Closure Certificates

Vishnu Murali
University of Colorado Boulder
Boulder, USA
vishnu.murali@colorado.edu

Ashutosh Trivedi
University of Colorado Boulder
Boulder, USA
ashutosh.trivedi@colorado.edu

Majid Zamani
University of Colorado Boulder
Boulder, USA
majid.zamani@colorado.edu

ABSTRACT

A *barrier certificate*, defined over the states of a dynamical system, is a real-valued function whose zero level set characterizes an inductively verifiable *state invariant* separating reachable states from unsafe ones. When combined with powerful decision procedures—such as sum-of-squares programming (SOS) or satisfiability-modulo-theory solvers (SMT)—barrier certificates enable an automated deductive verification approach to safety. The barrier certificate approach has been extended to refute LTL and ω-regular specifications by separating consecutive transitions of corresponding ω-automata in the hope of denying all accepting runs. Unsurprisingly, such tactics are bound to be conservative as refutation of *recurrence properties* requires reasoning about the well-foundedness of the transitive closure of the transition relation. This paper introduces the notion of *closure certificates* as a natural extension of barrier certificates from state invariants to transition invariants. We augment these definitions with SOS and SMT based characterization for automating the search of closure certificates and demonstrate their effectiveness over some case studies.

ACM Reference Format:
Vishnu Murali, Ashutosh Trivedi, and Majid Zamani. 2024. Closure Certificates. In *27th ACM International Conference on Hybrid Systems: Computation and Control (HSCC '24), May 14–16, 2024, Hong Kong SAR, China.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3641513.3650120

1 INTRODUCTION

As cyber-physical systems and internet-of-things continue to proliferate within critical infrastructure, the need for practical verification algorithms for infinite-state dynamical systems is ever-present. Structural induction over the transition structure of dynamical systems provides a lightweight yet powerful proof method to establish safety and invariance guarantees. However, when the invariant is not inductive, human ingenuity is required in strengthening the invariant to an inductive one. The notion of *barrier certificates* [28], when combined with automatic decision procedures automate the search for an inductive state invariant. This paper presents *closure certificates* as a generalization of barrier certificates to capture the *transitive closure* of transition relations to automate verification of linear temporal logic (LTL) and ω-regular specifications of discrete-time dynamical systems.

HSCC '24, May 14–16, 2024, Hong Kong SAR, China

ACM ISBN 979-8-4007-0522-9/24/05
https://doi.org/10.1145/3641513.3650120

Barrier Certificates for State Invariants. Intuitively, a barrier certificate [28] is a real-valued function over the state space that is negative over the initial states, positive over the unsafe states, and it does not increase with transitions. From this definition and the principle of structural induction, it follows that the zero level set of the barrier certificate over-approximate the set of reachable states. This, together with the positivity requirement over the unsafe states, provide a separation between reachable and unsafe states, guaranteeing safety. The results in [38] extended the barrier certificate based approach to refute violations of linear temporal logic (LTL) specifications expressed via ω-automata. In this so-called *state-triplet* approach, barrier certificates provide separation between consecutive transitions (involving three states) of the given ω-automaton in such a way that denies accepting runs. The approach has been extended for verification and synthesis for more general dynamical systems [1, 2, 15, 16]. These state-triplet approach are bound to suffer from conservatism as the verification of a general ω-regular property requires refutation of infinitely many visits to some state and that in turn requires a well-founded argument [6, 26] over transitive closure of transition relation.

Closure Certificates for Transition Invariants. Podelski and Rybalchenko, in an influential paper [26], introduced disjunctively well-founded *transition invariants* to verify programs against ω-regular properties. They defined the transition invariant as an over-approximation of the transitive closure of the transition relation, restricted to the set of reachable states. If the transition invariant restricted to pairs of accepting states is disjunctively well-founded, then they showed that no execution can visit these accepting states infinitely often, refuting the ω-regular specification. We introduce closure certificates as a functional analog of *transition invariants* and enable the use of SOS programming and SMT solvers to search for these certificates.

Intuitively, a closure certificate $\mathcal{T} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a real-valued function over the Cartesian product of the state set and itself (state pairs), such that $\mathcal{T}(x, x') \geq 0$ if x' is reachable from x. The closure certificate characterizes a transition invariant $T \subseteq \mathcal{X} \times \mathcal{X}$, with the set of initial states $\mathcal{X}_0$, in the following fashion:

$$T = \{(x, x') : \mathcal{T}(x, x') \geq 0 \text{ and } \mathcal{T}(x_0, x) \geq 0 \text{ for some } x_0 \in \mathcal{X}_0\}. \quad (1)$$

It is easy to see (Theorem 4) that the existence of a barrier certificate implies the existence of a closure certificate establishing the same property. On the other hand, to appreciate the utility of closure certificate, we show that, even for safety properties (state inviariants), it is often possible to construct a closure certificate of simpler shape (e.g., lower degree polynomials) than a barrier certificate. To demonstrate this, consider the simple finite state discrete example shown in Figure 1. Here we depict initial states with green filled circles ($\mathcal{X}_0 = \{1, 3, 5\}$) while unsafe states are shown with red filled circles ($\mathcal{X}_u = \{2, 4\}$). It is easy to see that starting from the

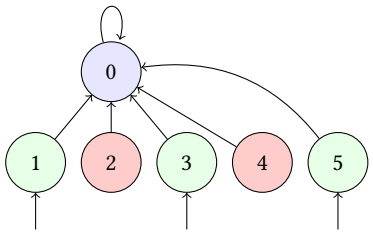

Figure 1: Illustrative example demonstrating the simplicity of closure certificates over barrier certificates

initial states, the system never visits any unsafe states. We show that while there is no *polynomial* barrier certificate of degree 2 that demonstrates the safety of the system, there is a *linear* closure certificate that does so. We note that this example can be modified to show the absence of barrier certificate for any fixed degree.

Let us suppose that there exists a polynomial barrier certificate $\mathcal{B}$ of degree 2 that acts as a proof of safety. We need $\mathcal{B}(x) \leq 0$ for every state $x \in \mathcal{X}_0$, and $\mathcal{B}(x) > 0$ for every state $x \in \mathcal{X}_u$. Applying intermediate value theorem, the function $\mathcal{B}$ needs to change signs in at least 3 points and must therefore have at least 3 roots. This supports our claim that there is no barrier certificate that is a polynomial of degree 2. On the other hand, the linear function $\mathcal{T} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defined as $\mathcal{T}(x, y) = -y$ is a closure certificate for this system. While we defer the details to later sections, from (1), it follows that this function corresponds to the transition invariant $T = \{(1, 0), (3, 0), (5, 0), (0, 0)\}$, and has no intersection with the set $\mathcal{X}_0 \times \mathcal{X}_u$. This ensures the safety of the system. We have chosen a finite-state example for illustrative purposes. This paper deals with continuous state spaces, and our case studies (cf. Section 6) will demonstrate similar advantages in continuous state spaces.

While barrier certificates can be employed to verify other, more complex objectives (e.g., liveness or general linear-time properties), their applications in such settings are often conservative [1, 15, 16]. We adapt closure certificates to validate or refute general linear-time properties. As an example, consider the so-called "persistence" property, where one wishes to verify that a system visits some region (denoted as $\mathcal{X}_{VF}$) only finitely often, or alternatively, it eventually stays within some region (the complement of $\mathcal{X}_{VF}$). We extend closure certificate conditions (Section3.2) with a "potential"-like argument. In particular, we require that for every initial state x_0 and every pair of states y and y' in the set $\mathcal{X}_{VF}$, if the system can reach from y to y', then the potential between (x_0, y') is less than the potential between (x_0, y) by a certain fixed amount. This, in turn, implies that for every execution starting from an initial state, the region $\mathcal{X}_{VF}$ can only be visited finitely often. This approach can be extended to general linear-time objectives (Section 3.3) by employing the classical automata-theoretic approach that reduces LTL verification to visiting certain states only finitely often.

Contributions. The contributions of the paper are listed next.

(1) This paper proposes a novel notion of closure certificates that act as a functional analog to transition invariants.
(2) We present SOS programming as well as SMT characterizations to search for a closure certificates within a given template (function class).
(3) We show that even when traditional barrier certificates of a some template fail to ensure safety, one can find closure certificates of the same template.
(4) We demonstrate how to use closure certificates to verify dynamical systems against LTL specifications described by ω-automata with our case studies.
(5) We show how closure certificates subsume existing barrier certificate based approaches to verify continuous-space dynamical systems against LTL specifications.

Related works. Prajna and Jadbabaie proposed the notion of barrier certificates [28] as a discretization-free approach to give guarantees of safety or reachability [30] for dynamical and hybrid systems. The results in [38] presented a state triplet approach that uses barrier certificates to verify linear temporal logic properties specified by ω-automata. This approach has since been used in the the verification and synthesis of stochastic and interconnected continuous-space systems against linear temporal logic properties [1, 2, 15, 16]. Unfortunately, the above approach is conservative in the sense that it treats the nondeterministic Büchi automaton corresponding to the negation of the LTL specification as a finite automaton and then searches for barrier certificates to disallow the transitions along accepting paths to show the accepting state is not visited. Thus, even if a system satisfies the property but visits the accepting state, then one cannot make use of the above approaches to verify the system. We show (cf. Section 5) that our approach subsumes this current approach. Podelski and Rybalchenko [26], proposed a notion of transition invariants and demonstrated their use in verifying the liveness properties of programs as well as in verifying programs against ω-regular properties. Transition invariants have also been used in [27] to give guarantees of stability for hybrid systems. Here, they make use of a reachability analysis tool to determine the over-approximation of reachable states and then establish a Lyapunov guarantee on the transition invariant to ensure the stability of the system. The results in [33] consider a notion of relational abstraction that is similar to transition invariants to give guarantees for safety. Due to lack of space, more detailed proofs and numerical values of coefficients for the closure certificates computed can be found in [22]. An extension of this work where neural networks are used to represent closure certificates can be found in [23].

While this paper focuses on abstraction-free approaches to verify LTL properties specified as ω-automata, abstraction-based techniques have also been used in the verification and synthesis of continuous-space dynamical systems against LTL properties such as the results in [14, 18, 20, 31]. These results build a finite state abstraction and then making use of model checking techniques [3, 36].

