AP-observation Automata for Abstraction-based Verification of Continuous-time Systems[⋆]

Sasinee Pruekprasert[1] and Clovis Eberhart[2]

[1] The University of Tokyo, Tokyo, Japan spruekprasert@g.ecc.u-tokyo.ac.jp
[2] Tohoku University, Sendai, Japan eberhart.clovis.d1@tohoku.ac.jp

Abstract. A key challenge in abstraction-based verification and control under complex specifications such as Linear Temporal Logic (LTL) is that abstract models retain significantly less information than their original systems. This issue is especially true for continuous-time systems, where the system state trajectories are split into intervals of discrete actions, and satisfaction of atomic propositions is abstracted to a whole time interval. To tackle this challenge, this work introduces a novel translation from LTL specifications to AP-observation automata, a particular type of Büchi automata specifically designed for abstraction-based verification. We also present a verification algorithm and an illustrative example for abstraction-based system verification under LTL specification utilizing the proposed structure.

Keywords: Linear temporal logic · Verification · Automata · Abstraction · Continuous-time system · Symbolic control

1 Introduction

The growing complexity of engineered physical systems has increased the need for formal methods that can specify and verify their desired behaviors. Many such properties can only be specified in temporal logics, a powerful framework for formalizing complex specifications of timed systems. In particular, *Linear Temporal Logic* (LTL) [1] strikes a good balance between expressivity and complexity of verification. Indeed, LTL is widely used for describing temporal specifications in many fields, such as verification [2–4] and control theory [5–7], thanks to its expressivity. Verification of LTL properties can be reduced to language emptiness of Büchi automata [8–10], which gives rise to efficient verification algorithms. However, these techniques are developed for discrete-time, discrete-state systems, while physical systems evolve in continuous space and time.

On the other hand, traditional control theory provides a wealth of methods for analyzing and designing controllers for continuous-time, continuous-state

[⋆] S. Pruekprasert is supported by JSPS KAKENHI Grant Numbers JP21K14191 and JP22KK0155. This work was partly done while S. Pruekprasert was with the National Institute of Advanced Industrial Science and Technology, Tokyo, Japan, and C. Eberhart was with the National Institute of Informatics, Tokyo, Japan.
Both authors have contributed equally.

systems [11, 12]. However, they focus primarily on specifications such as stability [13], robustness [14], and safety constraints [15]. Meanwhile, modern applications, such as autonomous systems, require temporal and logic-based properties [16–18], which conventional control methods are not designed for.

Abstraction-based control, or *symbolic control* [19–21], offers a framework to handle complex specifications by constructing a discrete abstraction, called a *symbolic model*, of the original continuous system. This approach allows us to leverage automata-theoretic techniques to prove properties of continuous systems. Although various abstraction-based frameworks have been explored, recent studies [22–27] have primarily addressed computational complexity and adaptability, a key challenge in this domain. However, this work focuses on another fundamental issue in abstraction-based approaches: the substantial loss of information in abstract models relative to their concrete counterparts. This is especially true for continuous-time systems, where trajectories are partitioned into discrete intervals and atomic proposition satisfaction is abstracted over these intervals, complicating verification under complex specifications like LTL.

Contribution. In this work, we introduce a novel approach for abstraction-based system verification for LTL specifications called *AP-observation automata*. They encode the abstract properties of *atomic propositions (APs)* along system trajectories through their transition labels, classifying them into four values. We define a new construction of a generalized Büchi automaton, which we call an AP-observation automaton, from an LTL formula for continuous-time systems.

Building on this structure, our verification framework soundly approximates the satisfiability of atomic propositions along system trajectories. Our approach is highly general, supporting non-deterministic, continuous-state, continuous-time systems without global stability assumptions. To the best of our knowledge, no existing technique provides formal verification for this broad class of systems under LTL specifications. Prior work has instead focused on discrete-time systems [5–7], imposed more restrictive dynamics or assumptions [19, 21, 23, 27], or addressed smaller classes of specifications [22, 25, 28–30]. To achieve this level of generality, we impose a constraint on the satisfaction zones of atomic propositions, a condition met by many practical systems.

Outline. The rest of the paper proceeds as follows. Section 2 introduces the verification problem. Section 3 presents the abstraction of dynamical systems into symbolic models. Section 4 presents AP-observation automata as an abstraction of the specification. Section 5 proposes the verification algorithm. Section 6 provides an illustrative application example. Section 7 presents the conclusions.

Notations. We write $\mathbb{R}$, $\mathbb{R}_{>0}$, $\mathbb{R}_{\geqslant 0}$, $\mathbb{Z}$, and $\mathbb{Z}_{\geqslant 0}$ for the sets of real, positive real, nonnegative real, integer, and nonnegative integer numbers, respectively. The infinity norm is $\|x\|_\infty = \max_{i=0}^{n-1} \|x_i\|$ for $x \in \mathbb{R}^n$. We also use "iff" as an abbreviation for "if and only if".

2 Dynamical Systems and LTL Specifications

In this section, we introduce dynamical systems, LTL specifications, and the verification problem.

2.1 Non-deterministic Dynamical Systems

We consider dynamical systems $\Sigma = (X, \xi, x_{\text{in}})$ where $X \subseteq \mathbb{R}^n$ is the set of considered n-dimensional system states, $\xi : (2^X \backslash \{\varnothing\}) \times \mathbb{R}_{\geqslant 0} \to 2^X \backslash \{\varnothing\}$ is the system evolution function, and $x_{\text{in}} \in X$ is the initial state. For a set $\mathfrak{x} \in 2^X \backslash \{\varnothing\}$ of states and a time instant $t \in \mathbb{R}_{\geqslant 0}$, the set $\xi(\mathfrak{x}, t) \in 2^X \backslash \{\varnothing\}$ contains all possible states reachable from some state in the set $\mathfrak{x}$ at time t. We require that ξ satisfy the following properties: $\xi(\mathfrak{x}, 0) = \mathfrak{x}$ and $\xi(\mathfrak{x}, t_1 + t_2) = \xi(\xi(\mathfrak{x}, t_1), t_2)$. Note that, unlike symbolic control approaches such as [25,28], we do not take control signals as inputs to the system's evolution function. Nevertheless, the results presented in this paper are applicable to controlled systems with fixed control strategies, as these systems can be modeled as dynamical systems Σ given above. For any state $x \in X$, by abuse of notation, we write $\xi(x, t)$ for $\xi(\{x\}, t)$.

A trajectory (*resp.* finite trajectory) of Σ from x is a function $\sigma \colon \mathbb{R}_{\geqslant 0} \to X$ (*resp.* $\sigma \colon [0, T] \to X$, where $T \in \mathbb{R}_{\geqslant 0}$) such that $\sigma(0) = x$ and $\sigma(t) \in \xi(\sigma(0), t)$ for all time $t \in \mathbb{R}_{\geqslant 0}$ (*resp.* $t \in [0, T]$). Let $\mathit{Traj}(\Sigma)$ denote all possible (infinite) trajectories of Σ starting from x_{in}.

2.2 Atomic Propositions (AP) and Assumptions on Trajectories

Atomic propositions (AP), statements about a state of the system, are the basic building blocks of temporal logic formulas for specifications in this paper. Examples of atomic propositions include properties such as whether the system is colliding with an obstacle or whether its position is in a desirable region. Let AP denote the finite set of considered atomic propositions, and $P : X \to 2^{AP}$ assigns the set of atomic propositions that hold at each system state: if $a \in AP$ represents the property that the system is safe, then $a \in P(x)$ means that the system is safe at state $x \in X$. In other words, $\{x \in X \mid p \in P(x)\}$ is the region of states that satisfies the atomic proposition $p \in \text{AP}$.

2.3 Classic LTL with Signal Semantics

Linear Temporal Logic (LTL) formulas are generated by the following grammar:

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathrm{X}\,\varphi \mid \varphi\,\mathrm{U}\,\varphi,$$

where $\top$ denotes *true* and $p \in \text{AP}$ is an atomic proposition.

Conventionally, the semantics of LTL is defined on words, i.e., in discrete time. For example, a classic LTL formula may contain $\mathrm{X}\,\varphi$ (*next* φ), which holds for $x_i x_{i+1} \ldots$ if φ holds at the *next* discrete step $x_{i+1} x_{i+2} \ldots$ (see [1] for a formal definition). However, we are interested in the property of a system trajectory σ :

$\mathbb{R}_{\geqslant 0} \to X$ defined on the continuous timeline. We consider *AP-signal* $\varsigma \colon \mathbb{R}_{\geqslant 0} \to 2^{\mathrm{AP}}$ where $\varsigma(t) = P(\sigma(t))$, i.e., ς indicates the atomic propositions that hold along σ. Note that trajectories $\sigma \colon \mathbb{R}_{\geqslant 0} \to X$ and AP-signals $\varsigma \colon \mathbb{R}_{\geqslant 0} \to 2^{\mathrm{AP}}$ have slightly different types. We say that a formula is *continuous-time* if it contains no subformulas of the form $\mathrm{X}\,\varphi$. Then, the *signal semantics* of continuous-time LTL is the relation $\models$ defined on ς as follows:

- $\varsigma, t \models \top$ always,
- $\varsigma, t \models \neg\varphi$ iff $\varsigma, t \models \varphi$ does not hold,
- $\varsigma, t \models p$ iff $\varsigma(t)(p) = \top$,
- $\varsigma, t \models \varphi \vee \psi$ iff $\varsigma, t \models \varphi$ or $\varsigma, t \models \psi$,
- $\varsigma, t \models \varphi\,\mathrm{U}\,\psi$ iff there exists $t' \geqslant t$ such that $\varsigma, t' \models \psi$ and for all $t'' \in [t, t')$, $\varsigma, t'' \models \varphi$.

The formula $\varphi\,\mathrm{U}\,\psi$ (*φ until ψ*) means that φ must remain true until ψ becomes true, and ψ must become true at some point. We also use the usual shorthands: $\varphi\,\mathrm{R}\,\psi = \neg(\neg\varphi\,\mathrm{U}\,\neg\psi)$, $\Diamond\,\varphi = \top\,\mathrm{U}\,\varphi$, and $\Box\,\varphi = \neg\Diamond\neg\varphi$. The formula $\varphi\,\mathrm{R}\,\psi$ (*φ release ψ*) means that ψ must remain true until φ becomes true, and ψ must remain true forever if φ never becomes true. The formula $\Diamond\,\varphi$ (*eventually φ*) means that φ will hold at some point, while $\Box\,\varphi$ (*globally φ*) means that φ holds all the time.

LTL is a very expressive logic. For example, a reach-avoid specification can be represented as $\Box\,\neg\mathrm{a} \wedge \Diamond\,\mathrm{r}$, where a is an atomic proposition that holds on the zone to avoid and r is the one that holds on the zone to reach.

2.4 System Verification for LTL Specifications

This work considers system verification under LTL specifications, i.e., checking whether the system only produces trajectories that satisfy a given specification. Formally, we consider the following problem.

Problem 1. Given system Σ, $P \colon X \to 2^{\mathrm{AP}}$, and a continuous-time LTL specification φ, our goal is to verify whether for all $\sigma \in \mathit{Traj}(\Sigma)$, $P \circ \sigma, 0 \models \varphi$.

A standard approach to verification is to construct a Büchi automaton corresponding to the LTL formula, as it is well-known [8,10] that LTL formulas can be translated to Büchi automata in the following sense.

