A Temporal Logic for Micro- and Macro-step-based Real-time Systems: Foundations and Applications

Matteo Rossi* and Dino Mandrioli* and Angelo Morzenti* and Luca Ferrucci†

*Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci, 32, 20133, Milano, Italy
e-mail: {firstname.lastname}@polimi.it

†ISTI-CNR, Pisa, Italy

Abstract

Many systems include components interacting with each other that evolve with possibly very different speeds. To deal with this situation many formal models adopt the abstraction of "zero-time transitions", which do not consume time. These, however, have several drawbacks in terms of naturalness and logic consistency, as a system is modeled to be in different states at the same time. We propose a novel approach that exploits concepts from non-standard analysis and pairs them with the traditional "next" operator of temporal logic to introduce a notion of micro- and macro-steps in an extension of the TRIO metric temporal logic, called X-TRIO. We study the expressiveness and decidability properties of the new logic; decidability is achieved through translation of a meaningful subset of X-TRIO into Linear Temporal Logic, which is a traditional means to support automatic verification. We illustrate the usefulness and the generality of our approach by applying it to provide a formal semantics of timed Petri nets which allows for their automated verification; we also give an overview of a formal semantics of Stateflow/Simulink diagrams that has been defined in terms of X-TRIO and used for automatic verification.

Keywords: metric temporal logic, formal and automatic verification, micro- and macro-steps, non-standard analysis, Petri nets, Stateflow/Simulink.

Contents

☆Work supported by the European Commission, Programme IDEAS-ERC, Project 227977-SMScom. This work contains material previously published in [1] and [2]. This work was done while Luca Ferrucci was at Politecnico di Milano.

1. Introduction

Modern, complex time-critical systems often require sophisticated time modeling approaches to support specification and verification of their properties. Traditional approaches to modeling time-dependent system behavior are roughly categorized into continuous and discrete ones: in the former case both system state and the time variable are ranging over a continuous domain, e.g., the reals, and system evolution is formalized as the state being a continuous function of the independent variable "time"; in the latter case both time and the system state range over a discrete domain, typically, the integers. Computing devices, which in most cases are synchronized by a clock, are traditionally formalized by such discrete models as automata of some type.

Such a traditional approach has proven incomplete, or at least partial, for most modern applications which involve complex systems with components whose time behavior and modeling needs are quite heterogeneous: think, e.g., of a continuous industrial process monitored and controlled by computing devices. In such systems, some components evolve through discrete steps, but their pace is determined by the environment in which they are embedded, which produces asynchronous stimuli to be reacted on with severe time constraints. As a consequence, such discrete steps often exhibit durations that may differ even by orders of magnitude – contrast, e.g., the switching of a transistor to a fire alarm and to the reaction of the automatic fire sprinklers.

To deal with such sharply different types of timing features in system components, many models studied in the literature offer the use of micro- and macro-steps, of which, normally, only the latter ones "consume time": micro-step

durations, being negligible w.r.t macro ones, are roughly assimilated to zero-time. Notions of zero-time transitions appear very naturally when reasoning about computations of embedded systems (see, e.g., [3], Chapter 6), and more generally of cyber-physical systems: they are also a natural mathematical tool to formalize the risk of Zeno behaviors, i.e., an accumulation of an infinite sequence of events with no "sensible time advancement". For instance, in [4], while developing and analyzing a formal model of an aerospace satellite system, the authors repeatedly emphasize the need for "algorithmic detection of Zeno behavior". Using zero-time transitions simplifies models and their analysis, but it is not without drawbacks, essentially due to the fact that the system can be in different, maybe even infinitely many, states, at the same time, with an obvious risk of engendering contradictions. We refer the reader to [5] for a more complete view of the time modeling issue in the literature.

An orthogonal but equally important issue in time modeling is model analyzability: not only do we need models that adequately represent system behavior, but they must also support mechanisms to effectively analyze – in a precise and possibly automated fashion – their properties. In fact, the fairly recent success of model-checking-based techniques has spurred a great interest in "push-button" analysis tools, which, by definition, are based on decidable formalisms. A flurry of (temporal) logic-based models has been developed in the last decades to reach a best compromise between expressiveness and naturalness on the one side, and decidability and complexity of the analysis on the other side [6, 7].

In this paper we propose a novel approach to **deal in a joint way with the three issues** mentioned above: **modeling systems that evolve stepwise** without a synchronizing clock; allowing for **steps whose durations may differ by orders of magnitude**, yet avoiding the logical trap and counterintuitive semantics of zero-time approximation; **supporting automatic analysis**, possibly by suitably restricting an original general but undecidable formalism. We will anchor our approach to our own temporal logic language TRIO [8], but we emphasize that it can be applied to other temporal logics such as, for example, MTL [9].

In a nutshell, we first extend the original TRIO language in two major ways to make it suitable to model systems with the above features. On the one hand, we introduce a "next-step" operator imported from classic temporal logic; since we use the traditional textual symbol X for this operator, we name our augmented language X-TRIO. On the other hand, we borrow from Nonstandard Analysis (NSA) [10] the concept of infinitesimal number to formalize the non-null but negligible duration of micro steps, as opposed to that of macro steps which is represented by standard numbers. In its full generality TRIO includes full arithmetic and is, therefore, undecidable; thus, we look for suitable restrictions that make it decidable, but still general enough to formalize and analyze the main properties of various systems of industrial relevance. Among the many possible ones, the approach used in this paper is based on a syntax inspired by decidable versions of Metric Temporal Logic (MTL) [9] and a "fine tuning" of the temporal domain over which X-TRIO formulae are interpreted. The decidability of the chosen X-TRIO subset is then demonstrated through translation

into Linear Temporal Logic, which enables the use of any LTL-based satisfiability solver such as, for example, Zot [11], to analyze its formulae. Even if we devote a necessary technical effort to come up with a decidable subset of X-TRIO suitable for a typical "push-button" verification, we also emphasize the applicability of X-TRIO to the modeling and –possibly semiautomatic– analysis of complex systems that require more expressive languages. A few examples in Sections 3 and 6 will be devoted to clarify the use of the various versions of X-TRIO.

This paper is structured as follows. Section 2 puts our work in the context of the existing literature, with particular attention to the formalisms based on timed words and 0-time transitions. Section 3 presents the general syntax and semantics of the X-TRIO language, with some examples of use, and Section 4 analyzes some properties of a propositional version thereof, which is shown to be undecidable. Section 5 introduces a further restricted fragment of X-TRIO to achieve decidability with a reasonable complexity. Section 6 presents two case studies of application of X-TRIO to formalize the semantics of two classical operational formalisms, namely timed Petri nets and Simulink/Stateflow diagrams, and to prove their properties, whether in fully automatic way or by means of deductive techniques based on a suitable axiomatization. Section 7 concludes and hints at future developments.

This work includes and extends material previously published in [1] and [2]. Precisely, [1] introduced a first propositional version of X-TRIO which is extended here to a more general version, and whose syntax, semantics and decidability issues have been investigated in more detail in this paper; [2] presents the complete case study on the formalization of Simulink/Stateflow diagrams which is summarized in Section 6.2.

2. Related Work

Notions of zero-time transitions, micro- and macro-steps appear very naturally when reasoning about computations of embedded systems, thus it is not surprising to find such concepts in the literature on real-time temporal logics. Since the very early developments in this field, approaches were introduced that admit zero-time transitions at the price of associating multiple states to single time instants [12]. Our approach is akin to that of [9], which introduces a general framework accommodating suitable time structures supporting the notion of micro- and macro-steps, focusing on naturalness and readability of the notation and without initially addressing issues of decidability and verification. In Metric Temporal Logic [6] a "metric next-time" operator is introduced to deal with discrete structures, so that formulae are interpreted over *timed words*, where each event is bound to a corresponding *timestamp* [13]. Timed words are *weakly monotonic* when it is possible that several, logically ordered events are associated with the same timestamp.

As already observed in previous literature [14, 5], logics adopting the weak monotonicity assumption are strongly connected to an operational formalism (such as, for example, Timed Automata [15, 16]) which, in turn, represents

the evolution of time by means of stamps attached to transitions, states, or enabling conditions. In a sense time becomes "yet another system variable" separated from the "physical time" which progresses continuously and independently, whereas in timed automata and metric temporal logics allowing for 0-time transitions time is reduced to a discrete, linear, totally but weakly ordered, sequence of events where no information is included in the time structure about the state of the modeled system outside such a sequence. The following examples illustrate a few inconveniences deriving from this modeling philosophy and hint at how our approach will avoid them.

As a first case, consider the timed automaton fragment of Figure 1 (see [16, 5] for an introduction to the formalism of Timed Automata), which accepts

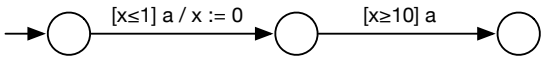


Figure 1: First example of Timed Automaton.

sequences such as $\ldots(a,1)(a,10)\ldots$ Suitable MTL formulae consistent with the automaton behavior may be written; no real "meaning", however, could be attached to facts predicated on times that are not "stamped" by the automaton. For instance, both formulae $\mathrm{G}_{[2,5]}\bot$ and $\mathrm{G}_{[2,5]}\top$ would be true in the origin. On the contrary, our semantics allows us to explicitly formalize, and to prove, that "nothing changes in the system state between two consecutive steps", so that system behavior is completely and coherently specified by logic formulae throughout the time domain.

The second example emphasizes that 0-time transitions, by permitting a system to be in different states at the same time, allow for counterintuitive and apparently contradictory logic expressions about system behavior. For instance, the timed automaton fragment of Figure 2 accepts the sequence $(a,1)(a,4)(\emptyset,4)\ldots$. Such sequence satisfies, at time 0, formula $\mathrm{F}_{[3,3]}(a) \wedge \mathrm{F}_{[3,3]}(\neg a)$. On the contrary, we will show that our formalism allows one to state that a and $\neg a$ hold "infinitely close" to each other, but not simultaneously.

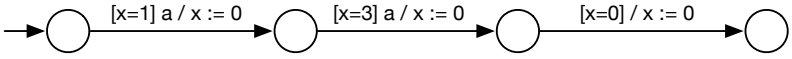


Figure 2: Second Example of Timed Automaton.

Things become more intricate when we want to formally describe and analyze system properties possibly involving some "pathological behavior". This is illustrated by our third example.

Suppose you want to express a property of "liveness at 0-time", i.e., "at the current time a given property a will occur infinitely often in the "future", but still at distance 0 from the current time". Such a property is properly captured by the MTL formula $\mathrm{G}_{[0,0]}\mathrm{X}_{[0,0]}\mathrm{F}_{[0,0]}(a)$ which imposes that after any

step occurring at time 0 there exists yet another step, again at time 0, when a holds; but this necessarily implies that the whole system is stuck in an infinite sequence of 0-time steps, i.e. a Zeno behavior where time cannot advance. On the contrary our model, which allows users to predicate system properties even out of the stamped instants, allows us also to state that a property holds in an (infinitesimal) future instant even when such an instant does not correspond to any system transition. We will see an example of such an "infinitesimal distance liveness" property, that does not necessarily incur Zeno behaviors, in Section 5.1.

The weakly monotonic semantics, normally associated with various forms of timed automata and/or temporal logics is certainly the most widely known and adopted approach to deal with micro- and macro-steps in system evolution. There are however other significant approaches.

The approach based on the concept of *super dense time* (SDT [17, 3]) exploits a temporal domain that is the product $\mathbb{R} \times \mathbb{N}$ (whose subset $\mathbb{N} \times \mathbb{N}$ is also of great interest); similar domains are also briefly treated in [9]. Roughly, micro-steps are represented along one dimension and macro ones along the other one. Such a concept, however, again departs sharply from the normal intuition of a time domain as a linear totally ordered one (which is the approach adopted in X-TRIO); for instance in such models we could have an *infinite* sequence of micro-steps (a Zeno behavior) *followed* by a macro one, a sort of "return from infinity". Furthermore, frameworks using SDT mostly focus on issues concerning the simulation of hybrid systems. In this paper we go further and develop a logic language and an associated decision procedure that allow us to formally model and verify systems whose underlying notion of time captures the same properties as SDT without the above inconveniences. [18] presents a variant of LTL interpreted over $\mathbb{N} \times \mathbb{N}$ that is shown to be decidable. This logic has some similarities with the variant of X-TRIO studied in Section 5 and originally introduced in [1].

Finally, the proposal in [19] provides notations for modeling micro-steps in the framework of Duration Calculus, which, unlike TRIO and other classical temporal logics which are based on time *points*, is based on time *intervals*: it defines a decidable fragment of the notation, but it does not provide algorithms or build tools supporting verification.

The distinguishing feature of exploiting NSA infinitesimals to formalize zero-time transitions in timed Petri nets was originally formulated in [20]: this is the clue to describe the effect of a micro-step "without stopping the time". Other works [21, 22] have used NSA to provide a formal and rigorous semantics to timing features of various kinds of notations for system modeling. In [21] NSA is used to describe a hybrid system modeled in Simulink, in presence of cascaded mode changes. In [22], a complete system theory is defined, adopting a theoretical approach to investigate computability issues.

Other works are only partially connected to ours, as they deal with issues concerning the modeling and development of embedded systems at various time scales: [23] and [24] deal with issues of sampling and digitization, [25], [26] and [27] discuss issues related with time granularity, and [28] provides a refinement

method based on assume-guarantee induction over different time scales. None of these papers, however, address the problem of modeling zero-time transitions, and of checking for the absence of Zeno behaviors. When different temporal granularities are present in the same system (e.g., a coarser one, with constants in the order of the seconds, and a finer one in the order of the milliseconds), one can model the system evolution in the faster dynamics as taking zero time with respect to the slower dynamics; still, transitions in the finer temporal model take finite, non-null time, so an infinite accumulation thereof is not allowed, hence ruling out Zeno behaviors *a priori*.

In summary, while features such as zero-time transitions, simultaneous events, co-existence of multiple states at the same time have been studied in the literature, we are not aware of approaches that allow users to capture and predicate over these issues in an abstract, uniform way, and at the same time provide mechanisms and tools for carrying out automated formal verification of these properties.

3. The General X-TRIO logic

In this section, we introduce X-TRIO in its full generality. After a short informal summary of the original TRIO language, we augment it through a "next step" operator that allows the user to model systems that evolve through a sequence of discrete steps even in an asynchronous way w.r.t. the environment; also, to capture at the semantic level the notion of steps that occur in sharply different time scales (micro and macro-steps) we augment the time domain over which X-TRIO formulae are interpreted with the notion of non-standard numbers. We conclude this section with a few preliminary examples of application of the logic.

3.1. Summary of the TRIO language

The original TRIO language [29][8][30] is a general-purpose specification language suitable for modeling real-time systems. It is a temporal logic supporting a metric on time. TRIO formulae are built out of the usual first-order connectives, operators, and quantifiers, and the single basic modal operator, Dist: for any formula ϕ and term t indicating a time distance, the formula $\mathrm{Dist}(\phi, t)$ specifies that ϕ holds at a time instant whose distance is exactly t time units from the current instant. TRIO formulae can be interpreted both in discrete and dense time domains.

TRIO allows the user to define a large set of derived operators to make formulae simpler and more intuitive. For instance, $\mathrm{Futr}(\phi, t)$ is equivalent to $t \geq 0 \wedge \mathrm{Dist}(\phi, t)$, while $\mathrm{Past}(\phi, t)$ is equivalent to $t \geq 0 \wedge \mathrm{Dist}(\phi, -t)$. Table 1 presents a meaningful sample of TRIO derived operators, where we use subscript [] to indicate that both endpoints of the interval are included; similarly for the other combinations.

In its full generality, TRIO allows users to write arithmetic formulae, hence it is trivially undecidable.

OPERATOR	DEFINITION
$\mathrm{Futr}(\phi, d)$	$d \geq 0 \wedge \mathrm{Dist}(\phi, d)$
$\mathrm{Past}(\phi, d)$	$d \geq 0 \wedge \mathrm{Dist}(\phi, -d)$
$\mathrm{AlwF}(\phi)$	$\forall d(0 \leq d \rightarrow \mathrm{Futr}(\phi, d))$
$\mathrm{SomF}(\phi)$	$\exists d(0 \leq d \wedge \mathrm{Futr}(\phi, d))$
$\mathrm{WithinF}_{[]}(\phi, \delta)$	$\exists d(0 \leq d \leq \delta \wedge \mathrm{Futr}(\phi, d))$
$\mathrm{WithinF}_{()}(\phi, \delta)$	$\exists d(0 < d < \delta \wedge \mathrm{Futr}(\phi, d))$
$\mathrm{WithinP}_{[)}(\phi, \delta)$	$\exists d(0 \leq d < \delta \wedge \mathrm{Past}(\phi, d))$
$\mathrm{Lasts}_{()}(\phi, \delta)$	$\forall d(0 < d < \delta \rightarrow \mathrm{Futr}(\phi, d))$
$\mathrm{Lasts}_{[)}(\phi, \delta)$	$\forall d(0 \leq d < \delta \rightarrow \mathrm{Futr}(\phi, d))$
$\mathrm{Lasted}_{[]}(\phi, \delta)$	$\forall d(0 \leq d \leq \delta \rightarrow \mathrm{Past}(\phi, d))$
$\mathrm{Lasted}_{[)}(\phi, \delta)$	$\forall d(0 \leq d < \delta \rightarrow \mathrm{Past}(\phi, d))$
$\mathrm{Until}(\phi, \psi)$	$\exists d(\mathrm{Futr}(\psi, d) \wedge \forall v(0 \leq v < d \rightarrow \mathrm{Futr}(\phi, v)))$
$\mathrm{Since}(\phi, \psi)$	$\exists d(\mathrm{Past}(\psi, d) \wedge \forall v(0 \leq v < d \rightarrow \mathrm{Past}(\phi, v)))$

Table 1: A sample of TRIO derived temporal operators.

