Project Report

Automatic Computation of Barrier Certificates

Frédéric Marcel Tchouli

Rémy Seassau

January, 2023

Contents

1 Introduction

An important topic of research in formal verification is the safety verification of hybrid systems. Hybrid systems include both discrete and continuous dynamics, and are used to model cyber-physical systems, for example. The importance of this topic is derived from its applications to safety critical systems such as embedded flight control systems [15] or life support devices [4]. In the case that interests us, the verification is done by showing that starting from some initial conditions, a system cannot evolve to some unsafe region in the state space. For example, for an autopilot system one could image the initial condition to be that we start at some altitude $A(0) > 500$ and we want to show that from $A(0)$, with our flight control system, we will never reach the state $A(t) \leq 0$ where we have crashed.

One method for safety verification is to use barrier certificates, which separate the initial states from the unsafe region. In this project, we study the automatic computation of different types of barrier certificates using computation methods. Note that barrier certificates can be extended to hybrid systems using multiple barriers indexed by the discrete state and linked using conditions on the discrete transitions. We thus focus on the safety verification of a single dynamical system.

1.1 Problem definition

In this section, we give fundamental definitions which shall be subsequently used in the study of the different kinds of Barrier Certificates which we shall consider.

1.1.1 Some preliminaries

Definition 1 (Semi-algebraic Sets)**.** *A set $S \subseteq \mathbb{R}^n$ is semi-algebraic iff it is characterized by a finite boolean combination of polynomial equations and inequalities:*

$$\bigvee_{i=1}^{l} \left(\bigwedge_{j=1}^{m_i} p_{ij} < 0 \wedge \bigwedge_{j=m_i+1}^{M_i} p_{ij} = 0 \right) \tag{1}$$

where $p_{ij} \in \mathbb{R}[x_1, \ldots, x_n]$ (i.e. p_{ij} are multivariate polynomials in the indeterminates $x_1, \ldots, x_n$, with real coefficients).

By quantifier elimination (Section 3.1), every first-order formula of real arithmetic characterizes a semi-algebraic set and can be expressed in the form (1) [8]. Let $n \in \mathbb{N}_{\geq 1}$.

Definition 2 (n-dimensional system of ODEs)**.** *An (autonomous) n-dimensional system of ODEs is a system of the form:*

$$\begin{aligned} x_1' &= f_1(x_1, x_2, \ldots, x_n) \\ &\vdots \\ x_n' &= f_n(x_1, x_2, \ldots, x_n) \end{aligned}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ is a real-valued (typically continuous) function for each $i \in \{1, \ldots, n\}$, and x'_i denotes the time derivative of x_i, i.e. $\frac{dx_i}{dt}$.

In applications, constraints are often used to specify the states where the system is allowed to evolve, i.e. the system may only be allowed to evolve inside some given set $\mathcal{X} \subseteq \mathbb{R}^n$, which is known as the evolution constraint. We can write down systems of constrained ODEs concisely by using vector notation, i.e. by writing $\boldsymbol{x}' = \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{X}$. Here we have $\boldsymbol{x}' = (x'_1, \ldots, x'_n)$ and $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ is a vector field generated by the system, i.e. $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x}))$ for all $\boldsymbol{x} \in \mathbb{R}^n$. If no evolution constraint is given, $\mathcal{X}$ is assumed to be the Euclidean space $\mathbb{R}^n$.

Definition 3 (Lie derivative)**.** *We define the Lie derivative of a differentiable scalar function $\varphi\colon \mathbb{R}^n \to \mathbb{R}$ with respect to a vector field f as*

$$\mathcal{L}_f \varphi = \frac{\partial \varphi}{\partial x} f(x) = \sum_{i=1}^{n} \frac{\partial \varphi}{\partial x_i} f_i(x)$$

The Lie derivative evaluates the change of a scalar function along the flow of a vector field. In our case, the scalar function often corresponds to the barrier certificate and the vector field is the function f representing our dynamical system. Additionally, we have the following equality $\frac{\partial \varphi}{\partial x} f(x) = \frac{d\varphi(x(t))}{dt}$ which we will use later.