2 PRELIMINARIES

We use $\mathbb{N}$ and $\mathbb{R}$ to denote the set of natural numbers and reals. For $a \in \mathbb{R}$, we use $\mathbb{R}_{\geq a}$ and $\mathbb{R}_{>a}$ to denote the intervals $[a, \infty)$ and (a, ∞), respectively, and similarly, for any natural number $n \in \mathbb{N}$, we use $\mathbb{N}_{\geq n}$ to denote the set of natural numbers greater than or equal to n. Given a set A, sets A^* and A^ω denote the set of finite and countably infinite sequences of elements in A, while $|A|$ denotes

the cardinality of the set. If $A \subseteq B$, and the set B can be inferred from the context, we denote the complement $B \setminus A$ simply as $\overline{A}$. We call a function $f : A \to \mathbb{R}$ *bounded* if there exists $l, u \in \mathbb{R}$, such that $l \leq f(a) \leq u$ for every $a \in A$.

Given a relation $R \subseteq A \times B$, and an element $a \in A$, we use $R(a)$ to denote the set $\{b \mid (a, b) \in R\}$. We use the notation $(a_1, a_2, \ldots, a_n) \in A^*$ for finite length sequences and $\langle a_0, a_1, \ldots, \rangle \in A^\omega$ for ω-sequences. Let $\mathsf{Inf}(s)$ be the set of infinitely often occurring elements in the sequence $s = \langle a_0, a_1, \ldots, \rangle$. Given an infinite sequence $s = \langle a_0, a_1, \ldots, \rangle$, and two natural numbers $i, j \in \mathbb{N}$ where $i \leq j$, we use $s[i, j]$ to indicate the finite sequence $(a_i, a_{i+1}, \ldots, a_j)$, and $s[i, \infty)$ to indicate the infinite sequence $\langle a_i, a_{i+1}, \ldots \rangle$. Finally, we use $s[i]$ to denote the ith element in the sequence s, *i.e.*, given an infinite sequence $s = \langle a_0, a_1, \ldots \rangle$, we have $s[i] = a_i$ for any $i \in \mathbb{N}$.

2.1 Discrete-time Dynamical System

A (discrete-time dynamical) system $\mathfrak{S}$ is a tuple $(\mathcal{X}, \mathcal{X}_0, f)$, where $\mathcal{X}$ (possibly infinite) denotes the state set, $\mathcal{X}_0 \subseteq X$ denotes a set of initial states, and $f \subseteq \mathcal{X} \times \mathcal{X}$ is the state transition relation. The state evolution of the system is given as the following:

$$\mathfrak{S} : x(t+1) \in f(x(t)). \tag{2}$$

If for every $x \in \mathcal{X}$, we have $|f(x)| = 1$, then we consider the transition relation f to be a *state transition function* that uniquely determines the next state. Abusing notation, we use f for both a set-valued map when it is a relation, and a transition function when it is a function. Throughout the paper, we assume that state sets of the systems under consideration are compact.

A *state sequence* is an infinite sequence $\langle x_0, x_1, \ldots, \rangle \in \mathcal{X}^\omega$ where $x_0 \in \mathcal{X}_0$, and $x_{i+1} \in f(x_i)$, for all $i \in \mathbb{N}$. We associate a labelling function $\mathcal{L} : \mathcal{X} \to \Sigma$ which maps each state of the system to a letter in a finite alphabet Σ. This naturally generalizes to mapping a state sequence of the system $\langle x_0, x_1, \ldots, \rangle \in \mathcal{X}^\omega$ to a trace or word $w = \langle \mathcal{L}(x_0), \mathcal{L}(x_1), \ldots, \rangle \in \Sigma^\omega$. For notational convenience, given a state $x \in \mathcal{X}$, we use x' to indicate a state in $f(x)$. Let $TR(\mathfrak{S}, \mathcal{L})$ denote the set of all traces of $\mathfrak{S}$ under the labeling map $\mathcal{L}$.

2.2 Specifications

We are interested in deductive verification of linear-time properties over discrete-time dynamical systems. We study increasingly complex specifications from safety, and persistence, to LTL and ω-regular specifications.

Safety. We say that a system is safe with respect to a set of unsafe states $\mathcal{X}_u \subseteq \mathcal{X}$ if for any state sequence $\langle x_0, x_1, \ldots, \rangle$ we have $x_i \notin \mathcal{X}_u$ for all $i \in \mathbb{N}$. An important technique to verify the safety of the system is to synthesize *barrier certificates* [28].

Definition 2.1 (Barrier Certificate). *A function $\mathcal{B} : \mathcal{X} \to \mathbb{R}$ is a barrier certificate for $\mathfrak{S} = (\mathcal{X}, \mathcal{X}_0, f)$ with respect to a set of unsafe states $\mathcal{X}_u$ if:*

$$\mathcal{B}(x) \leq 0 \qquad \text{for all } x \in \mathcal{X}_0 \tag{3}$$

$$\mathcal{B}(x) > 0 \qquad \text{for all } x \in \mathcal{X}_u \tag{4}$$

$$\big(\mathcal{B}(x) \leq 0\big) \implies \big(\mathcal{B}(x') \leq 0\big) \qquad \text{for all } x \in \mathcal{X}, \text{ and } x' \in f(x) \tag{5}$$

Theorem 1 (Barrier certificates imply safety [28]). *For a system $\mathfrak{S}$ with unsafe states $\mathcal{X}_u$, the existence of a barrier certificate $\mathcal{B}$ implies its safety.*

Persistence (refuting recurrence). We say that a system visits a region $\mathcal{X}_{VF} \subseteq \mathcal{X}$ only finitely often if for any state sequence $\langle x_0, x_1, \ldots, \rangle$ there exists some $i \in \mathbb{N}$, such that for all $j \geq i$, $j \in \mathbb{N}$, we have $x_j \notin \mathcal{X}_{VF}$. Observe that if a system is safe with respect to a set of unsafe states $\mathcal{X}_u$, then it satisfies the persistence objective. Thus one can make use of barrier certificates as a sound (not complete) way to ensure persistence. Another approach to ensure persistence is to fix the value of i to some constant value, and then search for a barrier certificate over the system and an augmented value. Such approaches are common in bounded verification and synthesis approaches as in [8, 34] for finite state systems.

Linear Temporal Logic (LTL). Formulae in LTL [25] are defined with respect to a set of finite atomic propositions AP that are relevant to our system. Let $\Sigma = 2^{AP}$ denote the powerset of atomic propositions. A trace $w = \langle w_0, w_1, \ldots, \rangle \in \Sigma^\omega$ is an infinite sequence of sets of atomic propositions. The syntax of LTL can be given via the following grammar:

$$\phi := \top \mid a \mid \neg\phi \mid \mathsf{X}\phi \mid \phi \mathsf{U} \phi,$$

where $\top$ indicates true, $a \in AP$ denotes an atomic proposition, symbols $\wedge$, $\neg$ denote the logical AND and NOT operators respectively. The temporal operators next, and until are denoted by X, and U respectively. The above operators are sufficient to derive the logical OR ($\vee$) and implication ($\implies$), and the temporal operators release (R), eventually (F) and always (G) respectively.

We inductively define the semantics of an LTL formula with respect to trace w as follows:

$$w \models a \qquad \text{if } a \in w[0] \tag{6}$$

$$w \models \phi_1 \wedge \phi_2 \qquad \text{if } w \models \phi_1 \text{ and } w \models \phi_2 \tag{7}$$

$$w \models \neg\phi \qquad \text{if } w \not\models \phi \tag{8}$$

$$w \models \mathsf{X}\phi \qquad \text{if } w[1, \infty) \models \phi \tag{9}$$

$$w \models \phi_1 \mathsf{U} \phi_2 \qquad \text{if there exists } i \in \mathbb{N} \text{ such that } w[0, i] \models \phi_1 \text{ and } w[i+1, \infty) \models \phi_2 \tag{10}$$

To reason about whether a system satisfies a property specified in LTL, we associate a labelling function $\mathcal{L} : \mathcal{X} \to \Sigma$ which maps each state of the system to a letter in the finite alphabet Σ. This naturally generalizes to mapping a state sequence of the system $\langle x_0, x_1, \ldots, \rangle \in \mathcal{X}^\omega$ to a trace $w = \langle \mathcal{L}(x_0), \mathcal{L}(x_1), \ldots, \rangle \in \Sigma^\omega$. Let $TR(\mathfrak{S}, \mathcal{L})$ denote the set of all traces of $\mathfrak{S}$ under the labeling map $\mathcal{L}$. Then the system $\mathfrak{S}$ satisfies an LTL property ϕ under labeling map $\mathcal{L}$ if for all $w \in TR(\mathfrak{S}, \mathcal{L})$, we have $w \models \phi$. We denote this as $\mathfrak{S} \models_{\mathcal{L}} \phi$ and infer the labeling map from context. As LTL subsume safety and persistence, one can formulate these as LTL formulae.

Nondeterminstic Büchi Automata. A nondeterminstic Büchi automaton (NBA) $\mathcal{A}$ is a tuple $(\Sigma, Q, Q_0, \delta, Acc)$, where:

- Σ is the alphabet,
- Q a finite set of states,
- $Q_0 \subseteq Q$ an initial set of states,
- $\delta \subseteq Q \times \Sigma \times Q$ the transition relation, and
- $Acc \subseteq Q$ denotes a set of accepting states.

A run of the automaton $\mathcal{A} = (\Sigma, Q, q_0, \delta, Acc)$ over a trace $w = \langle \sigma_0, \sigma_1, \sigma_2 \ldots, \rangle \in \Sigma^\omega$, is an infinite sequence of states characterized as $\rho = \langle q_0, q_1, q_2, \ldots, \rangle \in Q^\omega$ with $q_0 \in Q_0$ and $q_{i+1} \in \delta(q_i, \sigma_i)$. An NBA $\mathcal{A}$ is said to accept a trace w, if there exists a run ρ over w where $\mathsf{Inf}(\rho) \cap Acc \neq \emptyset$.

It is well known [37] that given an LTL formula ϕ over a set of atomic propositions AP, one can construct an NBA $\mathcal{A}$ such that $w \in L(\mathcal{A})$ iff $w \models \phi$. An automata-theoretic technique to determine whether $\mathfrak{S} \models_{\mathcal{L}} \phi$ is to first find the NBA $\mathcal{A}$ that represents $\neg\phi$, and then ensure that $\mathfrak{S} \not\models_{\mathcal{L}} \neg\phi$ by showing that no trace of the system is accepted by the NBA $\mathcal{A}$. While converting an LTL formula to an NBA is exponential in the size of the formula, negating an LTL formula has a complexity that is linear in its size.

