties: it can be derived automatically from the symbolic description, and it extends to and generalizes other versions of temporal logics. The main mathematical device at the core of this approach is Tarski-Collins' real quantifier elimination employed at each fixpoint iteration, whose high complexity is the crux of its unfortunate limitation. Along with these results, we prove the undecidability of this problem in the more powerful "real" Turing machine formalism of Blum, Shub and Smale. We then demonstrate a preliminary version of our model-checker Tolque on the Delta-Notch example.

1 Prologue

It has been said[4], "Biologists have generally eschewed the possibility, or even the value, of an overarching theory of life." Biology is considered complex and not amenable to systematic dissection to reveal a unifying principle. However, as complex interconnected interactions among various biological entities begin to be cataloged from a diverse set of experiments, patterns emerge: sequences are

* The work reported in this paper was supported by grants from NSF's ITR program, Defense Advanced Research Projects Agency (DARPA), the US Air Force (AFRL), National Institutes of Health (NIH) and New York State Office of Science, Technology & Academic Research (NYSTAR). C.P. was partially supported by the MIUR FIRB grant RBAU018RCZ and the MIUR PRIN'04 grant 2004013015.

[4] *Making Sense of Life*, E.F. Keller, Harvard Press, MA, 2002.

aligned; genes are clustered; genes are grouped in modules; proteins are placed in families; motifs of interaction are listed; polymorphisms are partitioned into blocks; chromosomal aberrations and methylation patterns are segmented. The picture, however, remains frozen in time.

On the other hand, much less has been inferred about the temporal interactions of these entities. There are two problems: a mathematically precise, but somewhat idealized description of these interactions is often presented in a form, that is neither succinct nor easy to analyze. We lack both theoretical frameworks and efficient implementations for developing automatic computational tools that will allow a scientist to explore important phenomenological properties of these models.

The subject *Algorithmic Algebraic Model Checking* focuses on these issues as it examines connections between systems biology, dynamical systems, modal logic and computability, and how they can be useful in the biological context. Towards this aim, we began by addressing the symbolic bounded reachability problem for a new class of hybrid models arising in systems biology – *semi-algebraic hybrid systems*, introduced in the first paper of this "AAMC" (Algorithmic Algebraic Model Checking) series [28]. There, we aimed to characterize the widest range of automata that admit sound albeit expensive mathematical techniques, as opposed to focusing on a very narrow class of systems that often prematurely sacrifice generalizability for the sake of efficiency. It was shown that the bounded reachability problem can be solved using real algebraic techniques like Taylor series approximation and quantifier elimination. It was found sufficiently powerful in analyzing such systems as the Delta-Notch protein interaction example [10, 14, 19]. It was expected that, building upon this algebraic bounded reachability algorithm [28] and other recent techniques (e.g. some of Fränzle's ideas [13]), we can address the algebraic model-checking problem over the dense time logic *TCTL* [1]. The current paper deals with this subject.

We build upon and integrate many existing ideas: we use Henzinger et al.'s characterization of the *Until* operator as a fixpoint expression involving the *one-step until* operator [17]. Exploiting the power of a symbolic[5] approach, we retain all parameters as variables thus obtaining an algebraic expression representing the possible solutions. The ability to perform an entirely symbolic analysis of arbitrary polynomial hybrid systems over a full temporal logic, limited only by computational power, distinguishes our approach from the other methods in literature. Furthermore, to study decidability, we use Blum et al.'s "real" Turing machine (or equivalently, finite-dimensional machine over a field) formalism [7] – a more apt approach to analyzing problems involving real computations. We find that reachability is undecidable even in this more powerful computational model. The rest of the paper is organized as follows: the main ingredients of the paper – semi-algebraic hybrid automata, the Blum-Shub-Smale model of "real" computation and TCTL – are reviewed in *Section 2*; the technical proofs of our main results along with a literature survey are provided in *Section 3*; we

[5] This is in contrast to traditional *symbolic* model-checking where we refer to the use of BDDs as opposed to explicit enumeration of states as being *symbolic*.

demonstrate our software system Tolque over the same Delta-Notch example in *Section 4* (additional results are recorded in the *Appendix*) and conclude with a discussion in *Section 5*.

2 Technical Preliminaries

The temporal properties of a network of interacting biochemicals are typically captured by relating two neighboring system-states at time instants t and $t + \delta$, and the biochemical interactions (synthesis, degradation, multimerization, etc.) which occur in that short time interval δ. The dynamics resulting from these interactions can be described as a set of differential equations and discrete states [25]. Nonetheless, a direct model of transitions and flows, given through their symbolic description, can be computationally manipulated (either numerically or symbolically) to derive logical conclusions about global temporal properties, that may not have been obvious in the instantaneous description. The exact structure of this approach depends on the complexity of the three underlying frameworks: description of the dynamical system, the expressivity of the temporal logic and the basic operations of the models of computation. In a conventional "numerical" approach, starting with an initial system-state, successive states are chased by an integration scheme (eg. Runge-Kutta). Conclusions about the behavior of the network are then made by tracing the trajectories over a suitable time-frame and verifying temporal properties (eg. the Simpathica tool [5]). The "symbolic" alternative to the numerical procedure instead uses algebraic methods to characterize the transition of the system with time. The appropriate frameworks for this setting consist of the following: semi-algebraic hybrid automata which allow polynomial expressions, TCTL logic to capture the continuous changes, and the "real" Turing machine model that computes a semi-algebraic operation in one unit step. Their formal definitions follow.

Definition 1. Semi-Algebraic Set[26] *Every quantifier-free boolean formula composed of polynomial equations and inequalities defines a semialgebraic set (i.e., unquantified first-order formulæ over the reals -* $(\mathbb{R}, +, \times, =, <)$*).* □

Definition 2. Semi-Algebraic Hybrid Automata [28] *A k-dimensional* hybrid automaton *is a 7-tuple,* $H = (Z,\ V,\ E,\ Init,\ Inv,\ Flow,\ Jump)$*, consisting of the following components:*

- $Z = \{Z_1,\ \ldots,\ Z_k\}$ *and* $Z' = \{Z'_1,\ \ldots,\ Z'_k\}$ *are two finite sets of variables ranging over the reals* $\mathbb{R}$
- (V, E) *is a directed graph of discrete states and transitions*
- *Each discrete state* $v \in V$ *is labeled by "Init"(initial), "Inv"(invariant) and "Flow" labels of the form* $Init_v[Z]$*,* $Inv_v[Z]$*, and* $Flow_v[Z, Z', t, h]$
- *Each edge* $e \in E$ *is labeled by a "Jump" condition of the form* $Jump_e[Z, Z'] \equiv Guard_e(Z) \wedge Reset_e(Z, Z')$
- *Init, Inv, Flow, and Jump are semi-algebraic.* □

For ease of expression and clarity, we have enhanced the semantics presented in [28].