Proposition 1. *[8] For all LTL formulas φ, there exists a Büchi automaton $\mathcal{B}$ such that for all words $w \colon \mathbb{Z}_{\geqslant 0} \to 2^{\mathrm{AP}}$, $w, 0 \models \varphi$ iff $w \in \mathcal{L}(\mathcal{B})$.*

We invite interested readers to refer to [8,10] for the translation algorithm of Proposition 1. However, we briefly explain the key concepts here. The translation heavily relies on the fact that $\varphi\,\mathrm{U}\,\psi$ is equivalent to $\psi \vee (\varphi \wedge \mathrm{X}(\varphi\,\mathrm{U}\,\psi))$, and similarly $\varphi\,\mathrm{R}\,\psi$ is equivalent to $\psi \wedge (\varphi \vee \mathrm{X}(\varphi\,\mathrm{R}\,\psi))$. For example, if p and $p\,\mathrm{U}\,q$ hold at time t, but q does not, then necessarily $p\,\mathrm{U}\,q$ must hold at time $t+1$. Using this fact, it is possible to build a generalized Büchi automaton for φ whose action labels are valuations of atomic propositions and whose states are valuations of subformulas of φ. The Büchi automaton's transitions reflect behaviors as described above: in a state where p and $p\,\mathrm{U}\,q$ hold but q does not, it can only transition to a state where $p \,\mathrm{U}\, q$ holds. Its accepting sets ensure that if $\varphi_1 \,\mathrm{U}\, \varphi_2$ holds at some point, then φ_2 must hold at some later point. The accepting states are the states that capture a property of operator U that cannot be verified by comparing two consecutive states in a run. In this case, the accepting states either map q to true or $p \,\mathrm{U}\, q$ to false due to the fact that if $p \,\mathrm{U}\, q$ is true in some state x_i, then eventually q later becomes true at some state x_j where $j \geqslant i$.

3 System Abstraction and Information Loss

A dynamical system, as described in the previous section, is a continuous-state, continuous-time system. In order to verify a system under an LTL specification by checking the system with the corresponding Büchi automaton, we need to abstract the system into a finite-state, discrete-time *symbolic model* that approximates the behavior of the dynamical system.

3.1 Time-abstraction and Signal Chopping

When abstracting a system, one of the most important losses of information comes from discretizing time. Indeed, since we are interested in complex temporal specifications, where the order in which atomic propositions are satisfied matters, discretizing time loses information about whether an LTL formula holds between two time instants. This information can be arbitrarily complex, and any abstraction into a finite number of patterns necessarily induces a loss of precision.

We discretize time by chopping an AP-signal $\varsigma\colon \mathbb{R}_{\geqslant 0} \to 2^{\mathrm{AP}}$ into slices of a fixed length $\tau \in \mathbb{R}_{>0}$. We abstract the satisfaction of each $p \in \mathrm{AP}$ within each of these slices into one of four possible patterns $\mathbb{O} = \{A, Z, E, N\}$, referred to as an *observation*. Conceptually, A means that the p holds at **A**ll time throughout the interval, Z means that p holds only at the beginning of the interval (time **Z**ero), E means that p holds only at the **E**nd of the interval, and N means that p holds at **N**one of the interval time points. Formally, we define the *signal chopping* of ς along τ, denoted $[\varsigma]_\tau\colon \mathbb{Z}_{\geqslant 0} \to \mathbb{O}^{\mathrm{AP}}$, as follows: for all $n \in \mathbb{Z}_{\geqslant 0}$,

- $[\varsigma]_\tau(n)(p) = A$ if for all $t \in [n\tau, (n+1)\tau]$, $\varsigma(t)(p) = \top$,
- $[\varsigma]_\tau(n)(p) = Z$ if there exists $t' \in [n\tau, (n+1)\tau)$ such that $\varsigma(t)(p) = \top$ for all $t \leqslant t'$ and $\varsigma(t)(p) = \bot$ for all $t > t'$.
- $[\varsigma]_\tau(n)(p) = E$ if there exists $t' \in [n\tau, (n+1)\tau)$ such that $\varsigma(t)(p) = \bot$ for all $t \leqslant t'$ and $\varsigma(t)(p) = \top$ for all $t > t'$.
- $[\varsigma]_\tau(n)(p) = N$ if for all $t \in [n\tau, (n+1)\tau]$, $\varsigma(t)(p) = \bot$.
- $[\varsigma]_\tau(n)(p)$ is undefined if $[\varsigma]_\tau(n)(p) \notin \{A, Z, E, N\}$.

To ensure that $[\varsigma]_\tau(n)$ is well-defined for all $p \in \mathrm{AP}$, we impose the following assumptions.

Assumption 1. *For all trajectories σ, all $p \in \mathrm{AP}$, all $t \in \mathbb{R}_{\geqslant 0}$, and all $t' \in [0, \tau]$,*

$$P(\sigma(t))(p) = P(\sigma(t+t'))(p) \Rightarrow \forall t'' \in [0, t'],\ P(\sigma(t))(p) = P(\sigma(t+t''))(p). \quad (1)$$

$$P(\sigma(t))(p) \neq P(\sigma(t+\tau))(p) \ \mathit{and}\ P(\sigma(t))(q) \neq P(\sigma(t+\tau))(q) \Rightarrow p = q. \quad (2)$$

The property in (1) restricts that, within time τ, a system trajectory cannot cross the border of each AP region twice. It is possible to ensure that the system trajectories have the property by appropriately designing or selecting a Lyapunov-like barrier function (see, e.g., [31, 32]) to enforce that any deviation of $\sigma(t)$ from the initial AP region results in a monotonic decrease in a certificate function over $[0, \tau]$, thus preventing the system from returning to its initial AP region within the time horizon. The property in (2) implies that a system trajectory can cross at most one AP region boundary within a time interval of length τ. To enforce this property, one may take τ small enough so that the minimum distance between the boundaries of any two AP regions is greater than the distance the system can travel in time τ. This is made formal by the following lemma.

Lemma 1. *System Σ has* bounded speed *if there exists $\Delta\colon \mathbb{R}_{>0} \to \mathbb{R}_{>0}$ such that for all $t \in \mathbb{R}_{>0}$, $x \in X$, $t' \leqslant t$, and $y \in \xi(x, t')$, $\|y - x\|_\infty < \Delta(t)$. It is* AP-separated *if for all $p \neq q \in \mathrm{AP}$, there exists $d_{p,q} > 0$ such that the distance from the (topological) boundary of $\{x \in X \mid p \in P(x)\}$ and $\{x \in X \mid q \in P(x)\}$ is greater than $d_{p,q}$. If Σ is speed-bounded and AP-separated, then for any choice of $\tau < \inf_{p \in \mathrm{AP}} \inf_{q \in \mathrm{AP}, q \neq p} \inf \Delta^{-1}(d_{p,q})$, (2) holds.*

The two properties in Assumption 1 are necessary because we want to prevent different subformulas from changing truth value at different times in the same time interval of length τ and also to prevent two subformulas from having Z and E as observations during the same time interval of length τ. Otherwise, we need to introduce new observations: B (if a formula holds at both ends of the interval, but not on the whole interval) and S (if it holds somewhere but not at the ends). In this paper, we can show that it is possible to deduce the observation of all subformulas from those of atomic propositions (see Lemma 4). If we add these new observations, the result no longer holds, and it is unclear how to construct a sound translation. From Assumption 1, we get the following lemma.

Lemma 2. *Assume both properties in Assumption 1. Given a trajectory σ, let $[\varsigma]_\tau : \mathbb{Z}_{\geqslant 0} \to \mathbb{O}^{\mathrm{AP}}$ be the chopped AP-signal of $\varsigma = P \circ \sigma$. For all $n \in \mathbb{Z}_{\geqslant 0}$,*

1. *For all $p \in \mathrm{AP}$, we have its observation $[\varsigma]_\tau(n)(p) \in \{A, Z, E, N\}$.*
2. *For all $p, q \in \mathrm{AP}$, $[\varsigma]_\tau(n)(p) \in \{Z, E\}$ and $[\varsigma]_\tau(n)(q) \in \{Z, E\}$ implies $p = q$.*

Lemma 2 indicates that at any time step $n \in \mathbb{Z}_{\geqslant 0}$: *1*. $[\varsigma]_\tau(n) \in \mathbb{O}$ for all $p \in \mathrm{AP}$, and *2*. the system trajectory crosses at most one AP-region border during the time interval $[n\tau, (n+1)\tau]$. By this lemma and the definition of signal slicing, we also have the following corollary.

Corollary 1. *Assume both properties in Assumption 1. Given a system trajectory σ, let $[\varsigma]_\tau$ be the chopped AP-signal of $\varsigma = P \circ \sigma$. For all $n \in \mathbb{Z}_{\geqslant 0}$,*

$$[\varsigma]_\tau(n)(p) \in \{A, E\} \iff [\varsigma]_\tau(n+1)(p) \in \{A, Z\}.$$

remark that it works because we consider closed intervals.

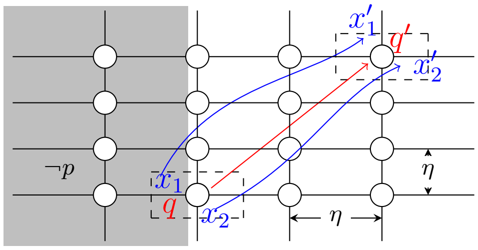

Fig. 1. A quantized state space using a quantization parameter $\eta \in \mathbb{R}_{>0}$. The circles symbolize discrete states. Each discrete state represents (e.g., q and q') a corresponding state region of the size $\eta \times \eta$ (e.g., the dashed boxes around q and q' circles, respectively). The atomic proposition p holds at all states, except those in the left gray half space.

3.2 Symbolic Models

We consider a symbolic model that serves as an abstraction of the dynamical system Σ, not only with respect to the time interval τ discussed in the previous section, but also by abstracting the continuous state space into a discrete set of states. A symbolic model is a labeled transition system [33] $\mathcal{S} = (Q, \delta, q_{\text{in}})$, where Q is a discrete set of states, $\delta \subseteq Q \times \mathbb{O}^{\text{AP}} \times Q$ is a transition relation, and $q_{\text{in}} \in Q$ is the initial state. If a transition $(q, o, q') \in \delta$ exists, this means that the system may move from state q to q' in exactly time τ, provided that the observations of atomic propositions are those given by the function $o\colon \text{AP} \to \mathbb{O}$. The transition system is non-deterministic in the sense that there may exist two transitions $(q, o, q'), (q, o, q'') \in \delta$ with $q' \neq q''$, meaning that the system may transition from q to q' or q''. Moreover, there may exist two transitions $(q, o, q'), (q, o', q') \in \delta$ with $o \neq o'$, meaning that the observation of an atomic proposition $p \in \text{AP}$ may be $o(p)$ or $o'(p)$.

Symbolic models have discrete executions defined in terms of runs, whereas dynamical systems have continuous executions defined in terms of trajectories. Namely, an *infinite (*resp. *finite) run* of the symbolic model $\mathcal{S}$ is a sequence $r_s = q_0 o_0 q_1 \ldots \in Q(\mathbb{O}^{\text{AP}} Q)^{\omega}$ (*resp.* $r_s = q_0 o_0 q_1 \ldots o_{n-1} q_n \in Q(\mathbb{O}^{\text{AP}} Q)^*$ with $n \in \mathbb{Z}_{\geqslant 0}$) such that there exists $(q_i, o_i, q_{i+1}) \in \delta$ for all $i \in \mathbb{Z}_{\geqslant 0}$ (*resp.* $i \in \{0, \ldots, n-1\}$). In what follows, we refer to infinite runs simply as *runs*, and specify *finite runs* explicitly when needed.