3 CLOSURE CERTIFICATES

Podelski and Rybalchenko [26] introduced the notion of transition invariants as an over-approximation of the transitive closure of the transition relation, restricted to the set of reachable states. If the transition invariant restricted to pairs of accepting states is disjunctively well-founded, then they showed that no execution can visit these accepting states infinitely often, refuting the ω-regular specification. In this section, we introduce closure certificates (CC) as a functional analog of *transition invariants*.

Recall that barrier certificates are functional analogs to inductive state invariants in the following way: all the initial states are within the zero level set of the barrier certificate, and, given any state that is within the zero level set, its successor according to the transition relation is also in the zero level set. Our definition of closure certificates are a functional analog to inductive *transition invaraints*. We study their use in the verification of safety, persistence (refuting recurrence), and LTL specifications.

3.1 Closure Certificates for Safety

We first define closure certificates for safety.

Definition 3.1 (Closure Certificate for Safety). *Consider a system $\mathfrak{S} = (\mathcal{X}, \mathcal{X}_0, f)$. A function $\mathcal{T} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a Closure Certificate (CC) for a set of unsafe states $\mathcal{X}_u$ if there exists a value $\xi \in \mathbb{R}_{>0}$ such that for all states $x, y \in \mathcal{X}$, $x' \in f(x)$, and states $x_0 \in \mathcal{X}_0$ and $x_u \in \mathcal{X}_u$, we have:*

$$\big(\mathcal{T}(x, x') \geq 0\big) \tag{11}$$

$$\big(\mathcal{T}(x', y) \geq 0\big) \implies \big(\mathcal{T}(x, y) \geq 0\big), \text{ and} \tag{12}$$

$$\big(\mathcal{T}(x_0, x_u) \leq -\xi\big). \tag{13}$$

The existence of a closure certificate implies the safety of the system $\mathfrak{S} = (\mathcal{X}, \mathcal{X}_0, f)$ as shown next.

Theorem 2 (Closure Certificate imply Safety). *Consider a system $\mathfrak{S}$. The existence of a function $\mathcal{T} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that satisfies conditions* (11)-(13) *implies its safety.*

Proof. Let us assume that there exists a trace of the system $\langle x_0, \ldots, x_u, \ldots \rangle$ that reaches an unsafe state $x_u \in \mathcal{X}_u$ from some initial state x_0. From condition (11), we have $\mathcal{T}(x_i, x_{i+1}) \geq 0$ for all $i \in \mathbb{N}$, and from condition (12) and induction, we have $\mathcal{T}(x_0, x_i) \geq 0$ for all $i \in \mathbb{N}$. Thus we must have $\mathcal{T}(x_0, x_u) \geq 0$ as $x_j = x_u$ for some $j \in \mathbb{N}$. According to condition (13), $\mathcal{T}(x_0, x_u) \leq -\xi$, where $\xi \in \mathbb{R}_{>0}$, which is in contradiction to the previous inequality. □

Observe that closure certificates are defined over pairs of states of the system rather than just over the states of the system. Hence, searching for a closure certificate suffers computationally more than a search for a barrier certificate. On the other hand, for a certain template of functions (e.g., linear or quadratic), one might be able to find closure certificates, even when barrier certificates of the same template do not exist. In particular, we have the following result:

Theorem 3 (Simplicity of Closure Certificates). *For any natural number $d \in \mathbb{N}$, there exists a system $\mathfrak{S}$ with unsafe set of states $\mathcal{X}_u$ that cannot be shown to be safe by a polynomial barrier certificate of degree d but can be shown to be safe by a linear closure certificate.*

Proof. Consider a system $\mathfrak{S} = (\mathcal{X}, \mathcal{X}_0, f)$, with $\mathcal{X} = [0, (2d+2)]$ as the state set, $\mathcal{X}_0 = \{1, 3, \ldots, (2d+1)\}$ as the initial set of states, and a constant transition relation $f(x) = \{0\}$ for every state $x \in \mathcal{X}$. Let the set of unsafe states be $\mathcal{X}_u = \{2, 4, \ldots, (2d+2)\}$. We observe that the system is trivially safe.

Let us suppose there exists a polynomial barrier certificate $\mathcal{B} : \mathcal{X} \to \mathbb{R}$ of degree d that acts as a proof of safety. From conditions (3) and (4), we have $\mathcal{B}(x) \leq 0$ for every state $x \in \mathcal{X}_0$, and $\mathcal{B}(x) > 0$ for every state $x \in \mathcal{X}_u$. Applying intermediate value theorem, the function $\mathcal{B}$ needs to change signs in at least $(d+1)$ points and must therefore have at least $(d+1)$ roots. This contradicts our assumption that $\mathcal{B}$ is a polynomial of degree d.

Consider the function $\mathcal{T} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defined as $\mathcal{T}(x, y) = -y$. Observe that $0 \in f(x)$ for all $x \in \mathcal{X}$, and that $\mathcal{T}(x, y) \geq 0$ only when $y \leq 0$. This implies conditions (11) and (12) are satisfied. Further for every state $x_u \in X_u$ and every state $x_0 \in X_0$, we have $\mathcal{T}(x_0, x_u) \leq -2$. Thus condition (13) also holds. We conclude that the function $\mathcal{T}$ is a closure certificate and acts as a proof that the system is safe. □

We should note that while the proof of the above Theorem relied on showing that no barrier certificate exists for a finite state system, one can employ similar techniques for a continuous space example. Consider the system $\mathfrak{S} = (\mathcal{X}, \mathcal{X}_0, f)$, where $\mathcal{X} = \mathbb{R}$ denotes the state set, $\mathcal{X}_0 = \{0, \frac{1}{4}, \ldots \frac{1}{2^{d+2}}\}$ indicates the initial set of states, and $f(x) = \{x+1\}$ denotes the transition relation for every state $x \in \mathcal{X}$. Let the set of unsafe states be $\mathcal{X}_u = \{\frac{1}{2}, \ldots, \frac{1}{2^{d+3}}\}$, then there exists no polynomial function of degree d that acts as a barrier certificate for the above function. However the function $\mathcal{T}(x, y) = y - x - 1$ acts as a closure certificate that ensures the system starting from the initial state does not reach the unsafe set of states. An illustration of this example for degree 2 can be found in [22, Appendix A].

Previously, we discussed how one can use closure certificates even when barrier certificates fail. We now show that if a system can be guaranteed to be safe via barrier certificates, then it can be guaranteed via closure certificates as well.

Theorem 4 (Expressiveness). *Consider a system $\mathfrak{S} = (\mathcal{X}, \mathcal{X}_0, f)$, with unsafe set of states $\mathcal{X}_u$. Given a barrier certificate $\mathcal{B} : \mathcal{X} \to \mathbb{R}$ (Definition 2.1), one can compute a closure certificate $\mathcal{T} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.*

Proof. Let $\gamma \in \mathbb{R}_{>0}$. We define the function $\mathcal{T} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ as:

$$\mathcal{T}(x,y) = \begin{cases} 0 & \text{if } \mathcal{B}(x) > 0 \text{ or } \mathcal{B}(y) \leq 0, \\ -\gamma & \text{otherwise.} \end{cases}$$

We now show that $\mathcal{T}$ is a CC with $\xi = \gamma$. Let us suppose that $\mathcal{T}(x,x') < 0$ for some $x \in \mathcal{X}$. For this to be true, we must have $\mathcal{B}(x) \leq 0$, and $\mathcal{B}(x') > 0$, however, this contradicts condition (5) and so $\mathcal{T}$ must satisfy condition (11). Second, suppose $\mathcal{T}(x',y) \geq 0$, and $\mathcal{T}(x,y) < 0$. Then $\mathcal{B}(x) \leq 0$, $\mathcal{B}(y) > 0$, and one of $\mathcal{B}(x') > 0$ or $\mathcal{B}(y) \leq 0$. Since both $\mathcal{B}(y) \leq 0$ and $\mathcal{B}(y) > 0$ cannot be true, we must have $\mathcal{B}(x) \leq 0$, and $\mathcal{B}(x') > 0$, which again contradicts condition (5), and so condition (12) must hold. Finally, consider $\mathcal{T}(x_0, x_u)$. From conditions (3) and (4), we have $\mathcal{B}(x_0) \leq 0$, and $\mathcal{B}(x_u) > 0$, and, hence, by definition $\mathcal{T}(x_0,x_u) = -\xi$ satisfies condition (13). □

3.2 Closure Certificates for Persistence

Similar to how closure certificates are used to guarantee safety, one may use closure certificates to show a region is visited finitely often. This relies on showing that the closure certificate is well-founded, similar to the condition used in [27].