Definition 3. Semantics of Hybrid Automata *Let $H = (Z, V, E, Init, Inv, Flow, Jump)$ be a hybrid automaton of dimension k.*

- *A* location *ℓ of H is a pair $\langle v,\ R\rangle$, where $v \in V$ is a discrete state and $R \in \mathbb{R}^k$ is an assignment of values to the variables of Z. A location $\langle v,\ R\rangle$ is said to be* admissible, *if $Inv_v(R)$ is satisfied.*
- *The* continuous reachability transition relation $\xrightarrow[\mathcal{C}]{h}$ *forces the discrete state invariant to hold at every location except the end-location, along the evolution curve determined by the flow equations during the $h(>0)$ time units from the current time t_0:*

  $$\langle v, R\rangle \xrightarrow[\mathcal{C}]{h} \langle v, S\rangle \quad \textit{iff}$$
  $$\Big(Flow_v(R, S, t_0, h)\ \wedge\ \forall Z', h' \in [0, h)\ Flow_v(R, Z', t_0, h') \Rightarrow Inv_v(Z')\Big),$$

  where $Flow_v(Z, Z', T, h)$ is the flow label of v.
- *The* discrete reachability transition relation $\xrightarrow[\mathcal{D}]{0}$ *ensures that both parts of the* zero-time *jump – the guard condition which needs to be satisfied just before the transition is taken, and the reset condition which determines the values after the transition, are satisfied.*

  $$\langle v, R\rangle \xrightarrow[\mathcal{D}]{0} \langle u, S\rangle \qquad \textit{iff} \qquad \langle v, u\rangle \in E\ \wedge\ Jump_{v,u}(R, S).$$

- *The* transition relation $\mathcal{T}$ *of H connects the possible values of the system variables before and after* one step - *a discrete step for a time $h = 0$ or a continuous evolution for any time period $h > 0$:*

  $$\mathcal{T}(\ell \xrightarrow{h} \ell') = \quad \{h = 0 \wedge \ell \xrightarrow[\mathcal{D}]{0} \ell'\}\ \vee\ \{h > 0 \wedge \ell \xrightarrow[\mathcal{C}]{h} \ell'\}.$$

- *A* trace *of H is a sequence $\ell_0, \ell_1, \ldots, \ell_n, \ldots$ of admissible locations such that*

  $$\forall i \geq 0,\ \exists h_i \geq 0,\ \mathcal{T}(\ell_i \xrightarrow{h_i} \ell_{i+1}). \quad \square$$

Remark 1. Few remarks about this definition of *trace* are in order: It admits two continuous transitions to occur consecutively, which is necessary for compositionality of traces. Further, two consecutive continuous transitions of time-steps h_1 and h_2 are *not* necessarily equivalent to one continuous transition of time-step $h_1 + h_2$ in the case of non-linear approximation errors in $\xrightarrow[\mathcal{C}]{h}$.

When a *semi-algebraic* relation $Flow_v(R, S, t, h)$ is used between the continuous states R at time t and S at time $t+h$ in a discrete state v, it may have been "derived" in two ways: (1) *Solution Is A Polynomial*: The equation describing the continuous evolution of the variables in a discrete state is a polynomial, say

$Y(t)$, and $Flow_v(Z, Z', t, h) \equiv \{\ Z = Y(t) \wedge Z' = Y(t{+}h)\ \}$. Or, (2) *Differential Equation Is A Polynomial*: Differential equations describing the continuous evolution are *approximated* in $Flow_v$ using one of the symbolic integration schemes (e.g., the *Taylor* series in [28] or based on a direct integration scheme such as the linear *Euler* or the higher degree *Runge-Kutta*). The error is controlled by an upper bound (say $\varDelta$) on the time spent in one continuous step, as we aim for over- or under-approximating the flow equations. The Lagrange Remainder Theorem can be used to estimate errors [23].

We now report the basic definitions of the temporal logics TCTL and $T\mu$-Calculus which we use to study properties of our semi-algebraic hybrid automata.

Definition 4. TCTL[1] *It has the following syntactic structure:*

$$\phi ::= p \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid \phi_1 \exists\mathcal{U} \phi_2 \mid \phi_1 \forall\mathcal{U} \phi_2 \mid z.\phi.$$

Its associated semantics are described below:

- **z.**: *The freeze quantification "z." binds the associated variable z to the current time. Thus the formula $z.\phi(z)$ holds at time t iff $\phi(t)$ does.*
- *$\phi_1 \forall\mathcal{U} \phi_2$ and $\phi_1\ \exists\mathcal{U}\ \boldsymbol{\phi_2}$: universal (on all paths) and existential (on at least one path) "until" operators. For $\phi_1\ \mathcal{U}\ \phi_2$ to be true on a path, ϕ_2 is required to be true somewhere along the path, and ϕ_1 is required to be true all along the path up to (but not necessarily at) that location.* □

Remark 2. The basic notations are often extended by the following syntactic abbreviations [1].

1. $p\ \exists\mathcal{U}_{\leq max}\ q\ \equiv\ p\ \exists\mathcal{U}\ (q \wedge z.(z \leq max))$ and $p\ \forall\mathcal{U}_{\leq max}\ q\ \equiv\ p\ \forall\mathcal{U}\ (q \wedge z.(z \leq max))$: "subscripted" *Until* operators (*max* is the time-bound).
2. $\forall\mathcal{F}\ \equiv\ true\ \forall\mathcal{U}\ p$ and $\exists\mathcal{F}\ \equiv\ true\ \exists\mathcal{U}\ p$: "eventuality" operators.
3. $\forall\mathcal{G}\ \equiv\ \neg\exists\mathcal{F}\neg p$ and $\exists\mathcal{G}\ \equiv\ \neg\forall\mathcal{F}\neg p$: "invariance" operators.

Definition 5. Single-Step Until Operator, $\triangleright$, [17]. *The formula $p \triangleright q$ holds if $p \vee q$ is true all along "one step" of the hybrid system and q is true at the end of the transition.* □

Definition 6. $T\mu$-Calculus Syntax: [17]. $\phi ::= X \mid p \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid \phi_1 \triangleright \phi_2 \mid z.\phi \mid \mu X.\phi$ *, where μ is the* least-fixpoint *operator. Thus,*

- *The* greatest-fixpoint *ν can be expressed as $\neg\mu X.(\neg\phi[X := \neg X])$.*
- Existential Until: $p\ \exists\mathcal{U}\ q\ =\ \mu X.(q \vee (p \triangleright X))$
- Universal Until: $p\forall\mathcal{U}q\ =\ \neg(\neg q\ \exists\mathcal{U}\ (\neg p \wedge \neg q))$ □

Notice that the translation of the universal until is valid only when q is "finitely variable" over all premodels [17].

The undecidability result we will prove is based on the model of finite-dimensional machines over a field $\mathcal{R}$, which in our case will be $\mathbb{R}$, and on the undecidability of the Mandelbrot set over these machines. (We only introduce these "real" Turing Machines here, and refer the interested reader to [7].)

Definition 7. Finite-Dimensional Machine Over $\mathcal{R}$: [7]. *A finite dimensional machine M over $\mathcal{R}$ consists of a finite directed connected graph with four types of nodes:* input, computation, branch *and* output*. The unique input node has no incoming edges and only one outgoing edge. All other nodes have possibly several incoming edges. Computation nodes have only one outgoing edge, branch nodes exactly two,* Yes *and* No*, and output nodes none. In addition the machine has three spaces:* input space $\mathcal{I}_M$, state space $\mathcal{S}_M$ *and* output space $\mathcal{O}_M$ *of the form $\mathcal{R}^n, \mathcal{R}^m, \mathcal{R}^l$, respectively, where n, m and l are positive integers. Associated with each node of the graph are maps of these spaces and* next node *assignments.*

1. *Associated with the* input node *is a linear map $I : \mathcal{I}_M \rightarrow \mathcal{S}_M$ and a unique next node β_1.*
2. *Each* computation node *η has an associated* computation map*, a polynomial (or rational) map $g_\eta : \mathcal{S}_M \rightarrow \mathcal{S}_M$ given by m polynomials (or rational functions) $g_j : \mathcal{R}^m \rightarrow \mathcal{R}, j = 1, \cdots, m$, and a unique next node β_η. If g is a rational map associated with a computation node (in the case $\mathcal{R}$ is a field), we assume each g_j is given by a* fixed *pair of polynomials (p_j, q_j), where $g_j(x) = (p_j(x))/(q_j(x))$.*
3. *Each* branch node *η has an associated* branching function*, a nonzero polynomial function $h_\eta : \mathcal{S}_M \rightarrow \mathcal{R}$. The next node along the* Yes *outgoing edge, β_η^+, is associated with the condition $h_\eta \geq 0$ and the next node along the* No *outgoing edge, β_η^-, with $h_\eta(z) < 0$.*
4. *Each* output node *η has an associated linear map $\mathcal{O}_\eta : \mathcal{S}_M \rightarrow \mathcal{O}_M$ and no next node.* □

Definition 8. The Mandelbrot Set [24], *$\mathcal{M}$ is the subset of the set of complex numbers $\mathcal{C}$ that remains bounded when subject to the following iterative procedure: $f_0(C) = C$, $f_{n+1}(C) = f_n(C)^2 + C$. Formally, the complement $\mathcal{M}'$ of the Mandelbrot set is defined as*

$$\mathcal{M}' = \{C \in \mathcal{C} | f_n(C) \rightarrow \infty \text{ as } n \rightarrow \infty\}. \quad \square$$

It is to be noted that $f_i(C) \geq 2$ implies that eventually $f_n(C) \rightarrow \infty$.