A symbolic state $q \in Q$ represents several or an infinite number of actual system states. We are interested in when a symbolic model soundly represents a dynamical system, i.e., when all behaviors of the dynamical system are modeled by those of the symbolic model. In Fig. 1, state q represents all states in the dashed square centered around it, including x_1 and x_2. In this example, both trajectories from x_1 to x'_1 and x_2 to x'_2 are abstracted to the transition from q to q'. However, the atomic proposition p holds throughout the entire trajectory from x_2, but not at the beginning of the trajectory from x_1. Therefore, there must exist two transitions, (q, o, q') and (q, o', q'), where $o(p) = E$ and $o'(p) = A$, reflecting the fact that the atomic proposition p may hold either only at the end or throughout the entire trajectory.

We consider symbolic models constructed by any method as long as they provide the following information.

1. An abstraction map $\gamma : X \to Q$ that maps each system state to its corresponding symbolic state. For the example in Fig. 1, the states in the dashed boxes around q and q' are mapped to q and q', respectively.
2. It must be so that,
$$\text{For all } x \in X \text{ and all } x' \in \xi(x, \tau), \text{ there exists } (\gamma(x), o, \gamma(x')) \in \delta. \tag{3}$$
Namely, there always exists a transition $(\gamma(x), o, \gamma(x')) \in \delta$ representing a trajectory from x to x'. This property is known as *approximate simulation* and can be ensured by constructing symbolic models using the methods proposed in [25, 28–30]. Note that a transition $(q, o, q') \in \delta$ may represent several trajectories from $\gamma^{-1}(q)$ to $\gamma^{-1}(q')$ under observations given by o.
3. Functions $\rho_Z, \rho_E : Q \times Q \times \mathrm{AP} \to \{+, -, ?\}$ from which we define P_Z, and P_E as follows:
$$P_Z(q, q', p) = \begin{cases} \{A, Z\} & \text{if } \rho_Z(q, q', p) = + \\ \{E, N\} & \text{if } \rho_Z(q, q', p) = - \\ \mathbb{O} & \text{otherwise} \end{cases}$$
$$P_E(q, q', p) = \begin{cases} \{A, E\} & \text{if } \rho_E(q, q', p) = + \\ \{Z, N\} & \text{if } \rho_E(q, q', p) = - \\ \mathbb{O} & \text{otherwise.} \end{cases}$$
They must be such that, for all trajectories σ from $x \in \gamma^{-1}(q)$ to $x' \in \gamma^{-1}(q')$, the observation of p along σ must belong to $P_Z \cap P_E$. Formally, for all $(x, x') \in \gamma^{-1}(q) \times \gamma^{-1}(q')$,
$$x' \in \xi(x, \tau) \implies \forall p \in \mathrm{AP}, [\varsigma]_\tau(0)(p) \in P_Z(q, q', p) \cap P_E(q, q', p), \tag{4}$$
where is $[\varsigma]_\tau$ the chopped AP-signal of $\varsigma = P \circ \sigma$, and $\sigma \colon [0, \tau] \to X$ is the finite trajectory from x to x'. Then, for all $o \colon \mathrm{AP} \to \mathbb{O}$, we require that there exists a transition $(q, o, q') \in \delta$ if
$$o(p) \in P_Z(q, q', p) \cap P_E(q, q', p), \text{for all } p \in \mathrm{AP}\,. \tag{5}$$

An intuitive explanation of the two functions is as follows. The function ρ_Z under-approximates the set of atomic propositions that hold and do not hold along the system trajectory at the beginning (at time **Z**ero). $\rho_Z(q, q', p) = +$ if we know p holds at the beginning of any trajectory from q to q' and $\rho_Z(p) = -$ if we know it never holds at the beginning. It returns ? if the approximation is too imprecise to give an answer or there exist a trajectory where p holds at the beginning and another where it does not. The function ρ_E is ρ_Z's counterpart for the **E**nd of trajectories (at time τ). We show example methods to construct these functions in Section 3.3.

The following theorem ensures that the reduction to symbolic models is sound.

Theorem 1. *Given a dynamical system Σ, let $\mathcal{S}$ be a symbolic model constructed as above. For a trajectory $\sigma\colon \mathbb{R}_{\geqslant 0} \to X$ of Σ, there exists a run $q_0 o_0 q_1 \ldots$ such that, for all $k \in \mathbb{Z}_{\geqslant 0}$,*

$$q_k = \gamma(\sigma(k\tau)) \text{ and } o_k = [\varsigma]_\tau(k),$$

where $[\varsigma]_\tau$ is the chopped AP-signal of $\varsigma = P \circ \sigma\colon \mathbb{R}_{\geqslant 0} \to 2^{\mathrm{AP}}$.

Proof. By induction, there exists a run $q_0 o_0 q_1 \ldots$ such that, for all $k \in \mathbb{Z}_{\geqslant 0}$, we have $q_k = \gamma(\sigma(k\tau))$ by (3), and $o_k = [\varsigma]_\tau(k)$ by (4) and (5). □

3.3 System Discretization

The most common way to construct $\mathcal{S}$ from Σ is to quantize the system state space X into a discrete finite state set Q using fixed-length grid cells. The quantization of space is illustrated in Fig. 1. Formally, $Q = \{(k_1\eta, \ldots, k_n\eta) \in X \mid k_i \in \mathbb{Z}\}$, and γ maps each state X to the closest state in Q (with an arbitrary choice for states at equal distance from several points in Q).

Using the above state abstraction process, one candidate pair of functions ρ_Z and ρ_E that satisfies the requirements in Section 3.2 is as follows:

$$\rho_Z(q, q', p) = \begin{cases} + & \text{if for all } x \in \mathcal{B}_{\eta/2}(q),\ P(x)(p) = \top \\ - & \text{if for all } x \in \mathcal{B}_{\eta/2}(q),\ P(x)(p) = \bot \\ ? & \text{otherwise} \end{cases}$$

$$\rho_E(q, q', p) = \begin{cases} + & \text{if for all } x \in \mathcal{B}_{\eta/2}(q'),\ P(x)(p) = \top \\ - & \text{if for all } x \in \mathcal{B}_{\eta/2}(q'),\ P(x)(p) = \bot \\ ? & \text{otherwise,} \end{cases}$$

where $\mathcal{B}_{\eta/2}(q) = \{x \in \mathbb{R}^n \mid \|q - x\|_\infty \leqslant \eta/2\}$. The intuition is that $\rho_Z(q, q', p)$ (*resp.* $\rho_E(q, q', p)$) should be $+$ if for all trajectories from $x \in \mathcal{B}_{\eta/2}(q)$ to $x' \in \mathcal{B}_{\eta/2}(q')$, p holds at the beginning (*resp.* the end) of the trajectory, i.e., exactly when for all $x \in \mathcal{B}_{\eta/2}(q)$, $P(x)(p) = \top$. By Lemma 2 and (3), using the above functions ρ_Z and ρ_E ensures (4).

4 *AP*-observation Automata

We introduce *AP-observation automata*, where the transitions are labeled by observations of atomic propositions. For a given LTL formula, we construct a generalized *AP*-observation automaton that approximates all observations of the subformulas. This construction is inspired by Vardi and Wolper's translation of LTL formulas to generalized Büchi automata [8], but is specifically adapted to our setting for continuous-time LTL, as we need to consider four observations in $\mathbb{O} = \{A, Z, E, N\}$, instead of the two values $\top$ and $\bot$.

4.1 Signal Word and *AP*-observation Automata

A *signal word* is a function $w\colon \mathbb{Z}_{\geqslant 0} \to \mathbb{O}^{\mathrm{AP}}$ such that for all $k \in \mathbb{Z}_{\geqslant 0}$ and $p \in \mathrm{AP}$, $w_k(p) \in \{A, E\}$ iff $w_{k+1}(p) \in \{A, Z\}$ (where w_k is a shorthand for $w(k)$). Notice that a chopped AP-signal $[\varsigma]_\tau$, defined in Section 3.1, is a word. The intuition is that a signal word is an abstraction of all possible signals that map to it through signal chopping.

In this section, we assume some given LTL formula φ, and we want to construct an automaton $\mathcal{B}_\varphi$ that is sound for φ, i.e., it only accepts words that represent signals that satisfy φ. Formally, we want to build $\mathcal{B}_\varphi$ such that, if w is in its recognized language, then for all signals ς and duration τ, if $[\varsigma]_\tau = w$, then $\varsigma, 0 \models \varphi$. We use $\mathrm{sub}(\varphi)$ to denote the set of subformulas of φ.

We now introduce *AP*-observation automata, which we use to verify that dynamical systems satisfy continuous-time LTL properties. They are very similar to classic Büchi automata used for verification of LTL, but one crucial difference is that they work on signal words on the alphabet $\mathbb{O}^{\mathrm{AP}}$, rather than words on the alphabet 2^{AP}.

A *non-deterministic AP-observation automaton* (or simply *AP-observation automaton*) is a tuple $\mathcal{B} = (B, \delta_b, b_{\mathrm{in}}, F)$, where B is a finite set of states, $\delta_b \subseteq B \times \mathbb{O}^{\mathrm{AP}} \times B$ is the transition relation, $b_{\mathrm{in}} \in B$ is the initial state, and $F \subseteq B$ is the set of accepting states. A *run* of a signal word w through $\mathcal{B}$ is an infinite sequence of states $b_0 b_1 \ldots$ such that $b_0 = b_{\mathrm{in}}$ and for all $k \in \mathbb{Z}_{\geqslant 0}$, $(b_k, w_k, b_{k+1}) \in \delta$. A run is accepting if it visits F infinitely many times. The *recognized language* of $\mathcal{B}$ is the set of signal words that induce at least one accepting run.

Like the original LTL-to-Büchi-automaton construction [8], we start by first building a generalized *AP*-observation automaton $\mathcal{G}_\varphi$, then turn it into a (non-deterministic) *AP*-observation automaton $\mathcal{B}_\varphi$. The following construction is the counterpart of generalized Büchi automata. A *generalized AP-observation automaton* is a tuple $\mathcal{B} = (B, \delta_b, b_{\mathrm{in}}, \mathcal{F})$, where $\mathcal{F} \subseteq \mathcal{P}(B)$ is a set of accepting sets. All definitions are similar to those of *AP*-observation automata, except that a run is accepting if it visits all $F \in \mathcal{F}$ infinitely often. Figure 2 shows an example of a generalized Büchi automaton $(\{q_0, q_1, q_2, q_3\}, \delta_b, q_0, \{\{q_2, q_3\}, \{q_1, q_2, q_3\}\})$, where δ_b can be derived from the picture. For example, $(q_0, g^N, q_1) \in \delta_b$, where g^N denotes the observation function that maps g to N.

4.2 Translation to Generalized *AP*-observation Automaton

Given an LTL formula φ, the construction of a generalized *AP*-observation automaton $\mathcal{G}_\varphi$ relies on the set $\mathrm{cs}(\varphi)$ of consistent subformula valuations of φ. We first present the construction of the automaton and define $\mathrm{cs}(\varphi)$ later. $\mathcal{G}_\varphi = (G_\varphi, \delta_\varphi, q_0, \mathcal{F}_\varphi)$ is constructed as follows.

- $G_\varphi = \mathrm{cs}(\varphi) \cup \{q_0\}$.
- For all states $\nu \in \mathrm{cs}(\varphi)$, $(\nu, o, \nu') \in \delta_\varphi$ iff (A1) for all $p \in \mathrm{AP}$, $\nu'(p) = o(p)$, and (A2) for all subformulas $\psi \in \mathrm{sub}(\varphi)$, $\nu(\psi) \in \{A, E\}$ iff $\nu'(\psi) \in \{A, Z\}$.
- There exists $(q_0, o, \nu') \in \delta_\varphi$ from the initial state q_0 iff (B1) for all $p \in \mathrm{AP}$, $\nu'(p) = o(p)$, and (B2) $\nu'(\varphi) \in \{A, Z\}$.