Definition 3.2 (Closure Certificates for Persistence). *Consider a system $\mathfrak{S} = (\mathcal{X}, \mathcal{X}_0, f)$. A bounded function $\mathcal{T} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a Closure Certificate (CC) for $\mathfrak{S}$ with set of states $\mathcal{X}_{VF} \subseteq \mathcal{X}$, that must be visited finitely often if there exists a value $\xi \in \mathbb{R}_{>0}$ such that for all states $x, y \in \mathcal{X}$, $x' \in f(x)$, $x_0 \in \mathcal{X}_0$, and all states $y', y'' \in \mathcal{X}_{VF}$ we have:*

$$\big(\mathcal{T}(x,x') \geq 0\big), \tag{14}$$

$$\big(\mathcal{T}(x',y) \geq 0\big) \implies \big(\mathcal{T}(x,y) \geq 0\big) \text{ and} \tag{15}$$

$$\big(\mathcal{T}(x_0,y') \geq 0\big) \wedge \big(\mathcal{T}(y',y'') \geq 0\big) \implies \big(\mathcal{T}(x_0,y'') \leq \mathcal{T}(x_0,y') - \xi\big). \tag{16}$$

Theorem 5 (Closure Certificates imply Persistence). *Consider a system $\mathfrak{S}$. The existence of a function $\mathcal{T} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that satisfies conditions* (14)-(16) *implies that the traces of the system visit the set $\mathcal{X}_{VF}$ finitely often.*

Proof. Let us suppose that there is some trajectory $\langle x_0, x_1, \ldots, \rangle$ of the system that starts from state $x_0 \in \mathcal{X}_0$ and visits $\mathcal{X}_{VF}$ infinitely often. Let the infinite sequence $\langle y_0, y_1, \ldots, \rangle$ denote the states that are visited in $\mathcal{X}_{VF}$ in that order, *i.e.*, the trajectory is $\langle x_0, \ldots, y_0, \ldots, y_1, \ldots \rangle$. From conditions (14) and (15), we have $\mathcal{T}(x_0, y_i) \geq 0$ and $\mathcal{T}(y_i, y_j) \geq 0$ for all indices $j > i$, $i, j \in \mathbb{N}$. As we assume the function $\mathcal{T}$ to be bounded, there exists some $\mathcal{T}^* \in \mathbb{R}$, such that $\mathcal{T}(x,y) \leq \mathcal{T}^*$ for every pair of states $x, y \in \mathcal{X}$. Note that $\mathcal{T}(x_0, y_0) \leq T^*$. From condition (16), and induction, we have

$$\mathcal{T}(x_0, y_i) \leq \mathcal{T}(x_0, y_0) - i\xi \leq T^* - i\xi.$$

As this is true for all $i \in \mathbb{N}$, and we have $\xi \in \mathbb{R}_{>0}$, there must exist some $j \in \mathbb{N}$ such that $\mathcal{T}(x_0, y_j) < 0$. This is a contradiction. □

3.3 Closure Certificates for LTL Specifications

To verify whether the system satisfies a desired LTL formula ϕ, we first construct the NBA $\mathcal{A} = (Q, Q_0, \delta, Q_{Acc})$ that represents the complement of the specification $\neg\phi$. Observe that the state set of the NBA is finite, and therefore we can denote the set Q as the set $\{0, 1, \ldots, |Q| - 1\}$. We then construct the product $\mathfrak{S} \otimes \mathcal{A} = (\mathcal{X}', \mathcal{X}_0', f')$ of the system $\mathfrak{S} = (\mathcal{X}, \mathcal{X}_0, f)$ with the NBA $\mathcal{A}$, where:

- $\mathcal{X}' = \mathcal{X} \times \{0, \ldots, |Q| - 1\}$ indicates the state set
- $\mathcal{X}_0' = \mathcal{X}_0' \times \{q_0 \mid q_0 \in Q_0\}$ indicate the initial set of states.
- the state transition relation f' is defined as :

$$f'((x,q_i)) = \big\{(x',q_j) \mid q_j \in \delta(q_i, \mathcal{L}(x)), \text{ and } x' \in f(x)\big\}.$$

To verify whether a given system satisfies a desired LTL property, we make use of a closure certificate on the product $\mathfrak{S} \otimes \mathcal{A}$.

Definition 3.3 (Closure Certificate for LTL). *Consider a system $\mathfrak{S} = (\mathcal{X}, \mathcal{X}_0, f)$ and NBA $\mathcal{A} = (Q, Q_0, \delta, Acc)$ representing the complement of an LTL formula ϕ. A bounded function $\mathcal{T} : \mathcal{X} \times Q \times \mathcal{X} \times Q \to \mathbb{R}$ is a closure certificate for $\mathfrak{S}$ and NBA $\mathcal{A}$ if there exists a value $\xi \in \mathbb{R}_{>0}$ such that for all states $x, y \in \mathcal{X}$, $x' \in f(x)$ and states $i, j \in Q$, and $i' \in \delta(i, \mathcal{L}(x))$, we have:*

$$\Big(\mathcal{T}\big((x,i),(x',i')\big) \geq 0\Big) \tag{17}$$

$$\Big(\mathcal{T}\big((x',i'),(y,j)\big) \geq 0\Big) \implies \Big(\mathcal{T}\big((x,i),(y,j)\big) \geq 0\Big) \tag{18}$$

and for all states $x_0 \in \mathcal{X}_0$, $s \in Q_0$, and $\ell, \ell' \in Acc$, we have:

$$\Big(\mathcal{T}\big((x_0,s),(y,\ell)\big) \geq 0\Big) \wedge \Big(\mathcal{T}\big((y,\ell),(y',\ell')\big) \geq 0\Big) \implies \Big(\mathcal{T}\big((x_0,s),(y',\ell')\big) \leq \mathcal{T}\big((x_0,s),(y,\ell)\big) - \xi\Big). \tag{19}$$

Now, we provide the next result of the paper on the verification of LTL specifications using closure certificates on $\mathfrak{S} \otimes \mathcal{A}$.

Theorem 6 (Closure Certificates verify LTL). *Consider a system $\mathfrak{S}$ and an LTL formula ϕ. Let NBA $\mathcal{A}$ represent the complement of the specification, i.e, $\neg\phi$. The existence of a closure certificate satisfying conditions* (17)-(19) *implies that $\mathfrak{S} \models_{\mathcal{L}} \phi$.*

Proof. Observe that a CC $\mathcal{T}$ satisfying conditions (17) to (19) is a CC for the product of $\mathfrak{S}$ and $\mathcal{A}$. From Theorem 5, we observe that the product system visits accepting states finitely often and so we infer that no trace of the system is in the language of the NBA $\mathcal{A}$. The proof is now complete. □

4 SYNTHESIZING CLOSURE CERTIFICATES

This section presents two approaches to synthesize closure certificates when the dynamical systems under study have state sets which are subsets of $\mathbb{R}^n$, *i.e.*, $\mathcal{X} \subseteq \mathbb{R}^n$, and the transition function f is a polynomial. The first approach we consider is using a counterexample guided approach via Satisfiability Modulo Theory (SMT) solvers [4], while the second makes use of standard sum-of-squares (SOS) [24] approaches to find closure certificates similar to barrier certificates. In the following sections we describe the relevant conditions for persistence and verifying ω-regular objectives. The conditions for safety can be recovered in a straightforward manner and are thus ommitted from the following discussion.

4.1 SMT-based Approach

Counterexample-guided Inductive Synthesis (CEGIS) [35] has seen significant use in the synthesis of barrier certificates. We thus consider conditions to provide a CEGIS approach to find closure certificates. To find a CC as in Definition 3.2, we first fix the template of

the CC to be a linear combination of user-defined basis functions:

$$\mathcal{T}(x,y) = \sum_{m=1}^{z} c_m p_m(x,y),$$

where functions p_m are user-defined analytical basis functions over the state variables x and y and $c_1, \ldots, c_z$ are the coefficients. As an example, we can consider $c_1, \ldots, c_z$ to be real values, and $\mathcal{T}(x,y)$ to be a polynomial. In such a case, the functions $p_1, \ldots, p_m$ are monomials over x and y. Note that if the values of $x, y \in \mathcal{X}$ are fixed, then the only decision variables in $\mathcal{T}(x,y)$ are the coefficients c_m, $m \in \{1, \ldots, z\}$.

We sample $2N$ points from the state set $\mathcal{X}$ of the system to create the sets $S_1 = \{x_1, \ldots, x_N\}$, and $S_2 = \{y_1, \ldots, y_N\}$, and sample $3N$ points from $\mathcal{X}_0$, $\mathcal{X}_{VF}$, and $\mathcal{X}_{VF}$, respectively, to create sets $S_3 = \{x_{0,1}, \ldots, x_{0,N}\}$, and $S_4 = \{z_1, \ldots, z_{2N}\}$, respectively. We then encode the constraints of the closure certificate for every pair of points as an SMT-query over the theory of linear real arithmetic (LRA) [7] using z3 [21] as follows:

$$\bigwedge_{x \in S_1} \Big(\mathcal{T}(x,x') \geq 0 \Big), \tag{20}$$

$$\bigwedge_{x \in S_1, y \in S_2} \Big(\big(\mathcal{T}(x',y) \geq 0\big) \implies \big(\mathcal{T}(x,y) \geq 0\big) \Big), \text{ and} \tag{21}$$

$$\bigwedge_{x_0 \in S_3, z, z' \in S_4} \Big(\big(\mathcal{T}(x_0,z) \geq 0\big) \wedge \big(\mathcal{T}(z,z') \geq 0\big) \tag{22}$$

$$\implies \big(\mathcal{T}(x_0,z') \leq \mathcal{T}(x_0,z) - \xi\big) \Big), \tag{23}$$

where $x' = f(x_{k_1})$ indicates the next state from x_{k_1} following the transition function. We lastly add a constraint of ξ being larger than some small positive value and then find values $c_1, \ldots, c_z$ for the coefficients and substitute them as a candidate CC $\mathcal{T}(x,y)$.

To determine if this candidate is in fact a CC, we now try to find elements $x, y, x_0, z, z' \in \mathcal{X}$ such that one of the conditions (14)-(16) does not hold. We do this by encoding the negation of these conditions as an SMT query. If such a counterexample is found, we add them to the respective set and repeat the process. If no counterexample is found, then we conclude that this is a CC.

Instead of using an SMT solver to find a candidate CC, we can instead run our CEGIS loop quicker by strengthening conditions (15)-(16) as inequalities of the form:

$$\tau_1 \mathcal{T}(x',y) \leq \mathcal{T}(x,y), \tag{24}$$

$$\mathcal{T}(x_0,y) - \xi - \mathcal{T}(x_0,y') \geq \tau_2 \mathcal{T}(x_0,y) + \tau_3 \mathcal{T}(y,y'), \tag{25}$$

for all states $x_0 \in \mathcal{X}_0$, and $y, y' \in \mathcal{X}_{VF}$. where $\tau_1, \tau_2, \tau_3 \in \mathbb{R}_{\geq 0}$ are fixed nonnegative values. The satisfaction of conditions (24) and (25) implies the satisfaction of conditions (15) and (16), and the search for a candidate CC can be cast as a linear program instead. This allows one to use a linear programming solver (such as Gurobi [12]) to find a candidate CC instead. We then find a counterexample via SMT queries similar to the earlier approach, and then add the counterexample to our linear program, and search for a candidate CC again. While conditions (24) and (25) are more conservative, the search for a candidate is much quicker.