A solution to the initial value problem (IVP) for the system of ODEs $\boldsymbol{x}' = \boldsymbol{f}(\boldsymbol{x})$ with initial value $\boldsymbol{x}_0 \in \mathbb{R}^n$ is a (differentiable) function $\boldsymbol{x} : (a, b) \to \mathbb{R}^n$ defined for all t in some open interval including zero, i.e. $t \in (a, b)$, where $a, b \in \mathbb{R} \cup \{\infty, -\infty\}, a < 0 < b$, and such that $\boldsymbol{x}(0) = \boldsymbol{x}_0$ and $\frac{d}{dt}\boldsymbol{x}(t) = \boldsymbol{f}(\boldsymbol{x}(t))$ for all $t \in (a, b)$. At time t, for solutions to IVPs with initial value $\boldsymbol{x}_0$, we shall write $\boldsymbol{x}(\boldsymbol{x}_0, t)$, or simply $\boldsymbol{x}(t)$ if the initial condition is understood from context. It is intuitive that if the solution $\boldsymbol{x}(\boldsymbol{x}_0, t)$ is available in closed-form, then one can study properties such as safety by analysing the closed-form expression. However, in nonlinear ODEs it is in practice highly uncommon for solutions to exist explicitly in closed-form [2, 5], and even if closed-form solutions can be found, transcendental functions in these expressions lead to an undecidable problem [13].

Remark 1 . As in Sogokon et al [14], we employ a slight abuse of notation for sets and formulas characterizing those sets, i.e. $\mathcal{X}$ denotes both a set $\mathcal{X} \subseteq \mathbb{R}^n$ and a formula $\mathcal{X}$ of real arithmetic with free variables $x_1, \ldots, x_n$ which characterizes this set. In the case of sub-level sets, i.e. sets characterized by predicates of the form $B \leq 0$ where B is a real valued function in the (dependent) variables $x_1, \ldots, x_n$, we will write $B(\boldsymbol{x}) \leq 0$ to mean $B \leq 0$ is true in state $\boldsymbol{x} \in \mathbb{R}^n$, and will explicitly use the independent time variable t to write $B(\boldsymbol{x}(t)) \leq 0$ when we are interested in evaluating the predicate along a solution $\boldsymbol{x}(t)$ of a differential equation.

1.1.2 The Safety Verification Problem

We now give a formal definition of the Safety Verification Problem, which is the focus of this research project.

Definition 4 (Safety in ODEs)**.** *Given a system of ODEs $x' = f(x)$ with evolution domain constraint $\mathcal{X} \subseteq \mathbb{R}^n$, and the sets $\mathcal{X}_0 \subseteq \mathbb{R}^n$, $\mathcal{X}_\mathrm{u} \subseteq \mathbb{R}^n$ of initial and unsafe states, respectively, the system is said to be safe if and only if:*

$$\forall \mathbf{x}_0 \in \mathcal{X}_0, \forall t \geq 0 : ((\forall \tau \in [0,t].\boldsymbol{x}(\boldsymbol{x}_0, \tau) \in \mathcal{X}) \Rightarrow \boldsymbol{x}(\boldsymbol{x}_0, t) \notin \mathcal{X}_\mathrm{u}).$$

Definition 5 (Continuous invariant)**.** . *Let $\mathcal{X} \subseteq \mathbb{R}^n$ be the evolution domain constraint on a given system $\mathbf{x}' = f(\mathbf{x})$. We say that the set $I \subseteq \mathbb{R}^n$ is a continuous invariant iff the following statement holds:*

$$\forall \mathbf{x}_0 \in I, \forall t \geq 0 : ((\forall \tau \in [0,t] : \mathbf{x}(\mathbf{x}_0, \tau) \in \mathcal{X}) \Longrightarrow \mathbf{x}(\mathbf{x}_0, t) \in I).$$