3.2. X-TRIO syntax and semantics

The original TRIO language is well suited to deal with both continuous systems, that evolve in a continuous time domain, and with discrete systems, where each step takes exactly one time unit. Moreover, it can deal with heterogeneous systems that combine both continuous and discrete components through suitable approximations [24]. More complex systems, however, evolve through discrete steps, whether in continuous or discrete time and state domains. Furthermore, different steps may require time durations that differ even by orders of magnitude from each other.

To accurately and naturally model such systems we enrich TRIO by a "next" operator as in classical temporal logic; however, whereas normally in metric frameworks this operator is associated with exactly one time unit (in TRIO terms this would mean $\mathrm{X}(\phi) \equiv \mathrm{Dist}(\phi, 1)$), here we do not associate a fixed time duration to the execution of a step. Instead, we distinguish between micro-steps that occur with a negligible but non-null time duration, and macro-steps, that take a finite but in general not *a priori* fixed time. Thus $\mathrm{X}(\phi)$ states that at the next step property ϕ holds, and this occurs after a time from the current instant that is at a distance from the current time that is either finite or infinitesimal. The "yesterday" operator $\mathrm{Y}(\phi)$ is defined symmetrically. We also introduce predicate $\mathrm{st}(d)$, which will be explained later, through which users can distinguish between infinitesimal and non-infinitesimal temporal distances. Derived temporal logic operators can be defined as in the original TRIO.

Typical examples of behaviors exhibiting micro- and macro-steps are the execution of a computer statement, say a sum, where a sequence of logic switches implements the micro-operations of a full adder and at the end of the clock period the result is made visible to the target register; or an avionic control system which within a given period reads a set of input data from a collection

of sensors, computes the corresponding output values, and at the end of the period supplies such values to the actuators.

We formalize the notion of infinitesimal time duration by exploiting the theory of non-standard analysis [10]. Let T be any temporal domain that is a subset of the reals. Intuitively, a number ϵ is infinitesimal if $\epsilon \geq 0$ and ϵ is smaller than any number in T that is greater than 0. The original values of T are classified as "standard" and are characterized by predicate st, that is, x is standard iff $\mathrm{st}(x)$ holds. T is then augmented with infinitesimal numbers and all numbers resulting from adding and subtracting infinitesimal non-zero numbers to and from standard ones. Predicate $\mathrm{ns}(x)$ denotes that x is non-standard. For each x, $\mathrm{ns}(x)$ holds if and only if $\mathrm{st}(x)$ does not hold, i.e., we have $\mathrm{ns}(x) \leftrightarrow \neg\mathrm{st}(x)$. We will also use predicate $\mathrm{inf}(\epsilon)$, which indicates that ϵ is infinitesimal, as an abbreviation for $\mathrm{ns}(\epsilon) \wedge \epsilon \geq 0 \wedge \forall x.(x > 0 \wedge \mathrm{st}(x) \rightarrow \epsilon < x)$. Notice that 0 is the only infinitesimal standard number and that non-standard numbers – which include, but are not limited to, the infinitesimals greater than 0 – are of the form $v \pm \epsilon$, where $\mathrm{st}(v)$ holds, and ϵ is infinitesimally greater than 0. The syntax of X-TRIO is the following (where v, k, f and p denote, respectively, a variable, a constant, a function, and a predicate; functions and predicates can have arity 0):

$$\begin{array}{lcl} \phi & := & p(\tau_1, \ldots, \tau_n) \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \mathrm{Dist}(\phi, \tau) \mid \forall v.\phi \mid \mathrm{X}(\phi) \mid \mathrm{Y}(\phi) \mid \mathrm{st}(\tau) \\ \tau & := & v \mid k \mid f(\tau_1, \ldots, \tau_n) \end{array} \quad (1)$$

Derived temporal operators can be defined as in Table 1.

NSA provides an axiomatization that allows one to apply all arithmetic operations and properties of traditional analysis in an intuitive way. For instance, the sum of two standard numbers is standard, the sum of two infinitesimal numbers is an infinitesimal and the sum of an infinitesimal with a standard number is a non-standard number. The theory of NSA introduces, in addition to the notion of infinitesimal numbers and operations on them, the notion of infinite numbers (which are, intuitively, greater than any value in T), plus a rich set of results that make NSA an appealing framework for reasoning on both familiar and new objects. In this paper, we exploit some of the terminology and concepts of NSA to provide an elegant characterization of "zero-time" steps, but we do not make use of the full power of the theory. For example, we do not deal with infinite numbers (i.e., we have that $\mathrm{ns}(x)$ iff $x = v \pm \epsilon$, with $\mathrm{st}(v)$ and ϵ infinitesimal), as they seem of little use when dealing with zero-time steps. Given a (standard) domain $T \subseteq \mathbb{R}$, we indicate with $\overline{T}$ its extension with infinitesimal numbers.

In keeping with the tradition of TRIO [8], X-TRIO can be interpreted over different temporal domains.

A model-theoretic semantics for X-TRIO is defined by following a fairly standard path on the basis of a temporal structure $S = \langle \overline{T}, \mathcal{D}, \beta, \nu, \sigma \rangle$, where:

- $\overline{T}$ is the time domain such that $\forall t \in \overline{T}$ it is $t \geq 0$;
- $\mathcal{D}$ is the union of the domains associated with functions and predicates (i.e., of their arguments and results);

- β associates, at every time instant, each function and predicate with its interpretation in that instant. For example, given a predicate p, $\beta(p,t)$ is the relation associated with p at instant t. β can be seen as the "system state", i.e., what holds in each time instant;

- $\nu : V \longrightarrow \mathcal{D}$, where V is a finite set of variables and constants, is an evaluation function that associates with each variable and constant in V a value in its domain;

- σ is the distinguishing element of the X-TRIO temporal structure. It is a (possibly infinite) sequence of time instants starting from the initial instant 0, called *History*. Intuitively, it represents the discrete sequence of instants when the system changes state. More precisely we have $\sigma = \{\sigma_i | i \in \mathbb{N}, \sigma_i \in \overline{T}, \sigma_0 = 0, \forall j \in \mathbb{N}$ s.t. $j < i$ it holds that $\sigma_j < \sigma_i$, and $\forall t \in \overline{T}$, if $\sigma_i < t < \sigma_{i+1}$ then for all function or predicate e it holds that $\beta(e, \sigma_i) = \beta(e,t))\}$.
  Thus, the X operator represents a step moving from σ_i to σ_{i+1}.

Given a term τ, its value at time $t \in \overline{T}$ is computed through a function α that is defined as follows:

- if τ is a variable $v \in V$ (resp., a constant $k \in V$), then $\alpha(v,t) = \nu(v)$ (resp., $\alpha(k,t) = \nu(k)$);

- if $\tau = f(\tau_1, \ldots, \tau_n)$, then $\alpha(\tau, t) = \beta(f,t)(\alpha(\tau_1,t), \ldots, \alpha(\tau_n,t))$.

Notice that the value of variables and constants of V does not depend on the time t.

Then, the satisfaction relation $\vDash$ of an X-TRIO formula ϕ by structure $S = \langle \overline{T}, \mathcal{D}, \beta, \nu, \sigma \rangle$ at a time instant $t \in \overline{T}$ is defined as follows:

$S, t \vDash p(\tau_1, \ldots, \tau_n)$ iff $\langle \alpha(\tau_1,t), \ldots, \alpha(\tau_n,t) \rangle \in \beta(p,t)$

$S, t \vDash \neg\phi$ iff $S, t \nvDash \phi$

$S, t \vDash \phi_1 \wedge \phi_2$ iff $S, t \vDash \phi_1$ and $S, t \vDash \phi_2$

$S, t \vDash \mathrm{Dist}(\phi, d)$ iff $t + \alpha(d,t) \in \overline{T}$ and $S, t + \alpha(d,t) \vDash \phi$

$S, t \vDash \mathrm{X}(\phi)$ iff there is $i \in \mathbb{N}$ s.t. $\sigma_i \leq t < \sigma_{i+1}$ and $S, \sigma_{i+1} \vDash \phi$

$S, t \vDash \mathrm{Y}(\phi)$ iff there is $i \in \mathbb{N}$ s.t. $\sigma_{i-1} < t \leq \sigma_i, i > 0$ and $S, \sigma_{i-1} \vDash \phi$

$S, t \vDash \forall v.\phi$ iff for all ν' that differ from ν at most for the value of v,
$\langle \overline{T}, \mathcal{D}, \beta, \nu', \sigma \rangle, t \vDash \phi$.

Finally a formula ϕ is *satisfiable* in a structure S iff $S, 0 \vDash \phi$. Notice that in this paper we consider system evolutions that conventionally begin from time 0.

The above definition of X-TRIO semantics shows a first distinguishing feature of this language w.r.t. approaches based on timed words: although in both cases system evolution is determined in a discrete sequence of instants (the σ_i

instants in X-TRIO and the *timestamps* in timed words semantics), X-TRIO, thanks to its typical Dist operator, allows for predicating system properties at any element of the time domain, not only at the "stamped ones"; in fact, whereas the system state (formalized by the β component) does not change outside σ, temporal distances between the various events keep changing throughout the flow of time. We will see, e.g., in Section 5.1, that this feature allows one to express some critical properties otherwise not expressible in approaches based on timed words.

3.3. Examples of usage of X-TRIO

In its present general version, X-TRIO allows users to express any system property of interest (remember that it has full Turing computational power since it includes first-order arithmetic). In this section, by following the typical TRIO style, we introduce some useful derived X-TRIO operators from those of Section 3.2. We then use them to express meaningful system properties in an intuitive way.

Consider as time domain the set of the reals augmented with infinitesimals, and then restricted to the nonnegative numbers. We denote it as $\overline{\mathbb{R}}_{\geq 0}$.

We start by identifying the origin of the temporal domain (which is, by definition, bounded to the left by the origin) through the following X-TRIO formula, where k is any constant in $\overline{\mathbb{R}}_{\geq 0}$:

$$\text{orig} = \forall d.(0 < d < k \rightarrow \text{Dist}(\neg \text{Dist}(\top, -k), d))$$

In fact, formula orig holds only at instant 0, since $\text{Dist}(\neg \text{Dist}(\top, -k), d)$ is true in an instant $t \in \overline{\mathbb{R}}_{\geq 0}$ iff at $t+d$ $\text{Dist}(\top, -k)$ is false, which occurs iff $t+d-k < 0$. Then, for $0 < t < k$, if $d = k - t/2$, then $t + d - k = t/2 > 0$, so orig does not hold (similarly if $t \geq k$).

We can also introduce an abbreviation for the duration of the current step, which is the difference of the "timestamps" (i.e., the distance from the origin) between the end and the start of the step:

$$\text{Dur}(d) = \exists d_1, d_2(\text{X}(\text{Past}(\text{orig}, d_1) \wedge \text{Y}(\text{Past}(\text{orig}, d_2))) \wedge d = d_1 - d_2)$$

We can distinguish micro-steps (which take an infinitesimal time) from macro-steps (which take a non-infinitesimal time) by means of the following new derived operators, where X_m (resp., X_M) stands for "the next step is a micro one" (resp., macro):

$$\begin{array}{rcl} \text{X}_\text{m}(\phi) & = & \text{X}(\phi) \wedge \forall d(\text{Dur}(d) \rightarrow \text{inf}(d)) \\ \text{X}_\text{M}(\phi) & = & \text{X}(\phi) \wedge \forall d(\text{Dur}(d) \rightarrow \neg\text{inf}(d)) \end{array}$$

We can also introduce operators to state whether the next step ends in a standard or in a nonstandard instant. For this, it is useful to introduce abbreviation $\text{NowST} = \exists t(t > 0 \wedge \text{Past}(\text{orig}, t) \wedge \text{st}(t))$, which holds exactly in all those instants that are standard.

$$\begin{array}{rcl} \text{X}_\text{st}(\phi) & = & \text{X}(\phi \wedge \text{NowST}) \\ \text{X}_\text{ns}(\phi) & = & \text{X}(\phi \wedge \neg\text{NowST}) \end{array}$$

Let us now use X-TRIO formulae to define some interesting behaviors of systems that evolve through micro- and macro-steps.

The following formula states that "the system keeps going forever", i.e. at any time, the system will make progress with further steps, whether micro or macro:

$$\mathrm{AlwF}(\mathrm{SomF}(\mathrm{X}(\top)))$$

Note that the presence of the AlwF operator in the formula above implies that macro-steps occur infinitely often. Conversely, the following one claims that at some point the system will stop forever:

$$\mathrm{SomF}(\mathrm{X}(\mathrm{AlwF}(\neg \mathrm{X}(\top))))$$

i.e., there will be a point at some time when the next step will cause the system to have no further steps. Thus the formula can be satisfied only by a finite sequence σ.

The similar formula $\mathrm{SomF}(\mathrm{AlwF}(\neg \mathrm{X}(\top)))$, instead, could also be satisfied by an infinite σ where the steps never advance past a certain time instant. This leads us to consider some typical "pathological" behaviors of systems that may exhibit unbounded sequences of steps with no corresponding time advancement. Such behaviors are well known in the literature as "Zeno behaviors". Next we show how X-TRIO allows us to formalize and distinguish in a natural way various forms of such pathological behaviors.

A first sufficient but not necessary condition to exhibit a Zeno behavior is that, from some point on, only micro-steps occur:

$$\mathrm{SomF}(\mathrm{X_m}(\top) \wedge \mathrm{AlwF}(\mathrm{X_m}(\top) \rightarrow \mathrm{X_m}(\mathrm{X_m}(\top)))) \tag{2}$$

On domain $\overline{\mathbb{R}}_{\geq 0}$, a Zeno behavior can occur also with a σ consisting exclusively of macro-steps, which however have an always decreasing duration. As an example, the following formula

$$\mathrm{SomF}(\mathrm{AlwF}(\neg \mathrm{X}(\top))) \wedge \mathrm{SomF}(\mathrm{X_M}(\top) \wedge \mathrm{AlwF}(\mathrm{X_M}(\top) \rightarrow \mathrm{X_M}(\mathrm{X_M}(\top)))) \tag{3}$$

can be satisfied, e.g., by a sequence σ whose steps σ_i occur at time instants $\sum_{k=0}^{i-1} \frac{1}{2^k}$.

If, instead, we restrict the time domain to be a discrete set augmented with infinitesimals, such as the one that will be used in Section 5, then only Formula (2) captures Zeno behaviors.

We can also specify so-called "Berkeley behaviors" [24], i.e., those where time keeps advancing, but the step duration is ever decreasing or, more precisely, it becomes shorter than any standard number:

$$\begin{gathered} \mathrm{AlwF}(\mathrm{SomF}(\mathrm{X_M}(\top))) \quad \wedge \\ \forall t(\mathrm{st}(t) \rightarrow \mathrm{SomF}(\mathrm{AlwF}(\mathrm{X_M}(\top) \rightarrow \forall d(\mathrm{Dur}(d) \rightarrow d < t)))) \end{gathered} \tag{4}$$

which is satisfied, e.g., by a sequence whose steps occur at time instants $\sum_{k=1}^{i} \frac{1}{k}$.

Notice that, whereas Zeno behaviors may occur through infinite sequences of both micro- and macro-steps, both within a discrete and a dense time domain, Berkeley behaviors are only possible in dense time domains and through macro-steps.

From the point of view of the physical intuition, the behaviors specified by formulae (2), (3), and (4) could all be considered as "pathological" and could look as "almost indistinguishable", but the differences in the mathematical formalization could be used, for instance, to separate cases in which an unstable clock ever increases its frequency (formulae (3) and (4)) from cases where an unacceptable number of gate switches is supposed to occur within a clock period.