We adopt a similar approach to find a CC for the synchronized product as in Definition 3.3, that acts as a proof that the traces of the system satisfy an LTL property whose negation is specified by the language of an NBA $\mathcal{A} = (\Sigma, Q, Q_0, \delta, Acc)$. In this setting, we assume our closure certificates to be piecewise with respect to pairs of states of NBA $\mathcal{A}$. Each piecewise component is then considered to be a linear-combination of some user-defined basis functions. For every pair of states $i, j \in Q$, we denote the corresponding piecewise component as $\mathcal{T}_{i,j}$. We define each piecewise component as:

$$\mathcal{T}_{i,j}(x,y) = \sum_{m=1}^{z} c_{m.i.j} p_{m,i,j}(x,y),$$

where the functions $p_{m,i,j}$ are user-defined basis functions over the states $x, y \in \mathcal{X}$, and $c_{m,i,j}$ are the coefficients. We then encode the constraints as the following conjunctions for every state $x \in S_1$, $y \in S_2$, $x_0 \in S_3$ and $z, z' \in S_2$ as well as every state $i, j \in Q$ such that $i' \in \delta(i, \mathcal{L}(x))$, and states $s \in Q_0$ and $\ell, \ell' \in Acc$:

$$\bigwedge_{x \in S_1} \Big(\mathcal{T}_{i,i'}(x,x') \geq 0 \Big), \tag{26}$$

$$\bigwedge_{x \in S_1, y \in S_2} \Big(\big(\mathcal{T}_{i',j}(x,y) \geq 0\big) \implies \big(\mathcal{T}_{i,j}(x,y) \geq 0\big) \Big), \text{ and} \tag{27}$$

$$\bigwedge_{x_0 \in S_3, z, z' \in S_2} \Big(\big(\mathcal{T}_{s,\ell}(x_0,z) \geq 0\big) \wedge \big(\mathcal{T}_{\ell,\ell'}(y,z') \geq 0\big) \implies \big(\mathcal{T}_{s,\ell'}(x_0,z') \leq \mathcal{T}_{s,\ell}(x_0,z) - \xi\big) \Big). \tag{28}$$

In general there is no guarantee of termination when using a CEGIS approach for uncountable state sets. However, one may strengthen the conditions as specified in [19] to guarantee termination of the CEGIS loop. Instead of using a CEGIS approach, one may also encode the conditions in an SMT solver over the nonlinear theory of reals [9] such as dReal [10] or z3 [17] to search for CCs. While all the above approaches are NP-hard [5, 10, 17], we find the CEGIS approach to work better in practice compared to searching for a solution in the nonlinear theory of reals.

Note that barrier certificates face many of the same challenges when using a CEGIS approach. Computationally, however, closure certificates take more time in practice as these are defined over pairs of states rather than over a single state, and so suffer more when the dimension of the state set increases. We should add that we have not considered the complexity for finding the NBA representing the complement of the specification, but rather assume this NBA to be given. While the complexity of NBA complementation is EXPTIME [32], it takes linear time to complement an LTL formula. However converting an LTL formula to an NBA has exponential complexiy in the size of the formula [37].

4.2 Sum-of-Squares based Approach

The technique of using semidefinite programming [24] and casting the search for a standard barrier certificates [28] as SOS polynomials is particularly important due to the simpler complexity of computation when compared to CEGIS approaches. We show how one may adopt a SOS approach to find closure certificates. To do so, we first note that a set $A \subseteq \mathbb{R}^n$ is semi-algebraic if it can be defined with the help of a vector of polynomial inequalities $h(x)$ as $A = \{x \mid h(x) \geq 0\}$, where the inequalities is interpreted component-wise.

To adopt a SOS approach to find CCs as in Definition 3.2, we consider the sets $\mathcal{X}$, $\mathcal{X}_0$, and $\mathcal{X}_{VF}$ to be semi-algebraic sets defined with the help of vectors of polynomial inequalities g_A, g_0, and g_{VF}, respectively. As these sets are semi-algebraic, the sets $\mathcal{X} \times \mathcal{X}$ and $\mathcal{X}_0 \times \mathcal{X}_{VF} \times \mathcal{X}_{VF}$ are semi-algebraic as well. Let their corresponding vectors be g_B and g_C, respectively. Furthermore, we assume that the user-defined basis functions p_m are monomials and again strengthen the implications in conditions (15)-(16) to conditions (24)-(25). Then the search for a CC as in Definition 3.2 reduces to showing that the following polynomials are sum-of-squares:

$$\mathcal{T}(x, x') - \lambda_A^T(x) g_A(x), \tag{29}$$

$$\mathcal{T}(x, y) - \tau_1 \mathcal{T}(x', y) - \lambda_B^T(x, y) g_B(x, y), \text{ and} \tag{30}$$

$$\mathcal{T}(x, y') - \xi - \tau_2 \mathcal{T}(x, y) - \tau_3 \mathcal{T}(y, y') - \lambda_C^T(x, y, y') g_C(x, y, y'), \tag{31}$$

where $x' = f(x)$, the multipliers λ_A, λ_B, λ_C, are sum-of-squares over the state variable x, the state variables x, y, and the state variables x, y, y' over the sets $\mathcal{X}$, $\mathcal{X} \times \mathcal{X}$, and $\mathcal{X}_0 \times \mathcal{X}_{VF} \times \mathcal{X}_{VF}$ respectively, and ξ, τ_1, τ_2, and $\tau_3 \in \mathbb{R}_{>0}$ are positive values.

Lemma 7. *Assume the sets $\mathcal{X}$, $\mathcal{X}_0$, and $\mathcal{X}_{VF}$ are semi-algebraic, and there exists a sum-of-squares polynomial $\mathcal{T}(x, y)$ satisfying conditions* (29)-(31). *Then the function $\mathcal{T}(x, y)$ is a CC satisfying conditions* (14)-(16).

Since there are finitely many letters $\sigma \in \Sigma$, without loss of generality, one can partition the set $\mathcal{X}$ into finitely many partitions $\mathcal{X}_{\sigma_1}, \ldots, \mathcal{X}_{\sigma_p}$, where for all $x \in \mathcal{X}_{\sigma_m}$, we have $\mathcal{L}(x) = \sigma_m$. Given an element $\sigma_m \in \Sigma$, we can uniquely characterize the relation δ_{σ_i} as $(q_i', q_i) \in \delta_{\sigma_i}$ if and only if $q_i' \in \delta(q_i, \sigma_i)$. Assume that the sets $\mathcal{X}$, $\mathcal{X}_0$, and $\mathcal{X}_{\sigma_m}$, for all σ_m, are semi-algebraic and characterized by polynomial vectors of inequalities $g(x)$, $g_0(x)$, and $g_{\sigma_m,A}(x)$, respectively. Furthermore, we consider polynomial vectors of inequalities $g_{(\sigma_m),B}(x, y)$ over the product space $\mathcal{X} \times \mathcal{X}$, and $g_{(\sigma_m,C)}(x, y, y')$ over $\mathcal{X}_0 \times \mathcal{X} \times \mathcal{X}$, respectively. Let the state transition function $f : \mathcal{X} \to \mathcal{X}$ be a polynomial function. Now, one can reduce the search for a CC to showing that the following polynomials are SOS for all states $x, y, y' \in \mathcal{X}$, $x' = f(x)$, and $x_0 \in \mathcal{X}_0$, and states $i, j \in Q$, $s \in Q_0$, and $\ell, \ell' \in Acc$, and letters $\sigma_m \in \Sigma$, such that $i' \in \delta_{\sigma_m}(i)$:

$$\mathcal{T}_{i',i}(x, x') - \lambda_{\sigma_m,A}^T(x) g_{\sigma_m,A}(x), \tag{32}$$

$$\mathcal{T}_{i',j}(x, y) - \tau_1 \mathcal{T}_{i,j}(x', y) - \lambda_{\sigma_m,B}^T(x, y) g_{\sigma_m,B}(x, y), \text{ and} \tag{33}$$

$$\mathcal{T}_{s,\ell'}(x_0, y') - \xi - \tau_2 \mathcal{T}_{s,\ell}(x_0, y) - \tau_3 \mathcal{T}_{\ell,\ell'}(y, y') - \lambda_{\sigma_m,C}^T(x_0, y, y') g_{\sigma_m,C}(x_0, y, y'), \tag{34}$$

are sum-of-squares, where $\lambda_{\sigma_i,A}^T$, $\lambda_{\sigma_i,B}^T$, and $\lambda_{\sigma_i,C}^T$ are sum-of-squares polynomials over their respective regions and $\tau_1, \tau_2, \tau_3, \xi \in \mathbb{R}_{>0}$ are positive values.

Lemma 8. *Assume the sets $\mathcal{X}$, $\mathcal{X}_0$, $\mathcal{X}_{VF}$, and $\mathcal{X}_{\sigma_i}$ for all σ_i are semi-algebraic, and there exists sum-of-squares polynomials $\mathcal{T}_{i,j}(x, y)$ satisfying conditions* (32)-(34) *for every $i, j \in Q$. Then the function $\mathcal{T}\big((x, i), (y, j)\big)$ defined piecewise as $\mathcal{T}_{i,j}(x, y)$ for all $i, j \in Q$ is a CC for the product satisfying conditions* (17)-(19).

To determine whether the above equations are SOS, one can make use of solvers such as [29]. The complexity of determining whether the above equations are SOS is $O\big(\binom{2n+d}{d} \times \binom{2n+d}{d}\big)$, when searching for CCs for safety or ensuring finite visits, where n is the dimension of the state set, and $2d$ is the degree of the polynomial. The complexity of verifying LTL specifications is[1] polynomial in $O\big(2^{|\phi|^2} \times \binom{2n+2d}{d} \times \binom{2n+2d}{d}\big)$, where $|\phi|$ indicates the size of the LTL formula. This is because the closure certificate is a function of pairs of the state set of the system and there are at most $|Q|^2$ many pairs of transitions in an automaton. The number of states of the NBA is $O\big(2^{|\phi|}\big)$, where $|\phi|$ is the size of the formula. On the other hand, the complexity of determining whether the equations for barrier certificates are SOS is polynomial in $O\big(\binom{n+d}{d} \times \binom{n+d}{d}\big)$ [28]. If the dimension of the system is fixed, then the complexity is polynomial in the degree $2d$ but exponential in the size of the formula ϕ. The key issue when using an SOS approach, however, is that there may be polynomials that satisfy the above constraints but are not SOS. Furthermore, one cannot directly encode the implication in SOS, and, hence, suffers from the conservatism of having to satisfy a stronger condition.