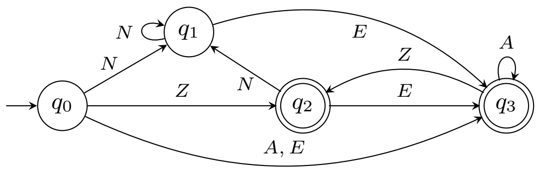

Fig. 2. The generalized AP-observation automaton for verifying the LTL formula $\Box \Diamond g$ ($\equiv \bot \,\mathrm{R}\, (\top \,\mathrm{U}\, g)$) in Section 6, but minimized by merging equivalent states (q_3 is a merged state). The subformulas are g, $\Diamond g$, and $\Box \Diamond g$, and the accepting state sets are $\mathcal{F} = \{F_{\Diamond g} = \{q_2, q_3\}, F_{\Box \Diamond g} = \{q_1, q_2, q_3\}\}$. Apart from q_0, the only non-accepting state (for $\Diamond g$) is q_1. This reflects the fact that the system violates the specification only if its trajectory never visits a region satisfying g after some point, i.e., the observation along the transitions is always N (g is never satisfied) after that point.

ψ_1	ψ_2	$\psi_1 \wedge \psi_2$	$\psi_1 \vee \psi_2$	$\psi_1 \,\mathrm{U}\, \psi_2$	$\psi_1 \,\mathrm{R}\, \psi_2$
$\underline{\mathbf{A}}$	A	A	A	A	A
	$\mathbf{Z}$	$\mathbf{Z}$	A	$\mathbf{AZ}$	Z
	E	E	A	A	E
	$\underline{N}$	N	A	$\underline{AN}$	N
Z	A	Z	A	A	AZ
	Z	Z	Z	Z	Z
	E	N	A	A	EN
	N	N	Z	N	N

ψ_1	ψ_2	$\psi_1 \wedge \psi_2$	$\psi_1 \vee \psi_2$	$\psi_1 \,\mathrm{U}\, \psi_2$	$\psi_1 \,\mathrm{R}\, \psi_2$
E	A	E	A	A	A
	Z	N	A	AZ	N
	E	E	E	E	E
	N	N	E	EN	N
N	A	N	A	A	AN
	Z	N	Z	Z	N
	E	N	E	E	EN
	N	N	N	N	N

Fig. 3. The consistency rules for generalized AP-observation automata. add a note for ZE and EZ, that they are just for cases such as p and neg p

- The set of accepting state-sets is

$$\mathcal{F}_\varphi = \{F_{\psi_1 \,\mathrm{U}\, \psi_2} \mid \psi_1 \,\mathrm{U}\, \psi_2 \in \mathrm{sub}(\varphi)\} \cup \{F_{\psi_1 \,\mathrm{R}\, \psi_2} \mid \psi_1 \,\mathrm{R}\, \psi_2 \in \mathrm{sub}(\varphi)\}\,, \qquad (6)$$

  where $F_{\psi_1 \,\mathrm{U}\, \psi_2} = \{\nu \in \mathrm{cs}(\varphi) \mid \nu(\psi_2) \neq N$ or $\nu(\psi_1 \,\mathrm{U}\, \psi_2) \neq A\}$ and $F_{\psi_1 \,\mathrm{R}\, \psi_2} = \{\nu \in \mathrm{cs}(\varphi) \mid \nu(\psi_2) \neq A$ or $\nu(\psi_1 \,\mathrm{R}\, \psi_2) \neq N\}$.

Next, we define $\mathrm{cs}(\varphi)$. A *subformula valuation* is a function $\nu\colon \mathrm{sub}(\varphi) \to \mathbb{O}$, where $\mathrm{sub}(\varphi)$ is the set of all subformulas of φ. For example, in Fig. 2, the valuations of all states map $\Diamond g (\equiv \top \,\mathrm{U}\, g)$ and $\Box \Diamond g (\equiv \bot \,\mathrm{R}\, \Diamond g)$ to A, state q_1 maps g to N, q_2 maps it to Z, and q_3 (which is a merge of two states) maps it either to A or E. The valuation ν is *consistent* if ν follows the rules given by Fig. 3. Then, $\mathrm{cs}(\varphi)$ is the set of consistent subformula valuations of φ.

The way to read the table in Fig. 3 is as follows: given observations for subformulas ψ_1 and ψ_2, the consistent observations of $\psi_1 \wedge \psi_2$, $\psi_1 \vee \psi_2$, $\psi_1 \,\mathrm{U}\, \psi_2$, and $\psi_1 \,\mathrm{R}\, \psi_2$ are given in the table. The intuition is that a valuation ν represents the current state of all subformulas. For example, let us consider the second row of the table (the one with bold blue text). It states in particular that, if

$\nu(\psi_1) = A$ and $\nu(\psi_2) = Z$, then $\nu(\psi_1 \wedge \psi_2) = Z$ and $\nu(\psi_1 \mathrm{U}\, \psi_2) \in \{A, Z\}$. Indeed, if ψ_1 holds on a whole time interval and ψ_2 holds at its beginning but not at the end, then $\psi_1 \wedge \psi_2$ also holds at the beginning and not at the end, and $\psi_1 \mathrm{U}\, \psi_2$ either holds only at the beginning (if ψ_2 never holds again, or ψ_1 stops holding before ψ_2 holds again) or on the whole interval (if ψ_1 keeps holding until ψ_2 holds again).

A noteworthy case is in the fourth row of the table (the underlined one): $\nu(\psi_1) = A$, $\nu(\psi_2) = N$, and $\nu(\psi_1 \mathrm{U}\, \psi_2) \in \{A, N\}$. If ψ_1 holds at all times and ψ_2 holds at none of the interval, then $\psi_1 \mathrm{U}\, \psi_2$ either holds for the whole interval (if ψ_1 keeps holding until ψ_2 holds) or does not hold in the interval (if ψ_1 stops holding before ψ_2 holds). Notice that a run $\nu_1\nu_2\ldots$ that assigns $\nu_k(\psi_1 \mathrm{U}\, \psi_2) = A$ and $\nu_k(\psi_2) = N$ at all time steps k is not an accepting run thanks to $F_{\psi_1 \mathrm{U}\, \psi_2}$.

Note in particular that the table has no rows $\psi_1 = Z$, $\psi_2 = E$ or $\psi_1 = E$, $\psi_2 = Z$. That is because it can never be the case that ψ_1 holds at the beginning of an interval and ψ_2 at the end by Assumption 1.

Formally, the connector $\wedge$ comes equipped with a function $c_\wedge : \mathbb{O} \times \mathbb{O} \to 2^{\mathbb{O}}$ described by the third column of Fig. 3 (and similarly for connectors $\vee$, U, and R). A subformula valuation $\nu\colon \operatorname{sub}(\varphi) \to \mathbb{O}$ is *consistent* if for all $\psi_1, \psi_2 \in \operatorname{sub}(\varphi)$ such that $\psi_1 \odot \psi_2 \in \operatorname{sub}(\varphi)$, $\nu(\psi_1 \odot \psi_2) \in c_\odot(\nu(\psi_1), \nu(\psi_2))$ for all connectors $\odot \in \{\wedge, \vee, \mathrm{U}, \mathrm{R}\}$.

The following lemma states that the table in Fig. 3 is sound and complete. Note that the AP-signals ς referred to in this lemma are general, and not necessarily those generated by Σ.

Lemma 3. *For all connectors $\odot \in \{\wedge, \vee, \mathrm{U}, \mathrm{R}\}$, formulas $\psi = \psi_1 \odot \psi_2$, and signals $\varsigma\colon \mathbb{R}_{\geqslant 0} \to 2^{\mathrm{AP}}$ and $\tau \in \mathbb{R}_{>0}$ that satisfy Assumption 1, $[\varsigma]_\tau(k)(\psi) \in c_\odot([\varsigma]_\tau(k)(\psi_1), [\varsigma]_\tau(k)(\psi_2))$ for all $k \in \mathbb{Z}_{\geqslant 0}$.*

Moreover, for all $\psi = \psi_1 \odot \psi_2$, $o_1, o_2 \in \mathbb{O}$ and $o \in c_\odot(o_1, o_2)$, there exists a signal $\varsigma\colon \mathbb{R}_{\geqslant 0} \to 2^{\mathrm{AP}}$ and $\tau \in \mathbb{R}_{>0}$ that satisfy Assumption 1 and such that, for all $k \in \mathbb{Z}_{\geqslant 0}$, $[\varsigma]_\tau(k)(\psi_i) = o_i$ for $i \in \{1, 2\}$ and $[\varsigma]_\tau(k)(\psi) = o$.

The proof can be found in Appendix A.1. The following theorem proves that the generalized AP-observation automaton construction of $\mathcal{G}_\varphi$ is sound.

Theorem 2. *For all continuous-time LTL formulas φ, AP-signals $\varsigma\colon \mathbb{R}_{\geqslant 0} \to 2^{\mathrm{AP}}$, and durations τ, if $[\varsigma]_\tau$ is in the recognized language of $\mathcal{G}_\varphi$, then $\varsigma, 0 \vDash \varphi$.*

We relegate the full proof to the appendix and only state a few crucial lemmas. The following lemma gives a fundamental property of $\mathcal{G}_\varphi$: given a word, there exists exactly one non-initial state and one accepting run along that word from that state. Its proof in A.2 heavily relies on Lemma 3 in order to show by induction on subformulas ψ that there is a unique possibility for $\nu_k(\psi)$.

Lemma 4. *For all words $w\colon \mathbb{Z}_{\geqslant 0} \to \mathbb{O}^{\mathrm{AP}}$ such that*

$$\forall k \in \mathbb{Z}_{\geqslant 0}.\ \forall p \in \mathrm{AP}\,.\ w_k(p) \in \{A, E\} \iff w_{k+1}(p) \in \{A, Z\}\,, \tag{7}$$

$$\forall k \in \mathbb{Z}_{\geqslant 0}.\ \forall p, q \in \mathrm{AP}\,.\ w_k(p), w_k(q) \in \{Z, E\} \Rightarrow p = q, \tag{8}$$

there exists a unique accepting run $\nu_0\nu_1\dots$ *such that for all* $k \in \mathbb{Z}_{\geqslant 0}$ *and* $p \in \mathrm{AP}$, $\nu_k(p) = w_k(p)$.

The following corollary demonstrates that accepting runs are, in fact, exactly valuations of signal choppings. Its proof is relegated to Appendix A.3, but it crucially uses Assumption 1 to show that signal choppings have the shape described in the proof of Lemma 4.

Corollary 2. *Given* $\varsigma\colon \mathbb{R}_{\geqslant 0} \to 2^{\mathrm{AP}}$ *and* $\tau \in \mathbb{R}_{>0}$, *a run* $\nu_0\nu_1\dots$ *such that* $\nu_k(p) = [\varsigma]_\tau(k)(p)$, *for all* $k \in \mathbb{Z}_{\geqslant 0}$ *and* $p \in \mathrm{AP}$, *is an accepting run iff* $[\varsigma]_\tau(k)(\psi) = \nu_k(\psi)$, *for all* $k \in \mathbb{Z}_{\geqslant 0}$ *and* $\psi \in \mathrm{sub}(\varphi)$.

Theorem 2 follows directly from Corollary 2, using the shape of transitions from the initial state, as we only have transitions from q_0 to ν with $\nu(\varphi) \in \{A, Z\}$.