For any given set of initial states $\mathcal{X}_0 \subseteq \mathbb{R}^n$, a continuous invariant I such that $\mathcal{X}_0 \subseteq I$ provides a sound over-approximation of the states reachable by the system from $\mathcal{X}_0$ by following the solutions to the ODEs within the evolution domain constraint $\mathcal{X}$. Indeed, the exact set of states reachable by a continuous system from $\mathcal{X}_0$ provides the smallest such invariant. [4] While Def. 1 above features the solution $\mathbf{x}(\mathbf{x}_0, t)$, which may not be available explicitly, a crucial advantage afforded by continuous invariants is the possibility of checking whether a given set is a continuous invariant without computing the solution, i.e. by working directly with the ODEs.

2 Barrier certificates

First introduced by S. Prajna and A. Jadbabaie [10], barrier certificates are a function satisfying a set of inequalities on both itself and its derivative. The term barrier originates from the fact that the zero level set of the barrier certificate $\coloneqq \{ x \mid B(x) = 0 \}$ is what separates the unsafe space from the trajectories starting from our initial space. Hence, if such a barrier certificate exists, we know that our system is safe. The main advantage of such a technique is that we can find a barrier without having to compute the reachable sets, since showing the existence of the barrier is sufficient. For systems modeled by robustly safe ODEs we know that there exists a corresponding barrier certificate [12]. What remains difficult is methods for computing these barrier certificates.

We can extend the use of barrier certificates to handle hybrid systems by constructing different barriers for different system locations and linking them by conditions on the discrete transitions between these system locations. It is known that if the vector field of the system is polynomial, then we can try to construct a polynomial barrier certificate using some numerical solvers. This also holds for hybrid systems if we add the condition that the whole system is composed of semi-algebraic sets. For simplification purposes, we will focus on barrier certificates in a single continuous system.

The following subsections focus on the definition of different types of barrier certificates. Section 2.1 goes over the initial barrier certificate introduced by Prajna and Jadbabaie. We call this the strict barrier certificate. In Section 2.3, we introduce another type of barrier certificate we will call the exponential barrier certificate [7]. From now, consider $x' = f(x)$ to be a continuous dynamical system where $x \in \mathcal{X}$ is a state of the system, $\mathcal{X}_0$ is the initial space and $\mathcal{X}_\mathrm{u}$ is our unsafe space.

2.1 Strict

In this subsection we formally define the strict barrier certificate and prove that its existence certifies the safety of the dynamical system. We report the following theorem and proof from Prajna and Jadbabaie [10], giving the conditions on the barrier certificate in order to ensure safety:

Theorem 1. *Suppose there exists a function $B\colon \mathcal{X} \to \mathbb{R}$ that is differentiable with respect to its argument and satisfies the following conditions for all $x \in \mathcal{X}$:*

$$x(t) \in \mathcal{X}_\mathrm{u} \implies B(x) > 0 \tag{2}$$

$$x(t) \in \mathcal{X}_0 \implies B(x) \leq 0 \tag{3}$$

$$B(x(t)) = 0 \implies \frac{\partial B}{\partial x} f(x) \leq 0 \tag{4}$$

then the safety of the system is guaranteed, i.e., there exists no trajectory of the system contained in $\mathcal{X}$ that starts from an initial state $\mathcal{X}_0$ and reaches a state in $\mathcal{X}_\mathrm{u}$.

Proof. Assume that a barrier B fulfilling the previous conditions exists. Let $x(t) \in \mathcal{X}$ such that $x(0) \in \mathcal{X}_0$. Then we have by condition (3) that $B(x(0)) \leq 0$. Then, condition (4) tells us that that $B(x(t))$ may never become positive. By (2), any such trajectory is thus guaranteed to never enter an unsafe state. □

In the above proof, it may be easier to understand that the reason that (4) implies that the barrier stays negative is analogous to the fact that for $f \in \mathcal{C}^1(\mathbb{R})$, $f'(x) < 0$ implies that the function is decreasing. Thus if the function is decreasing when $f(x) = 0$, we know that the function may never become positive once it is negative. Note also that we have simplified the proof by not considering disturbances to the dynamical system.