4. Towards decidable versions of X-TRIO

Since the advent of model checking, much research effort has been devoted to the definition of logic languages that exhibit a good trade-off among naturalness (ease of usage), expressiveness (computational power), and decidability[1] effort (complexity of decision procedures). In the case of TRIO, such an effort has produced several "LTL-oriented" versions of the original language and supporting tools[2]. In this section, we trace a path to achieve the same goal for the new X-TRIO language. To this end, several directions are possible, depending on the combination of syntactic and domain restrictions chosen. In fact, the full generality of temporal domains augmented with infinitesimal numbers as in NSA is both too powerful to achieve decidability and often even useless in practical cases (we already excluded infinite numbers since we are not interested in behaviors that take, say, an infinite time to perform a single step); thus, there are several ways of restricting time domains which clearly affect decidability and computational complexity properties. Some of these are effective only when combined with suitable syntactic restrictions, which in turn depend on the domain chosen. For instance, in most practical cases we can assume that a macro step ends always in a standard element of the time domain, i.e., $\mathrm{X_M}(\top) \rightarrow \mathrm{X_{st}}(\top)$. In this paper, we will focus on one particular version of X-TRIO that exhibits a good trade-off between expressiveness and computational effort: it can be translated into LTL formulae (with past operators), it is PSPACE-complete, and it can be (and has been) implemented in satisfiability solvers for LTL, such as Zot. The investigation of other approaches is left for future work.

The path to achieve our goal, however, is not straightforward and faces several technical difficulties. Thus, we will proceed through various versions both of the syntax of the original X-TRIO language and of its interpretation domain, in some sense "zooming-in" into the features of the language operators and

[1] As usual, in this paper by *decidable logic language* we mean a language whose satisfiability problem is decidable.

[2] `http://github.com/fm-polimi/zot`

their interpretation. According to the typical approach to achieve decidability in MTL-like languages, our first step will be to restrict to a propositional version of the language. A careful analysis of the expressiveness and decidability limits of the first version of the language in the time domain $\overline{\mathbb{R}}_{\geq 0}$ will drive us to select an appropriate set of basic X-TRIO operators, a suitable discrete time domain (where, in some sense, even nonstandard time instants are "discretized") and a key hypothesis for its interpretation that guarantees its decidability, obtained by translating its formulae into "equisatisfiable" LTL ones.

Table 2 at the end of Section 5 provides a summary of the various options considered in this paper.

4.1. A propositional X-TRIO

From now on, we restrict X-TRIO to a propositional, MTL-like syntax, where the metric Until and Since, which in the original TRIO are derived operators, become primitive. In addition, we specialize the X operator in two separate cases, $\mathrm{X_{st}}$ and $\mathrm{X_{ns}}$. We will also introduce a condition on the history σ that will entail that a "jump" from σ_i to σ_{i+1} corresponds to a macro-step (resp., micro-step) if and only if $\mathrm{st}(\sigma_{i+1})$ (resp., $\mathrm{ns}(\sigma_{i+1})$).

All in all, we introduce the following syntax for X-TRIO, where $p \in AP$, the set of propositional letters, a and b are constant distances such that $a \leq b$, $a < \infty$, $b \leq \infty$, $\langle$ is either (or [, $\rangle$ is either) or], and $\rangle \neq]$ if $b = \infty$:

$$\begin{array}{rcl} \phi & := & p \,|\, \neg\phi \,|\, \phi_1 \wedge \phi_2 \,|\, \mathrm{Until}_{\langle \mathrm{a,b} \rangle}(\phi_1, \phi_2) \,|\, \mathrm{Since}_{\langle \mathrm{a,b} \rangle}(\phi_1, \phi_2) \,| \\ & & \mathrm{X_{st}}(\phi) \,|\, \mathrm{X_{ns}}(\phi) \,|\, \mathrm{Y_{st}}(\phi) \,|\, \mathrm{Y_{ns}}(\phi) \end{array} \tag{5}$$

As a consequence of the syntax simplification, the structures over which formulae are to be interpreted are also simpler. More precisely, a temporal structure is now a triple $S = \langle \overline{T}, \beta, \sigma \rangle$, where:

- $\overline{T}$ is the time domain (as before, $\forall t \in \overline{T}$ it holds that $t \geq 0$).
- $\beta : \overline{T} \longrightarrow 2^{AP}$ associates with every time instant the propositions that hold in that instant.
- σ is defined as before, with the additional constraint that if $i > 0$ and $\mathrm{ns}(\sigma_i)$, then for all $\sigma_{i-1} < t < \sigma_i$, it holds that $\mathrm{ns}(t)$.

Then, the satisfaction relation $\vDash$ on a structure $S = \langle \overline{T}, \beta, \sigma \rangle$ at a time

instant $t \in \overline{T}$ is defined as follows:

$S, t \vDash p$ iff $p \in \beta(t)$
$S, t \vDash \neg\phi$ iff $S, t \nvDash \phi$
$S, t \vDash \phi_1 \wedge \phi_2$ iff $S, t \vDash \phi_1$ and $S, t \vDash \phi_2$
$S, t \vDash \mathrm{X}_{\mathrm{st}}(\phi)$ iff there is $i \in \mathbb{N}$ s.t. $\sigma_i \leq t < \sigma_{i+1}$, $S, \sigma_{i+1} \vDash \phi$ and $\mathrm{st}(\sigma_{i+1})$
$S, t \vDash \mathrm{X}_{\mathrm{ns}}(\phi)$ iff there is $i \in \mathbb{N}$ s.t. $\sigma_i \leq t < \sigma_{i+1}$, $S, \sigma_{i+1} \vDash \phi$ and $\mathrm{ns}(\sigma_{i+1})$
$S, t \vDash \mathrm{Y}_{\mathrm{st}}(\phi)$ iff there is $i \in \mathbb{N}$ s.t. $\sigma_{i-1} < t \leq \sigma_i, i > 0$, $S, \sigma_{i-1} \vDash \phi$ and $\mathrm{st}(\sigma_{i-1})$
$S, t \vDash \mathrm{Y}_{\mathrm{ns}}(\phi)$ iff there is $i \in \mathbb{N}$ s.t. $\sigma_{i-1} < t \leq \sigma_i, i > 0$, $S, \sigma_{i-1} \vDash \phi$ and $\mathrm{ns}(\sigma_{i-1})$
$S, t \models \mathrm{Until}_{\langle \mathrm{a,b} \rangle}(\phi, \psi)$ iff $\exists t' \in \overline{T}$ s.t. $t + a \prec_1 t' \prec_2 t + b$ and $S, t', \models \psi$
and $\forall t'' \in \overline{T}$ s.t. $t \leq t'' < t'$, it holds that $S, t'' \models \phi$
$S, t \models \mathrm{Since}_{\langle \mathrm{a,b} \rangle}(\phi, \psi)$ iff $\exists t' \in \overline{T}$ s.t. $t - b \prec_2 t' \prec_1 t - a$ and $S, t' \models \psi$
and $\forall t'' \in \overline{T}$ s.t. $t' < t'' \leq t$, it holds that $S, t'' \models \phi$

In the definition of the semantics of the metric Until and Since, $\prec_1$ (resp., $\prec_2$) is $\leq$ or $<$ depending on whether the left (resp., right) endpoint is included or excluded.

In the rest of this section, we study some relevant properties of the X-TRIO logic defined by syntax (5). In particular, we study how its expressiveness and decidability are affected by the choice of temporal domain and by possible restrictions on the use of temporal operators.

4.2. Expressiveness

In this section, we present some results concerning the expressiveness of X-TRIO in relation to the set of allowed temporal operators and to the time domain $\overline{T}$ used. To this end, we study the problem of stating that "the current instant corresponds to a standard value". In fact, when formalizing and analyzing systems that evolve through micro- and macro-steps, it is often useful to write a formula F that holds in an instant t if, and only if, t is standard.

First of all, we introduce the following abbreviations:

$$\begin{aligned}
\mathrm{Until}(\phi, \psi) &= \mathrm{Until}_{(0,\infty)}(\phi, \psi) \\
\mathrm{SomF}(\phi) &= \mathrm{Until}_{[0,\infty)}(\top, \phi) \\
\mathrm{Futr}(\phi, d) &= \mathrm{Until}_{[\mathrm{d,d}]}(\top, \phi) \\
\mathrm{WithinF}_{\langle\rangle}(\phi, d) &= \mathrm{Until}_{\langle 0,\mathrm{d} \rangle}(\top, \phi) \\
\mathrm{Lasts}_{\langle\rangle}(\phi, d) &= \neg\mathrm{WithinF}_{\langle\rangle}(\neg\phi, d)
\end{aligned}$$

We can similarly define operators Since, SomP, Past, WithinP and Lasted.

It is easy to see that the following equivalences hold (similar ones hold for the past operators):

$$\begin{aligned}
\mathrm{Until}_{(\mathrm{a},\infty)}(\phi, \psi) &\equiv \mathrm{Lasts}_{[)}(\phi, a) \wedge \mathrm{Futr}(\mathrm{Until}(\phi, \psi), a) \\
\mathrm{Until}_{[\mathrm{a},\infty)}(\phi, \psi) &\equiv \mathrm{Lasts}_{[)}(\phi, a) \wedge \mathrm{Futr}(\psi \vee \mathrm{Until}(\phi, \psi), a) \\
\mathrm{Until}_{(\mathrm{a,b}\rangle}(\phi, \psi) &\equiv \mathrm{Lasts}_{[)}(\phi, a) \wedge \mathrm{Futr}\big(\mathrm{Until}(\phi, \psi) \wedge \mathrm{WithinF}_{\langle)}(\psi, b - a), a\big) \\
\mathrm{Until}_{[\mathrm{a,b}\rangle}(\phi, \psi) &\equiv \mathrm{Lasts}_{[)}(\phi, a) \wedge \\
&\quad \mathrm{Futr}\big((\psi \vee \mathrm{Until}(\phi, \psi)) \wedge \mathrm{WithinF}_{[)}(\psi, b - a), a\big)
\end{aligned} \tag{6}$$

Thanks to the equivalences above, without loss of generality in the rest of this section we use Until (non-metric), Futr and WithinF (resp. Since, Past and WithinP) as basic temporal operators, instead of the metric $\text{Until}_{\langle \text{a,b} \rangle}$ (resp. $\text{Since}_{\langle \text{a,b} \rangle}$).

Let us now consider, for the temporal domain, the following subset of $\overline{\mathbb{R}}_{\geq 0}$: $\overline{\mathbb{R}}_{+} = \{x \in \overline{\mathbb{R}}_{\geq 0} | \exists v, \epsilon \geq 0 \text{ s.t. } \text{st}(v), \inf(\epsilon), \text{ and } x = v + \epsilon\}$, which is comprised of the numbers of the form $v+\epsilon$, where v is a nonnegative real, and ϵ is a nonnegative infinitesimal – i.e., with respect to $\overline{\mathbb{R}}_{\geq 0}$, it does not include nonstandard numbers of the form $v - \epsilon$.

Our first results show the impact of the past operators on expressiveness, when the time domain is $\overline{\mathbb{R}}_{+}$. We call an X-TRIO formula F *metric-future-only* if it does not contain any instances of operator $\text{Since}_{\langle \text{a,b} \rangle}$ (or, equivalently, if it does not contain any instances of operators Since, Past and WithinP). Notice that a metric-future-only X-TRIO formula can still have instances of the stepwise past operators Y_{st} and Y_{ns}. Similarly, *metric-future-only* X-TRIO is the restriction of X-TRIO that comprises only metric-future-only formulae. We have the following theorem.

Theorem 4.1. *Over the $\overline{\mathbb{R}}_{+}$ temporal domain, X-TRIO (with past operators) is strictly more expressive than its metric-future-only counterpart.*

To prove Theorem 4.1, we introduce a pair of intermediate lemmata.

Lemma 4.2. *If $AP = \varnothing$ and $S = \langle \overline{\mathbb{R}}_{+}, \beta, \sigma \rangle$ is such that the history σ only contains standard instants (i.e., $\forall i \in \mathbb{N}$ it holds that $\text{st}(\sigma_i)$), any metric-future-only X-TRIO formula F with less than n instances of the Y_{st} operator is such that, given two instants t_1, t_2, such that $\sigma_n < t_1 < t_2$, $S, t_1 \models F$ holds iff $S, t_2 \models F$ holds.*

Proof. First, note that in structure S, formulae $\text{X}_{\text{ns}}(\phi)$ and $\text{Y}_{\text{ns}}(\phi)$ are always false, independent of ϕ. The proof is by induction on the structure of F.

The base cases, in which $F = \top$ or $F = \bot$, are trivial, and so are the cases $F = \phi_1 \wedge \phi_2$ and $F = \neg\phi$.

Suppose F is $\text{Futr}(\phi, k)$; then, we have $S, t_1 \models F$ iff $S, t_1 + k \models \phi$; since $\sigma_n < t_1 < t_1 + k$, by inductive hypothesis this holds iff $S, t_2 + k \models \phi$, i.e., iff $S, t_2 \models \text{Futr}(\phi, k)$.

If $F = \text{WithinF}_{()}(\phi, k)$, $S, t_1 \models F$ iff there is $0 < d < k$ s.t. $S, t_1 + d \models \phi$. Since $t_2 + d > t_1 + d > \sigma_n$, by inductive hypothesis, $S, t_1 + d \models \phi$ iff $S, t_2 + d \models \phi$, hence $S, t_2 \models F$. The cases for other variants of the WithinF operator and for the Until are similar.

If $F = \text{X}_{\text{st}}(\phi)$, $S, t_1 \models F$ iff $S, \sigma_i \models \phi$, where σ_i is the first element of σ such that $\sigma_i > t_1$ (note that, by hypothesis, it holds that $\text{st}(\sigma_i)$). If σ_j is the smallest element of σ s.t. $\sigma_j > t_2$ (also, $\text{st}(\sigma_j)$ holds), by inductive hypothesis, we have $S, \sigma_i \models \phi$ iff $S, \sigma_j \models \phi$, hence iff $S, t_2 \models \text{X}_{\text{st}}(\phi)$.

If $F = \text{Y}_{\text{st}}(\phi)$, $S, t_1 \models F$ iff $S, \sigma_i \models \phi$, where σ_i is the biggest element of σ such that $\sigma_i < t_1$ (by hypothesis, it holds that $\text{st}(\sigma_i)$). If σ_j is the biggest element of σ s.t. $\sigma_j < t_2$ (also, $\text{st}(\sigma_j)$ holds), since the number of operators Y_{st}

in ϕ is $n-1$, and $\sigma_j \geq \sigma_i > \sigma_{n-1}$, by inductive hypothesis $S, \sigma_i \models \phi$ holds iff $S, \sigma_j \models \phi$ does, hence iff $S, t_2 \models \mathrm{Y}_{\mathrm{st}}(\phi)$ holds. □

Thanks to Lemma 4.2, we can easily show that the next result holds.

Lemma 4.3. *There is no metric-future-only X-TRIO formula F, with $AP = \varnothing$, such that, for any structure $S = \langle \overline{\mathbb{R}}_+, \beta, \sigma \rangle$, $S, t \models F$ holds iff* $\mathrm{st}(t)$ *also holds (i.e., F holds exactly in all standard instants t of $\overline{\mathbb{R}}_+$).*

Proof. The proof is by contradiction. Assume there is a metric-future-only formula F such that $AP = \varnothing$ and, given a structure S, it holds that $S, t \models F$ iff $\mathrm{st}(t)$. If n is the number of instances of operator Y_{st} in F, given a structure S that satisfies the hypotheses of Lemma 4.2, two instants t_1, t_2 such that $\sigma_n < t_1 < t_2$, $\mathrm{st}(t_1)$, and $\mathrm{ns}(t_2)$ hold, by Lemma 4.2, $S, t_1 \models F$ holds iff $S, t_2 \models F$, which contradicts the assumption. □

The proof of Theorem 4.1 is now straightforward.

Proof of Theorem 4.1. Thanks to past temporal operators, we can define X-TRIO formula $\mathrm{NowST} = \mathrm{Lasts}_{[)}(\neg \mathrm{Past}(\top, \epsilon), \epsilon)$, where $\epsilon > 0$ is any infinitesimal constant, which is true exactly at the instants $t \in \overline{\mathbb{R}}_+$ such that $\mathrm{st}(t)$ holds. However, by Lemma 4.3, there is no metric-future-only formula equivalent to NowST. □

Let us change the temporal domain from $\overline{\mathbb{R}}_+$ to $\overline{\mathbb{R}}_{\geq 0}$, which includes non-negative instants of the form $v - \epsilon$. For domain $\overline{\mathbb{R}}_{\geq 0}$, it is not possible anymore to separate standard and non-standard instants, as the following result shows.

Theorem 4.4. *There is no X-TRIO formula F, with $AP = \varnothing$, such that, for any structure $S = \langle \overline{\mathbb{R}}_{\geq 0}, \beta, \sigma \rangle$, $S, t \models F$ holds iff* $\mathrm{st}(t)$ *also holds.*

The proof of Theorem 4.4, is based on a pair of lemmata, which are somewhat inspired by the results in [31].