5 SUBSUMING EXISTING APPROACHES

We show that our approach generalizes the existing class of techniques using state triplet introduced in [38] for the verification of continuous-space systems against linear temporal logic properties. The state triplet technique has been used for the verification and synthesis for stochastic systems [15, 16], for networks of systems [1, 2], and in motion-planning for nonlinear systems [13]. Here, the transition map is a function, and the state set is a subset of $\mathbb{R}^n$. First, we present the details of the state triplet approach briefly, in Section 5.1. Then in Section 5.2, we show how one can use closure certificates to guarantee satisfaction of LTL properties when the state triplet approach provides a guarantee as well.

5.1 The State Triplet Approach

Consider a system $\mathfrak{S} = (\mathcal{X}, \mathcal{X}_0, f)$, where $\mathcal{X} \subseteq \mathbb{R}^n$, and f is a state transition function. Consider a NBA $\mathcal{A} = (\Sigma, Q, Q_0, \delta, Acc)$ that represents the complement of the desired LTL formula ϕ, and a labeling function $\mathcal{L} : \mathcal{X} \to \Sigma$. The key idea of the state triplet approach is to find barrier certificates between edges of the automaton to disallow the system from visiting an accepting state. This ensures that $\mathfrak{S} \not\models_{\mathcal{L}} \neg\phi$, and so we have $\mathfrak{S} \models_{\mathcal{L}} \phi$.

The steps of the approach are as follows:

(1) Consider all the simple paths in the NBA that start from an initial state and reach an accepting state.
(2) Break these paths into a sequence of state triplets (q_m, q_m', q_m'') (or edge pairs (e_m, e_m')).
(3) Search for a barrier certificate to "cut" at least one triplet from each path.
(4) If we can cut at least one triplet along each path, we can conclude that $\mathfrak{S} \models_{\mathcal{L}} \phi$, and if not this approach is inconclusive.

To help illustrate this approach consider a system $\mathfrak{S} = (\mathcal{X}, \mathcal{X}_0, f)$, and a finite alphabet $\Sigma = \{a_0, a_1\}$; the labeling map $\mathcal{L}$ naturally

[1] Determining whether a polynomial in n variables and degree d are SOS can be reduced to a semidefinite program in $O\big(\binom{n+d}{d} \times \binom{n+d}{d}\big)$ variables [24].

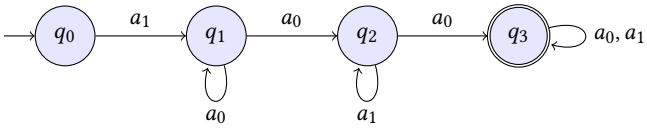

Figure 2: Example NBA $\mathcal{A}$ which represents the complement of a safety to illustrate the state triplet approach.

partition the set $\mathcal{X}$ into two sets $\mathcal{X}_{a_1}$ and $\mathcal{X}_{a_2}$. Let the NBA $\mathcal{A} = (\Sigma, Q, Q_0, \delta, Acc)$ in Figure 2 represent the complement of an LTL specification of interest, where $\Sigma = \{a_0, a_1\}$, $Q = \{q_0, q_1, q_2, q_3\}$, $Q_0 = \{q_0\}$, $Acc = \{q_3\}$, and δ is specified by the edges in the graph. This NBA accepts those words which start with a a_1 and have at least two a_0's in them. It represents the LTL formula $\phi = a_1 \wedge \mathsf{XF}(a_0 \wedge \mathsf{F}a_0)$. There is one simple path starting from the initial state q_0 that reaches the accepting state q_3. This path corresponds to the sequence of states (q_0, q_1, q_2, q_3) and can be broken into two triplets (q_0, q_1, q_2) and (q_1, q_2, q_3). The first state triplet corresponds to the edge pair $((q_0, q_1), (q_1, q_2))$ which are labeled by the pair of letters (b, a), and the second to the edge pair $((q_1, q_2), (q_2, q_3))$ which are labeled by the pair of letters (a_0, a_0).

To cut the transitions along the first state triplet (q_0, q_1, q_2), we try to find a barrier certificate, where the initial set of states are all the states with the label a_1 (corresponding to the edge (q_0, q_1)), *i.e.* all the states of the system in $\mathcal{X}_{a_1}$. The unsafe states are all the states with a label of a_0 (corresponding to the edge (q_1, q_2)), *i.e.* all the states in the set $\mathcal{X}_{a_0}$. The existence of a barrier certificate, proves that no trace of the system can visit a state with label a_0, after visiting a state with label a_1, and so cannot correspond to the run (q_0, q_1, q_2) in the automaton. This cuts the path from the initial state q_0 to the accepting state q_3 of the automaton and shows that no trace of the system can take this corresponding path in the NBA. As there are no other simple paths to the accepting state, we conclude that no trace of the system is in the language of the NBA. If we fail to find a barrier certificate for the first triplet, we then search for a barrier certificate in the next triplet (q_1, q_2, q_3). As the edges of the states in the triplet have the same label a_0, we cannot find a barrier certificate where the initial set and unsafe set are both $\mathcal{X}_{a_0}$. If we fail to find a barrier certificate for both triplets, then our approach is inconclusive.

As the state triplet approach proves that no trace of the system can reach the accepting state, one expects that it can be leveraged in a similar fashion to bounded model checking. Ideally unrolling the automaton for k-steps would allow one to verify that no trace of the system visits the accepting state more than k times. Unfortunately, this is not true, and the state triplet approach does not benefit when one unrolls more than once.

Lemma 9. *Consider a NBA $\mathcal{A} = (\Sigma, Q, Q_0, \delta, Acc)$, whose simple paths from the initial states Q_0 to the accepting states Acc have been divided into k state triplets (q_m, q'_m, q''_m) for all $1 \leq m \leq k$, such that $q'_m \in \delta(q, a_m)$ and $q''_m \in \delta(q, b_m)$, for some $a_m, b_m \in \Sigma$. Unrolling the automaton more than once does not lead to finding a new triplet with labels that have not been considered before.*

A proof of Lemma 9 can be found in [22, Appendix B].

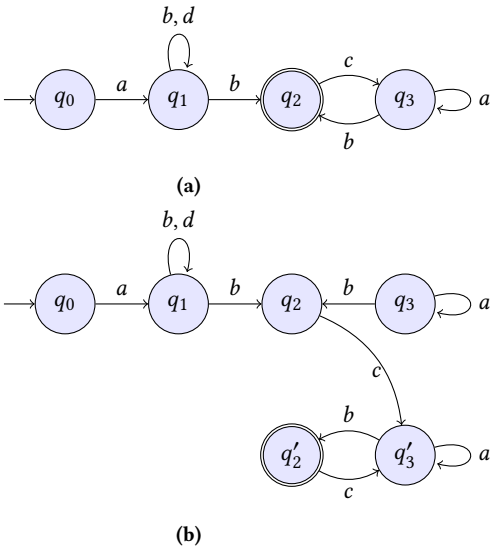

Figure 3: Example NBA $\mathcal{A}$ (Figure 3a) and its unrolling (Figure 3b) from Section 5.1.

Now, with an example, we demonstrate why unrolling the automaton once might help in finding state triplets. Consider the NBA $\mathcal{A} = (\Sigma, Q, Q_0, \delta, Acc)$ in Figure 3a, with $\Sigma = \{a, b, c, d\}$, $Q = \{q_0, q_1, q_2, q_3\}$, $Acc = \{q_2\}$, and the transition relation specified by the edges in the graph. We unroll this automaton to get the automaton $\mathcal{A}'$ in Figure 3b. Unrolling the automaton once allows us to consider the triplet (q_2, q'_3, q'_2) whose edge labels correspond to the pair of letters (c, b). Observe that no state triplet in the original NBA corresponds to this pair of letters. Thus even if one was not able to find a barrier certificate for a state triplet in NBA $\mathcal{A}$ (for the state triplet (q_0, q_1, q_2)), one may still find a barrier certificate for a state triplet in NBA $\mathcal{A}'$ (the state triplet (q_2, q'_3, q'_2)). Hence, no trace of the system can visit the accepting state more than once.

We observe that unrolling once does have an impact since we can now consider those state triplets along the simple cycles of the NBA. Unfortunately, unrolling twice does not help. Thus, one is unable to verify those traces of a system which reach and cycle on accepting states more than twice, even if they visit accepting states finitely often.

The state triplet approach is conservative in the following direction: *independently of the state runs in the automaton and of the initial states of the system $\mathfrak{S}$, one is required to break the edge pairs of every simple path regardless of what states of the automaton may be encountered before or after.*

5.2 CC Subsumes State Triplet Approach

We now show that our approach generalizes the earlier state triplet one. Consider a NBA $\mathcal{A} = (\Sigma, Q, Q_0, \delta, Acc)$, and let us assume that there exist k barrier certificates $\mathcal{B}_1, \mathcal{B}_2 \ldots, \mathcal{B}_k$ associated with state triplets (q_m, q'_m, q''_m), (or edge pairs (e_m, e'_m)) for each $1 \leq m \leq k$, that act as a proof that every trace of the system is in $L(\mathcal{A})$. Furthermore, let (a_m, b_m) be the pairs of letters associated with these triplets. We divide the states of the NBA $\mathcal{A}$ into two sets Q_l,

and Q_r. A state $q \in Q$ is in the set Q_l if It is not the middle element of a state triplet and there is a path from q to the middle element of *some* state triplet. A state $q \in Q$ is in the set Q_r if there is a path from the middle element of every state triplet to q. The only states that are in neither of the sets are middle elements of the triplets. As the state triplet approach "cuts" the transitions of the NBA, we observe that no trace of the system starting from a state in Q_l can reach a state in Q_r. Furthermore, we note that every state $s \in Q_0$ is in the set Q_l, and every state $\ell \in Acc$ is in the set Q_r. We now show how one may use closure certificates to provide guarantees of satisfaction for LTL specifications when the state triplet approach can guarantee the satisfaction of the same specification.