2.1.1 Example 1

To give further intuition, we report the following example of a two-dimensional second-order system [6, p. 180] used in Prajna and Jadbabaie. Let $\mathcal{X} = \mathbb{R}^2$ and consider the vector field given by

$$x_1' = x_2$$

$$x_2' = -x_1 + \frac{1}{3}x_1^3 - x_2$$

Let $\mathcal{X}_0 = \{\, x \in \mathbb{R}^2 : (x_1 - 1.5)^2 + x_2^2 \leq 0.25 \,\}$ and $\mathcal{X}_\mathrm{u} = \{\, x \in \mathbb{R}^2 : (x_1 + 1)^2 + (x_2 + 1)^2 \leq 0.16 \,\}$. It is possible to find a barrier quartic certificate that guarantees that a trajectory starting at $\mathcal{X}_0$ never enters $\mathcal{X}_\mathrm{u}$.

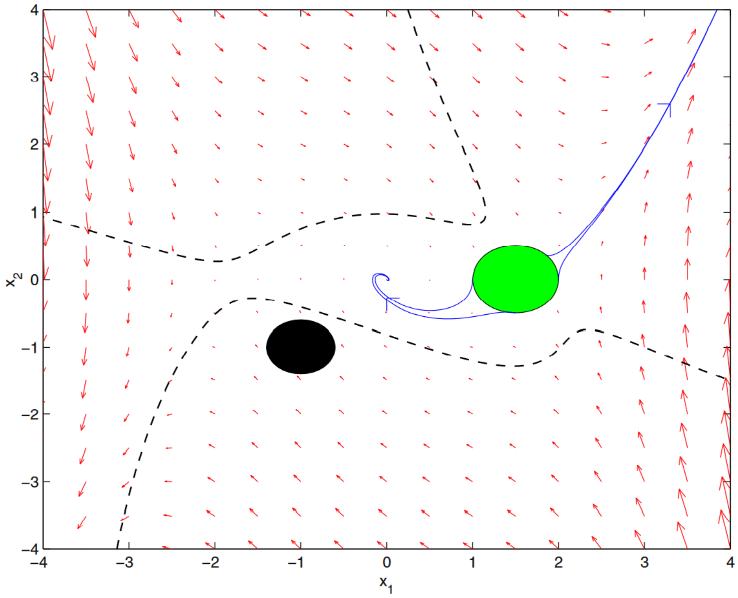

Figure 1: Phase portrait of the example [10, p. 12]: the red arrows indicate the flow of the vector field, the black set is $\mathcal{X}_u$, the green set is $\mathcal{X}_0$, the dashed lines are the zero level set of the barrier and the blue lines are some trajectories starting from $\mathcal{X}_0$.

The above figure offers a graphical intuition of why the example of the barrier guarantees safety. A trajectory starting in $\mathcal{X}_0$ will never enter $\mathcal{X}_u$ as it would have to cross the zero level set to do so, which is impossible by (4).

2.2 Convex

The convex barrier certificate is stronger than the strict barrier certificate. Indeed, one has that the set of strict barrier certificates is not convex. Thus, once cannot find a strict barrier certificate using convex optimization. This can be resolved by introducing a new barrier certificate type we will call the convex barrier certificate.

Definition 6 (Convex Barrier Certificate)**.** *We call a function a "convex barrier certificate" if it satisfies conditions* (2)-(3) *and a new condition that is stronger than* (4)*, given by:*

$$\forall x \in \mathcal{X},\ \frac{\partial B}{\partial x} f(x) \leq 0 \tag{5}$$

By the same proof as for the strict barrier certificate, the existence of a convex barrier certificate guarantees the system's safety. The added benefit of this stronger condition is that the set of convex barrier certificates is convex, meaning that for any $\theta \in [0, 1]$ and any two barrier B_1, B_2 satisfying conditions (2),(3) and (5), we have that $B(x) = \theta B_1(x) + (1 - \theta)B_2(x)$ is also a convex barrier certificate. In particular, this enables the computation of convex barrier certificates using convex optimization.