Lemma 4.5. *Let F be an X-TRIO formula over the set of atomic propositions $AP = \{p\}$. Let δ_{F_+} and δ_{F_-} be the sum of all time bounds appearing in future and past operators in F, respectively, and $\delta_F = \delta_{F_+} + \delta_{F_-}$. Let ε be an infinitesimal and let $S_i = \langle \overline{\mathbb{R}}_{\geq 0}, \beta_i, \sigma_i \rangle$ and $S_j = \langle \overline{\mathbb{R}}_{\geq 0}, \beta_j, \sigma_j \rangle$ be two structures such that: (a) $j > i > 1 + \delta_F$; (b) for all $k \in \mathbb{N}$ it holds that $\sigma_{i,k} = \sigma_{j,k} = k\varepsilon$; (c) for all $t \in \overline{\mathbb{R}}_{\geq 0}$, it holds that $\beta_i(t) = \{p\}$ (resp., $\beta_j(t) = \{p\}$), except for i (resp., j), where $\beta_i(i) = \emptyset$ (resp., $\beta_j(j) = \emptyset$). Then, $S_i, 0 \models F$ holds iff $S_j, 0 \models F$ also holds.*

Proof. Structures S_i, S_j correspond to Zeno behaviors in which all $\sigma_{i,k}, \sigma_{j,k}$ are at an infinitesimal distance from the origin. Then, for all t, ϵ (with ϵ infinitesimal) such that $t \geq 1 - \epsilon$ and any formula ϕ, it holds that $S_i, t \not\models \mathrm{X}_{\mathrm{ns}}(\phi)$, $S_j, t \not\models \mathrm{X}_{\mathrm{ns}}(\phi)$, and similarly for $\mathrm{X}_{\mathrm{st}}(\phi)$, $\mathrm{Y}_{\mathrm{st}}(\phi)$, $\mathrm{Y}_{\mathrm{ns}}(\phi)$. In fact, $\mathrm{X}_{\mathrm{st}}(\phi)$ is always false, and $\mathrm{Y}_{\mathrm{st}}(\phi)$ is false for all $t > \varepsilon$.

We show the following:

- for all t such that $t \leq \delta_{F_-} + 1$, it holds that $S_i, t \models F$ iff $S_j, t \models F$;
- for all $t \geq i - \delta_{F_+}$, $S_i, t \models F$ holds iff $S_j, j - i + t \models F$ holds, i.e., starting from δ_{F_+} instants before p becomes false in each structure S_i, S_j, F has the same values in S_i and S_j if the same offset is considered;
- for all $1 + \delta_{F_-} \leq t < i - \delta_{F_+}$, it holds that $S_i, t \models F$ iff $S_i, 1 + \delta_{F_-} \models F$; i.e., in all instants from $1 + \delta_{F_-}$ (included) to $i - \delta_{F_+}$ (excluded) F has the same value in structure S_i.

The proof is by induction on the structure of F.

The property holds in the base case $F = p$, as S_i and S_j are the same except in instants i and j, and $j > i > 1 + \delta_{F_-} + \delta_{F_+}$. The cases in which F is $\neg\phi$ or $\phi_1 \wedge \phi_2$ are trivial.

If $F = \text{Futr}(\phi, d)$, it holds that $S_i, t \models F$ iff $S_i, t+d \models \phi$. If $t \geq i - \delta_{F_+}$, then $t + d \geq i - \delta_{\phi_+}$, and by induction hypothesis $S_i, t+d \models \phi$ iff $S_j, j - i + t + d \models \phi$, i.e. $S_j, j - i + t \models F$. If $t + d < 1 + \delta_{F_-}$, then $S_i, t + d \models \phi$ iff $S_j, t + d \models \phi$, hence the result. If $1 + \delta_{F_-} \leq t < i - \delta_{F_+}$ or $1 + \delta_{F_-} \leq t + d < i - \delta_{F_+}$, then $t + d < i - \delta_{\phi_+} < j - \delta_{\phi_+}$, as $t < i - \delta_{F_+}$ and $\delta_{F_+} = \delta_{\phi_+} + d$, so $S_i, t + d \models \phi$ iff $S_i, 1 + \delta_{F_-} \models \phi$, iff $S_j, 1 + \delta_{F_-} \models \phi$, iff $S_j, t + d \models \phi$, hence the result. Similarly for $\text{WithinF}_{\langle\rangle}(\phi, d)$.

The reasoning is symmetric if $F = \text{Past}(\phi, d)$, for which $S_i, t \models F$ iff $t - d \in \overline{\mathbb{R}}_{\geq 0}$ and $S_i, t - d \models \phi$, when one considers that $\delta_{F_-} = \delta_{\phi_-} + d$. Similarly for $\text{WithinP}_{\langle\rangle}(\phi, d)$.

If $F = \text{X}_{\text{ns}}(\phi)$, $S_i, t \not\models F$, and $S_j, t \not\models F$ if $t > 1 - \epsilon$ for some infinitesimal ϵ. If $t = \epsilon$ for some infinitesimal ϵ, then $S_i, t \models F$ and $S_j, t \models F$ iff $S_i, \sigma_{i,k+1} \models \phi$, with $\sigma_{i,k} \leq t < \sigma_{i,k+1}$, as σ_i and σ_j are the same. Similarly for $\text{X}_{\text{st}}(\phi)$, $\text{Y}_{\text{st}}(\phi)$, and $\text{Y}_{\text{ns}}(\phi)$.

If $F = \text{Until}(\phi, \psi)$, then $S_i, t \models F$ holds if there is $t' > t$ such that $S_i, t' \models \psi$ holds. We separate several cases. If $t \geq i - \delta_{F_+}$, then by induction hypothesis $S_i, t' \models \psi$ iff $S_j, j - i + t' \models \psi$, and also by induction hypothesis we have that for all $t \leq t'' < t'$ it holds that $S_i, t'' \models \phi$ iff for all $j - i + t \leq t'' < j - i + t'$ it holds that $S_j, t'' \models \phi$, hence the result. Similarly if $t < t' < 1 + \delta_{F_-}$. If $t < i - \delta_{F_+} \leq t'$, then by induction hypothesis $S_i, t' \models \psi$ iff $S_j, j - i + t' \models \psi$, $S_i, t \models \phi$ iff $S_j, t \models \psi$, and, for all $t \leq t'' < t'$, $S_i, t'' \models \phi$ holds iff for all $t \leq t'' < j - i + t'$ it holds that $S_j, t'' \models \phi$, hence the result. The other cases are similar, and are not detailed here for brevity. The case $F = \text{Since}(\phi, \psi)$ is dual.

$S_i, 0 \models F$ iff $S_j, 0 \models F$ follows by observing that $0 < 1 + \delta_{\phi_-}$. □

The next lemma shows that there is no X-TRIO formula that can express the property "p holds in all standard instants".

Lemma 4.6. *There is no X-TRIO formula F, with $AP = \{p\}$, such that, for any structure $S = \langle \overline{\mathbb{R}}_{\geq 0}, \beta, \sigma \rangle$, $S, 0 \models F$ holds iff, for every standard instant t in $\overline{\mathbb{R}}_{\geq 0}$, it holds that $p \in \beta(t)$.*

Proof. The proof is, as usual, by contradiction. Suppose there is an X-TRIO formula F such that $S, 0 \models F$ holds iff, for every $t \in \overline{\mathbb{R}}_{\geq 0}$ such that $\text{st}(t)$

holds, $p \in \beta(t)$ also holds. Consider two structures, S_i and S_j, that satisfy the hypotheses of Lemma 4.5, where instants i, j are such that $\mathrm{st}(i)$ and $\mathrm{ns}(j)$ hold (i.e., in S_i there is a standard instant in which p does not hold, whereas in S_j p holds in all standard instants). Then, by Lemma 4.5, we have that $S_i, 0 \models F$ holds iff $S_j, 0 \models F$ also holds, which contradicts the assumption. □

Thanks to Lemma 4.6, the proof of Theorem 4.4 is now straightforward.

Proof of Theorem 4.4. Suppose there is an X-TRIO formula F such that, for any structure $S = \langle \overline{\mathbb{R}}_{\geq 0}, \beta, \sigma \rangle$ and for every $t \in \overline{\mathbb{R}}_{\geq 0}$, $S, t \models F$ holds iff $\mathrm{st}(t)$ holds. Then, X-TRIO formula $\mathrm{AlwF}(F \rightarrow p)$ would express the property "p holds in all standard instants", thus contradicting Lemma 4.6. □

4.3. Decidability

Theorem 4.7. *The satisfiability problem of X-TRIO formulae over the temporal domain $\overline{\mathbb{R}}_+$ is undecidable.*

Proof. To show the undecidability of X-TRIO, we reduce the satisfiability problem of MTL, which is known to be undecidable [6], to that of X-TRIO. First of all, notice that MTL is a syntactic fragment of X-TRIO, the one obtained by avoiding in syntax (5) the use of X, Y, and of any non-standard constants. Given an MTL formula F, we transform it in the X-TRIO formula F', which is obtained through the following translation τ:

- if ϕ is a propositional letter $p \in AP$, then $\tau(p) = p$;
- if $\phi = \neg\psi$, then $\tau(\phi) = \neg\tau(\psi)$;
- if $\phi = \psi_1 \wedge \psi_2$, then $\tau(\phi) = \tau(\psi_1) \wedge \tau(\psi_2)$;
- if $\phi = \mathrm{Until}_{\langle \mathrm{a,b} \rangle}(\psi_1, \psi_2)$, then $\tau(\phi) = \mathrm{Until}_{\langle \mathrm{a,b} \rangle}(\mathrm{NowST} \rightarrow \psi_1, \mathrm{NowST} \wedge \psi_2)$;
- if $\phi = \mathrm{Since}_{\langle \mathrm{a,b} \rangle}(\psi_1, \psi_2)$, then $\tau(\phi) = \mathrm{Since}_{\langle \mathrm{a,b} \rangle}(\mathrm{NowST} \rightarrow \psi_1, \mathrm{NowST} \wedge \psi_2)$.

Then, we need to show that ϕ is satisfiable if, and only if, $\tau(\phi)$ is. In fact, we prove a slightly stronger property. Let ϕ be an MTL formula, $\pi : \mathbb{R}_{\geq 0} \rightarrow 2^{AP}$ an interpretation for ϕ, and $S = \langle \overline{\mathbb{R}}_+, \beta, \sigma \rangle$ a structure for $\tau(\phi)$ such that, for all $p \in AP$, and for all $t \in \mathbb{R}_{\geq 0}$ (i.e., t is a standard number), $p \in \pi(t)$ if, and only if, $p \in \beta(t)$ (σ can be any, as it is only needed to evaluate X and Y operators, which do not appear in $\tau(\phi)$). Then, $\pi, t \models \phi$ if, and only if, $S, t \models \tau(\phi)$ (where $\models$ stands for the satisfiability relation of MTL in the former case, and for the one of X-TRIO in the latter). The proof is carried out by induction on the structure of formulae. The base case, where $\phi = p \in AP$ holds by hypothesis. The cases $\phi = \neg\psi$ and $\phi = \psi_1 \wedge \psi_2$ trivially hold by induction.

Let us focus on the case $\phi = \mathrm{Until}_{\langle \mathrm{a,b} \rangle}(\psi_1, \psi_2)$. By definition, ϕ holds in t if, and only if, there is $t' \in \langle t + a, t + b \rangle$ such that $\pi, t' \models \psi_2$ holds, and for all $t'' \in [t, t')$ ψ_1 holds, where $t', t'' \in \mathbb{R}_{\geq 0}$. By inductive hypothesis, this holds if, and only if, $S, t' \models \mathrm{NowST} \wedge \psi_2$, and for all $\bar{t} \in \overline{\mathbb{R}}_+$ such that $\bar{t} \in [t, t')$ either $\bar{t}$ is a

non-standard number (in which case ¬NowST holds in $\bar{t}$), or $S, \bar{t} \models \text{NowST} \wedge \psi_1$. This, in turn, is equivalent to $S, t \models \text{Until}_{\langle \text{a,b} \rangle}(\text{NowST} \rightarrow \psi_1, \text{NowST} \wedge \psi_2)$.

The case for $\phi = \text{Since}_{\langle \text{a,b} \rangle}(\psi_1, \psi_2)$ is similar. □

Note that Theorem 4.7 holds also when past operators are forbidden, or when the domain is $\overline{\mathbb{R}}_{\geq 0}$. In fact, to carry out its proof, one only needs to be able to identify standard instants and separate them from non-standard ones. It does not matter whether this is achieved through an X-TRIO formula that holds exactly in all standard instants, without introducing new propositional letters, or, conversely, by augmenting the alphabet with additional propositional letters. If the domain is $\overline{\mathbb{R}}_+$ and past temporal operators are allowed, we can follow the former path, by using the formula defined in the proof of Theorem 4.4. If past operators are forbidden, or the domain is $\overline{\mathbb{R}}_{\geq 0}$, we can follow the second path, and introduce a fresh atomic proposition NowST that holds exactly in all standard instants. This can be easily achieved through the following formula

$$\text{NowSTdef} = \text{NowST} \wedge \text{Alw}\big(\text{NowST} \rightarrow (\text{Futr}(\text{NowST}, 1) \wedge \text{Lasts}_{()}(\neg\text{NowST}, 1))\big)$$

which does not involve past operators, and holds both in $\overline{\mathbb{R}}_+$ and $\overline{\mathbb{R}}_{\geq 0}$. Then, an MTL formula ϕ is satisfiable if, and only if, X-TRIO formula $\tau(\phi) \wedge \text{NowSTdef}$ is satisfiable. Hence, we have the following results.

Theorem 4.8. *The satisfiability problem of X-TRIO formulae, without past operators, over the temporal domain $\overline{\mathbb{R}}_+$ is undecidable.*

Theorem 4.9. *The satisfiability problem of X-TRIO formulae over the temporal domain $\overline{\mathbb{R}}_{\geq 0}$ is undecidable.*

To achieve decidability many different choices are possible. For example, in the standard MTL case, one way to make the logic decidable is to limit the kinds of intervals that can be written in the metric Until modality [32]. Decidability is also often obtained by considering a discrete temporal domain. In the next sections, we explore this second path, without renouncing infinitesimals, however.

5. A Decidable fragment of X-TRIO

In this section, we focus our attention on discrete subsets of $\overline{\mathbb{R}}_{\geq 0}$. In particular, we consider domain $\overline{\mathbb{N}}_+ = \{x \in \overline{\mathbb{R}}_{\geq 0} | \exists v, k \in \mathbb{N} \text{ s.t. } x = v + k\varepsilon\}$, with ε a fixed positive infinitesimal. Note that $\overline{\mathbb{N}}_+$ is not the set of the "hypernaturals" [33], which does not include infinitesimal numbers, but, rather, a particular subset of $\overline{\mathbb{R}}_{\geq 0}$.

It is easy to see that the proof of Lemma 4.2 works also when the temporal domain is $\overline{\mathbb{N}}_+$. Then, Theorem 4.3 holds also for temporal domain $\overline{\mathbb{N}}_+$. In addition, to show that Theorem 4.1 holds for temporal domain $\overline{\mathbb{N}}_+$, it is enough to consider that, over $\overline{\mathbb{N}}_+$, formula $\neg\text{Past}(\top, \varepsilon)$ holds exactly in all standard instants.

In the syntax shown in (5), let us consider distances for the bounds a, b of the metric Until and Since operators that have the form $v \pm k\varepsilon$. On domain $\overline{\mathbb{N}}_+$, in addition to equivalences 6, we have a further set of results that allow us to consider only a subset of the operators introduced in syntax (5) (which includes, for example, an unbounded number of forms for operator Futr, one for each possible bound), without loss of generality. More precisely, one can show that through operators $\mathrm{Futr}(\bullet, 1)$, $\mathrm{Futr}(\bullet, \varepsilon)$, $\mathrm{WithinF}_{()}(\bullet, 1)$ and Until (and their past counterparts) one can express all other temporal operators. Let us show some of the most interesting equivalences.

First of all, it is easy to show that $\mathrm{Futr}(\bullet, v \pm k\varepsilon)$ can be expressed in terms of $\mathrm{Futr}(\bullet, 1)$ and $\mathrm{Futr}(\bullet, \varepsilon)$, as in the following (where $k \geq 1$, $\mathrm{st}(v)$, and $v \geq 1$):

$$\begin{array}{rcl} \mathrm{Futr}(\phi, v + k\varepsilon) & \equiv & \mathrm{Futr}(\mathrm{Futr}(\phi, k\varepsilon), v) \\ \mathrm{Futr}(\phi, v - k\varepsilon) & \equiv & \mathrm{Futr}(\mathrm{Past}(\phi, k\varepsilon), v) \\ \mathrm{Futr}(\phi, v) & \equiv & \mathrm{Futr}(\mathrm{Futr}(\phi, v-1), 1) \\ \mathrm{Futr}(\phi, k\varepsilon) & \equiv & \mathrm{Futr}(\mathrm{Futr}(\phi, (k-1)\varepsilon), \varepsilon) \end{array} \tag{7}$$

Similar equivalences hold for the Past operator.