This construction is inspired by the LTL-to-Büchi-automaton construction by Vardi and Wolper [8], which we briefly discussed in Section 2.4. However, there is a fundamental difference in that there are no explicit constraints on transitions. Indeed, in the original construction, where states are consistent valuations $\nu\colon\ \mathrm{sub}(\varphi) \to 2$, there can be a transition from ν to ν' only if they "agree" on the value of all formulas $\psi_1 \,\mathrm{U}\, \psi_2$ and $\psi_1 \,\mathrm{R}\, \psi_2$. This uses the fact that, in discrete time, $\psi_1 \,\mathrm{U}\, \psi_2 \iff \psi_2 \vee (\psi_1 \wedge \mathrm{X}(\psi_1 \,\mathrm{U}\, \psi_2))$, so for example if $\psi_1 \,\mathrm{U}\, \psi_2$ holds in ν, then either ψ_2 should hold in ν, or ψ_1 should hold in ν and $\psi_1 \,\mathrm{U}\, \psi_2$ in ν'. Similarly, for Release, using the fact that $\psi_1 \,\mathrm{R}\, \psi_2 \iff \psi_2 \wedge (\psi_1 \vee \mathrm{X}(\psi_1 \,\mathrm{R}\, \psi_2))$.

In our translation, this constraint comes from the fact that the consistency rules in Fig. 3 also contain constraints on Until and Release, while the original translation only has constraints on conjunction and disjunction. This, coupled with a generalization of (7) to all subformulas, gives constraints between valuations in ν and ν'.

talk about the size

However, we need to add another constraint to only retain good behaviors. Indeed, while the original construction only has accepting sets for Until subformulas, here we also need to add accepting sets for the Release subformulas to make up for the constraints in the original construction.

Following the translation from generalized Büchi automata to non-deterministic Büchi automata [10], we can translate generalized AP-observation automata to their *non-deterministic AP-observation automata* counterparts. In our implementation for the example in Section 6, before applying this translation, we prune the generalized AP-observation automaton $\mathcal{G}_\varphi$ by removing states that cannot lead to any accepting run (i.e., that fail to reach at least one state in each $F \in \mathcal{F}(\mathcal{G}_\varphi)$), and merge states that are equivalent with respect to acceptance conditions and outgoing transitions. The generalized AP-observation automaton shown in Fig. 2 reflects the outcome of this pruning and minimization.

5 System Verification

We know how to build a symbolic model $\mathcal{S} = (Q, \delta_s, q_{\mathrm{in}})$ that over-approximates the behaviors of the dynamical system Σ in the sense of Theorem 1 and a non-deterministic AP-observation automaton $\mathcal{B}_\varphi = (B, \delta_b, b_{\text{in}}, F)$ whose language is that of φ in the sense of Theorem 2.

Both $\mathcal{S}$ and $\mathcal{B}_\varphi$ are non-deterministic. The non-determinism in $\mathcal{S}$ is demonic and comes from that of Σ: if $(q, o_1, q_1), (q, o_2, q_2) \in \delta_s$, then we cannot choose whether the system goes to q_1 by reading o_1 and to q_2 by reading o_2. The non-determinism in $\mathcal{B}_\varphi$ is angelic: a word is recognized if there exists an accepting run. In order to mix these two forms of non-determinism, we build a Büchi game [34], which is a particular type of parity game [35] (with parities 1 and 2).

A *Büchi game* is a tuple $\mathcal{G} = (G, G_0, \delta_g, F_g, g_{\text{in}})$, where G is a set of *vertices*, $G_0 \subseteq G$ is the set of *Player* vertices, $G_1 = G \backslash G_0$ is that of *Opponent* vertices, $\delta_g \subseteq G \times G$ is a set of *edges*, and $F_g \subseteq G$ is the set of *Büchi vertices*. A *run* is a sequence $g_0 g_1 \ldots$ of states such that $(g_i, g_{i+1}) \in \delta_g$ for all $i \in \mathbb{Z}_{\geqslant 0}$. It is *accepting* if it reaches F_g infinitely often or ends in $g_i \in G_1$ such that there are no $(g_i, g') \in \delta_g$. A *Player strategy* is a function $\pi \colon G_0 \to G$ such that for all $g \in G_0$, $(g, \pi(g)) \in \delta_g$, and similarly for *Opponent strategy*. The run *induced* by a Player strategy π_0 and an Opponent strategy π_1 from a state g_0 is the sequence $g_0 g_1 \ldots$ such that for all $i \in \mathbb{Z}_{\geqslant 0}$, if $g_i \in G_k$, then $\pi_k(g_i) = g_{i+1}$. A Player strategy is (from g) *winning* if for all Opponent strategies, the resulting run from g is accepting. A state g is winning if there exists a winning Player strategy from g, and $\mathcal{G}$ is winning if g_{in} is.

Given a non-deterministic AP-observation automaton $\mathcal{B}$ and a symbolic model $\mathcal{S}$, we build $\mathcal{S} \times \mathcal{B} = (G, G_0, \delta_g, g_{\text{in}})$ as follows:

- $G = \{(q, b) \mid q \in Q, b \in B\} \cup \{(q, o, b) \mid q \in Q, o \in \mathbb{O}, b \in B\}$,
- $G_0 = \{(q, o, b) \mid q \in Q, o \in \mathbb{O}, b \in B\}$,
- $((q, b), (q', o, b)) \in \delta_g$ if and only if $(q, o, q') \in \delta_s$, and $((q, o, b), (q, b')) \in \delta_g$ if and only if $(b, o, b') \in \delta_b$,
- $F_g = \{(q, b) \mid b \in F\}$,
- $g_{\text{in}} = (q_{\text{in}}, b_{\text{in}})$.

Theorem 3. *Given a symbolic model $\mathcal{S}$ an AP-observation automaton $\mathcal{B}$, $\mathcal{S} \times \mathcal{B}$ is winning iff all runs $q_{\text{in}} \xrightarrow{o_0} q_1 \xrightarrow{o_1} \ldots$ of $\mathcal{S}$ are such that $o_0 o_1 \ldots$ is in the recognized language of $\mathcal{B}$.*

Proof. It is well-known that positional strategies are optimal [35]. In particular, if a positional Player strategy π_0 wins against all positional Opponent strategies, then it wins against all (general) Opponent strategies $\pi_1 \colon G^* G_1 \to G$ that map each play to a next state. Assuming that π_0 is a winning strategy, given a run $q_{\text{in}} \xrightarrow{o_0} q_1 \xrightarrow{o_1} \ldots$ of $\mathcal{S}$, we define

$$\pi_1((q_{\text{in}}, b_{\text{in}}), (q_1, o_0, b_{\text{in}}), \ldots, (q_n, b_n)) = (q_{n+1}, o_n, b_n).$$

Because π_0 wins against π_1, the induced play visits $F_g = \{(q, b) \mid b \in F\}$ infinitely often, so $b_{\text{in}} \xrightarrow{o_0} b_1 \xrightarrow{o_1} \ldots$ is an accepting run of $\mathcal{B}$, and therefore $o_0 o_1 \ldots$ is in the recognized language of $\mathcal{B}$. □

Putting Theorems 1, 2, and 3 together, we get the following corollary.

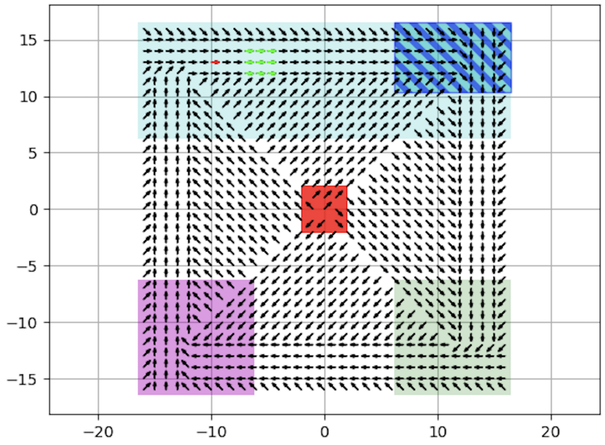

Fig. 4. A surveillance drone system flying from the position $(-10, 13)$ (base of the red arrow). The atomic propositions are assigned to xy-coordinates as follows: c (cyan) for $y \geqslant 6.21$, b (blue) for $(x, y) \geqslant (6.21, 10.32)$, p (pink) for $(x, y) \leqslant (6.21, 6.21)$, g (green) for $x \geqslant 6.21$ and $y \leqslant 6.21$, and r (not red) for $|x| > 2.1$ and $|y| > 2.1$. The states reached in a single transition from the initial state are at the bases of the green arrows.

Corollary 3. *Given a dynamical system Σ, a symbolic model $\mathcal{S}$ that soundly represents Σ (constructed as in Section 3.2), and a continuous-time LTL formula φ, if $\mathcal{S} \times \mathcal{B}_\varphi$ is winning, then for all trajectories σ of Σ from x_{in}, $\sigma, 0 \models \varphi$.*

This gives a sound algorithm for Problem 1. If the game is winning, then all trajectories σ of Σ from x_{in} are such that $\sigma, 0 \models \varphi$, but otherwise we cannot conclude that there exists a trajectory such that $\sigma, 0 \not\models \varphi$. The completeness of this result cannot be guaranteed, as the exact values of atomic propositions along trajectories are lost during the discretization of the system's state space. To mitigate this theoretic limitation, we demonstrate the practical feasibility of our approach by verifying several specifications in Section 6.

6 Illustrative Example

To illustrate a potential application of the proposed structure, we present an illustrative application example of verifying a surveillance drone system. As shown in Fig. 4, the drone flies in an area of $33 \times 33\text{m}^2$ and can move in eight directions: the four cardinal directions and the four diagonal directions, following linear dynamics at a speed of 4m/s. The system's non-determinism comes from environmental disturbances impacting the drone's speed and direction. Its speed may vary by up to 0.1m/s and its angle by up to 0.08 radians.

We construct a symbolic model $\mathcal{S}$ using $\eta = 1\text{m}$ and $\tau = 1\text{s}$, and fix the moving direction for each symbolic state as depicted by the arrows in Fig. 4. We implement Zielonka's algorithm [35] to solve the game described in Section 5.

Specification	$\mathcal{B}_\varphi$		Game construction		Game solving	**Total Time**
	Size	**Time(s)**	**Size**	**Time(s)**	**Time(s)**	
$\Box\, r$	2	0.01	651 + 642	0.23	0.35	1.34
$\Diamond\, p$	5	0.01	757 + 680	0.52	0.46	1.75
$c\,\mathrm{U}\,b$	7	0.04	830 + 691	0.69	0.55	2.04
$b\,\mathrm{R}\,c$	7	0.04	814 + 685	0.69	0.53	2.03
$\Diamond\Box\, r$	6	0.02	1,933+1,924	0.60	2.05	3.44
$\Box\Diamond\, g$	7	0.02	4,640+2,631	1.93	2.48	5.20
$\Diamond(g \wedge \Diamond\, p)$	33	0.40	4,856+2,687	8.98	3.05	13.19
$\Box\, r \wedge (\Diamond\, p \wedge \Diamond\, c)$	46	51.39	7,723+4,144	29.66	7.54	89.36
$\Box\, r \wedge \Diamond(g \wedge \Diamond\, p)$	49	53.86	7,279+4,030	25.49	7.06	87.18

Fig. 5. Sizes (number of states) and construction times (averaged over 10 runs) for the deadlock-free reachable parts of the nondeterministic AP-observation automata $\mathcal{B}_\varphi$ and the corresponding parity games. For the games, sizes are reported as the total number of states controlled by Player (angelic nondeterminism in $\mathcal{B}_\varphi$) and Opponent (demonic nondeterminism in $\mathcal{S}$). The table also shows average game construction times for the deadlock-free reachable parts, game-solving times, and total times per specification. Total times include the construction of the symbolic model $\mathcal{S}$, which has 1,089 states and takes an average of 0.77 seconds to build.