2.3 Exponential

The exponential barrier certificate introduced by Kong et al. [7] is a generalization of the convex barrier certificate. It introduces an additional constraint on the lie derivative of the barrier and is defined through the following theorem. We report here the statement of the theorem defining the convex barrier certificate using the same notation for the dynamical system as before. The proof is given in [7].

Theorem 2. *For any given $\lambda \in \mathbb{R}$, if there exists $B\colon\ \mathcal{X} \to \mathbb{R}$ satisfying the following conditions*

$$x(t) \in \mathcal{X}_{\mathrm{u}} \implies B(x) > 0 \tag{6}$$

$$x(t) \in \mathcal{X}_0 \implies B(x) \leq 0 \tag{7}$$

$$\frac{\partial B}{\partial x} f(x) - \lambda B(x) \leq 0 \tag{8}$$

then the safety of the system is guaranteed.

Remark that the exponential condition is convex, as it is a generalisation of the convex barrier certificate. This means that we can reduce the problem of constructing a barrier certificate into a convex optimization problem which can be solved by semidefinite programming (more in Section 3).

The obvious question to ask now is which barrier certificate is "better". Obviously in terms of use, all barrier certificates serve the exact same purpose and do so equally well. We thus turn to the question of computability: is it easier to find one of the barrier certificates types? In other words, is there a type that we can construct more reliably and is there a type that we can construct faster.

3 Computation Methods

3.1 Quantifier Elimination

Given a quantified first-order logic formula in real arithmetic, Quantifier Elimination (QE) is an algorithmic procedure that converts such formula into a quantifier-free yet logically equivalent one. Because the necessary conditions that polynomial Barrier Certificates need to satisfy are in fact quantified first-order logic formulas in real arithmetic in terms of the "template coefficients" to be determined (see Section 2), it is natural to think of ways in which these formulas can be reduced to their quantifier-free versions if the latter exist, in which case the problem is more approachable from a computational point of view.

In fact, a 1949 seminal and constructive result by Alfred Tarski and Abraham Seidenberg showed that Quantification Elimination is possible over the reals (an important corollary is the decidability of the first-order theory of real arithmetic). This is a very powerful result in the context of this research project, since it exactly means that once Quantification Elimination is performed on the Barrier Certificate conditions, we can obtain the exact full set of possible Barrier Certificates, each of which proves the safety of the system at hand. Hence, this method is the least conservative in the sense that it reveals all possible solutions.

Theorem 3 (Quantifier Elimination over real arithmetic OR Tarski-Seidenberg (1949))**.** *Given a semi-algebraic set $S \subset \mathbb{R}^{n+1}$, we define a projection map $\pi : S \longrightarrow \mathbb{R}^n$ that sends every $(x_1, x_2, ..., x_{n+1})$ to $(x_1, ..., x_n)$ in $\mathbb{R}^n$.*
Then, the image of S by π, $\pi(S) \subset \mathbb{R}^n$, is semi-algebraic.

An iterated application of the Tarski-Seidenberg theorem above then yields a quantifier-free Boolean formula. While Tarski had proposed an exact algorithm to perform QE, it could not be implemented on a computer. In [3] however, Collins and Hong provided an innovative (and optimal) doubly-exponential time QE algorithm. We refer the reader to this paper for more details.