The equivalences for operator WithinF must take into account the peculiarities of the underlying domain. In a standard, discrete domain such as $\mathbb{N}$, $\mathrm{WithinF}_{[]}(\phi, 1)$, for example, would simply be $\phi \vee \mathrm{Futr}(\phi, 1)$. However, in domain $\overline{\mathbb{N}}_+$, between 0 and 1 there is an infinity of non-standard numbers $k\varepsilon$, so $\mathrm{WithinF}_{[]}(\phi, 1)$ evaluated at 0 actually reads "ϕ holds either in the current instant, or 1 instant from now, or in one of the non-standard instants between now and 1 instant from now". On the other hand, $\mathrm{WithinF}_{[]}(\phi, 2\varepsilon)$ is still equivalent to $\phi \vee \mathrm{Futr}(\phi, \varepsilon) \vee \mathrm{Futr}(\phi, 2\varepsilon)$. We have the following lemma.

Lemma 5.1. *The following equivalences hold (where* $\mathrm{st}(v)$*,* $v > 1$*,* $k \geq 0$*,* $k' > 0$*,* $k'' > 1$*, and* $\mathrm{WithinF}_{()}(\phi, \varepsilon)$ *is trivially false).*

$$\mathrm{WithinF}_{()}(\phi, k''\varepsilon) \equiv \mathrm{Futr}\big(\phi \vee \mathrm{WithinF}_{()}(\phi, (k''-1)\varepsilon), \varepsilon\big) \tag{8}$$

$$\begin{array}{rcll} \mathrm{WithinF}_{()}(\phi, v \pm k\varepsilon) & \equiv & \mathrm{WithinF}_{()}(\phi, 1) & (9) \\ & & \vee\ \mathrm{Futr}\big(\phi \vee \mathrm{WithinF}_{()}(\phi, v - 1 \pm k\varepsilon), 1\big) & \\ \mathrm{WithinF}_{()}(\phi, 1 + k'\varepsilon) & \equiv & \mathrm{WithinF}_{()}(\phi, 1) & (10) \\ & & \vee\ \mathrm{Futr}\big(\phi \vee \mathrm{WithinF}_{()}(\phi, (k'-1)\varepsilon), 1\big) & \\ \mathrm{WithinF}_{()}(\phi, 1 - k\varepsilon) & \equiv & \mathrm{Futr}(\neg\mathrm{Until}(\neg\phi, \mathrm{NowST}), \varepsilon) & (11) \\ & & \vee\ \mathrm{Futr}(\mathrm{Since}(\neg\mathrm{NowST}, \phi), 1 - k\varepsilon) & \\ \mathrm{WithinP}_{()}(\phi, 1 - k\varepsilon) & \equiv & \mathrm{Since}(\neg\mathrm{NowST}, \phi) \ \vee & (12) \\ & & \mathrm{Past}(\neg\mathrm{Until}(\neg\phi, \mathrm{NowST}), 1 - k\varepsilon - \varepsilon) & \end{array}$$

Proof sketch. Let us focus on equivalence (11). $S, t \models \mathrm{WithinF}_{()}(\phi, 1 - k\varepsilon)$ if $S, t' \models \phi$ for any of the infinite instants in $[t + \varepsilon, t + 1 - k\varepsilon)$. Recall that $t \in \overline{\mathbb{N}}_+$ has the form $v + n\varepsilon$, so $v + 1$ is the first standard number greater than t. Then, $[t + \varepsilon, v + 1 - k\varepsilon) = [t + \varepsilon, v + 1) \cup [t + 1, v + 1 - k\varepsilon)$. If ϕ holds sometimes in $[t + \varepsilon, v + 1)$, then it cannot be that there is an instant $t' > t$ such that $\mathrm{st}(t')$

and, for all $t'' \in [t+\varepsilon, t')$ it holds that $S, t' \not\models \phi$. This corresponds to the first disjunct in equivalence (11). If ϕ holds in $[v+1, t+1-k\varepsilon)$, then it must be $t+1-k\varepsilon > v+1$ (notice that it could be that $t+1-k\varepsilon-\varepsilon = v+1$), and there must be $t' \in [v+1, t+1-k\varepsilon)$ such that $S, t' \models \phi$, and for all $t'' \in (t', t+1-k\varepsilon]$ it holds that $\mathrm{ns}(t'')$. This is captured by the second disjunct of equivalence (11). □

All other forms of operators WithinF and WithinP ($\mathrm{WithinF}_{[]}$, $\mathrm{WithinF}_{(]}$, etc.) can easily be expressed in terms of $\mathrm{WithinF}_{()}$ and $\mathrm{WithinP}_{()}$.

Equivalences (6)-(12) suggest the following equivalent syntax for X-TRIO, which is more suitable for our purposes.

$$\begin{array}{rcl} \phi & := & p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \mathrm{Futr}(\phi, 1) \mid \mathrm{Past}(\phi, 1) \mid \mathrm{Futr}(\phi, \varepsilon) \mid \mathrm{Past}(\phi, \varepsilon) \mid \\ & & \mathrm{Until}(\phi_1, \phi_2) \mid \mathrm{Since}(\phi_1, \phi_2) \mid \mathrm{X}_{\mathrm{st}}(\phi) \mid \mathrm{X}_{\mathrm{ns}}(\phi) \mid \mathrm{Y}_{\mathrm{st}}(\phi) \mid \mathrm{Y}_{\mathrm{ns}}(\phi) \end{array} \quad (13)$$

We name syntax (13) $\text{X-TRIO}_{\mathbb{N}}$.

Despite the restrictions introduced in $\text{X-TRIO}_{\mathbb{N}}$, however, the logic is still undecidable:

Theorem 5.2. *The satisfiability problem of the X-TRIO$_{\mathbb{N}}$ logic is undecidable.*

Proof sketch. Since, when interpreted over the time domain $\mathbb{N}$, MTL is decidable, unlike in the proof of Theorem 4.7 we cannot resort to a reduction from MTL to show the undecidability of $\text{X-TRIO}_{\mathbb{N}}$. Instead, in classic fashion (see, e.g., [34]), we reduce the halting problem of a 2-counter machine to the satisfiability problem of $\text{X-TRIO}_{\mathbb{N}}$ formulae, by defining a set of $\text{X-TRIO}_{\mathbb{N}}$ formulae that formalize the increment and decrement of the 2 counters.

We associate one counter with the sequence of even standard numbers, and one with the sequence of odd standard numbers, as detailed below. We associate two propositional letters, E and O, with each standard instant of σ so that when the current standard instant is even (resp., odd) then only E (resp., O) holds. They do not hold in non-standard instants. These constraints are represented by the following $\text{X-TRIO}_{\mathbb{N}}$ formulae (we show the case of even instants):

$$\begin{array}{ll} E \rightarrow & \mathrm{Futr}(\mathrm{Until}(\neg\mathrm{NowST} \wedge \neg O \wedge \neg E, \mathrm{NowST} \wedge O), \varepsilon) \\ E \leftrightarrow & \mathrm{Futr}(O, 1) \end{array}$$

Given two consecutive standard instants σ_j and σ_i in σ (i.e., such that $\sigma_i = \sigma_j + 1$), there is a finite nonempty sequence $\sigma_{[j,i)}$ of length $i-(j+1)$ of non-standard instants in σ between them since σ is discrete. We introduce $\text{X-TRIO}_{\mathbb{N}}$ formulae to partition $\sigma_{[j,i)}$ into two subsequences such that, at each instant, either propositional letter A or propositional letter B holds. We use letters A and B to "mark" each instant in $\sigma_{[j,i)}$ as shown in Figure 3. The sequence of B's ends in the last non-standard instant of $\sigma_{[j,i)}$. The following $\text{X-TRIO}_{\mathbb{N}}$ formulae formalize the behavior of propositions A and B:

$$\begin{array}{l} A \rightarrow \mathrm{Until}(A \wedge \mathrm{X}_{\mathrm{ns}}(\top), B) \\ B \rightarrow \mathrm{Until}(B, \neg\mathrm{X}_{\mathrm{ns}}(\top)) \\ A \leftrightarrow \neg B. \end{array}$$

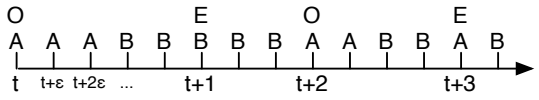


Figure 3: Part of a history σ representing counters.

We use the sequence of A's and B's to represent the two counters: the number of A's starting from a standard number marked with E (resp., O) represents the first (resp., second) counter. We encode the operations *increase/decrease/check if 0*, by manipulating the length of the sequence of A's. For example, the counter associated with E increases its current value if the sequence of A's that starts at the next even standard instant is such that the last A of that sequence is at distance $2 + \varepsilon$ from the last A of the current sequence of A's. This is encoded through the following X-TRIO$_{\mathbb{N}}$ formula (recall that all natural numbers belong to the temporal domain $\overline{\mathbb{N}}_+$):

$$\begin{aligned} E \rightarrow \quad & (A \rightarrow \mathrm{Until}(A, B \wedge \mathrm{Futr}(A \wedge \mathrm{Futr}(B, \varepsilon), 2))) \wedge \\ & (B \rightarrow \mathrm{Futr}(A \wedge \mathrm{Futr}(B, \varepsilon), 2)) \end{aligned}$$

The other cases are omitted for brevity.

The counter is zero when the sequence of A's is empty. In the case of the counter associated with even standard numbers, we can encode this check with the formula $E \wedge B$.

Finally, at the initial instant of the sequence σ, which is an even number, E holds and the corresponding counter value is 0 (i.e., $E \wedge B$ holds at 0).

The halting of the formalized machine is expressed simply as reachability of a final state. Hence, we conclude that the satisfiability problem of X-TRIO$_{\mathbb{N}}$ is undecidable. □

Next, by closely inspecting the essence of the above negative result, we introduce a sufficient condition that makes X-TRIO$_{\mathbb{N}}$ decidable, but still expressive enough for our purposes.

5.1. A decision procedure for X-TRIO

We show the decidability of a fragment of X-TRIO$_{\mathbb{N}}$ by reducing its satisfiability problem to that of PLTLB (LTL with both future and past operators). The transformation is effective and has been implemented in the ℤot satisfiability checker.

PLTLB extends classic LTL [35] with past operators; its syntax (as used in the rest of this paper) is the following:

$$\phi := p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \mathrm{X_L}(\phi) \mid \mathrm{Y_L}(\phi) \mid \phi_1 \,\mathrm{U_L}\, \phi_2 \mid \phi_1 \,\mathrm{S_L}\, \phi_2$$

We use also the standard abbreviations $\mathrm{F_L}(\phi) = \top \,\mathrm{U_L}\, \phi$ and $\mathrm{G_L}(\phi) = \neg \mathrm{F_L}(\neg\phi)$. The semantics of PLTLB is defined over discrete *traces*. A trace is an infinite

word $\pi = \pi(0)\pi(1)\ldots$ over the finite alphabet $\Sigma = 2^{AP}$, where each $\pi(i)$ represents the set of atomic propositions that are true in i. $\models_{\mathrm{L}}$ denotes the satisfiability relation of PLTLB. The definition of $\models_{\mathrm{L}}$ is straightforward if one considers that, for any ϕ, $\mathrm{Y}_{\mathrm{L}}(\phi)$ is false at 0, and that $\phi_1 \,\mathrm{U}_{\mathrm{L}}\, \phi_2$ holds in i in the case in which ϕ_2 holds in i (similarly for S_{L}) [36].

As a first step to encode X-TRIO$_{\mathbb{N}}$ into PLTLB, we restrict histories σ according to the following constraints:

C1. Histories σ are infinite.

C2. If σ_{i+1} is non-standard ($\mathrm{ns}(\sigma_{i+1})$), then $\sigma_{i+1} - \sigma_i = k\varepsilon$ for some $k \in \mathbb{N}_{>0}$.

These constraints are not overly restrictive and they help reducing the number of cases to be considered in the encoding. For example, a finite history σ must be such that, after the last element of the sequence, the state does not change, which can be also represented as an infinite history in which, from a certain point on, all $\beta(\sigma_i)$ are the same. Notice also that if σ_{i+1} is standard ($\mathrm{st}(\sigma_{i+1})$), then between σ_i and σ_{i+1} there is an infinite sequence of nonstandard numbers $\sigma_i + \varepsilon, \sigma_i + 2\varepsilon, \ldots$ such that, for all $k \in \mathbb{N}$, $\beta(\sigma_i + k\varepsilon) = \beta(\sigma_i)$.

To reduce the satisfiability problem of X-TRIO$_{\mathbb{N}}$ (which is in general undecidable) to that of PLTLB (which is decidable), we need to apply further restrictions to the former. As mentioned above, the key to encoding a counting mechanism in X-TRIO$_{\mathbb{N}}$ is to evaluate formulae of the form $\mathrm{Futr}(\phi, 1)$ in non-standard instants. Then, to avoid this, every occurrence of $\mathrm{Futr}(\phi, 1)$ will be intended as an abbreviation for $\mathrm{Futr}(\phi, 1) \wedge \mathrm{NowST}$. Hence, the value of $\mathrm{Futr}(\phi, 1)$ in non-standard instants does not affect the value of the formula. Similar considerations hold for the Past operator.

We also assume that the value of formulae is meaningful only as far as there is some σ_i following the current time. This means "forever" in the case of non-Zeno behaviors; for Zeno behaviors, instead, there are instants $t \in \overline{T}$ such that, for all i, $\sigma_i < t$. This assumption does not cause a real loss of generality since, in the case of Zeno behaviors, any conventional definition of the semantics "outside sigma" can be adopted without altering the essence of system behavior; in our case we state by convention that formulae that are evaluated after one such accumulation point are false.

This can be achieved by considering every subformula ψ of an X-TRIO$_{\mathbb{N}}$ formula ϕ as an abbreviation for $\psi \wedge \mathrm{SomF}(\mathrm{X}_{\mathrm{st}}(\top) \vee \mathrm{X}_{\mathrm{ns}}(\top))$.

To summarize, we indicate by X-TRIO$_{\mathbb{N}}^{\mathrm{dec}}$ the fragment of X-TRIO$_{\mathbb{N}}$ such that:

- Every occurrence of operator $\mathrm{Futr}(\phi, 1)$ in a formula ϕ is intended as an abbreviation for $\mathrm{Futr}(\phi, 1) \wedge \mathrm{NowST}$; similarly for operator $\mathrm{Past}(\phi, 1)$.
- Every subformula ψ in a formula ϕ is an abbreviation for $\psi \wedge \mathrm{SomF}(\mathrm{X}_{\mathrm{st}}(\top) \vee \mathrm{X}_{\mathrm{ns}}(\top))$.
- Interpretations obey conditions C1-C2.

Next, we show that X-TRIO$^{\text{dec}}_{\mathbb{N}}$ is decidable, with a PSPACE-complete decision problem. A first preliminary result is given by the following lemma.

Lemma 5.3. *Given an X-TRIO$^{dec}_{\mathbb{N}}$ formula ϕ, and given two structures $S_1 = \langle \overline{\mathbb{N}}_+, \beta_1, \sigma \rangle$, $S_2 = \langle \overline{\mathbb{N}}_+, \beta_2, \sigma \rangle$ (i.e., which have the same history σ) such that, for all $t \in \overline{\mathbb{N}}_+$ for which there is $i \in \mathbb{N}$ such that $t < \sigma_i$, it holds that $\beta_1(t) = \beta_2(t)$, then $S_1, 0 \models \phi$ iff $S_2, 0 \models \phi$.*

Proof. We show a stronger result, from which Lemma 5.3 descends as a corollary. More precisely, we show that, given any $t \in \overline{\mathbb{N}}_+$, $S_1, t \models \phi$ iff $S_2, t \models \phi$. First, notice that if for each $t \in \overline{\mathbb{N}}_+$ there is a σ_i such that $t < \sigma_i$, then for all $t \in \overline{\mathbb{N}}_+$ it holds that $\beta_1(t) = \beta_2(t)$, hence the desired result. In addition, notice that, in this case, condition $\text{SomF}(\text{X}_{\text{st}}(\top) \vee \text{X}_{\text{ns}}(\top))$ is true for all $t \in \overline{\mathbb{N}}_+$, so the value of ϕ does not depend on it.

Consider now the case in which there are instants t such that, for all i, $\sigma_i < t$. The set of such instants can be shown to have a minimum, which we indicate with $\bar{t}$, such that $\text{st}(\bar{t})$. Then, history σ accumulates at $\bar{t}$, and we separate two cases: $t < \bar{t}$ and $t \geq \bar{t}$. If $t \geq \bar{t}$, $\text{SomF}(\text{X}_{\text{st}}(\top) \vee \text{X}_{\text{ns}}(\top))$ is false, hence for all ϕ both $S_1, t \not\models \phi$ and $S_2, t \not\models \phi$. If, instead, $t < \bar{t}$, the proof is by induction on the structure of ϕ. All cases are proved essentially by applying the definitions of the operators. We outline one of them, the others are similar.