We implement the algorithm in Python 3.11.2 and run it on a MacBook Pro (Apple M2 Max, 64GB). We verify several LTL formulas as shown in Fig. 5.

For complex specifications, the main bottleneck lies in the construction of the AP-observation automaton. This step is computationally expensive due to the enumeration of all possible valuations of subformulas, which grows exponentially with the size of the formula. For example, the formula $\Box\, r \wedge \Diamond(g \wedge \Diamond\, p)$ has ten subformulas (including $\top$ and $\bot$). Therefore, we consider all 4^{10} valuations and check if they are consistent before pruning deadlocked states and minimizing the automaton. This implementation serves as a simple baseline to demonstrate the feasibility of the method, and we leave it as future work to optimize it, for example, by considering only valuations at reachable states.

7 Conclusions

We have introduced a novel translation of LTL formulas to AP-observation automata, which is specifically designed for verification of continuous-time systems by abstracting truth values on an interval to four possible patterns. We have presented a verification algorithm that uses this translation for the abstraction-based verification of nonlinear, non-deterministic, continuous-time, continuous-state systems without global stability assumptions.

In the future, we plan to adapt this framework to tackle the symbolic controller synthesis problem. We also want to weaken the constraints imposed on the system, specifically that when an atomic proposition holds, it should continue to hold for a certain amount of time. This can be done by introducing other observation patterns, which makes the construction more complex.

Acknowledgments The authors would like to thank Jérémy Dubut for letting them reuse part of his implementation.

References

1. A. Pnueli, "The temporal logic of programs," in *18th annual symposium on foundations of computer science (SFCS 1977)*. ieee, 1977, pp. 46–57.
2. C. Kern and M. R. Greenstreet, "Formal verification in hardware design: a survey," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 4, no. 2, pp. 123–193, 1999.
3. A. Pnueli, "Applications of temporal logic to the specification and verification of reactive systems: a survey of current trends," *Current Trends in Concurrency: Overviews and Tutorials*, pp. 510–584, 2005.
4. K. Y. Rozier, "Linear temporal logic symbolic model checking," *Computer Science Review*, vol. 5, no. 2, pp. 163–203, 2011.
5. C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G. J. Pappas, "Symbolic planning and control of robot motion [grand challenges of robotics]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 61–70, 2007.
6. C. Belta and S. Sadraddini, "Formal methods for control synthesis: An optimization perspective," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, no. 1, pp. 115–140, 2019.
7. A. Banerjee and V. Choppella, "Challenges and opportunities in the industrial usage controller synthesis tools: A review of ltl-based opensource tools for automated control design," *Results in Control and Optimization*, p. 100511, 2025.
8. M. Y. Vardi and P. Wolper, "An automata-theoretic approach to automatic program verification," in *1st Symposium in Logic in Computer Science (LICS)*. IEEE Computer Society, 1986.
9. R. Gerth, D. Peled, M. Y. Vardi, and P. Wolper, "Simple on-the-fly automatic verification of linear temporal logic," in *International Conference on Protocol Specification, Testing and Verification*. Springer, 1995, pp. 3–18.
10. P. Gastin and D. Oddoux, "Fast LTL to Büchi automata translation," in *Computer Aided Verification: 13th International Conference, CAV 2001 Paris, France, July 18–22, 2001 Proceedings 13*. Springer, 2001, pp. 53–65.
11. K. Ogata *et al.*, *Modern control engineering*. Prentice Hall India, 2009.
12. H. K. Khalil and J. W. Grizzle, *Nonlinear systems*. Prentice hall Upper Saddle River, NJ, 2002, vol. 3.
13. K. Gu and S.-I. Niculescu, "Survey on recent results in the stability and control of time-delay systems," *J. Dyn. Sys., Meas., Control*, vol. 125, no. 2, pp. 158–165, 2003.
14. S. Sastry and M. Bodson, *Adaptive control: stability, convergence and robustness*. Courier Corporation, 2011.
15. A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.
16. P. Doherty, J. Kvarnström, and F. Heintz, "A temporal logic-based planning and execution monitoring framework for unmanned aircraft systems," *Autonomous Agents and Multi-Agent Systems*, vol. 19, pp. 332–377, 2009.
17. S. Jha, V. Raman, D. Sadigh, and S. A. Seshia, "Safe autonomy under perception uncertainty using chance-constrained temporal logic," *Journal of Automated Reasoning*, vol. 60, pp. 43–62, 2018.
18. N. Arechiga, "Specifying safety of autonomous vehicles in signal temporal logic," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 58–63.

19. P. Tabuada, “Approximate simulation relations and finite abstractions of quantized control systems,” in *International Workshop on Hybrid Systems: Computation and Control*. Springer, 2007, pp. 529–542.
20. G. Pola, A. Girard, and P. Tabuada, “Approximately bisimilar symbolic models for nonlinear control systems,” *Automatica*, vol. 44, no. 10, pp. 2508–2516, 2008.
21. P. Tabuada, “An approximate simulation approach to symbolic control,” *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1406–1418, 2008.
22. E. Macoveiciuc and G. Reissig, “Memory efficient symbolic solution of quantitative reach-avoid problems,” in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 1671–1677.
23. P.-J. Meyer and D. V. Dimarogonas, “Hierarchical decomposition of LTL synthesis problem for nonlinear control systems,” *IEEE transactions on automatic control*, vol. 64, no. 11, pp. 4676–4683, 2019.
24. Y. Bai, K. Mallik, A.-K. Schmuck, D. Zufferey, and R. Majumdar, “Incremental abstraction computation for symbolic controller synthesis in a changing environment,” in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 6261–6268.
25. S. Pruekprasert, C. Eberhart, and J. Dubut, “Fast synthesis for symbolic self-triggered control under right-recursive LTL specifications,” in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 1321–1328.
26. E. Macoveiciuc and G. Reissig, “On-the-fly symbolic synthesis with memory reduction guarantees,” *IEEE Transactions on Automatic Control*, vol. 68, no. 4, pp. 2576–2583, 2022.
27. W. Ren, R. M. Jungers, and D. V. Dimarogonas, “Zonotope-based symbolic controller synthesis for linear temporal logic specifications,” *IEEE Transactions on Automatic Control*, 2024.
28. S. Pruekprasert, C. Eberhart, and J. Dubut, “Symbolic self-triggered control of continuous-time non-deterministic systems without stability assumptions for 2-LTL specifications,” in *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2020, pp. 548–554.
29. M. Zamani, G. Pola, M. Mazo, and P. Tabuada, “Symbolic Models for Nonlinear Control Systems Without Stability Assumptions,” *IEEE Trans. Autom. Control*, vol. 57, no. 7, pp. 1804–1809, 2012.
30. K. Hashimoto and D. V. Dimarogonas, “Synthesizing Communication Plans for Reachability and Safety Specifications,” *IEEE Trans. Autom. Control*, vol. 65, no. 2, pp. 561–576, 2019.
31. J. Lygeros, K. H. Johansson, S. N. Simic, J. Zhang, and S. S. Sastry, “Dynamical properties of hybrid automata,” *IEEE Transactions on automatic control*, vol. 48, no. 1, pp. 2–17, 2003.
32. D. Panagou, D. M. Stipanović, and P. G. Voulgaris, “Distributed coordination control for multi-robot networks using lyapunov-like barrier functions,” *IEEE Transactions on Automatic Control*, vol. 61, no. 3, pp. 617–632, 2015.
33. C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press, 2008.
34. K. Chatterjee and M. Henzinger, “Efficient and dynamic algorithms for alternating büchi games and maximal end-component decomposition,” *Journal of the ACM (JACM)*, vol. 61, no. 3, pp. 1–40, 2014.
35. W. Zielonka, “Infinite games on finitely coloured graphs with applications to automata on infinite trees,” *Theoretical Computer Science*, vol. 200, no. 1-2, pp. 135–183, 1998.

A Omitted Proofs

A.1 Proof of Lemma 3

For soundness, we proceed by case distinction on $\odot$ in $\psi = \psi_1 \odot \psi_2$, then on case distinction on the values $[\varsigma]_\tau(k)(\psi_1)$ and $[\varsigma]_\tau(k)(\psi_2)$.

- If $\psi = \psi_1 \wedge \psi_2$, we only show that if $[\varsigma]_\tau(k)(\psi_1) = [\varsigma]_\tau(k)(\psi_2) = A$, then $[\varsigma]_\tau(k)(\psi) = A$, the other cases are similar. We have that for all $t \in [k\tau, (k+1)\tau]$, $\varsigma, t \models \psi_1$ and $\varsigma, t \models \psi_2$, so for all $t \in [k\tau, (k+1)\tau]$, $\varsigma, t \models \psi$.
- If $\psi = \psi_1 \,\mathrm{U}\, \psi_2$, we only show that if $[\varsigma]_\tau(k)(\psi_1) = A$ and $[\varsigma]_\tau(k)(\psi_2) = N$, then $[\varsigma]_\tau(k)(\psi) \in \{A, N\}$, the other cases are simpler. We have that for all $t \in [k\tau, (k+1)\tau]$, $\varsigma, t \models \psi_1$ and $\varsigma, t \not\models \psi_2$. Therefore, either there exists $t > (k+1)\tau$ such that $\varsigma, t \models \psi_2$ and for all $t' \in ((k+1)\tau, t')$, $\varsigma, t' \models \psi_1$, in which case $[\varsigma]_\tau(k)(\psi) = A$, or for all $t' > (k+1)\tau$ there exists $t' \in ((k+1)\tau, t')$ such that $\varsigma, t' \not\models \psi_1$, so $[\varsigma]_\tau(k)(\psi) = N$.
- The other cases are similar.

For completeness, it is just a matter of exhibiting signals that have the desired property, which is simple and we do not make explicit.

A.2 Proof of Lemma 4

First, we exhibit an accepting run. We build $\nu_k(\psi)$ by induction on subformulas ψ and simultaneously prove the following properties:

$$\psi = p \in \mathrm{AP}, \text{ then for all } k \in \mathbb{Z}_{\geqslant 0},\ \nu_k(\psi) = w_k(p) \tag{9}$$

$$\psi = \psi_1 \odot \psi_2, \text{ then for all } k \in \mathbb{Z}_{\geqslant 0},\ \nu_k(\psi) \in c_\odot(\nu_k(\psi_1), \nu_k(\psi_2)) \tag{10}$$

$$\begin{aligned}&\text{for all } k \in \mathbb{Z}_{\geqslant 0}, \text{ if } \nu_k(\psi) \in \{E, Z\},\\ &\qquad \text{then there exists } p \in \mathrm{AP} \cap \mathrm{sub}(\psi) \text{ such that } \nu_k(p) = \nu_k(\psi)\end{aligned} \tag{11}$$

$$\text{for all } k \in \mathbb{Z}_{\geqslant 0},\ \nu_k(\psi) \in \{A, E\} \text{ iff } \nu_{k+1}(\psi) \in \{A, Z\} \tag{12}$$

- If $\psi = p$, we define $\nu_k(p) = w_k(p)$. Equations (9) and (11) obviously hold, Equation (10) is void, and Equation (12) holds by Equation (7).
- If $\psi = \psi_1 \wedge \psi_2$, we define $\nu_k(\psi_1 \wedge \psi_2) = c_\wedge(\nu_k(\psi_1), \nu_k(\psi_2))$, by which we mean the unique element of $c_\wedge(\nu_k(\psi_1), \nu_k(\psi_2))$. Equations (9) is void and Equation (10) holds by construction.
  We now want to show Equation (11), so we assume that $\nu_k(\psi) \in \{Z, E\}$. By scrutinizing Fig. 3, we see that if $\nu_k(\psi) \in \{Z, E\}$, then $\nu_k(\psi_1) = \nu_k(\psi)$ or $\nu_k(\psi_2) = \nu_k(\psi)$ by Lemma 3. Without loss of generality, we assume that $\nu_k(\psi_1) = \nu_k(\psi)$. By induction hypothesis, we know that there exists $p \in \mathrm{sub}(\psi_1) \subseteq \mathrm{sub}(\psi)$ such that $\nu_k(p) = \nu_k(\psi_1) = \nu_k(\psi)$ as desired.
  We now want to show Equation (12). By scrutinizing Fig. 3, we see that $\nu_k(\psi)$ is in $\{A, E\}$ iff both $\nu_k(\psi_1)$ and $\nu_k(\psi_2)$ are in $\{A, E\}$. By induction hypothesis, this is equivalent to both $\nu_{k+1}(\psi_1)$ and $\nu_{k+1}(\psi_2)$ being in $\{A, Z\}$, which is equivalent to $\nu_{k+1}(\psi)$ being in $\{A, Z\}$ again by scrutinizing Fig. 3.