3.2 The Sum of Squares (SOS) Method

Given the very high computational complexity of QE, it is interesting to consider more computationally attractive methods (which may come with trade-offs). For Barrier Certificate conditions exhibiting convexity, Sum-of-squares programming [11] provides a tractable way of solving for the Certificate's polynomial template coefficients as revealed by the following theorem:

Theorem 4 (SOS condition for exponential-type Barrier Certificates [11])**.**
As previously, we suppose that $\mathcal{X}_0 = \bigvee_{i=1}^{s} \left(\bigwedge_{j=1}^{m_i} p_{ij} < 0 \wedge \bigwedge_{j=m_i+1}^{M_i} p_{ij} = 0 \right)$ and
$\mathcal{X}_\mu = \bigvee_{k=1}^{t} \left(\bigwedge_{l=1}^{m_k} q_{kl} < 0 \wedge \bigwedge_{l=m_k+1}^{M_k} q_{kl} = 0 \right)$ are semi-algebraic.
Let $p_{\boldsymbol{a},\boldsymbol{d}}$ be the Barrier Certificate polynomial template where $\boldsymbol{a}$ is the vector of coefficients and $\boldsymbol{d}$ is the degree of the template. Then:

Inequalities (6), (7) and (8) are respectively implied by the following SOS inequalities, where $\varepsilon > 0$ is a small positive constant and $\sigma_{p_{i,j}}, \sigma_{q_{k,l}}$ are template SOS polynomials:

$$-p_{\mathrm{a},d} - \sum_{i,j} \sigma_{p_{i,j}} p_{i,j} \geq 0$$

$$p_{\mathrm{a},d} - \sum_{k,l} \sigma_{q_{k,l}} q_{k,l} - \varepsilon \geq 0$$

$$\lambda p_{\mathrm{a},d} - (p_{\mathrm{a},d})' \geq 0$$

The proof follows immediately by construction.
This problem can then be solved using a solver like Mathematica[16] or SOSTOOLS [11].

4 Implementation

Our objective being to compare the use of sum of squares and quantifier elimination to generate barrier certificates, we would need to build a tool to test different computation techniques.

Working with Prof. Mover, we set to extend a tool designed for semi-algebraic abstraction for the verification of polynomial dynamical systems [9]. The use of quantifier elimination to generate strict barrier certificates was already implemented. We thus added an encoding of convex barrier certificate computation to sum of squares optimization and an encoding for exponential barrier certificate computation using quantifier elimination.

All of the following implementations are based off of a polynomial template on which we enforce constraints. These constraints are implemented with either satisfiability modulo theories solvers for QE, such as Mathematica [16], or semidefinite programming solvers for SOS, such as CVXOPT [1]. The key problem here is what we choose the degree of our polynomial template to be. Indeed, one method is template enumeration, where we incrementally increase the degree and use our solves to try and find a solution [14], but this is computationally heavy.

The implementation procedure for Quantifier Elimination is quite straightforward as it analogous to writing down the conditions for either barrier certificate type in first order logic. The implementation for the sum of squares revolves around making sure that the inequalities described in (2)-(4) (or (6)-(8) for an exponential barrier) are true. This is done by forcing either the non-zero or the negation of the non-zero side of the inequality to be a sum of squares, i.e. positive.

5 Conclusion

During our project, we have encountered multiple types of barrier certificates and seen their individual characteristics. Of these, convexity of the set of barrier certificates has been particularly important as it enables convex optimization solvers to compute barriers. In general, we have seen that for any safe system there exists a barrier certificate proving this safety. It is however not guaranteed that we can always find such a barrier automatically with existing computation techniques. Although we have implemented two new types of computations, it remains to test these on various examples to observe empirically if some techniques appear to be ”better” than others.

Through our task of implementation, we have found particular topics regarding the automatic computation of barrier certificates that deserve further investigation. One important question for the efficiency of the computation is whether we can find a good heuristic for the template degree when we do template enumeration. If some can be found, they will greatly improve the scalability of the method as more complex barriers will be found faster.

Another direction could be to look at a weaker variation of the exponential barrier certificate akin to the strict barrier where we no longer require condition (8) on the whole space but rather on some smaller set. Doing this, we would lose the convexity of the set which would render computation through sum of squares impossible. However, it may increase the speed of quantifier elimination by enforcing a weaker condition which effectively filters the domain on which we are searching for a solution.

References

[1] M. S. Andersen, J. Dahl, and L. Vandenberghe. *CVXOPT: A Python package for convex optimization, version 1.2.* `http://cvxopt.org/`. 2018.