If $\psi = \text{X}_{\text{st}}(\zeta)$, then $S_1, t \models \psi$ iff there is $i \in \mathbb{N}$ such that $\text{st}(\sigma_{i+1})$, $\sigma_i < t \leq \sigma_{i+1}$ and $S_1, \sigma_{i+1} \models \zeta$; by inductive hypothesis this holds iff $S_2, \sigma_{i+1} \models \zeta$, hence the result. □

To introduce our PLTLB-based decision procedure for X-TRIO$^{\text{dec}}_{\mathbb{N}}$, we need a further intermediate result. We show that, in each interval $(t, t+1)$, where it holds that $\text{st}(t)$, there is an instant $\bar{t}$ such that the subformulae of ϕ have the same value for all $t' \in [\bar{t}, t+1)$. In addition, let δ_ϕ (precisely defined below) be the maximum nesting depth of $\text{Past}(\bullet, \varepsilon)$ in ϕ. Then, if σ_i is the greatest element of σ in interval $(t, t+1)$, $\bar{t}$ dists from σ_i a number of non-standard instants that is equal to δ_ϕ (i.e., $\bar{t} = \sigma_i + \delta_\phi \varepsilon$). Otherwise, if there are no elements of σ in $(t, t+1)$, then $\bar{t} = t + \delta_\phi \varepsilon$. More precisely, given a formula ϕ, the nesting δ_ϕ of $\text{Past}(\bullet, \varepsilon)$ operators is defined as follows: $\delta_p = 0$; $\delta_{\psi_1 \wedge \psi_2} = \max(\delta_{\psi_1}, \delta_{\psi_2})$, and similarly for the Until and Since operators; $\delta_{\neg\psi} = \delta_\psi$ (similarly for $\text{Futr}(\psi, 1)$, $\text{Past}(\psi, 1)$, and $\text{Futr}(\psi, \varepsilon)$); finally, $\delta_{\text{Past}(\psi, \varepsilon)} = 1 + \delta_\psi$. Then, we have the following result.

Lemma 5.4. *Let ϕ be an X-TRIO$^{dec}_{\mathbb{N}}$ formula, $S = \langle \overline{\mathbb{N}}_+, \beta, \sigma \rangle$ be a structure, and t be a standard instant (i.e., $\text{st}(t)$ holds) such that there is $j \in \mathbb{N}$ such that $t < \sigma_j$ and $\text{st}(\sigma_j)$ holds. If there is $i \in \mathbb{N}$ such that $t \leq \sigma_i < t + 1 \leq \sigma_{i+1}$, let $\hat{t} = \sigma_i$, otherwise let $\hat{t} = t$. Then, for any two instants $t_1, t_2 \in \overline{\mathbb{N}}_+$ such that $\hat{t} + \delta_\phi \varepsilon < t_1 < t_2 < t + 1$, $S, t_1 \vDash \phi$ iff $S, t_2 \vDash \phi$.*

Proof. Let us indicate by σ_{i+1} the smallest element of σ such that both $t < \sigma_{i+1}$ and $\text{st}(\sigma_{i+1})$ hold. Note that by hypothesis such an element exists, whether $t \leq \sigma_i$ or not. In addition, $\sigma_i < t + 1 \leq \sigma_{i+1}$, otherwise there would be a $j \in \mathbb{N}$ such that $\sigma_j < t + 1 \leq \sigma_{j+1}$ and both $\text{ns}(\sigma_j)$ and $\text{ns}(\sigma_{j+1})$ hold, which

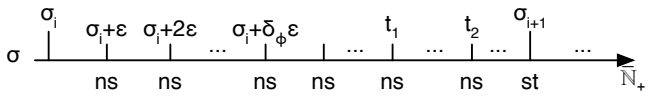


Figure 4: Constancy of formulae after $\sigma_i + \delta_\phi \varepsilon$.

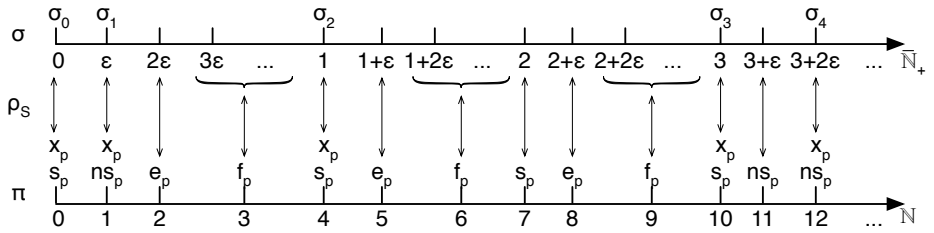


Figure 5: An example of ρ_{S} (with $\delta_\phi = 1$).

is impossible by condition C2; it also holds that $\sigma_i \leq \hat{t} + \delta_\phi \varepsilon < t + 1$. The situation captured by the lemma is exemplified in Figure 4. $\mathrm{ns}(t_1)$ and $\mathrm{ns}(t_2)$ must hold, since $t < t_1 < t_2 < t + 1$. Then, the proof proceeds by induction on the structure of ϕ.

The proof is intuitive for all subformulae in which the main operator is a Boolean connective or a future operator, as their truth depends on instants after $\hat{t} + \delta_\phi \varepsilon$, where the inductive hypothesis is easily applied. Let us detail three significant cases (including one for a future operator).

If $\phi = \mathrm{X}_{\mathrm{st}}(\psi)$, $S, t_1 \vDash \phi$ iff $S, \sigma_{i+1} \vDash \psi$, as $\mathrm{st}(\sigma_{i+1})$. We have also $S, t_2 \vDash \phi$ iff $S, \sigma_{i+1} \vDash \psi$, hence the result.

If $\phi = \mathrm{Y}_{\mathrm{st}}(\psi)$, both $S, t_1 \vDash \phi$ and $S, t_2 \vDash \phi$ hold iff $\mathrm{st}(\sigma_i)$ and $S, \sigma_i \vDash \psi$. Similarly for the case $\phi = \mathrm{Y}_{\mathrm{ns}}(\psi)$.

If $\phi = \mathrm{Past}(\psi, \varepsilon)$, $S, t_1 \models \phi$ iff $S, t_1 - \varepsilon \models \psi$; since $t_1 - \varepsilon > \hat{t} + (\delta_\phi - 1)\varepsilon$ and $\delta_\phi = \delta_\psi + 1$, then $t_2 - \varepsilon > t_1 - \varepsilon > \hat{t} + \delta_\psi \varepsilon$ hence, by inductive hypothesis, we have $S, t_2 - \varepsilon \models \psi$, i.e., $S, t_2 \models \phi$. □

As a consequence of Lemmata 5.3 and 5.4, given the restrictions introduced above, to determine whether an X-TRIO$^{\mathrm{dec}}_{\mathbb{N}}$ formula is satisfiable or not, we only need to focus on the sequence σ, and we can disregard the instants following an accumulation point, if any.

The basic idea of the encoding is, given an X-TRIO$^{\mathrm{dec}}_{\mathbb{N}}$ formula ϕ, to build a corresponding PLTLB formula $\tau(\phi)$ such that each model $S = \langle \overline{\mathbb{N}}_+, \beta, \sigma \rangle$ of ϕ corresponds to a trace π that is a model of $\tau(\phi)$, where each $t \in \overline{\mathbb{N}}_+$ such that there is $\sigma_i > t$ is mapped onto an element $\rho_{\mathrm{S}}(t)$, and $\beta(t) = \pi(\rho_{\mathrm{S}}(t))$, where $\rho_{\mathrm{S}} : \overline{\mathbb{N}}_+ \mapsto \mathbb{N}$ is monotonic. Then, we represent the transition $\sigma_i \longmapsto \sigma_{i+1}$ through PLTLB operators $\mathrm{X_L}$ and $\mathrm{U_L}$.

The encoding "flattens" the history σ over π: it represents each $\beta(\sigma_i)$ through an element of π. Then, to separate the elements of π that represent standard instants from those that represent non-standard ones, we introduce a PLTLB propositional letter s_p such that $s_p \in \pi(i)$ iff i in π corresponds to a standard number. We also need to separate the positions in π that correspond to elements of σ from those that do not. For this reason, we introduce propositional letter x_p such that $x_p \in \pi(i)$ iff i in π corresponds to an element of σ. In addition, we need to introduce "filling" elements in π to represent the (infinite) non-standard instants between t and $t+1$ when $\mathrm{st}(t)$. Lemma 5.4 suggests that the required number of these elements is finite, equal to $\delta_\phi + 1$; in fact, all non-standard instants such that $\hat{t} + \delta_\phi < t' < t + 1$ (where $\hat{t}$ is defined as in Lemma 5.4) are equivalent from the point of view of the truth of subformulae, hence they can be "condensed" in one single element. We mark the first δ_ϕ elements of π following the one corresponding to $\hat{t}$ with proposition e_p, and the element corresponding to all instants $\hat{t} + \delta_\phi < t < \sigma_{i+1}$ with f_p. Figure 5 depicts an example of history σ, and its corresponding trace π.

Then, trace π must obey the following PLTLB constraint, where ns_p is an abbreviation for $\neg s_p \wedge \neg f_p \wedge \neg e_p$:

$$\pi_{\mathrm{constr}} = s_p \wedge x_p \wedge \mathrm{G_L} \left(\begin{array}{l} (e_p \rightarrow \neg f_p \wedge \neg s_p \wedge \neg x_p) \wedge (f_p \rightarrow \neg s_p \wedge \neg x_p) \wedge \\ s_p \rightarrow \left(\begin{array}{l} \mathrm{X_L} \left(ns_p \, \mathrm{U_L} \left(ns_p \wedge x_p \wedge \bigwedge_{k=1}^{\delta_\phi} \mathrm{X_L^k}(e_p) \wedge \mathrm{X_L^{\delta_\phi+1}}(f_p \wedge \mathrm{X_L}(s_p)) \right) \right) \\ \vee \\ \mathrm{X_L}(\mathrm{G_L}(ns_p)) \end{array} \right) \wedge \\ \neg x_p \rightarrow \left(\bigwedge_{q \in AP} q \leftrightarrow \mathrm{Y_L}(q) \right) \wedge \\ x_p \rightarrow \mathrm{X_L}(\neg x_p \, \mathrm{U_L} \, (x_p \wedge s_p)) \vee \mathrm{X_L}(ns_p \, \mathrm{U_L} \, (x_p \wedge ns_p)) \end{array} \right) \tag{14}$$

PLTLB Formula (14) imposes, respectively, that

(i) s_p and x_p hold in $\pi(0)$ (i.e., the first instant corresponds to a standard one and it is in σ);

(ii) e_p, f_p and s_p are mutually exclusive; in addition, elements marked with e_p or f_p are not in σ;

(iii) each element marked s_p is followed either by an infinity of ns_p elements or by a finite number of ns_p elements, until there is an element marked with both ns_p and x_p (i.e., a nonstandard element of σ), immediately followed by a sequence of exactly δ_ϕ e_p elements followed, in turn, by a f_p element, which is in turn followed by a s_p element;

(iv) if $x_p \notin \pi(i+1)$, then all propositions that hold in $\pi(i+1)$ also hold in $\pi(i)$;

$$
\begin{aligned}
&\tau(p) = p\\
&\tau(\neg\phi) = \neg\tau(\phi)\\
&\tau(\phi_1 \wedge \phi_2) = \tau(\phi_1) \wedge \tau(\phi_2)\\
&\tau(\mathrm{X}_{\mathrm{ns}}(\phi)) = \mathrm{X}_{\mathrm{L}}(\neg x_p \,\mathrm{U}_{\mathrm{L}}\, (ns_p \wedge x_p \wedge \tau(\phi)))\\
&\tau(\mathrm{X}_{\mathrm{st}}(\phi)) = \mathrm{X}_{\mathrm{L}}(\neg x_p \,\mathrm{U}_{\mathrm{L}}\, (s_p \wedge x_p \wedge \tau(\phi)))\\
&\tau(\mathrm{Y}_{\mathrm{ns}}(\phi)) = \mathrm{Y}_{\mathrm{L}}(\neg x_p \,\mathrm{S}_{\mathrm{L}}\, (ns_p \wedge x_p \wedge \tau(\phi)))\\
&\tau(\mathrm{Y}_{\mathrm{st}}(\phi)) = \mathrm{Y}_{\mathrm{L}}(\neg x_p \,\mathrm{S}_{\mathrm{L}}\, (s_p \wedge x_p \wedge \tau(\phi)))\\
&\tau(\mathrm{Futr}(\phi, 0)) = \tau(\mathrm{Past}(\phi, 0)) = \tau(\phi)\\
&\tau(\mathrm{Futr}(\phi, \varepsilon)) = (\neg f_p \wedge \mathrm{X}_{\mathrm{L}}(\tau(\phi))) \vee (f_p \wedge \tau(\phi))\\
&\tau(\mathrm{Past}(\phi, \varepsilon)) = \ \neg s_p \ \wedge \ ((\neg f_p \wedge \mathrm{Y}_{\mathrm{L}}(\tau(\phi))) \vee (f_p \wedge \tau(\phi)))\\
&\tau(\mathrm{Futr}(\phi, 1)) = s_p \wedge \mathrm{X}_{\mathrm{L}}(\neg s_p \,\mathrm{U}_{\mathrm{L}}\, (s_p \wedge \tau(\phi)))\\
&\tau(\mathrm{Past}(\phi, 1)) = s_p \wedge \mathrm{Y}_{\mathrm{L}}(\neg s_p \,\mathrm{S}_{\mathrm{L}}\, (s_p \wedge \tau(\phi)))\\
&\tau(\mathrm{Until}(\phi, \psi)) = \tau(\phi) \wedge ((f_p \wedge \tau(\psi)) \ \vee \ \mathrm{X}_{\mathrm{L}}(\tau(\phi) \,\mathrm{U}_{\mathrm{L}}\, \tau(\psi)))\\
&\tau(\mathrm{Since}(\phi, \psi)) = \tau(\phi) \wedge ((f_p \wedge \tau(\psi)) \ \vee \ \mathrm{Y}_{\mathrm{L}}(\tau(\phi) \,\mathrm{S}_{\mathrm{L}}\, (\tau(\psi) \wedge (f_p \rightarrow \tau(\phi)))))
\end{aligned}
$$

Figure 6: Translation schema τ.

(v) if $x_p \in \pi(i)$, then either the next element marked by x_p is also marked by s_p (i.e., it corresponds to a standard instant), or all elements until then are marked by ns_p (i.e., they are nonstandard).

In other words, traces π have one of the following two forms (up to a homomorphism erasing x_p): $s_p(ns_p^*\ e_p^{\delta_\phi}\ f_p\ s_p)^\omega$ or $s_p(ns_p^*\ e_p^{\delta_\phi}\ f_p\ s_p)^*(ns_p)^\omega$. In addition, x_p can appear in π only in conjunction with s_p or ns_p, and there cannot be a sequence of the form $x_p(\neg x_p)^* s_p(\neg x_p)^*\{x_p, ns_p\}$.

Transformation τ of Figure 6 takes an X-TRIO$^{\mathrm{dec}}_{\mathbb{N}}$ formula ϕ and produces an equisatisfiable PLTLB formula ϕ_L.

Given a structure $S = \langle \overline{\mathbb{N}}_+, \beta, \sigma \rangle$, for all $t \in \overline{\mathbb{N}}_+$ such that there is $\sigma_i > t$, we define function $\rho_{\mathrm{S}} : \overline{\mathbb{N}}_+ \mapsto \mathbb{N}$ as follows (see Figure 5 for a graphical representation):

(i) $\rho_{\mathrm{S}}(0) = 0$ ($\sigma_0 = 0$);

(ii) if $\mathrm{ns}(t)$ and $\exists i, k \in \mathbb{N}$ s.t. $t = \sigma_i + k\varepsilon$ and $\sigma_{i+1} - \sigma_i = k'\varepsilon$ for some $k' \in \mathbb{N}_{>0}$, with $k < k'$, then $\rho_{\mathrm{S}}(t) = \rho_{\mathrm{S}}(\sigma_i) + k$;

(iii) if $\mathrm{ns}(t)$ and $\exists i, k \in \mathbb{N}$ s.t. $t = \sigma_i + k\varepsilon$ and there is no $k' \in \mathbb{N}$ s.t. $\sigma_{i+1} - \sigma_i = k'\varepsilon$ (i.e., the distance $\sigma_{i+1} - \sigma_i$ is not infinitesimal), then if $k \leq \delta_\phi$ we have that $\rho_{\mathrm{S}}(t) = \rho_{\mathrm{S}}(\sigma_i) + k$, otherwise $\rho_{\mathrm{S}}(t) = \rho_{\mathrm{S}}(\sigma_i) + \delta_\phi + 1$;

(iv) if $\mathrm{ns}(t)$ and $\nexists i, k' \in \mathbb{N}$ s.t. $t = \sigma_i + k'\varepsilon$ (i.e., the distance of t from the previous element of σ is noninfinitesimal), and $t = v + k\varepsilon$ (with $\mathrm{st}(v)$) for

some $k \in \mathbb{N}$, then if $k \leq \delta_\phi$ we have that $\rho_{\mathrm{S}}(t) = \rho_{\mathrm{S}}(v) + k$, otherwise $\rho_{\mathrm{S}}(t) = \rho_{\mathrm{S}}(v) + \delta_\phi + 1$;

(v) if $t > 0$ and $\mathrm{st}(t)$ and $\exists i$ s.t. $0 < t - \sigma_i \leq 1$ and $t \leq \sigma_{i+1}$ (i.e., σ_i is the last element of σ before t, and it is greater than or equal to $t-1$), then $\rho_{\mathrm{S}}(t) = \rho_{\mathrm{S}}(\sigma_i) + \delta_\phi + 2$;

(vi) if $t > 0$ and $\mathrm{st}(t)$ and $\nexists i$ s.t. $0 < t - \sigma_i \leq 1$ (i.e., the last element of σ before t is less than $t-1$), then $\rho_{\mathrm{S}}(t) = \rho_{\mathrm{S}}(t-1) + \delta_\phi + 2$.