In what follows, when we refer to the Mandelbrot set we mean the 2-dimensional set of real numbers corresponding to the Mandelbrot set, i.e., the set of pairs of the form $\langle C_r, C_i \rangle$ such that $C = C_r + iC_i$ is in the Mandelbrot set.

Theorem 1. Undecidability Of The Mandelbrot Set: [7]. *The Mandelbrot set cannot be expressed as the countable union of semi-algebraic sets over $\mathbb{R}$, and hence not decidable over $\mathbb{R}$.* □

3 Symbolic Algebraic Model Checking

Our main results for semi-algebraic hybrid systems may be summarized thus: (1) Reachability is *undecidable* even in Blum et al.'s "real" Turing machine formalism. (2) The "existential" segment of TCTL (including reachability) and the negation of the "universal" segment are *semi-decidable*. Further, all subscripted operators become *decidable* in the absence of zeno-paths. (3) Finally, a *quantifier elimination* tool (e.g. Qepcad [18],Redlog [12]) may be used to perform the fixpoint iterations of a TCTL query. The technical details are presented below.

The symbolic route to model-checking TCTL-specifications of hybrid systems is via the fixpoint expression for the *until* operator, which uses the standard *single-step until* operator $\triangleright$ [17] (also, see [28, 13]). The exact expression for the $\triangleright$ operator for semi-algebraic hybrid systems proves the basis of our approach: $\triangleright$ *corresponds to a semi-algebraic expression and is hence decidable.*

Definition 9. $\triangleright$ **for Semi-Algebraic Hybrid Systems.** *The expression* $p \triangleright q$ *is* True *at the current continuous state* R *if* q *is true now,* or

- *For one of the possible current discrete states* v*, there exists at least one discrete state* u *to which a transition can be taken such that* q *holds at the end,* or
- *For one of the possible current discrete states* v*, there exists a continuous transition (of at most* Δ *time units when we need to upper-bound the flow-approximation error) all along which* $p \vee q$ *holds, with* q *being true at the end*[6]*.*

$$
\begin{aligned}
p \triangleright q \ = \ q(R) \ & \textstyle\bigvee_{\forall v} \\
& \Big(\{\exists S \ \textstyle\bigvee_{\forall u} \langle v, R\rangle \xrightarrow[\mathcal{D}]{0} \langle u, S\rangle \wedge q(S)\} \ \bigvee \\
& \quad \{\exists S, h \ (0 < h \leq \Delta) \wedge \langle v, R\rangle \xrightarrow[\mathcal{C}]{h} \langle v, S\rangle \wedge q(S) \ \wedge \\
& \quad\quad \forall S', h' \ ((0 \leq h' < h) \wedge \langle v, R\rangle \xrightarrow[\mathcal{C}]{h'} \langle u, S'\rangle) \Rightarrow (p(S') \vee q(S'))\} \Big) \quad \square
\end{aligned}
$$

Remark 3. The upperbound Δ on h should be *omitted* if there is no error in the $Flow_v$ expression. Also, since the discrete jump is instantaneous, $p(R)$ does not appear in the discrete-jump expression (second line).

Theorem 2. *The* one-step-until *operator* $\triangleright$ *is decidable for semi-algebraic hybrid systems if* p *and* q *are also semi-algebraic.*

Proof. Semi-algebraic sets are closed under boolean operations and quantifier elimination. Since *Jump*, *Inv* and *Flow* are semi-algebraic, so are the expressions $\xrightarrow[\mathcal{C}]{t}$ and $\xrightarrow[\mathcal{D}]{0}$. Thus $p \triangleright q$ is semi-algebraic since p and q are also semi-algebraic. Since quantifier elimination over semi-algebraic sets is decidable [32], $p \triangleright q$ is decidable. □

Corollary 1. *For semi-algebraic hybrid systems:*

1. $\exists\mathcal{U}$*,* $\exists\mathcal{F}$*,* $\exists\mathcal{G}$ *and their subscripted versions* $\exists\mathcal{U}_{\leq z}$*,* $\exists\mathcal{F}_{\leq z}$ *and* $\exists\mathcal{G}_{\leq z}$ *are* semi-decidable.

[6] The last term in the formula, $p(S') \vee q(S')$, can be replaced with just $p(S')$ for evaluating $\exists\mathcal{U}$ over semi-algebraic hybrid systems.

2. *The* negations *of $\forall\mathcal{U}$, $\forall\mathcal{F}$, $\forall\mathcal{G}$ and their subscripted versions $\forall\mathcal{U}_{\leq z}$, $\forall\mathcal{F}_{\leq z}$ and $\forall\mathcal{G}_{\leq z}$ are* semi-decidable.
3. *All subscripted operators become* decidable *in the absence of* zeno *paths.*

Proof. The conclusions can be drawn as follows:

- The $\exists\mathcal{U}$ operator can be evaluated by iterating (indefinitely) over the decidable "one-step-until" operator $\rhd$ as per the fixpoint characterization $p\ \exists\mathcal{U}\ q\ \equiv\ \mu X.(q \vee (p \rhd X))$. Hence it is semi-decidable i.e. the computation procedure is guaranteed to converge if the query is *True*.
- Since $p\forall\mathcal{U}q\ \equiv\ \neg(\neg q\ \exists\mathcal{U}\ (\neg p \wedge \neg q))$, it can be guaranteed to converge only when it is *False*. Thus the negation of $\forall\mathcal{U}$ is semi-decidable by our procedure.
- Since $\exists\mathcal{F}p\ \equiv\ \mathit{true}\ \exists\mathcal{U}\ p$, reachability is semi-decidable.
- $\forall\mathcal{F}p\ \equiv\ \mathit{true}\ \forall\mathcal{U}\ p$ and is not semi-decidable since $\forall\mathcal{U}$ is not.
- Since $\exists\mathcal{G}p\ \equiv\ \neg\forall\mathcal{F}\neg p$, we can guarantee that it will converge if it is *True* since $\forall\mathcal{F}$ is guaranteed to converge if it is *False*. Thus it is semi-decidable.
- Since $\forall\mathcal{G}p\ \equiv\ \neg\exists\mathcal{F}\neg p$, it is guaranteed to converge only when it is *False*.
- A new variable *time* is introduced, with *initial* value 0, *flow* 1 in all discrete states and identity resets. This allows the interpretation of *freeze* $(z.X)$ and *subscripted until* $(U_{\leq a})$ operators.
- In non-zeno systems, every path of a specified time-length can be explored fully. Hence all subscripted operators are decidable. □

Remark 4. Purely symbolic reachability cannot be convergent as many sets (including the Mandelbrot set) cannot be expressed as the finite union of semi-algebraic sets [7]. Similarly, the solution of many coupled, non-linear differential equations and simple discrete difference equations are inexpressible even using exponential and trigonometric terms [30], let alone as a finite union of polynomial inequalities. However, the conventional semi-decidability notion only applies to cases where the query can be answered as *True* or *False*. It was under this default assumption (also used by Fränzle while discussing "polynomial" hybrid systems [13]) that the above results were derived.