[2] Garrett Birkhoff and Gian-Carlo Rota. *Ordinary differential equations*. English. Introductions to Higher Mathematics. Boston-New York-Chicago: Ginn and Company, VII, 318 p. (1962). 1962.

[3] George E. Collins and Hoon Hong. "Partial Cylindrical Algebraic Decomposition for Quantifier Elimination". In: *J. Symb. Comput.* 12.3 (1991), pp. 299–328. DOI: `10.1016/S0747-7171(08)80152-6`. URL: `https://doi.org/10.1016/S0747-7171(08)80152-6`.

[4] Sonja Glavaski, Antonis Papachristodoulou, and Kartik B. Ariyur. "Safety Verification of Controlled Advanced Life Support System Using Barrier Certificates". In: *HSCC*. 2005.

[5] Jack K. Hale and Joseph P. LaSalle. "Differential Equations: Linearity vs. Nonlinearity". In: *SIAM Review* 5.3 (1963), pp. 249–272. DOI: `10.1137/1005068`. eprint: `https://doi.org/10.1137/1005068`. URL: `https://doi.org/10.1137/1005068`.

[6] Hassan K Khalil. *Nonlinear systems; 3rd ed.* The book can be consulted by contacting: PH-AID: Wallet, Lionel. Upper Saddle River, NJ: Prentice-Hall, 2002. URL: `https://cds.cern.ch/record/1173048`.

[7] Hui Kong et al. "Exponential-Condition-Based Barrier Certificate Generation for Safety Verification of Hybrid Systems". In: *CoRR* abs/1303.6885 (2013). arXiv: `1303.6885`. URL: `http://arxiv.org/abs/1303.6885`.

[8] Bhubaneswar Mishra. *Algorithmic Algebra*. Texts and Monographs in Computer Science. Springer, 1993. ISBN: 978-3-540-94090-6. DOI: `10.1007/978-1-4612-4344-1`. URL: `https://doi.org/10.1007/978-1-4612-4344-1`.

[9] Sergio Mover et al. "Implicit Semi-Algebraic Abstraction for Polynomial Dynamical Systems". In: July 2021, pp. 529–551. DOI: `10.1007/978-3-030-81685-8_25`.

[10] Stephen Prajna and Ali Jadbabaie. "Safety Verification of Hybrid Systems Using Barrier Certificates". In: *Hybrid Systems: Computation and Control*. Ed. by Rajeev Alur and George J. Pappas. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 477–492. ISBN: 978-3-540-24743-2.

[11] Stephen Prajna et al. "SOSTOOLS and its control applications". In: vol. 312. Aug. 2005, pp. 580–580. ISBN: 978-3-540-23948-2. DOI: `10.1007/10997703_14`.

[12] Stefan Ratschan. "A Robust Finite-Time Converse Theorem for Inductive Safety Certificates of Ordinary Differential Equations". In: *CoRR* abs/1701.03948 (2017). arXiv: `1701.03948`. URL: `http://arxiv.org/abs/1701.03948`.

[13] Daniel Richardson. "Some undecidable problems involving elementary functions of a real variable". In: *Journal of Symbolic Logic* 33.4 (1969), pp. 514–520. DOI: `10.2307/2271358`.

[14] Andrew Sogokon et al. “Pegasus: sound continuous invariant generation”. en. In: *Formal Methods in System Design* (Jan. 2021). ISSN: 0925-9856, 1572-8102. DOI: 10.1007/s10703-020-00355-z. URL: http://link.springer.com/10.1007/s10703-020-00355-z (visited on 10/15/2021).

[15] C.J. Tomlin et al. “Computational techniques for the verification of hybrid systems”. In: *Proceedings of the IEEE* 91.7 (2003), pp. 986–1001. DOI: 10.1109/JPROC.2003.814621.

[16] Inc. Wolfram Research. *Mathematica, Version 12.3.1*. Champaign, IL, 2021. URL: https://www.wolfram.com/mathematica.