Notice that when rules (iii) and (iv) are applied to a nonstandard instant $t < t' < t+1$ that belongs to interval $(\hat{t} + \delta_\phi \varepsilon, t+1)$ (where $\hat{t}$ is defined as in Lemma 5.4), t' is mapped onto a "filling" element f_p in π. As a consequence, rules (v) and (vi) map a standard instant onto $\rho_{\mathrm{S}}(\hat{t}) + \delta_\phi + 2$, i.e., to the element in π that follows a "filling" one. Then, for any pair t, t' such that $t < t'$, we have $\rho_{\mathrm{S}}(t) \leq \rho_{\mathrm{S}}(t')$. Finally, we can prove the main result, which implies the decidability of X-TRIO$^{\mathrm{dec}}_{\mathbb{N}}$.

Theorem 5.5. *Given an X-TRIO$^{dec}_{\mathbb{N}}$ formula ϕ, there is a structure $S = \langle \overline{\mathbb{N}}_+, \beta, \sigma \rangle$ such that $S, 0 \vDash \phi$ iff there exists a trace π such that $\pi \vDash_{\mathrm{L}} \tau(\phi) \wedge \pi_{\mathrm{constr}}$.*

Before delving into the proof of Theorem 5.5, let us present an example of X-TRIO$^{\mathrm{dec}}_{\mathbb{N}}$ formula, its equisatisfiable translation, and an example of model for the translated formula.

Consider the following formula (to be evaluated in the origin), which states that, starting with the first instant, a sequence of micro-steps occurs in which p and $\neg p$ alternate, until a macro-step is taken:

$$EX = \mathrm{Until}(p \leftrightarrow \mathrm{X}_{\mathrm{ns}}(\neg p)\,, \mathrm{X}_{\mathrm{st}}(\top))$$

The translation $\tau(EX)$ is the following:

$$\tau(EX) = \left(\begin{array}{l} p \leftrightarrow \mathrm{X_L}(\neg x_p \,\mathrm{U_L}\, (ns_p \wedge x_p \wedge \neg p)) \\ \wedge \\ \left(\begin{array}{l} f_p \wedge \mathrm{X_L}(\neg x_p \,\mathrm{U_L}\, (s_p \wedge x_p \wedge \top)) \\ \vee \\ \mathrm{X_L} \left(\begin{array}{l} p \leftrightarrow \mathrm{X_L}(\neg x_p \,\mathrm{U_L}\, (ns_p \wedge x_p \wedge \neg p)) \\ \mathrm{U_L} \\ \mathrm{X_L}(\neg x_p \,\mathrm{U_L}\, (s_p \wedge x_p \wedge \top)) \end{array} \right) \end{array} \right) \end{array} \right) \wedge \pi_{\mathrm{constr}}$$

Notice that, for formula EX, we have that $\delta_{EX} = 0$, since the formula does not include any instance of operator $\mathrm{Past}(\bullet, \varepsilon)$. Figure 7 shows an example of model for formula $\tau(EX)$.[3]

[3]Notice that $\neg p \rightarrow \neg \mathrm{X}_{\mathrm{ns}}(\neg p)$ implies that, if $\neg p$ holds (as in position 3 in Figure 7) and the next time instant in σ is nonstandard, then p holds there.

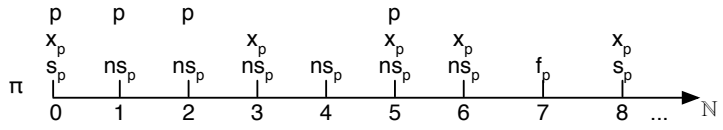


Figure 7: An example of model for formula $\tau(EX)$.

Proof of Theorem 5.5. Suppose we have a structure $S = \langle \overline{\mathbb{N}}_+, \beta, \sigma \rangle$. The corresponding infinite word π is built from S as follows: for each $p \in AP$ and $t \in \overline{\mathbb{N}}_+$ such that there is $\sigma_i > t$ (hence $\rho_S(t)$ is defined), $p \in \beta(t)$ iff $p \in \pi(\rho_S(t))$. In addition, $x_p \in \pi(\rho_S(t))$ iff there is $i \in \mathbb{N}$ s.t. $t = \sigma_i$; $s_p \in \pi(\rho_S(t))$ iff $\mathrm{st}(t)$ holds; $f_p \in \pi(\rho_S(t))$ iff there are i, t' s.t. $\mathrm{st}(t')$ and either $t' < \sigma_i + \delta_\phi \varepsilon < t < t' + 1 \leq \sigma_{i+1}$, or $\sigma_i < t' \leq t' + \delta_\phi \varepsilon < t < t' + 1 \leq \sigma_{i+1}$; and $e_p \in \rho_S(t)$ iff there are i, t' s.t. $\mathrm{st}(t')$ and either $t' \leq \sigma_i < t \leq \sigma_i + \delta_\phi \varepsilon < t' + 1 \leq \sigma_{i+1}$, or $\sigma_i < t' < t \leq t' + \delta_\phi \varepsilon < t' + 1 \leq \sigma_{i+1}$. It can be shown that trace π built in this way satisfies Formula (14).

Dually, if π is such that $\pi \vDash_L \tau_e(\phi) \wedge \pi_{\mathrm{constr}}$, structure $S = \langle \overline{\mathbb{N}}_+, \beta, \sigma \rangle$ is obtained in the following way. Given an $l \in \mathbb{N}$, if $x_p \in \pi(l)$, $\langle l_0 \dots l_v \rangle$ (where $\forall j \in [0, v)$ it holds that $l_j < l_{j+1}$) are all the elements of the infinite word π such that $l_v \leq l$ and $\forall j \in [0, v] : s_p \in \pi(l_j)$ (i.e., they are all the elements that correspond to standard instants in π preceding element l), $\langle b_0 \dots b_h \rangle$ (where $\forall j \in [0, h)$, it holds that $b_j < b_{j+1}$) are all the elements of the infinite word π such that $b_h = l$ and $\forall j \in [0, h] : x_p \in \pi(b_j)$ (i.e., they are all the elements that correspond to instants of σ in π preceding element l), then $\sigma_h = v + k\varepsilon$. This entails that, since σ_l is a standard number iff it is of the form $v + 0\varepsilon$, we have that $\mathrm{st}(\sigma_l)$ iff $s_p \in \pi(l)$. In addition, by (14), $s_p \in \pi(0)$, hence $\sigma_0 = 0$ as expected. This defines the points of the history σ of S, which in turn defines ρ_S. β is defined as follows: for each t such that there is $\sigma_i > t$, for each $p \in AP$, $p \in \beta(t)$ iff $p \in \pi(\rho_S(t))$. If, instead, there is no $\sigma_i > t$, we can choose the value of $\beta(t)$ arbitrarily, as, by Lemma 5.3, it does not affect the truth of ϕ in S. Then, in this case we have $p \notin \beta(t)$ for all $p \in AP$.

We prove the equisatisfiability of ϕ and $\tau(\phi) \wedge \pi_{\mathrm{constr}}$ by induction on the structure of ϕ. By Lemma 5.3, if there is an accumulation point $\bar{t}$, instants $t \geq \bar{t}$ do not affect the satisfiability of ϕ, hence we need only analyze instants $t < \bar{t}$. Then, we show that, for all $t < \bar{t}$ (where $\bar{t} = +\infty$, if there is no such an accumulation point), $S, t \models \psi$ (where ψ is a subformula of ϕ) holds iff $\pi, \rho_S(t) \vDash_L \tau(\psi)$ holds.

If $\psi = p$, then $S, t \models p$ iff $p \in \beta(t)$, which holds, by construction, iff $p \in \pi(\rho_S(t))$, i.e., iff $\pi, \rho_S(t) \vDash_L p$, and $\tau(p) = p$, hence the result. The cases $\psi = \neg\zeta$ and $\psi = \psi_1 \wedge \psi_2$ are immediate.

If $\psi = \mathrm{X}_{\mathrm{ns}}(\zeta)$, $S, t \models \psi$ iff it holds that $\sigma_i \leq t < \sigma_{i+1}$, $\mathrm{ns}(\sigma_{i+1})$ and $S, \sigma_{i+1} \models \zeta$; by inductive hypothesis, this holds iff $\pi, \rho_S(\sigma_{i+1}) \vDash_L \tau(\zeta)$, and by construction $s_p, f_p, e_p \notin \pi(\rho_S(\sigma_{i+1}))$ and $x_p \in \pi(\rho_S(\sigma_{i+1}))$, hence $\pi, \rho_S(\sigma_{i+1}) \vDash_L \tau(\zeta) \wedge ns_p \wedge x_p$; in addition, by construction $\rho_S(t) < \rho_S(\sigma_{i+1})$, and for all j such that $\rho_S(t) <$

$j < \rho_S(\sigma_{i+1})$ we have that $x_p \notin \pi(j)$, hence $\pi, \rho_S(t) \vDash_L X_L(\neg x_p \, U_L \, (\tau(\zeta) \wedge ns_p \wedge x_p))$. The cases for the X_{st}, Y_{ns} and Y_{st} operators are similar.

$\mathrm{Futr}(\psi, 0)$ (and $\mathrm{Past}(\psi, 0)$) is equivalent to ψ, hence this case is trivial.

If $\psi = \mathrm{Futr}(\zeta, \varepsilon)$, $S, t \models \psi$ iff $S, t + \varepsilon \models \zeta$, which, by inductive hypothesis, holds iff $\pi, \rho_S(t + \varepsilon) \vDash_L \tau(\zeta)$. We separate two cases: there exist i, t' s.t. $\mathrm{st}(t')$ and (a) $t' \leq \sigma_i \leq t < t' + 1 \leq \sigma_{i+1}$, or (b) $\sigma_i < t' \leq t < t' + 1 \leq \sigma_{i+1}$. Both cases are further split in two parts. Since cases (a) and (b) are very similar, we show only the first one. Consider, then, the two further subcases of case (a): $\sigma_i \leq t \leq \sigma_i + \delta_\phi \varepsilon < t' + 1$ and $\sigma_i + \delta_\phi \varepsilon < t < t' + 1$ (notice that $\mathrm{st}(t' + 1)$). In the first case, $f_p \notin \rho_S(t)$, $\rho_S(t + \varepsilon) = \rho_S(t) + 1$, hence $\pi, \rho_S(t) \vDash_L \neg f_p \wedge X_L(\tau(\zeta))$. In the second case, $f_p \in \rho_S(t)$, and by Lemma 5.4 $S, t + \varepsilon \models \zeta$ iff $S, t \models \zeta$, which in turn holds iff $\pi, \rho_S(t) \vDash_L \tau(\zeta)$, hence $\pi, \rho_S(t) \vDash_L f_p \wedge \tau(\zeta)$. Finally, $\pi, \rho_S(t) \vDash_L \neg f_p \wedge X_L(\neg s_p \wedge \tau(\zeta)) \vee f_p \wedge \tau(\zeta)$, i.e., $\pi, \rho_S(t) \vDash_L \tau(\psi)$.

The case $\psi = \mathrm{Past}(\zeta, \varepsilon)$ is similar to the previous one, with the addition that $S, t \models \psi$ only if $t - \varepsilon \in \overline{\mathbb{N}}_+$ (i.e., $\mathrm{ns}(t)$) which, by inductive hypothesis, holds iff $s_p \notin \pi(\rho_S(t))$, i.e., $\pi, \rho_S(t) \vDash_L \neg s_p$.

If $\psi = \mathrm{Futr}(\zeta, 1)$, $S, t \models \psi$ iff $\mathrm{st}(t)$, $t + 1 < \bar{t}$, and $S, t + 1 \models \zeta$. It holds that $\mathrm{st}(t)$ iff $s_p \in \pi(\rho_S(t))$. Also, $t+1 < \bar{t}$ holds iff there is $\sigma_i > t+1$, i.e., iff $\rho_S(t+1)$ is defined, and $s_p \in \pi(\rho_S(t+1))$. In addition, when $\rho_S(t+1)$ is defined, $S, t+1 \models \zeta$ iff $\pi, \rho_S(t + 1) \vDash_L \tau(\zeta)$, by inductive hypothesis. As $\rho_S(t + 1) > \rho_S(t)$, and there are no standard instants in between, then $\pi, \rho_S(t) \vDash_L s_p \wedge X_L(\neg s_p \, U_L \, (s_p \wedge \tau(\zeta)))$, i.e., $\pi, \rho_S(t) \vDash_L \tau(\psi)$. The case $\psi = \mathrm{Past}(\zeta, 1)$ is similar.

If $\psi = \mathrm{Since}(\psi_1, \psi_2)$, then $S, t \models \psi$ iff there is $0 \leq t' < t$ s.t. $S, t' \models \psi_2$ and for all $t' < t'' \leq t$, $S, t'' \models \psi_1$ holds. By inductive hypothesis we have $\pi, \rho_S(t) \vDash_L \tau(\psi_1)$ and also $\pi, \rho_S(t') \vDash_L \tau(\psi_2)$. As in the proof for the encoding of $\mathrm{Futr}(\zeta, \varepsilon)$, we need to separate several cases. (a) If there are $i, \bar{t}$ such that $\mathrm{st}(\bar{t})$ and $\sigma_i < \bar{t} \leq \bar{t} + \delta_\phi \varepsilon < t' < t < \bar{t} + 1 \leq \sigma_{i+1}$ hold, then, by inductive hypothesis, $\pi, \rho_S(t) \vDash_L \tau(\psi_2)$, and $f_p \in \pi(\rho_S(t))$, hence $\pi, \rho_S(t) \vDash_L \tau(\psi_1) \wedge \tau(\psi_2) \wedge f_p$. The same is true if it holds that $\bar{t} \leq \sigma_i \leq \sigma_i + \delta_\phi \varepsilon < t' < t < \bar{t} + 1 \leq \sigma_{i+1}$. Otherwise, (b) if there are $i, \bar{t}$ such that $\mathrm{st}(\bar{t})$, $\sigma_i < \bar{t} \leq t' \leq \bar{t} + \delta_\phi \varepsilon < \bar{t} + 1 \leq \sigma_{i+1}$ and $\bar{t} + 1 \leq t$ hold, then for all $t' < t'' \leq t$ it also holds that $\rho_S(t') < \rho_S(t'') \leq \rho_S(t)$, hence, by inductive hypothesis, $f_p \notin \pi(\rho_S(t'))$ and for all $\rho_S(t') < \rho_S(t'') \leq \rho_S(t)$ it holds that $\pi, \rho_S(t'') \vDash_L \tau(\psi_1)$, hence $\pi, \rho_S(t) \vDash_L \tau(\psi_1) \wedge Y_L(\tau(\psi_1) \, S_L \, (\tau(\psi_2) \wedge \neg f_p))$. The same is true if it holds that $\bar{t} \leq \sigma_i \leq t' \leq \sigma_i + \delta_\phi \varepsilon < \bar{t} + 1 \leq \sigma_{i+1}$ and also $\bar{t} + 1 \leq t$. Finally, (c) if $i, \bar{t}$ are such that $\mathrm{st}(\bar{t})$, $\sigma_i < \bar{t} \leq \bar{t} + \delta_\phi \varepsilon < t' < \bar{t} + 1 \leq \sigma_{i+1}$, and $\bar{t} + 1 \leq t$ hold, then there are some t'' s.t. $t' < t'' < \bar{t} + 1$ hence, by Lemma 5.4, it holds that $S, t' \models \psi_1$, and also $\rho_S(t') = \rho_S(t'')$ and $f_p \in \pi(\rho_S(t'))$. Then, by inductive hypothesis, we have $\pi, \rho_S(t') \vDash_L \tau(\psi_2)$ and for all $\rho_S(t') \leq \rho_S(t'') \leq \rho_S(t)$ it holds that $\pi, \rho_S(t'') \vDash_L \tau(\psi_1)$. Then, $\pi, \rho_S(t) \vDash_L \tau(\psi_1) \wedge Y_L(\tau(\psi_1) \, S_L \, (f_p \wedge \tau(\psi_1) \wedge \tau(\psi_2)))$. The same is true if we have that $\bar{t} \leq \sigma_i \leq \sigma_i + \delta_\phi \varepsilon < t' < \bar{t} + 1 \leq \sigma_{i+1}$, and $\bar{t} + 1 \leq t$. Overall, we have that $\pi, \rho_S(t) \vDash_L \tau(\psi)$. The case for the Until operator is similar. □

Finally, from translation schema τ and Theorem 5.5, we can prove the following complexity result.