3.1 General Undecidability Of Reachability

System-state (or equivalently, "location") reachability is undecidable for hybrid automata with just two clocks [16], as the Turing machine halting-problem can be encoded as a reachability query. It becomes pertinent to ask if this undecidability result holds for the more powerful "real" computing machines of Blum et al.[7], where semi-algebraic sets appear naturally in the computability definition (see *Path Decomposition Theorem* [7]). In the following construction, we present a semi-algebraic hybrid system and encode the Mandelbrot set as a reachability query. Since Blum and Smale have proved that the *Mandelbrot set is undecidable* [7], this proves that reachability over semi-algebraic hybrid systems is also undecidable, *even* under the "real" Turing Machine model.

Definition 10. The Mandelbrot Hybrid Automaton *Let $C = \langle C_r, C_i \rangle$ be a pair of real numbers. The* Mandelbrot Hybrid Automaton *M_C consists of*

- *One discrete state s_0 with invariant* False *and two continuous variables Z_1 and Z_2.*
- *Flow*$_1$: $\{\ Z_1' = Z_1\ \wedge\ Z_2' = Z_2\ \}$ *(no continuous evolution).*
- *One Discrete State Transition:* $1 \rightarrow 1$ *with Jump*$_1$: $(Z_1' = Z_1^2 - Z_2^2 + C_r) \wedge (Z_2' = 2Z_1Z_2 + C_i)$. $\square$

Notice that in M_C the only possible trace is the infinite zeno path of self-loops.

Theorem 3. General Undecidability Of Reachability *For semi-algebraic hybrid systems, reachability is undecidable even in Blum et al.'s "real" Turing machine formalism.*

Proof. Consider the Mandelbrot hybrid automaton M_C defined above. Let $S(t) = (Z_1(t), Z_2(t))$ be the point reached after t discrete transitions from the initial location $\langle s_0, (0,0)\rangle$. After one more discrete transition (self-loop), we get

$$\begin{aligned} S(t+1) = S'(t) &= \{Z_1(t)^2 - Z_2(t)^2 + C_r\} + \imath.\{2Z_1(t)Z_2(t) + C_i\} \\ &= \{Z_1(t) + \imath.Z_2(t)\}^2 + \{C_r + \imath.C_i\} \end{aligned}$$

In other words, if we consider the pairs of real numbers as complex numbers, we have $S'(t) = S^2(t) + C$ which is the defining equation of the *Mandelbrot Set.* Clearly, there exists an evolution where $|S(t)| \geq 2$ if and only if $C = C_r + iC_i$ does not belong to the Mandelbrot set, i.e., the decidability of the reachability query[7] $(Z_1^2 + Z_2^2 \geq 4)$ would imply the decidability of the Mandelbrot set, thus resulting in a contradiction. $\square$

3.2 Literature Review

While semi-algebraic hybrid systems have been suggested in one form on another before [20, 4, 13, 22], the full potential of this formalization is only beginning to be appreciated [28]. Beyond timed, multirate and initialized rectangular automata [2, 29], the linearity of continuous dynamics is another extensively studied restriction [3, 6]. Controllable linear systems [31], some families of linear vector fields [22] and o-minimal hybrid automata [21] have also been shown to be decidable for the reachability query. In the case of o-minimal hybrid automata, the decidability is guaranteed by the decidability of the underlying theory and by the fact that the resets are constant. In semi-algebraic hybrid automata, we do not have any restriction on the resets. However, o-minimal systems admit more complex functions (beyond polynomials) in the flows, invariants and guards.

While the above methods find efficient solutions by restricting the dynamics, over- or under-approximating methods assume that the reachable region has a (mathematically) convenient geometric shape such as a polyhedron, a level set or an ellipsoid [6, 8, 9]. Bisimulation on the other hand is an intelligent partitioning of the concrete system-state space of the hybrid system into fewer abstract discrete-states such that the properties of interest continue to hold in the simpler

[7] *Reachable*$(p) \equiv\ \exists\mathcal{F}(p)$.

smaller model [15]. Predicate abstraction has also been frequently used to map a hybrid automaton into a discrete one [33, 3].

On the algebraic side, Jirstrand [20] demonstrated the use of Qepcad for problems in control system design. Anai [4] and Fränzle [13] independently suggested the use of quantifier elimination for the verification of polynomial (semi-algebraic) hybrid systems, while Lafferiere et al. [22] have described a quantifier-elimination-centric method for symbolic reachability computation of linear vector fields.

4 Tolque: A Symbolic Algebraic TCTL Model Checker

A preliminary version of a symbolic algebraic model checker that uses the TCTL model-checking approach outlined in the previous section has been implemented. This quantifier-elimination-centric model checker, christened Tolque, takes as input a semi-algebraic hybrid automaton specification (with the flow equations already approximated if necessary) and an *Existential Until* ($p\ \exists\mathcal{U}\ q$) query. It then computes the fixpoint $p\ \exists\mathcal{U}\ q\ =\ \mu X.(q \vee (p \triangleright X))$ [17] by using Qepcad [18] to perform the quantifier elimination in $p \triangleright X$. The entire process is automated in this *C/C++* implementation that runs in *Linux*.

A Case Study: The Delta-Notch Protein Signaling

Here we examine the Delta-Notch protein interaction system, the primary basis of biological pattern formation. Ghosh et al. [14, 19] analyzed a simplified piecewise linear hybrid automaton model (derived from Collier et al.'s work [10]) with the following properties: (1) The Delta (concentration v_D) production is turned on by low Notch concentration (v_N) in the same cell i.e. when $-v_N > h_D$; (2) The Notch production is turned on by high Delta concentration in the cell environment (neighbors) i.e. when $\Sigma_i u_D^i > h_N$. Here, h_D and h_N are the thresholds, and u_D^i denotes the Delta concentration in each (i-th) neighbor.

In this section, we show how some interesting properties of the one-cell and two-cell Delta-Notch model of Ghosh et al. [14, 19] can be formulated as temporal logic queries, that Tolque can answer. Unfortunately, Qepcad cannot support the queries necessary to analyze system properties more complex than those documented here. Approximate methods (such as those discussed in AAMC-III [27]), reduction in the computational complexity of quantifier elimination, and greater computing power will help overcome this computational bottleneck. Rather than providing new insight about the model, at this point, Tolque is only seen to support a more elegant and general way of thinking about system properties. (Please see the appendix for a complete list of results.)

One-Cell System In the hybrid automaton modeling the one-cell system [14], there are 2 dynamic variables v_D and v_N corresponding to the Delta and Notch concentration in the cell, 4 discrete states corresponding to the 2×2 possibilities resulting from Delta and Notch production being switched "on" or "off". The external variable u_N is assumed to be static. We will denote the upper bound on the continuous time-step by $\varDelta$.