Theorem 10. *Consider a System $\mathfrak{S} = (\mathcal{X}, \mathcal{X}_0, f)$, labeling map $\mathcal{L} : \mathcal{X} \to \Sigma$, and NBA $\mathcal{A} = (\Sigma, Q, Q_0, \delta, Q_{Acc})$ representing the complement of a desired LTL formula ϕ. Suppose that there exists barrier certificates $\mathcal{B}_1, \ldots, \mathcal{B}_k$ that show $\mathfrak{S} \models_{\mathcal{L}} \phi$ via the state triplet approach. Then there exists a closure certificate $\mathcal{T}$ that also acts as a proof that $\mathfrak{S} \models_{\mathcal{L}} \phi$.*

Proof (sketch). We construct a closure certificate $\mathcal{T} : \mathcal{X} \times Q \times \mathcal{X} \times Q \to \mathbb{R}$ such that, for $x, y \in \mathcal{X}$, and $i, j \in Q$, we have that $\mathcal{T}(x, i, y, j) \geq 0$, if:

- $i \in Q_r$;
- $i \in Q_l$ and $j \in Q_l$;
- i is the middle element of triplet m, and $\mathcal{B}_m(x) \leq 0$, $j \in Q_l$ and $\mathcal{B}_m(y) \leq 0$;
- i is the middle element of triplet m, and $\mathcal{B}_m(x) > 0$; or
- $i \in Q_l$, j is the middle element of some triplet m, and $\mathcal{B}_m(y) \leq 0$.

Moreover, $\mathcal{T}(x, i, y, j) < 0$, otherwise. This certificate clearly guarantees that no trace of the system can reach the accepting states. A detailed proof is given in [22, Appendix C]. □

6 CASE STUDIES

We experimentally demonstrate the utility of closure certificates on Kuramoto oscillators and a two-room temperature model. In the first example, we consider the problem of safety verification. Here, we show that we can verify the safety a 1 dimensional Kuramoto oscillator via a linear closure certificate when a linear barrier certificate cannot do the same. We then verify the safety of a 2 dimensional Kuramoto oscillator by converting the safety objective to an LTL specification. We then search for a closure certificate over the product of the system and the NBA representing the complement of the specification. In the second, we consider the problem of verifying the persistence of a two-room temperature model. To do so, we convert the objective to an LTL specification, after which we search for a closure certificate over the product of the system and the NBA.

6.1 Kuramoto Oscillator

Kuramoto model [11] has been used widely to describe chemical and biological oscillators, with applications in neuroscience and modern power system analysis. As a first case study, we consider a system $\mathfrak{S} = (\mathcal{X}, \mathcal{X}_0, f)$ to model a Kuramoto oscillator whose dynamics are taken from [2], where $\mathcal{X} = [0, 2\pi]$ indicates the state set, $\mathcal{X}_0 = [\frac{4\pi}{9}, \frac{5\pi}{9}]$ the initial set of states, and $\mathcal{X}_u = [\frac{7\pi}{9}, \frac{8\pi}{9}]$ denotes the unsafe set of states. The transition function f is defined as:

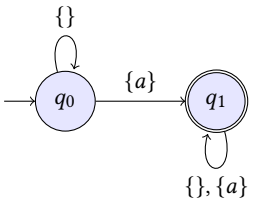

Figure 4: A (nondeterministic) Büchi automaton $\mathcal{A}$ representing the LTL formula $\mathsf{F}a$.

$$f(x) = x + \tau\Omega + t_s K \sin(-x) - 0.532x^2 + 1.69,$$

where $x \in \mathcal{X}$ indicates the phase of the oscillator, $t_s = 0.1$ is the sampling time, $\Omega = 0.01$ is the natural frequency, and $K = 0.0006$ is the coupling strength.

We then search for a linear closure certificate as in Defintion 3.1 to ensure the safety of the system. To do so, we strengthen the implication in condition (12) to condition (24), with $\tau_1 = 1$, and sample 50 points from the initial, unsafe, and entire state set. We then solve a linear program to find a candidate closure certificate. As z3 [21] cannot handle the function $\sin(-x)$, we instead use dReal [10] to find counterexamples. We add these counterexamples to the set of samples and repeat the procedure until we find no counterexamples. We find the closure certificate $\mathcal{T}(x, y) = 10 - 4.094y$ that demonstrates safety. The time taken for our CEGIS loop to terminate is around 10 minutes on a machine running MacOS 11.2 (Intel i9-9980HK with 64 GB of RAM). We should note that the linear program encoding the barrier certificate conditions is infeasible when we consider a linear barrier certificate.

We now cast the problem of safety verification as a problem of verifying a system against the LTL formula $\mathsf{G}\neg a$, over the set of atomic propositions $AP = \{a\}$, where a state is marked with label $\{a\}$ if it is unsafe. The complement of this specification is $\mathsf{F}a$, and the NBA $\mathcal{A}$ representing this is described in Figure 4. We consider the system $\mathfrak{S} = (\mathcal{X}, \mathcal{X}_0, f)$ to be a two-dimensional Kuramoto oscillator, where $\mathcal{X} = [0, \frac{8\pi}{9}] \times [0, \frac{8\pi}{9}]$ denotes the state set. $\mathcal{X}_0 = [0, \frac{\pi}{9}] \times [0, \frac{\pi}{9}]$ denotes the initial set of states and the transition function f is defined as:

$$f(x_1, x_2) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \tau\Omega + 1.69 \\ \tau\Omega + 1.69 \end{bmatrix} + Kt_s \begin{bmatrix} sin(x_2 - x_1) \\ sin(x_1 - x_2) \end{bmatrix} - 0.532\tau \begin{bmatrix} x_1^2 \\ x_2^2 \end{bmatrix},$$

where $(x_1, x_2) \in \mathcal{X}$ indicates the phase of the oscillators, and the remaining constants have the same values as the one dimensional case. We consider the alphabet $\Sigma = \{\{\}, \{a\}\}$, and the labeling function $\mathcal{L}$ as $\mathcal{L}(x_1, x_2) = \{a\}$ if either $x_1 \in [\frac{15\pi}{18}, \frac{8\pi}{9}]$ or $x_2 \in [\frac{15\pi}{18}, \frac{8\pi}{9}]$. All the other states are assigned a label of the empty set $\{\}$. We consider the template of the piecewise components of the closure certificate to be:

$$\mathcal{T}_{i,j}\big((x_1, x_2), (y_1, y_2)\big) = c_{0,i,j} + c_{1,i,j} y_1 \mathbb{I}_0(x_1, x_2) + c_{2,i,j} y_2 \mathbb{I}_0(x_1, x_2) + c_{3,i,j} y_1 \mathbb{I}_a(x_1, x_2) + c_{4,i,j} y_2 \mathbb{I}_a(x_1, x_2) + c_{5,i,j} y_1 + c_{6,i,j} y_2, \quad (35)$$

for all states $(x_1, x_2), (y_1, y_2) \in \mathcal{X}$ and NBA states $i, j \in Q$, where the functions $\mathbb{I}_0$, and $\mathbb{I}_a$ are indicator functions over the initial set of states, and states with label $\{a\}$ respectively. We then search for the

piecewise components of the closure certificate via a counterexample-guided approach by collecting round 400 points from the system. To do so, we encode the conditions as a linear program, and set the s-procedure coefficients of $\tau_1 = 1$ for the conditions (24) and the values of $\tau_2 = 1$, and $\tau_3 = 0$ for conditions (25) to find a candidate closure certificate. To speed up the search for counterexamples, we randomly sample points and check if the conditions fail to hold. If so we have found a counterexample. If no such counterexample is found, we then formulate a query in dReal to search for a valid counterexample. We repeat this process until no counterexamples are found. The coefficients of the resulting closure certificate are displayed as a table in [22, Appendix D.1]. The time taken for this CEGIS loop to terminate is around 1 hour and 50 minutes on the reference machine. We find the value of ξ to be 1.

6.2 Two Room Temperature Model

As a second case study, we consider our system $\mathfrak{S} = (\mathcal{X}, \mathcal{X}_0, f)$ to be an interconnected two-room temperature model adapted from [1], where $\mathcal{X} = [20, 34] \times [20, 34] \in \mathbb{R}^2$ indicate the temperature of the two rooms, $\mathcal{X}_0 = [21, 24] \times [21, 24]$ indicate the initial set of states, and the transition function is defined as:

$$f(x_1, x_2) = A \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \mu T_h \begin{bmatrix} u(x_1) \\ u(x_2) \end{bmatrix} + \theta \begin{bmatrix} T_e \\ T_e \end{bmatrix},$$

where x_i represents the temperature of room i, for all $i \in \{1, 2\}$, the matrix A is

$$A := \begin{bmatrix} 1 - 2\alpha - \theta - \mu u(x_1) & \alpha \\ \alpha & 1 - 2\alpha - \theta - \mu u(x_2) \end{bmatrix},$$

where constants $\alpha = 0.004$, $\theta = 0.01$, and $\mu = 0.15$ represent the conduction factors, and $u(x)$ denotes the temperature controller, and is defined as $u(x_i) = 0.59 - 0.011x_i$. The value $T_h = 40$C denotes the heater temperature, and $T_e = 0$C represents the ambient temperature. Let the LTL formula to be verified be $a_0 \implies \mathsf{FG}\neg a_1$. This property requires that a system that starts from a state with atomic proposition a_0 does not visit the states with atomic proposition a_1 infinitely often. We consider the atomic propositions $AP = \{a_0, a_1\}$, and the alphabet $\Sigma = \{\{\}, \{a_0\}, \{a_1\}, \{a_0, a_1\}\}$. In this setting, we require that if a state sequence of the system starts from $\mathcal{X}_0$ then it must visit the region $([20, 26] \times \mathcal{X}) \cup (\mathcal{X} \times [20, 26])$ finitely often. The complement of this specification is $a_0 \wedge \mathsf{GF}a_1$ and the NBA $\mathcal{A}$ in Figure 5 denotes this complement. Here, we mark the states $(x_1, x_2) \in \mathcal{X}_0$ with the atomic proposition a_0. We mark a state $(x_1, x_2) \in \mathcal{X}$ with atomic proposition a_1, if $(x_1, x_2) \in ([20, 26] \cup \mathcal{X}) \times (\mathcal{X} \cup [20, 26])$. All other states are not marked with any atomic proposition. Observe that a state (x_1, x_2) may be marked with both atomic propositions a_0, and a_1, or neither. We define the labeling map as:

$$\mathcal{L}(x_1, x_2) = \begin{cases} \{a_0, a_1\} & \text{if } (x_1, x_2) \text{ is marked with both } a_0, a_1 \\ \{a_0\} & \text{if } (x_1, x_2) \text{ is marked with only } a_0 \\ \{a_1\} & \text{if } (x_1, x_2) \text{ is marked with only } a_1 \\ \{\} & \text{otherwise,} \end{cases}$$

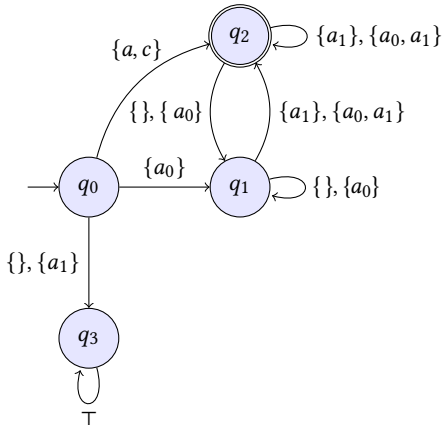

Figure 5: A (nondeterministic) Büchi automaton $\mathcal{A}$ for the two-room temperature case study from Section 6. The automata represents the LTL formula $a_0 \wedge \mathsf{GF}a_1$. Here $\top$ indicates any letter in the alphabet.