- If $\psi = \psi_1 \vee \psi_2$, we define $\nu_k(\psi_1 \vee \psi_2) = \neg(c_\wedge(\neg\nu_k(\psi_1), \neg\nu_k(\psi_2)))$, where $\neg\colon \mathbb{O} \to \mathbb{O}$ is the involution such that $\neg A = N$ and $\neg Z = E$, from which all properties follow directly by the same arguments as above.
- If $\psi = \psi_1 \,\mathrm{U}\, \psi_2$, we define

$$\nu_k(\psi_1 \,\mathrm{U}\, \psi_2) = \begin{cases} A & \text{if } \nu_k(\psi_2) = A \text{ or} \\ & \quad \exists k' > k.\ \nu_{k'}(\psi_2) \neq N \text{ and } \forall k \leqslant k'' < k'.\ \nu_{k''}(\psi_1) = A \\ Z & \text{if } \nu_k(\psi_2) = Z \text{ and} \\ & \quad \forall k' > k.\ (\nu_{k'}(\psi_2) \neq N \Rightarrow \exists k \leqslant k'' < k'.\ \nu_{k''}(\psi_1) \neq A) \\ E & \text{if } (\nu_k(\psi_2) = E \text{ and } \nu_k(\psi_1) \in \{E, N\}) \text{ or} \\ & \quad (\nu_k(\psi_2) = N,\ \nu_k(\psi_1) = E, \text{ and } \exists k' > k.\ \nu_{k'}(\psi_2) \neq N \\ & \qquad \text{and } \forall k < k'' < k'.\ \nu_{k''}(\psi_1) = A) \\ N & \text{if } \forall k' \geqslant k.\ \nu_{k'}(\psi_2) \neq N \Rightarrow \exists k \leqslant k'' < k'.\ \nu_{k''}(\psi_1) \neq A. \end{cases}$$

Equation (9) is void. It is not directly obvious that $\nu_k(\psi)$ is well-defined, so we first show that it is, as well as Equation (10), by case distinction on $\nu_k(\psi_1)$ and $\nu_k(\psi_2)$.

- If $\nu_k(\psi_2) = A$, then the case for $\nu_k(\psi) = A$ holds, the cases for $\nu_k(\psi) \in \{Z, E\}$ obviously do not hold, and the case for $\nu_k(\psi) = N$ does not hold since for $k' = k$, $\nu_{k'}(\psi_2) = A \neq N$, but there is not $k < k'' \leqslant k' = k$. Therefore, $\nu_k(\psi_2) = A$ is well-defined, and Equation (10) holds by scrutinizing Fig. 3.
- If $\nu_k(\psi_1) = A$, $\nu_k(\psi_2) = Z$, then the case for $\nu_k(\psi) = A$ holds iff $\exists k' > k.\ \nu_{k'}(\psi_2) \neq N$ and $\forall k \leqslant k'' < k'.\ \nu_{k''}(\psi_1) = A$, the case for $\nu_k(\psi) = Z$ holds iff $\forall k' > k.\ (\nu_{k'}(\psi_2) \neq N \Rightarrow \exists k \leqslant k'' < k'.\ \nu_{k''}(\psi_1) \neq A)$, which is the negation of the above, so exactly one of them holds. Moreover, the case for $\nu_k(\psi) = E$ obviously does not hold, and the case for $\nu_k(\psi) = N$ does not hold (take $k' = k$). Therefore, $\nu_k(\psi) \in \{A, Z\}$ is well-defined and Equation (10) holds.
- If $\nu_k(\psi_1) = E$, $\nu_k(\psi_2) = Z$, then by induction hypothesis, Equation (11) holds for ψ_1 and ψ_2, so there is $p \in \mathrm{AP} \cap \mathrm{sub}(\psi_1)$ such that $w_k(p) = \nu_k(p) = \nu_k(\psi_1) = E$ and $q \in \mathrm{AP} \cap \mathrm{sub}(\psi_2)$ such that $w_k(q) = \nu_k(q) = \nu_k(\psi_2) = Z$, which contradicts Equation (8), so this case never happens.
- The other cases are similar to the three cases above.

To prove Equation (11), we know by Lemma 3 that $\nu_k(\psi) \in \{Z, E\}$ implies that $\nu_k(\psi) = \nu_k(\psi_1)$ or $\nu_k(\psi) = \nu_k(\psi_2)$. Without loss of generality, if we assume $\nu_k(\psi) = \nu_k(\psi_1)$, then by induction hypothesis there exists $p \in \mathrm{AP} \cap \mathrm{sub}(\psi_1) \subseteq \mathrm{AP} \cap \mathrm{sub}(\psi)$ such that $\nu_k(p) = \nu_k(\psi_1) = \nu_k(\psi)$.

Now, we want to show that Equation (12) holds. We can show that

$$\begin{aligned} \nu_k(\psi) \in \{A, E\} \iff{} & \nu_k(\psi_2) \in \{A, E\} \vee \\ & (\nu_k(\psi_1) \in \{A, E\} \wedge \\ & \quad \exists k' > k.(\nu_{k'}(\psi_2) \neq N \wedge \forall k \leqslant k'' < k'.\ \nu_{k''}(\psi_1) = A)), \\ \nu_k(\psi) \in \{A, Z\} \iff{} & \nu_k(\psi_2) \in \{A, Z\} \vee \\ & \exists k' > k.(\nu_{k'}(\psi_2) \neq N \wedge \forall k \leqslant k'' < k.\ \nu_{k''}(\psi_1) = A). \end{aligned}$$

Therefore,

$$\begin{aligned}\nu_{k+1}(\psi) \in \{A, Z\} \iff\; & \nu_{k+1}(\psi_2) \in \{A, Z\} \vee \\ & \exists k' > k+1.(\nu_{k'}(\psi_2) \neq N \wedge \\ & \qquad\qquad \forall k+1 \leqslant k'' < k.\ \nu_{k''}(\psi_1) = A) \\ \iff\; & \nu_k(\psi_2) \in \{A, E\} \vee \\ & \exists k' > k+1.(\nu_{k'}(\psi_2) \neq N \wedge \\ & \qquad\qquad \forall k+1 \leqslant k'' < k.\ \nu_{k''}(\psi_1) = A)\end{aligned}$$

It is thus obvious that $\nu_k(\psi) \in \{A, E\}$ implies that $\nu_{k+1}(\psi) \in \{A, Z\}$. Moreover, if there exists $k' > k+1$ such that $\nu_{k'}(\psi_2) \neq N$ and for all $k+1 \leqslant k'' < k$, $\nu_{k''}(\psi_1) = A$, then $\nu_k(\psi_1) \in \{A, E\}$ by (12) on ψ_1, so $\nu_{k+1}(\psi) \in \{A, Z\}$ implies $\nu_k(\psi) \in \{A, E\}$.

- If $\psi = \psi_1 \mathrm{R}\, \psi_2$, we define $\nu_k(\psi_1 \vee \psi_2) = \neg(c_{\mathrm{U}}(\neg\nu_k(\psi_1), \neg\nu_k(\psi_2)))$, from which all properties follow directly by the same arguments as above.

By Equations (10) and (12), $\nu_0\nu_1\ldots$ is a run of $\mathcal{G}_\varphi$, and $\nu_k(p) = w_k(p)$ for all $k \in \mathbb{Z}_{\geqslant 0}$ and $p \in \mathrm{AP}$ by Equation (9).

We now prove that $\nu_0\nu_1\ldots$ is accepting, i.e., that for each $\psi = \psi_1 \mathrm{U}\, \psi_2 \in \mathrm{sub}(\varphi)$, it visits F_ψ infinitely often (and similarly for $\psi_1 \mathrm{R}\, \psi_2$). We only prove the case $\psi_1 \mathrm{U}\, \psi_2$, the other case is symmetric. By contradiction, assume that $\nu_0\nu_1\ldots$ stops visiting F_ψ after index k_0, then for all $k > k_0$, $\nu_k(\psi) = A$ and $\nu_k(\psi_2) = N$. But $\nu_k(\psi) = A$ and $\nu_k(\psi_2) = N$ iff $\nu_k(\psi_2) = N$ and $\exists k' > k.(\nu_{k'}(\psi_2) \neq N \wedge \forall k \leqslant k'' < k'.\ \nu_k(\psi_1) = A)$, so in particular $\nu_k(\psi_2) = N$ for all $k > k_0$ and $\exists k' > k_0 + 1.\nu_{k'}(\psi_2) \neq N$, hence a contradiction, as desired.

Finally, we prove that, if $\nu'_0\nu'_1\ldots$ is an accepting run such that for all $k \in \mathbb{Z}_{\geqslant 0}$ and $p \in \mathrm{AP}$, $\nu'_k(p) = w_k(p)$, then $\nu'_k = \nu_k$. We prove by induction on $\psi \in \mathrm{sub}(\varphi)$ that for all $k \in \mathbb{Z}_{\geqslant 0}$, $\nu'_k(\psi) = \nu_k(\psi)$.

- If $\psi = p$, then the result is obvious.
- If $\psi = \psi_1 \wedge \psi_2$ or $\psi = \psi_1 \vee \psi_2$, then the result holds by induction hypothesis on ψ_1 and ψ_2, using the fact that $c_\wedge(o_1, o_2)$ and $c_\vee(o_1, o_2)$ contain a unique element.
- If $\psi = \psi_1 \mathrm{U}\, \psi_2$, then by induction hypothesis $\nu'_k(\psi_1) = \nu_k(\psi_1)$ and $\nu'_k(\psi_2) = \nu_k(\psi_2)$, then we proceed by case distinction on $\nu_k(\psi_1)$ and $\nu_k(\psi_2)$. In most cases, $c_{\mathrm{U}}(\nu_k(\psi_1), \nu_k(\psi_2))$ contains only one element, so the result holds directly, so we only detail the other cases.
  - If $\nu_k(\psi_1) = A$, $\nu_k(\psi_2) = Z$, and $\nu_k(\psi) = A$, then because $\nu_k(\psi) = A$, we know that either $\nu_k(\psi_2) = A$ (which is not true) or there exists $k' > k$ such that $\nu_{k'}(\psi_2) \neq N$ and for all $k \leqslant k'' < k'$, $\nu_{k''}(\psi_1) = A$. Let k' be the smallest such index, then for all $k < k'' < k'$, $\nu_{k''}(\psi_2) = N$, and $\nu_{k'}(\psi_2) \in \{E, N\}$ by (12), so $\nu_{k'}(\psi_2) = E$. Therefore, we have for all $k < k'' < k'$, $\nu'_{k''}(\psi_1) = A$ and $\nu'_{k''}(\psi_2) = N$, so $\nu'_{k''}(\psi) \in \{A, N\}$. We also have $\nu'_{k'}(\psi_1) \in \{A, Z\}$ by (12) and $\nu'_{k'}(\psi_2) = E$, so $\nu'_{k'}(\psi_1) = A$ by (8). Therefore, $\nu'_{k'}(\psi) = A$, whence $\nu'_{k''}(\psi) = A$ for all $k < k'' < k'$ (by induction on $k' - k''$). Finally, because $\nu'_k(\psi) \in \{A, Z\}$, $\nu'_k(\psi) = A$ by (12).