1. **Pruned Transition Map** When the state invariants are non-overlapping, an evolution path from discrete state i to j is possible iff $Inv_i \wedge \{Inv_i \; \exists\mathcal{U} \; Inv_j\}$. Notice that invariants can be made non-overlapping by introducing a new environmental variable "discrete-state" that is reset to the destination discrete state number during discrete state transitions, with flow always 0.
   - **Discrete Transition** $1 \xrightarrow[\mathcal{D}]{0} 2$
     $[-v_N \leq h_D \wedge u_N \leq h_N] \; \exists\mathcal{U} \; [-v_N \geq h_D \wedge u_N \leq h_N]$
     After k iterations, we get the requirement $v_N \leq -h_D/(1-\varDelta l_N)^k$ which is *True* when $k \; (\geq -\log{(h_D/v_N)}/\log{(1-\varDelta l_N)})$. Thus the transition from 1 to 2 is possible.
   - **Discrete Transition** $2 \xrightarrow[\mathcal{D}]{0} 1$
     $[-v_N \geq h_D \wedge u_N \leq h_N] \; \exists\mathcal{U} \; [-v_N < h_D \wedge u_N < h_N]$ converges after two iterations to *False*. Thus, it is not possible to jump to state 1 from state 2.
2. **Estimating Continuous-State Equilibrium Concentrations** When the state invariants are non-overlapping, an equilibrium of the continuous state exists in state i iff $Inv_i \; \wedge \; \neg\{Inv_i \; \exists\mathcal{U} \; (v'_D \neq v_D \; \vee \; v'_N \neq v_N)\}$, where v'_D and v'_N are the values after one step of the hybrid automaton.

   Remark 5. We have extended the TCTL notation to allow more complex temporal queries that can describe the values of the variables before and after one step of evolution. The semi-algebraic quantifier elimination based model-checking supports this without *any* additional work.

   State 1: $\neg\{[-v_N \leq h_D \wedge u_N \leq h_N] \; \exists\mathcal{U} \; [v'_D \neq v_D \vee v'_N \neq v_N]\}$ converges to *False* – implying the non-existence of an equilibrium in this state.

   State 2: $\neg\{[-v_N \geq h_D \wedge u_N \leq h_N] \; \exists\mathcal{U} \; [v'_D \neq v_D \vee v'_N \neq v_N]\}$ converges to $v_D l_D - r_D = 0 \wedge v_N \leq 0$. Thus we get the equilibrium concentrations as $v^*_D = r_D/l_D, v^*_N = 0$.
3. **Discrete State Equilibria** When the invariants are non-overlapping, a system can stay forever in the discrete state i iff $Inv_i \; \wedge \; \neg \; \{Inv_i \; \exists\mathcal{U} \; \neg Inv_i\}$.

   State 1: $[-v_N \leq h_D \wedge u_N \leq h_N] \; \exists\mathcal{U} \; [-v_N > h_D \vee u_N > h_N]$ returns $v_N \leq -h_D/(1-\varDelta l_N)^k$ after k iterations, effectively evaluating to *True*. Thus the system always evolves out of state 1 and hence it does not correspond to any equilibrium.

   State 3: $[-v_N \leq h_D \wedge u_N \geq h_N] \; \exists\mathcal{U} \; [-v_N < h_D \vee u_N > h_N]$ is non-convergent and returns $v_N \leq (-h_D - \varDelta r_N)/(1-\varDelta l_N)$ after one iteration. So, for such a path out of state 3 to not exist, there should be no way of satisfying the above inequality when $-v_N < h_D$. So we get $(-h_D - \varDelta r_N)/(1-\varDelta l_N) < -h_D$ which simplifies to $h_D > -r_N/l_N$.

Two-Cell System The above exercise can be repeated for a two cell model, where there are 4 dynamic variables n_1, d_1, n_2 and d_2, which stand for the Notch and Delta concentrations in cell 1 and 2 respectively. Due to the limitations of Qepcad, we use the numerical parameter values courtesy Hwang et al. [19] to demonstrate our approach. In particular, we set $\lambda_N = \lambda_D = r_N = r_D = 1, h_D = -\frac{1}{2}, h_N = \frac{1}{5}, \Delta = \frac{1}{2}$.

1. **Equilibrium Concentration Estimation**
   State q_{10} (3,2): $\neg\{[-2n_1 > -1 \wedge 5d_2 < 1 \wedge -2n_2 < -1 \wedge 5d_1 > 1] \exists\mathcal{U} [d_1' \neq d_1 \vee n_1' \neq n_1 \vee d_2' \neq d_2 \vee n_2' \neq n_2]\}$ converges to $[n_1 \leq 0 \wedge d_2 \leq 0 \wedge d_1 - 1 = 0 \wedge n_2 - 1 = 0]$. Thus $n_1^* = d_2^* = 0$ and $d_1^* = n_2^* = 1$.
   State q_{15} (4,3): $\neg\{[-2n_1 > -1 \wedge 5d_2 > 1 \wedge -2n_2 < -1 \wedge 5d_1 > 1] \exists\mathcal{U} [d_1' \neq d_1 \vee n_1' \neq n_1 \vee d_2' \neq d_2 \vee n_2' \neq n_2]\}$ converges to *False*, implying that in this discrete state the variables can never be in equilibrium.
2. **Are Equilibria Reversible?**
   State q_7 (2,3): $[-2n_1 > -1 \wedge 5d_2 < 1 \wedge -2n_2 < -1 \wedge 5d_1 > 1]\ \exists\mathcal{U}\ [-2n_1 = -1 \vee 5d_2 = 1 \vee -2n_2 = -1 \vee 5d_1 = 1]$ converges to *False* after 2 iterations implying that this is an irreversible discrete state equilibrium.
   State q_{16} (4,4): $[-2n_1 > -1 \wedge 5d_2 > 1 \wedge -2n_2 > -1 \wedge 5d_1 > 1]\ \exists\mathcal{U}\ [-2n_1 = -1 \vee 5d_2 = 1 \vee -2n_2 = -1 \vee 5d_1 = 1]$ converges to *True* implying that the two-cell Delta-Notch system will always leave this discrete state.
3. **Choice Of Equilibrium** We can "verify" that the wrong equilibrium cannot be reached from a given initial relation between n_1 and n_2, and d_1 and d_2. When the invariants are non-overlapping, the initial conditions that allow a path to discrete state i but not to discrete state j are given by $\{True\ \exists\mathcal{U}\ Inv_i\} \wedge \neg\{True\ \exists\mathcal{U}\ Inv_j\}$.
   State q_7 (2,3): At iteration 2 of *True* $\exists\mathcal{U}$ $[-2n_1 > -1 \wedge 5d_2 < 1 \wedge -2n_2 < -1 \wedge 5d_1 > 1]$, we get: $n_1 - 1 \leq 0 \wedge [[2n_1 - 5d_1 \leq 0 \wedge 5d_2 - 1 \leq 0 \wedge 8n_2 - 5d_2 - 3 \geq 0 \wedge n_2 + n_1 - 1 = 0] \vee [8n_1 - 5d_1 - 3 \leq 0 \wedge 4d_2 + d_1 - 1 = 0 \wedge 2n_2 - 1 \geq 0 \wedge 8n_2 + 5d_1 - 5 \geq 0] \vee [5d_1 - 1 \geq 0 \wedge 2n_1 - 5d_1 \leq 0 \wedge 5d_2 + 2n_1 - 2 \leq 0 \wedge 2n_2 - 1 \geq 0] \vee [5d_1 - 1 \geq 0 \wedge 2n_1 - 1 \leq 0 \wedge 5d_2 - 1 \leq 0 \wedge 8n_2 - 5d_2 - 3 \geq 0] \vee [2n_1 - 1 \leq 0 \wedge 5d_2 - 1 \leq 0 \wedge 8n_2 - 5d_2 - 3 \geq 0 \wedge 8n_2 + 5d_1 - 5 \geq 0] \vee [2n_1 - 5d_1 \leq 0 \wedge 5d_2 - 1 \leq 0 \wedge 2n_2 - 1 \geq 0 \wedge 8n_2 + 5d_1 - 5 \geq 0]]\ \equiv f_7$.
   State q_{10} (3,2): At iteration 2 of *True* $\exists\mathcal{U}$ $[-2n_1 < -1 \wedge 5d_2 > 1 \wedge -2n_2 > -1 \wedge 5d_1 < 1]$, we get: $n_2 - 1 \leq 0 \wedge [[2n_1 - 1 \geq 0 \wedge 5d_2 + 8n_1 - 5 \geq 0 \wedge d_2 + 4d_1 - 1 = 0 \wedge 2n_2 + 5d_1 - 2 \leq 0] \vee [2n_1 - 1 < 0 \wedge 8n_1 - 5d_1 - 3 \geq 0 \wedge 5d_2 + 8n_1 - 5 \geq 0 \wedge n_2 + n_1 - 1 = 0] \vee [8n_1 - 5d_1 - 3 \geq 0 \wedge 5d_2 + 8n_1 - 5 < 0 \wedge 5d_2 + 2n_1 - 2 \geq 0 \wedge n_2 + n_1 - 1 = 0] \vee [2n_1 - 1 \geq 0 \wedge 5d_2 - 1 \geq 0 \wedge 2n_2 + 5d_1 - 2 \leq 0 \wedge n_2 + n_1 - 1 < 0] \vee [5d_1 - 1 \leq 0 \wedge 2n_1 - 1 \geq 0 \wedge 5d_2 + 8n_1 - 5 \geq 0 \wedge 2n_2 - 5d_2 \leq 0] \vee [5d_1 - 1 \leq 0 \wedge 2n_1 - 1 \geq 0 \wedge 5d_2 + 8n_1 - 5 \geq 0 \wedge 2n_2 - 1 \leq 0] \vee [8n_1 - 5d_1 - 3 \geq 0 \wedge 5d_2 - 1 \geq 0 \wedge 2n_2 + 5d_1 - 2 \leq 0 \wedge 2n_2 - 1 \leq 0]]\ \equiv f_{10}$.
   State q_7 and not State q_{10}: The initial conditions that lead only to q_7 and not q_{10} are thus given by $f_7 \wedge \neg f_{10}$. Since we have assumed no upper bound on the initial values and since we have been able to compute only two iterations, this formula does *not* evaluate to *True* given the correct initial partition $n_1 < n_2 \wedge d_1 > d_2$. However, when Qepcad simplifies the above formula assuming that $n_1 > n_2 \wedge d_1 < d_2$, it evaluates to *False*.