We consider the template of the piecewise components of the closure certificate to be specified as:

$$\begin{aligned} \mathcal{T}_{i,j}\big((x_1, x_2), (y_1, y_2)\big) = {} & c_{0,i,j} + c_{1,i,j}x_1 + c_{2,i,j}x_2 + c_{3,i,j}y_1 \\ & + c_{4,i,j}y_2 + c_{5,i,j}\max(x_1, x_2) + c_{6,i,j}\max(y_1, y_2) \\ & + c_{7,i,j}x_1^2 + c_{8,i,j}x_2^2 + c_{9,i,j}y_1^2 + c_{10,i,j}y_2^2, \end{aligned} \quad (36)$$

for all states (x_1, x_2), and $(y_1, y_2) \in \mathcal{X}$ and all states i, j of the NBA $\mathcal{A}$ in Figure 5. We then search for the piecewise components of the closure certificate using a CEGIS approach. To speed up this, we first solve the linear program with around 100 points, where we set the values of $\tau_1 = 1$, $\tau_2 = 0.4$, and $\tau_3 = 0.1$. We then search for counterexamples by first randomly sampling points, after which we use z3 to find counterexamples. The resulting coefficients are described in a table in [22, Appendix D.2]. The time taken to find the closure certificate is around 1.5 hours on the reference machine. Finally, we find the value of ξ to be 0.5 in this example.

7 CONCLUSION

We proposed a notion of so-called closure certificates that act as a function analog of transition invariants. Our notion of closure certificates provide an abstraction-free approach to verify dynamical systems against ω-regular properties. Our approach of using closure certificates to verify ω-regular properties subsume existing approaches that use barrier certificate to verify ω-regular properties. As future work, we plan to investigate how one may use approaches such as k-induction to allow for a larger class of functions to act as closure certificates. We also plan on investigating data driven approaches to find these closure certificates as well as investigate their use in synthesizing controllers.

8 ACKNOWLEDGEMENTS

The authors thank Mateo Perez and Sriram Sankaranarayanan for valuable discussions as well as the anonymous reviewers for their constructive comments. This work was supported by NSF CAREER

awards CCF-2146563, and CNS-2145184, and grants ECCS-2015403, CNS-2039062, and CNS-2111688.

REFERENCES

[1] Mahathi Anand, Abolfazl Lavaei, and Majid Zamani. 2021. Compositional Synthesis of Control Barrier Certificates for Networks of Stochastic Systems against *omega*-Regular Specifications. *arXiv preprint arXiv:2103.02226* (2021).

[2] Mahathi Anand, Abolfazl Lavaei, and Majid Zamani. 2022. From small-gain theory to compositional construction of barrier certificates for large-scale stochastic systems. *IEEE Trans. Automat. Control* 67, 10 (2022).

[3] Christel Baier and Joost-Pieter Katoen. 2008. *Principles of model checking*. MIT press.

[4] Clark Barrett and Cesare Tinelli. 2018. *Satisfiability modulo theories*. Springer.

[5] George E Collins. 1975. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Automata Theory and Formal Languages: 2nd GI Conference Kaiserslautern, May 20–23, 1975*. Springer, 134–183.

[6] Byron Cook. 2009. Priciples of program termination. *Engineering Methods and Tools for Software Safety and Security* 22, 161 (2009), 125.

[7] Bruno Dutertre and Leonardo de Moura. 2006. A fast linear-arithmetic solver for DPLL (T). In *International Conference on Computer Aided Verification*. Springer, 81–94.

[8] Emmanuel Filiot, Naiyong Jin, and Jean-François Raskin. 2009. An antichain algorithm for LTL realizability. In *International Conference on Computer Aided Verification*. Springer, 263–277.

[9] Sicun Gao, Jeremy Avigad, and Edmund M. Clarke. 2012. δ-complete decision procedures for satisfiability over the reals. In *Automated Reasoning (Lecture Notes in Computer Science)*. Springer, 286–300.

[10] Sicun Gao, Soonho Kong, and Edmund M. Clarke. 2013. dReal: An SMT solver for nonlinear theories over the reals. In *Automated Deduction – CADE-24 (Lecture Notes in Computer Science)*. Springer, 208–214.

[11] Yufeng Guo, Dongrui Zhang, Zhuchun Li, Qi Wang, and Daren Yu. 2021. Overviews on the applications of the Kuramoto model in modern power system analysis. *International Journal of Electrical Power & Energy Systems* 129 (2021), 106804.

[12] Gurobi Optimization, LLC. 2021. Gurobi Optimizer Reference Manual. https://www.gurobi.com

[13] Binghan He, Jaemin Lee, Ufuk Topcu, and Luis Sentis. 2020. BP-RRT: Barrier pair synthesis for temporal logic motion planning. In *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 1404–1409.

[14] Thomas A Henzinger, Pei-Hsin Ho, and Howard Wong-Toi. 1997. HyTech: A model checker for hybrid systems. In *Computer Aided Verification: 9th International Conference, CAV'97 Haifa, Israel, June 22–25, 1997 Proceedings 9*. Springer, 460–463.

[15] Pushpak Jagtap, Sadegh Soudjani, and Majid Zamani. 2018. Temporal logic verification of stochastic systems using barrier certificates. In *ATVA*. 177–193.

[16] Pushpak Jagtap, Sadegh Soudjani, and Majid Zamani. 2020. Formal synthesis of stochastic systems via control barrier certificates. *IEEE Trans. Automat. Control* 66, 7 (2020), 3097–3110.

[17] Dejan Jovanović and Leonardo De Moura. 2013. Solving non-linear arithmetic. *ACM Communications in Computer Algebra* 46, 3/4 (2013), 104–105.

[18] Mahmoud Khaled and Majid Zamani. 2021. OmegaThreads: symbolic controller design for ω-regular objectives. In *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control* (Nashville, Tennessee) *(HSCC '21)*. Association for Computing Machinery, New York, NY, USA, Article 25, 7 pages. https://doi.org/10.1145/3447928.3456652

[19] Soonho Kong, Armando Solar-Lezama, and Sicun Gao. 2018. Delta-decision procedures for exists-forall problems over the reals. In *Computer Aided Verification: 30th International Conference, CAV 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings, Part II 30*. Springer, 219–235.

[20] Morteza Lahijanian, Sean B Andersson, and Calin Belta. 2011. Temporal logic motion planning and control with probabilistic satisfaction guarantees. *IEEE Transactions on Robotics* 28, 2 (2011), 396–409.

[21] Leonardo de Moura and Nikolaj Bjørner. 2008. Z3: An efficient SMT solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*. 337–340.

[22] Vishnu Murali, Ashutosh Trivedi, and Majid Zamani. 2024. Closure Certificates. arXiv:2305.17519 https://arxiv.org/abs/2305.17519

[23] Alireza Nadali, Vishnu Murali, Ashutosh Trivedi, and Majid Zamani. 2024. Neural Closure Certificates. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[24] Pablo A. Parrilo. 2003. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming* 96 (2003), 293–320.

[25] Amir Pnueli. 1977. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science*. IEEE, 46–57.

[26] Andreas Podelski and Andrey Rybalchenko. 2004. Transition invariants. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, 2004*. IEEE, 32–41.

[27] Andreas Podelski and Silke Wagner. 2006. Model checking of hybrid systems: From reachability towards stability. In *International Workshop on Hybrid Systems: Computation and Control*. Springer, 507–521.

[28] Stephen Prajna and Ali Jadbabaie. 2004. Safety verification of hybrid systems using barrier certificates. In *International Workshop on Hybrid Systems: Computation and Control*. Springer, 477–492.

[29] Stephen Prajna, Antonis Papachristodoulou, and Pablo A Parrilo. 2002. Introducing SOSTOOLS: A general purpose sum of squares programming solver. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002*. IEEE, 741–746.

[30] Stephen Prajna and Anders Rantzer. 2007. Convex programs for temporal verification of nonlinear dynamical systems. *SIAM Journal on Control and Optimization* (2007), 999–1021.

[31] Matthias Rungger and Majid Zamani. 2016. SCOTS: A tool for the synthesis of symbolic controllers. In *Proceedings of the 19th international conference on hybrid systems: Computation and control*. 99–104.

[32] Shmuel Safra. 1988. On the complexity of ω-automata. In *Proc. 29th IEEE Symp. Found. of Comp. Sci.* IEEE, 319–327.

[33] Sriram Sankaranarayanan and Ashish Tiwari. 2011. Relational abstractions for continuous and hybrid systems. In *International Conference on Computer Aided Verification*. Springer, 686–702.

[34] Sven Schewe and Bernd Finkbeiner. 2007. Bounded synthesis. In *International symposium on automated technology for verification and analysis*. Springer, 474–488.

[35] Armando Solar-Lezama. 2008. *Program Synthesis by Sketching*.

[36] Paulo Tabuada. 2009. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer Science & Business Media.

[37] Moshe Y Vardi. 2005. An automata-theoretic approach to linear temporal logic. *Logics for concurrency: structure versus automata* (2005), 238–266.

[38] Tichakorn Wongpiromsarn, Ufuk Topcu, and Andrew Lamperski. 2015. Automata theory meets barrier certificates: Temporal logic verification of nonlinear systems. *IEEE Trans. Automat. Control* 61, 11 (2015), 3344–3355.