- If $\nu_k(\psi_1) = A$, $\nu_k(\psi_2) = Z$, and $\nu_k(\psi) = Z$, then because $\nu_k(\psi) = Z$, we know that $\nu_k(\psi_2) = Z$ and for all $k' > k$, if $\nu_{k'}(\psi_2) \neq N$, then there exists $k \leqslant k'' < k'$ such that $\nu_{k''}(\psi_1) \neq A$.
  - $*$ If for all $k' > k$, $\nu_{k'}(\psi_2) = N$, and we assume that $\nu'_k(\psi) \neq Z$, then $\nu'_k(\psi) = A$, then because $\nu'_{k'}(\psi_1) = N$, we have $\nu'_{k'}(\psi) \in \{A, E, N\}$ according to Lemma 3. By induction and (12), we have that $\nu'_{k'}(\psi) = A$. Therefore, we have for all $k' > k$ that $\nu'_{k'}(\psi) = A$ and $\nu'_{k'}(\psi_2) = N$, which contradicts the fact that $\nu'_0\nu'_1 \ldots$ is accepting.
  - $*$ Otherwise, let $k' > k$ be the minimal index such that $\nu_{k'}(\psi_2) \neq N$, then we have that for all $k < k'' < k'$, $\nu_{k''}(\psi_2) = N$, and $\nu_{k'}(\psi_2) \in \{E, N\}$ by (12), hence $\nu_{k'}(\psi_2) = E$. Let $k \leqslant k'' < k'$ be the minimal index such that $\nu_{k''}(\psi_1) \neq A$, then for all $k \leqslant k''' < k''$, $\nu_{k'''}(\psi_1) = A$, and $\nu_{k''}(\psi_1) \in \{A, Z\}$ by (12), hence $\nu_{k''}(\psi_1) = Z$. Now, for all $k < k''' < k''$, $\nu'_{k'''}(\psi_1) = A$ and $\nu'_{k'''}(\psi_2) = N$, hence $\nu'_{k'''}(\psi) \in \{A, N\}$ by Lemma 3. If we assume that $\nu'_k(\psi) = A$, then $\nu'_{k'''}(\psi) = A$ by induction using (12). Moreover, $\nu'_{k''}(\psi_1) = Z$ and $\nu'_{k''}(\psi_2) = N$, hence $\nu'_{k''}(\psi) = N$, which contradicts $\nu'_{k''-1} = A$.
- If $\nu_k(\psi_1) = A$, $\nu_k(\psi_2) = N$, and $\nu_k(\psi) = A$, then because $\nu_k(\psi) = A$, we know that either $\nu_k(\psi_2) = A$ (which is not true) or there exists $k' > k$ such that $\nu_{k'}(\psi_2) \neq N$ and for all $k \leqslant k'' < k'$, $\nu_{k''}(\psi_1) = A$. Let k' be the minimal such index, then for all $k < k'' < k'$, $\nu_{k''}(\psi_2) = N$, and $\nu_{k'}(\psi_2) \in \{E, N\}$ by (12), hence $\nu_{k'}(\psi_2) = E$. Moreover, $\nu_{k'}(\psi_1) \in \{A, Z\}$ by Lemma 3, hence $\nu_{k'}(\psi_1) = A$ by (8). We thus have that, $\nu'_{k'}(\psi_1) = A$, $\nu'_{k'}(\psi_2) = E$, so $\nu'_{k'}(\psi) = A$ by Lemma 3, and for all $k < k'' < k'$, $\nu'_{k''}(\psi_1) = A$, $\nu'_{k''}(\psi_2) = N$, so $\nu'_{k''}(\psi) \in \{A, N\}$, thus $\nu'_{k''}(\psi) = A$ by induction on $k' - k''$ using (12). By Lemma 3, $\nu'_k(\psi) \in \{A, N\}$, hence $\nu'_k(\psi) = A$ by (12).
- If $\nu_k(\psi_1) = A$, $\nu_k(\psi_2) = N$, and $\nu_k(\psi) = N$, then because $\nu_k(\psi) = N$, we know that for all $k' > k$, if $\nu_{k'}(\psi_2) \neq N$, then there exists $k \leqslant k'' < k'$ such that $\nu_{k''}(\psi_1) \neq A$.
  - $*$ If for all $k' > k$, $\nu_{k'}(\psi_2) = N$, then for all $k' > k$, $\nu'_{k'}(\psi) \neq Z$ by Lemma 3. If $\nu_k(\psi) = A$, then for all $k' > k$, $\nu'_{k'}(\psi) = A$ by induction on k' using (12), therefore $\nu'_0\nu'_1 \ldots$ is not accepting.
  - $*$ Otherwise, let $k' > k$ be the minimal index such that $\nu_{k'}(\psi_2) \neq N$, then for all $k \leqslant k'' < k'$, $\nu_{k''}(\psi_2) = N$, and $\nu_{k'}(\psi_2) \in \{E, N\}$ by (12), hence $\nu_{k'}(\psi_2) = E$. Moreover, let $k \leqslant k'' < k'$ by the minimal index such that $\nu_{k''}(\psi_1) \neq A$, then for all $k \leqslant k''' < k''$, $\nu_{k'''}(\psi_1) = A$, and $\nu_{k''}(\psi_1) \in \{A, Z\}$ by (12), hence $\nu_{k''}(\psi_1) = Z$. Therefore, $\nu'_{k''}(\psi_1) = Z$ and $\nu'_{k''}(\psi_2) = N$, so $\nu'_{k''}(\psi) = N$ by Lemma 3. For all $k \leqslant k''' < k''$, $\nu'_{k'''}(\psi_1) = A$ and $\nu'_{k'''}(\psi_2) = N$, so $\nu'_{k'''}(\psi) \in \{A, N\}$, and therefore $\nu'_{k'''}(\psi) = N$ by induction on $k'' - k'''$, hence $\nu'_k(\psi) = N$.

A.3 Proof of Corollary 2

It is enough to show that $[\varsigma]_\tau$ satisfies the definitions in the proof of Lemma 4, which we show for all $k \in \mathbb{Z}_{\geqslant 0}$ and $\psi \in \mathrm{sub}(\varphi)$ by induction on ψ.

- If $\psi = p$, then the result holds directly.
- If $\psi = \psi_1 \wedge \psi_2$, then

$$\begin{aligned}
[\varsigma]_\tau(k)(\psi) = A &\iff \forall t \in [k\tau, (k+1)\tau].\, \varsigma, t \vDash \psi \\
&\iff \forall t \in [k\tau, (k+1)\tau].\, \varsigma, t \vDash \psi_1 \wedge \psi_2 \\
&\iff \forall t \in [k\tau, (k+1)\tau].\, \varsigma, t \vDash \psi_1 \text{ and } t \vDash \psi_2 \\
&\iff \forall t \in [k\tau, (k+1)\tau].\, \varsigma, t \vDash \psi_1 \text{ and} \\
&\qquad \forall t \in [k\tau, (k+1)\tau].\, \varsigma, t \vDash \psi_2 \\
&\iff [\varsigma]_\tau(k)(\psi_1) = A \text{ and } [\varsigma]_\tau(k)(\psi_2) = A
\end{aligned}$$

  as desired. The other cases are similar.
- If $\psi = \psi_1 \,\mathrm{U}\, \psi_2$, then

$$\begin{aligned}
[\varsigma]_\tau(k)(\psi) = A &\iff \forall t \in [k\tau, (k+1)\tau].\, \varsigma, t \vDash \psi \\
&\iff \forall t \in [k\tau, (k+1)\tau].\, \varsigma, t \vDash \psi_1 \,\mathrm{U}\, \psi_2 \\
&\iff \forall t \in [k\tau, (k+1)\tau]. \\
&\qquad \exists t' \geqslant t.\, (\varsigma, t' \vDash \psi_2 \wedge \forall t'' \in [t, t').\, \varsigma, t'' \vDash \psi_1). \qquad (13)
\end{aligned}$$

  On the other hand

$$\begin{aligned}
&[\varsigma]_\tau(k)(\psi_2) = A \text{ or } \exists k' > k.([\varsigma]_\tau(k')(\psi_2) \neq N \text{ and } \forall k \leqslant k'' < k'.\, [\varsigma]_\tau(k'') = A) \\
&\iff \forall t \in [k\tau, (k+1)\tau].\, \varsigma, t \vDash \psi_2 \text{ or} \\
&\qquad \exists k' > k.(\exists t' \in [k'\tau, (k'+1)\tau].\, \varsigma, t' \vDash \psi_2 \text{ and} \\
&\qquad\qquad \forall k \leqslant k'' < k'.\, \forall t'' \in [k''\tau, (k''+1)\tau].\, \varsigma, t'' \vDash \psi_1). \qquad (14)
\end{aligned}$$

  We want to show that (13) and (14) are equivalent. We distinguish four cases:
  - If $\varsigma, k\tau \vDash \psi_2$ and $\varsigma, (k+1)\tau \vDash \psi_2$, then by Assumption 1.(1), for all $t \in [k\tau, (k+1)\tau]$, $\varsigma, t \vDash \psi_2$, so both (13) and (14) hold.
  - If $\varsigma, k\tau \vDash \psi_2$ and $\varsigma, (k+1)\tau \nvDash \psi_2$, then by Assumption 1.(2), there exists $t \in (k\tau, (k+1)\tau)$ such that $\varsigma, t' \vDash \psi_2$ for all $t' \in [k\tau, t)$ and $\varsigma, t' \nvDash \psi_2$ for all $t' \in (t, (k+1)\tau]$. If (13) holds, then there exists $t' > (k+1)\tau$ such that $\varsigma, t' \vDash \psi_2$ and $\varsigma, t'' \vDash \psi_1$ for all $t'' \in [t, t')$. Therefore so does (14) by taking $k' = \lceil t/\tau \rceil$ and $t' = k'\tau$.
    Conversely, if (14) holds, then there exists $k' > k$ and $t' \in [k'\tau, (k'+1)\tau]$, $\varsigma, t' \vDash \psi_2$, and for all $k \leqslant k'' < k'$ and $t'' \in [k''\tau, (k''+1)\tau]$, $\varsigma, t'' \vDash \psi_1$. In particular, $\varsigma, ((k'-1)+1)\tau \vDash \psi_1$, so $[\varsigma]_\tau(k')(\psi_1) \in \{A, Z\}$. If $[\varsigma]_\tau(k')(\psi_1) = A$, then (13) also directly holds. If $[\varsigma]_\tau(k')(\psi_1) = Z$, then by Assumption 1.(2), because $\varsigma, t' \vDash \psi_2$, $\varsigma, t'' \vDash \psi_2$ for all $t'' \in [k'\tau, t']$ (because $[\varsigma]_\tau(k')(\psi_2) \neq E$ and formulas can only change values at one time during a time interval of length τ). Therefore, (13) holds.
  - The other cases are similar to one of the two cases above.

  The other cases are similar.
- If $\psi = \psi_1 \vee \psi_2$ or $\psi = \psi_1 \,\mathrm{R}\, \psi_2$, the result follows using the same arguments as above.