5 Conclusion

The real limitation of this quantifier-elimination-based model-checking comes from the computational complexity of Collins' cylindrical algebraic decomposition (CAD) algorithm, with its double-exponential dependence on the number of variables [11]. In our experience, Qepcad failed to support fully symbolic analysis of the two-cell Delta-Notch system. However, it is to be noted that even this preliminary version of Tolque was able to support a very uniform way of asking about a good spectrum of interesting temporal properties of a biologically significant hybrid system. We are in the process of rewriting Tolque in *Lisp* and integrating it with Simpathica[5]. These modifications will allow biochemical networks to be easily represented, stored and analyzed in keeping with our initial "Systems Biology" motivation. Based on the results of this paper, we can focus on complexity improvement through other meaningful approximations. The next paper in the AAMC series focuses on approximate methods like bisimulation-partitioning, space discretization (using grids and polyhedra) and time discretization [27]. Eventually, we plan to implement our own symbolic algebra system to work hand in hand with the different *quantifier elimination*, *Gröbner basis* and *characteristic set* tools that can systematically simplify the formulæ at each fixpoint iteration.

To summarize, the "semi-algebraic" method, outlined here, enables sophisticated symbolic algebraic model checking of a large class of hybrid automata, well beyond the capabilities of current applications of symbolic methods in this area. The semi-decidability results for the TCTL operators and the introduction of the Blum-Shub-Smale model are expected to spark further investigations of the relations between dynamical systems, topology and complexity. Our approach is general: it can be extended beyond TCTL model-checking to dense-time LTL; and it can be further enhanced by allowing non-linear (but polynomial) expressions in the temporal queries that can involve the values before and after one step of the hybrid system.

Finally, although the state of the art of algebraic hybrid systems model-checking can only be compared to that of *boolean finite-state model-checking* in the early 80s, we believe that the approach will make quick and important strides, and yield deep insights in biological areas before the end of this decade.

References

1. R. Alur, C. Courcoubetis, and D. Dill. Model-Checking for Real-Time Systems. In *International Symposium on Logic in Computer Science*, 5, pages 414–425. IEEE Computer Press, 1990.
2. R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The Algorithmic Analysis of Hybrid Systems. *Theoretical Computer Science*, 138:3–34, 1995.
3. R. Alur, T. Dang, and F. Ivancic. Progress on Reachability Analysis of Hybrid Systems Using Predicate Abstraction. In O. Maler and A. Pnueli, editors, *Hybrid Systems: Computation and Control (HSCC'03)*, volume 2623 of *LNCS*, pages 4–19. Springer-Verlag, 2003.

4. Hirokazu Anai. Algebraic approach to analysis of discrete-time polynomial systems. In *ECC Karlsure (Germany)*, 1999.
5. M. Antoniotti, A. Policriti, N. Ugel, and B. Mishra. Reasoning about Biochemical Processes. *Cell Biochemistry and Biophysics*, 38:271–286, 2003.
6. E. Asarin, T. Dang, O. Maler, and O. Bournez. Approximate Reachability Analysis of Piecewise-Linear Dynamical Systems. In B. Krogh and N. Lynch, editors, *Hybrid Systems: Computation and Control (HSCC'00)*, volume 1790 of *LNCS*, pages 20–31. Springer-Verlag, 2000.
7. L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation*. Springer-Verlag, 1997.
8. O. Bournez, O. Maler, and A. Pnueli. Orthogonal Polyhedra: Representation and Computation. In F. Vaadrager and J. van Schuppen, editors, *Hybrid Systems: Computation and Control (HSCC 1999)*, volume 1596 of *LNCS*, pages 19–30. Springer-Verlag, 1999.
9. A. Chutinan and B. Krogh. Verification of Polyhedral-Invariant Hybrid Automata Using Polygonal Flow Pipe Approximations. In F. W. Vaandrager and J. H. van Schuppen, editors, *Hybrid Systems: Computation and Control (HSCC'99)*, volume 1569 of *LNCS*, pages 76–90. Springer-Verlag, 1999.
10. J. R. Collier, N. A. M. Monk, P. K. Maini, and J. H. Lewis. Pattern Formation by Lateral Inhibition with Feedback: a Mathematical Model of Delta-Notch Intercellular Signalling. *Journal of Theor. Biology*, 183:429–446, 1996.
11. G. E. Collins. Quantifier Elimination for the Elementary Theory of Real Closed Fields by Cylindrical Algebraic Decomposition. In *Proceedings of the Second GI Conference on Automata Theory and Formal Languages*, volume 33 of *LNCS*, pages 134–183. Springer-Verlag, 1975.
12. Andreas Dolzmann and Thomas Sturm. REDLOG: Computer algebra meets computer logic. *SIGSAM Bulletin (ACM Special Interest Group on Symbolic and Algebraic Manipulation)*, 31(2):2–9, 1997.
13. Martin Fränzle. What will be eventually true of polynomial hybrid automata? In Naoki Kobayashi and Benjamin C. Pierce, editors, *Theoretical Aspects of Computer Software, 4th International Symposium, TACS 2001, Sendai, Japan, October 29-31, 2001, Proceedings*, volume 2215 of *Lecture Notes in Computer Science*, pages 340–359. Springer, 2001.
14. R. Ghosh and C. Tomlin. Lateral Inhibition through Delta-Notch signaling: A Piecewise Affine Hybrid Model. In M. D. D. Benedetto and A. Sangiovanni-Vincentelli, editors, *Int.l Workshop on Hybrid Systems: Computation and Control (HSCC'01)*, volume 2034 of *LNCS*, pages 232–246. Springer-Verlag, 2001.
15. Esfandiar Haghverdi, Paulo Tabuada, and George J. Pappas. Bisimulation relations for dynamical, control, and hybrid systems. *Theoretical Computer Science*, November 2003.
16. T. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya. What's Decidable about Hybrid Automata. In *Symposium on the Theory of Computing (STOC)*, pages 373–382, 1995.
17. T. A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic Model Checking for Real-time Systems. In *7th Annual IEEE Symposium on Logic in Computer Science*, pages 394–406. IEEE, IEEE Computer Society Press, June 1992.
18. H. Hong. Quantifier elimination in elementary algebra and geometry by partial cylindrical algebraic decomposition, version 13. *WWW site* `www.eecis.udel.edu/∼saclib`, 1995.

19. Inseok Hwang, Hamsa Balakrishnan, Ronojoy Ghosh, and Claire Tomlin. Reachability analysis of delta-notch lateral inhibition using predicate abstraction. *Lecture Notes in Computer Science*, 2552:715–724, Jan 2002.
20. Mats Jirstrand. Nonlinear control system design by quantifier elimination. *J. Symb. Comput.*, 24(2):137–152, 1997.
21. G. Lafferiere, G. J. Pappas, and S. Sastry. O-minimal Hybrid Systems. *Mathematics of Control, Signals, and Systems*, 13(1):1–21, March 2000.
22. Gerardo Lafferriere, George J. Pappas, and Sergio Yovine. Symbolic reachability computation for families of linear vector fields. *J. Symb. Comput.*, 32(3):231–253, 2001.
23. R. Lanotte and S.Tini. Taylor approximation for hybrid systems. In *HSCC*. LNCS, 2005.
24. B. Mandelbrot. *The Fractal Geometry of Nature*. Freeman Co., San Francisco, 1982.
25. B. Mishra. A Symbolic Approach to Modeling Cellular Behavior. In S. Sahni, V. K. Prasanna, and U. Shukla, editors, *High Performance Computing (HiPC'02)*, volume 2552 of *LNCS*, pages 725–732. Springer-Verlag, 2002.
26. B. Mishra. *Computational Real Algebraic Geometry.* CRC Press, Boca Raton, FL, 2004.
27. V. Mysore and B. Mishra. Algorithmic Algebraic Model Checking III: Approximate Methods. In *Third International Symposium on Automated Technology for Verification and Analysis (ATVA)*, 2005.
28. C. Piazza, M. Antoniotti, V. Mysore, A. Policriti, F. Winkler, and B. Mishra. Algorithmic Algebraic Model Checking I: The Case of Biochemical Systems and their Reachability Analysis. In *17th International Conference on Computer Aided Verification (CAV)*, 2005.
29. A. Puri and P. Varaiya. Decidebility of hybrid systems with rectangular differential inclusions. *Computer Aided Verification*, pages 95–104, 1994.
30. C. Robinson. *Dynamical Systems: Stability, Symbolic Dynamics, and Chaos.* CRC Press, Boca Raton, 1995.
31. Paulo Tabuada and George J. Pappas. Model checking ltl over controllable linear systems is decidable. *Hybrid Systems : Computation and Control, Lecture Notes in Computer Science*, 2623, April 2003.
32. A. Tarski. *A Decision Method for Elementary Algebra and Geometry.* University of California Press, second edition, 1948.
33. A. Tiwari and G. Khanna. Series of Abstraction for Hybrid Automata. In C. J. Tomlin and M. Greenstreet, editors, *Hybrid Systems: Computation and Control (HSCC'02)*, volume 2289 of *LNCS*, pages 465–478. Springer-Verlag, 2002.

Appendix

One-Cell Delta-Notch Analysis in Tolque

1. **Pruned Transition Map**
   - **Discrete Transition** $1 \xrightarrow[\mathcal{D}]{0} 2$

     $[-v_N \leq h_D \wedge u_N \leq h_N]\ \exists\mathcal{U}\ [-v_N \geq h_D \wedge u_N \leq h_N]$

     After k iterations, we get the requirement $v_N \leq -h_D/(1-\Delta l_N)^k$ which is *True* when k $(\geq -\log{(h_D/v_N)}/\log{(1-\Delta l_N)})$. Thus the transition from 1 to 2 is possible.

- **Discrete Transition** $2 \xrightarrow[\mathcal{D}]{0} 1$
  $[-v_N \geq h_D \wedge u_N \leq h_N]\ \exists\mathcal{U}\ [-v_N < h_D \wedge u_N < h_N]$ converges after two iterations to *False*. Thus, it is not possible to jump to state 1 from state 2.

2. **Estimating Continuous-State Equilibrium Concentrations**
   State 1: $\neg\{[-v_N \leq h_D \wedge u_N \leq h_N]\ \exists\mathcal{U}\ [v'_D \neq v_D \vee v'_N \neq v_N]\}$ converges to *False* – implying the non-existence of an equilibrium in this state.
   State 2: $\neg\{[-v_N \geq h_D \wedge u_N \leq h_N]\ \exists\mathcal{U}\ [v'_D \neq v_D \vee v'_N \neq v_N]\}$ converges to $v_D l_D - r_D = 0 \wedge v_N \leq 0$. Thus we get the equilibrium concentrations as $v^*_D = r_D/l_D, v^*_N = 0$.
   State 3: $\neg\{[-v_N \leq h_D \wedge u_N \geq h_N]\ \exists\mathcal{U}\ [v'_D \neq v_D \vee v'_N \neq v_N]\}$ converges to $v_D \leq 0 \wedge v_N l_N - r_N = 0$. Thus $v^*_D = 0, v^*_N = r_N/l_N$ are the equilibrium values.
   State 4: $\neg\{[-v_N \geq h_D \wedge u_N \geq h_N]\ \exists\mathcal{U}\ [v'_D \neq v_D \vee v'_N \neq v_N]\}$ converges to the equilibrium condition $v^*_N l_N - r_N = 0 \wedge h_D + v^*_N \neq 0 \wedge v^*_D l_D - r_D = 0 \wedge h_N - u_N \neq 0$.
3. **Discrete State Equilibria**
   State 1: $[-v_N \leq h_D \wedge u_N \leq h_N]\ \exists\mathcal{U}\ [-v_N > h_D \vee u_N > h_N]$ returns $v_N \leq -h_D/(1 - \Delta l_N)^k$ after k iterations, effectively evaluating to *True*. Thus the system always evolves out of state 1 and hence it does not correspond to any equilibrium.
   State 2: $[-v_N \geq h_D \wedge u_N \leq h_N]\ \exists\mathcal{U}\ [-v_N < h_D \vee u_N > h_N]$ converges to *False*. Thus there is no path out of state 2 and hence it corresponds to an equilibrium. Note that the transition from 2 to 4 recorded in [14] is not possible in a one-cell model where u_N is not modeled as a dynamic variable.
   State 3: $[-v_N \leq h_D \wedge u_N \geq h_N]\ \exists\mathcal{U}\ [-v_N < h_D \vee u_N > h_N]$ is non-convergent and returns $v_N \leq (-h_D - \Delta r_N)/(1 - \Delta l_N)$ after one iteration. So, for such a path out of state 3 to not exist, there should be no way of satisfying the above inequality when $-v_N < h_D$. So we get $(-h_D - \Delta r_N)/(1 - \Delta l_N) < -h_D$ which simplifies to $h_D > -r_N/l_N$.
   State 4: $[-v_N \geq h_D \wedge u_N \geq h_N]\ \exists\mathcal{U}\ [-v_N < h_D \vee u_N < h_N]$ is non-convergent and returns $l_N h_D + r_N > 0 \wedge h_D - \Delta v_N l_N + \Delta r_N + v_N \geq 0$ after the second iteration. The second term is just a lower bound on the starting value of v_N which continues to drop with each iteration - effectively being *True*. Hence, for an equilibrium to exist in State 4, the first term must not be satisfiable i.e. $l_N h_D + r_N \leq 0$ which is equivalent to $h_D \leq -r_N/l_N$.

Two-Cell Delta-Notch Analysis in Tolque

1. **Equilibrium Concentration Estimation**
   State q_{10} (3,2): $\neg\{[-2n_1 > -1 \wedge 5d_2 < 1 \wedge -2n_2 < -1 \wedge 5d_1 > 1]\exists\mathcal{U}[d'_1 \neq d_1 \vee n'_1 \neq n_1 \vee d'_2 \neq d_2 \vee n'_2 \neq n_2]\}$ converges to $[n_1 \leq 0 \wedge d_2 \leq 0 \wedge d_1 - 1 = 0 \wedge n_2 - 1 = 0]$. Thus $n^*_1 = d^*_2 = 0$ and $d^*_1 = n^*_2 = 1$.

State q_7 **(2,3):** $\neg\{[-2n_1 < -1 \wedge 5d_2 > 1 \wedge -2n_2 > -1 \wedge 5d_1 < 1] \exists\mathcal{U} [d_1' \neq d_1 \vee n_1' \neq n_1 \vee d_2' \neq d_2 \vee n_2' \neq n_2]\}$ converges to $[n_2 \leq 0 \wedge d_1 \leq 0 \wedge d_2 - 1 = 0 \wedge n_1 - 1 = 0]$. Thus $n_2^* = d_1^* = 0$ and $d_2^* = n_1^* = 1$.

State q_{15} **(4,3):** $\neg\{[-2n_1 > -1 \wedge 5d_2 > 1 \wedge -2n_2 < -1 \wedge 5d_1 > 1] \exists\mathcal{U} [d_1' \neq d_1 \vee n_1' \neq n_1 \vee d_2' \neq d_2 \vee n_2' \neq n_2]\}$ converges to *False*, implying that in this discrete state the variables can never be in equilibrium.

2. **Are Equilibria Reversible?**

State q_7 **(2,3):** $[-2n_1 > -1 \wedge 5d_2 < 1 \wedge -2n_2 < -1 \wedge 5d_1 > 1] \; \exists\mathcal{U} \; [-2n_1 = -1 \vee 5d_2 = 1 \vee -2n_2 = -1 \vee 5d_1 = 1]$ converges to *False* after 2 iterations implying that this is an irreversible discrete state equilibrium.

State q_{10} **(3,2):** $[-2n_1 < -1 \wedge 5d_2 > 1 \wedge -2n_2 > -1 \wedge 5d_1 < 1] \; \exists\mathcal{U} \; [-2n_1 = -1 \vee 5d_2 = 1 \vee -2n_2 = -1 \vee 5d_1 = 1]$ also converges to *False* after 2 iterations implying that the equilibrium is irreversible.

State q_{16} **(4,4):** $[-2n_1 > -1 \wedge 5d_2 > 1 \wedge -2n_2 > -1 \wedge 5d_1 > 1] \; \exists\mathcal{U} \; [-2n_1 = -1 \vee 5d_2 = 1 \vee -2n_2 = -1 \vee 5d_1 = 1]$ converges to *True* implying that the two-cell Delta-Notch system will always leave this discrete state.

3. **Choice Of Equilibrium**

State q_7 **(2,3):** At iteration 2 of *True* $\exists\mathcal{U} \; [-2n_1 > -1 \wedge 5d_2 < 1 \wedge -2n_2 < -1 \wedge 5d_1 > 1]$, we get: $n_1 - 1 \leq 0 \wedge [[2n_1 - 5d_1 \leq 0 \wedge 5d_2 - 1 \leq 0 \wedge 8n_2 - 5d_2 - 3 \geq 0 \wedge n_2 + n_1 - 1 = 0] \vee [8n_1 - 5d_1 - 3 \leq 0 \wedge 4d_2 + d_1 - 1 = 0 \wedge 2n_2 - 1 \geq 0 \wedge 8n_2 + 5d_1 - 5 \geq 0] \vee [5d_1 - 1 \geq 0 \wedge 2n_1 - 5d_1 \leq 0 \wedge 5d_2 + 2n_1 - 2 \leq 0 \wedge 2n_2 - 1 \geq 0] \vee [5d_1 - 1 \geq 0 \wedge 2n_1 - 1 \leq 0 \wedge 5d_2 - 1 \leq 0 \wedge 8n_2 - 5d_2 - 3 \geq 0] \vee [2n_1 - 1 \leq 0 \wedge 5d_2 - 1 \leq 0 \wedge 8n_2 - 5d_2 - 3 \geq 0 \wedge 8n_2 + 5d_1 - 5 \geq 0] \vee [2n_1 - 5d_1 \leq 0 \wedge 5d_2 - 1 \leq 0 \wedge 2n_2 - 1 \geq 0 \wedge 8n_2 + 5d_1 - 5 \geq 0]] \; \equiv f_7$.

State q_{10} **(3,2):** At iteration 2 of *True* $\exists\mathcal{U} \; [-2n_1 < -1 \wedge 5d_2 > 1 \wedge -2n_2 > -1 \wedge 5d_1 < 1]$, we get: $n_2 - 1 \leq 0 \wedge [[2n_1 - 1 \geq 0 \wedge 5d_2 + 8n_1 - 5 \geq 0 \wedge d_2 + 4d_1 - 1 = 0 \wedge 2n_2 + 5d_1 - 2 \leq 0] \vee [2n_1 - 1 < 0 \wedge 8n_1 - 5d_1 - 3 \geq 0 \wedge 5d_2 + 8n_1 - 5 \geq 0 \wedge n_2 + n_1 - 1 = 0] \vee [8n_1 - 5d_1 - 3 \geq 0 \wedge 5d_2 + 8n_1 - 5 < 0 \wedge 5d_2 + 2n_1 - 2 \geq 0 \wedge n_2 + n_1 - 1 = 0] \vee [2n_1 - 1 \geq 0 \wedge 5d_2 - 1 \geq 0 \wedge 2n_2 + 5d_1 - 2 \leq 0 \wedge n_2 + n_1 - 1 < 0] \vee [5d_1 - 1 \leq 0 \wedge 2n_1 - 1 \geq 0 \wedge 5d_2 + 8n_1 - 5 \geq 0 \wedge 2n_2 - 5d_2 \leq 0] \vee [5d_1 - 1 \leq 0 \wedge 2n_1 - 1 \geq 0 \wedge 5d_2 + 8n_1 - 5 \geq 0 \wedge 2n_2 - 1 \leq 0] \vee [8n_1 - 5d_1 - 3 \geq 0 \wedge 5d_2 - 1 \geq 0 \wedge 2n_2 + 5d_1 - 2 \leq 0 \wedge 2n_2 - 1 \leq 0]] \; \equiv f_{10}$.

State q_7 **and not State** q_{10}**:** The initial conditions that lead only to q_7 and not q_{10} are thus given by $f_7 \wedge \neg f_{10}$. Since we have assumed no upper bound on the initial values and since we have been able to compute only two iterations, this formula does *not* evaluate to *True* given the correct initial partition $n_1 < n_2 \wedge d_1 > d_2$. However, when Qepcad simplifies the above formula assuming that $n_1 > n_2 \wedge d_1 < d_2$, it evaluates to *False*.

State q_{10} **and not State** q_7 Similarly, $\neg f_7 \wedge f_{10}$ evaluates to *False* assuming $n_1 < n_2 \wedge d_1 > d_2$.