Theorem 5.6. *The satisfiability problem of X-TRIO$^{dec}_{\mathbb{N}}$ is PSPACE-complete.*

X-TRIO variant & domain	Can express "now is standard"?	Decidable?
Future-only operators, over $\overline{\mathbb{R}}_+$	No (Lem. 4.3)	No (Th. 4.8)
Future and past operators, over $\overline{\mathbb{R}}_+$	Yes (Th. 4.1)	No (Th. 4.7)
Future and past operators, over $\overline{\mathbb{R}}_{\geq 0}$	No (Th. 4.4)	No (Th. 4.9)
Future and past operators, over $\overline{\mathbb{N}}_+$	Yes (Sect. 5)	No (Th. 5.2)
Future and past, over $\overline{\mathbb{N}}_+$, $\mathrm{Futr}(\bullet, 1)$ and $\mathrm{Past}(\bullet, 1)$ false in t if $\mathrm{ns}(t)$	Yes (Sect. 5)	Yes (Th. 5.5)

Table 2: Summary of results for propositional X-TRIO.

Proof. To show the PSPACE-hardness of the satisfiability problem for X-TRIO$^{\mathrm{dec}}_{\mathbb{N}}$ we reduce the satisfiability problem of PLTLB, which is PSPACE-complete [36], to that of X-TRIO$^{\mathrm{dec}}_{\mathbb{N}}$. To achieve this, given a PLTLB formula ϕ_L, we can build a corresponding X-TRIO$^{\mathrm{dec}}_{\mathbb{N}}$ formula simply by applying the following transformation: $\phi \,\mathrm{U_L}\, \psi \mapsto \psi \vee \mathrm{Until}(\phi, \psi)$, $\phi \,\mathrm{S_L}\, \psi \mapsto \psi \vee \mathrm{Since}(\phi, \psi)$, $\mathrm{X_L}(\phi) \mapsto \mathrm{Futr}(\phi, 1)$, $\mathrm{Y_L}(\phi) \mapsto \mathrm{Past}(\phi, 1)$, and by including the constraint $\mathrm{AlwF}(\neg \mathrm{X_{ns}}(\top))$.

To show the PSPACE-completeness, it is enough to note that, given an X-TRIO$^{\mathrm{dec}}_{\mathbb{N}}$ formula ϕ, transformation τ produces an equisatisfiable PLTLB formula ϕ_L, whose size is polynomial in the size of ϕ. □

Table 2 summarizes the expressiveness and decidability results of Section 4 and Section 5.

Remark

We emphasize that, despite the limitations introduced to achieve decidability, X-TRIO$^{\mathrm{dec}}_{\mathbb{N}}$ still retains some distinguishing features of the general version X-TRIO that allow us to escape a few typical traps of traditional temporal logics based on 0-time transitions. For instance, in Section 2 we observed that the MTL formula $\mathrm{G}_{[0,0]}\mathrm{X}_{[0,0]}\mathrm{F}_{[0,0]}(a)$, which expresses a "liveness at 0-time" property, necessarily implies a Zeno behavior; here we show that, on the contrary, in X-TRIO$^{\mathrm{dec}}_{\mathbb{N}}$ we can express a "liveness at infinitesimal time" property which does not necessarily imply zenoness.

We first introduce, as an abbreviation and by exploiting predicate NowST, a variant of the Until operator that only considers instants at infinitesimal distance from the current one; we call this variant $\mathrm{Until_{inf}}$.

$$\mathrm{Until_{inf}}(\phi, \psi) \stackrel{\mathrm{def}}{=} \psi \vee (\phi \wedge \mathrm{Dist}(\mathrm{Until}(\neg \mathrm{NowST} \wedge \phi, \neg \mathrm{NowST} \wedge \psi)\,, \varepsilon)). \quad (15)$$

From the $\mathrm{Until_{inf}}$ operator, we can define operators $\mathrm{SomF_{inf}}$ and $\mathrm{AlwF_{inf}}$ as usual as the following abbreviations:

$$\mathrm{SomF_{inf}}(\phi) \stackrel{\mathrm{def}}{=} \mathrm{Until_{inf}}(\top, \phi) \quad (16)$$

$$\mathrm{AlwF_{inf}}(\phi) \stackrel{\mathrm{def}}{=} \neg \mathrm{SomF_{inf}}(\neg \phi)\,. \quad (17)$$

Finally, we can express the property of "liveness at infinitesimal distance" through the following formula:

$$\mathrm{AlwF}_{\mathrm{inf}}(\mathrm{SomF}_{\mathrm{inf}}(\phi)) \tag{18}$$

which is satisfied both by infinite sequences of infinitesimal steps – a Zeno behavior – but also by sequences of any (unbounded) finite number of micro-steps eventually followed by a macro one: in fact, after the last micro-step where ϕ holds, ϕ remains true in all following infinitesimals preceding the macro-step. Notice also that Formula (18) has a natural meaning that is easily understandable even by an "end user" who may be totally unaware of the technicalities that formalize its semantics.

6. Exploiting X-TRIO to formalize languages with zero-time transitions

Often logic formalisms are exploited in a *dual language approach* by pairing them with operational models that describe the structure and the behavior of systems to be analyzed. The semantics of the operational model is formalized through a suitable *axiomatization* in terms of the logic language; then its properties are specified and proved (or disproved) as theorems of the logic formalism. Model checking in its many formulations offered in the literature is the most widely known example of automatically supported application of this approach, where various forms of state machines are paired with (and axiomatized in terms of) various forms of temporal logics, but, starting with the classical Hoare's method to prove program correctness, the dual language approach has been widely applied also by exploiting Turing-complete, and therefore undecidable formalisms.

X-TRIO has been designed both to be used as a "stand alone" formalism and to be paired with any operational one within the dual language scheme, whenever the systems to be modeled and analyzed progresses stepwise by means of micro- and macro-steps.

To show the wide applicability of our language – and, in general, of the approach based on infinitesimals – to various operational formalisms based on micro- and macro-steps, first, in Section 6.1, we sketch how to exploit X-TRIO to formalize the semantics and to prove properties of timed Petri nets (TPN) allowing for transitions that may fire in 0-time. Then, in Section 6.2 we briefly report on the use of X-TRIO to give formal semantics to the Stateflow notation [37], including the composition of several cooperating modules [2], and on the application of a tool supporting the automatic verification of X-TRIO properties to a case study in the field of Flexible Manufacturing Systems (FMS).

6.1. X-TRIO for the semantics of timed Petri nets

The original version of TRIO has already been used to support the proof of TPN properties [38]; we also already exploited NSA to deal with 0-time transition firing [20]. In this section, instead, we apply all the novelties of

X-TRIO by assimilating 0-time firings to micro-steps and non-0-time ones to macro-steps, still maintaining the typical asynchronous nature of the model, which is sharply different from the synchronous one typical of formalisms rooted in the finite state machines such as Statechart-like ones.

Among the many versions of TPNs we refer here to the one introduced by Merlin and Farber [39]. In their definition, each transition is associated with both a lowerbound L and an upperbound U, such that and $0 \leq L \leq U \leq \infty$, with $L \neq \infty$. The intuitive semantics is that a transition, once enabled by the presence of tokens in its input places, can fire only at a time enclosed between the two bounds; in addition, it *must* fire when the upperbound expires, unless a conflicting transition fired earlier. A detailed analysis of the technical intricacies of this informal semantics is out of the scope of this paper and can be found in [38, 5]; for the purpose of illustrating the use of X-TRIO, we introduce the following simplifying assumptions on the nets' topology and semantics, which still cover many practical cases:

- 1-bounded PNs, i.e., nets for which it is known a priori that at most one token can be in any place during net's evolution.
- Nonnegative integer lower- and upper-bounds attached to all transitions.
- Nets without input and output conflicts on places, i.e., with only one ingoing and one outgoing arc for every place; we also forbid that the same transition is both input and output to the same place: This will allow us to define the X-TRIO formalization of the nets by referring only to the fragments in Figure 8.

We remark that only the first two assumptions are necessary to allow us to exploit X-TRIO$^{\text{dec}}_{\mathbb{N}}$; the third one, instead, has been introduced only to keep the example as simple as possible while still showing the essential aspects of the application; furthermore, at the end of this subsection we provide a few hints on exploiting more powerful, yet undecidable, versions of X-TRIO.

We now provide an X-TRIO$^{\text{dec}}_{\mathbb{N}}$ formalization of such TPNs. Thus the time domain is $\overline{\mathbb{N}}_+$, i.e., the set of values of type $v+k\varepsilon$, ε being a constant positive infinitesimal, v and k being nonnegative integers; remember also that formulae of type $\text{Futr}(\phi, 1)$ must be intended as an abbreviation for $\text{Futr}(\phi, 1) \wedge \text{NowST}$ (similarly for $\text{Past}(\phi, 1)$ and all formulae built upon them, such as $\text{Lasted}_{[]}(\phi, k)$).

In essence, our formalization assumes that transition firings in 0-time take the infinitesimal constant time ε to occur; firings at non-null time instead occur at standard integer values. Thus, all and only the firings labeled as 0-time occur at nonstandard times. This implies a kind of "resynchronization" after a sequence of micro-steps (which cannot be unbounded if 0-time loops are forbidden). We will see that this leads to a different semantics than the one adopted in [20], though both can be reasonably adopted as a formalization of TPN behavior.

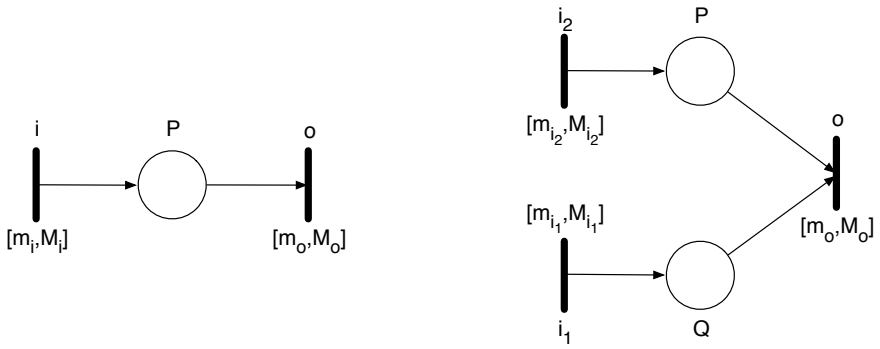


Figure 8: Fragments of timed Petri nets.

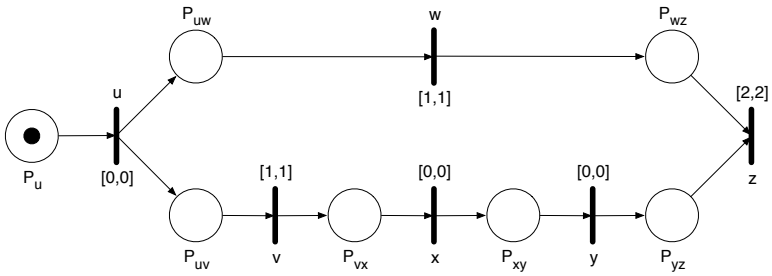


Figure 9: A richer fragment of Petri net.

X-TRIO$^{dec}_{\mathbb{N}}$ axioms for TPNs. We refer to the sample fragments of Figure 8 to define the axioms specifying TPNs semantics. The axioms for fragments of type (a) are easily generalized to the case in which transition i has more than one place in its postset, like transition u in the net of Figure 9. Following [38], for each place p in the net, we introduce predicate $\texttt{Marked}_\texttt{p}$, which holds if and only if the place is marked with a token; similarly, for each transition t we introduce a predicate $\texttt{fire}_\texttt{t}$, which holds exactly in those instants at which the transition fires. With respect to [38] and other weakly monotonic semantics of TPNs, replacing 0-time firing with ε firing avoids counterintuitive behaviors where, e.g., several subsequent transitions such as v, x, y of Figure 9 fire simultaneously, whereas in the "nature" of Petri nets only concurrent transitions can fire simultaneously.

Formulae (19)-(20) state that for any place P that is in the postset of any transition i and in the preset of transition o, P is marked starting from the

instant when i fires (not included), up to the instant when o fires (included) [4]

$$\texttt{fire}_\texttt{i} \rightarrow \text{Until}_{(]}(\texttt{Marked}_\texttt{P}, \texttt{fire}_\texttt{o}) \tag{19}$$

$$\texttt{Marked}_\texttt{P} \rightarrow \text{Since}_{[)}(\neg\texttt{fire}_\texttt{o}, \texttt{fire}_\texttt{i}) \tag{20}$$

where $\text{Until}_{(]}(A, B)$ is an abbreviation for $\text{Futr}(\text{Until}(A, A \wedge B), \varepsilon)$ and, symmetrically, $\text{Since}_{[)}(A, B)$ is an abbreviation for $\text{Past}(\text{Since}(A, A \wedge B), \varepsilon)$. This kind of axiom applies to the places of fragments of both types (a) and (b) of Figure 8. Notice that to avoid inconsistencies it is essential that the same transition is not both input and output for the same place. Such an hypothesis, however, is not restrictive, as one can split a "self-loop" transition into a sequence of two ones: one empties the place and the other one fills it back, the latter being (almost) 0-time.

The following formulae state that any transition o fires at standard time instants, unless it fires in 0-time. Thus, when $m_o > 0$, for fragments of both types we have

$$\texttt{fire}_\texttt{o} \rightarrow \text{NowST}. \tag{21}$$

When $m_o = 0$, instead, we separate the cases (a) and (b) in, respectively,

$$(\texttt{fire}_\texttt{o} \wedge \neg\text{NowST}) \rightarrow \texttt{Marked}_\texttt{P} \wedge \neg\text{Past}(\texttt{Marked}_\texttt{P}, \varepsilon) \tag{22}$$

and

$$(\texttt{fire}_\texttt{o} \wedge \neg\text{NowST}) \rightarrow \texttt{Marked}_\texttt{P} \wedge \texttt{Marked}_\texttt{Q} \wedge \neg\text{Past}(\texttt{Marked}_\texttt{P} \wedge \texttt{Marked}_\texttt{Q}, \varepsilon)\,. \tag{23}$$

Thus, if a transition fires in 0-time, i.e., at a nonstandard time, at least one of its input places is marked exactly and only when it fires.

The next formulae capture the meaning of the lower and upper bounds associated with transitions when both m_o and M_o are > 0. We separate the cases (a) and (b). In the former we have the following formulae:

$$\texttt{fire}_\texttt{o} \rightarrow \text{Lasted}_{[]}(\texttt{Marked}_\texttt{P}, m_o - 1) \tag{24}$$

$$\text{Lasted}_{[]}(\texttt{Marked}_\texttt{P}, M_o - 1) \rightarrow \texttt{fire}_\texttt{o}. \tag{25}$$

Notice that, since in this case o fires at standard times, if i fired at a nonstandard time, we deduce that it also holds that $\text{Lasted}_{[)}(\texttt{Marked}_\texttt{P}, m_o - k\varepsilon)$ (resp., $\text{Lasted}_{[)}(\texttt{Marked}_\texttt{P}, M_o - k\varepsilon)$), where k is the number of 0-time firings that preceded i's firing. For fragments of type (b), instead, we have the following:

$$\texttt{fire}_\texttt{o} \rightarrow \text{Lasted}_{[]}(\texttt{Marked}_\texttt{P} \wedge \texttt{Marked}_\texttt{Q}, m_o - 1) \tag{26}$$

$$\text{Lasted}_{[]}(\texttt{Marked}_\texttt{P} \wedge \texttt{Marked}_\texttt{Q}, M_o - 1) \rightarrow \texttt{fire}_\texttt{o} \tag{27}$$

Similarly to case (a), if i_1 or i_2 fired at a nonstandard time, this implies that $\text{Lasted}_{[)}(\texttt{Marked}_\texttt{P}, m_o - k\varepsilon)$ or $\text{Lasted}_{[)}(\texttt{Marked}_\texttt{Q}, m_o - k\varepsilon)$, etc.

[4] This convention is symmetric to the one adopted elsewhere (e.g., [5]), where marking is assumed to hold in left-closed and right-open intervals.

Finally, we need to consider the cases in which the bounds are 0. First, when the lower bound is 0 (i.e., $m_o = 0$), we have, for fragments of type (a),

$$\texttt{fire}_\texttt{o} \rightarrow \texttt{Marked}_\texttt{P} \tag{28}$$

or, equivalently (since $\text{Lasted}_{[)}(\alpha, \varepsilon) \equiv \alpha$),

$$\texttt{fire}_\texttt{o} \rightarrow \text{Lasted}_{[)}(\texttt{Marked}_\texttt{P}, \varepsilon)\,. \tag{29}$$

For fragments of type (b), instead, we have:

$$\texttt{fire}_\texttt{o} \rightarrow \texttt{Marked}_\texttt{P} \wedge \texttt{Marked}_\texttt{Q}. \tag{30}$$

When the upper bound is 0 (i.e., $M_o = 0$) the following holds for fragments of type (a)

$$\texttt{Marked}_\texttt{P} \rightarrow \texttt{fire}_\texttt{o} \tag{31}$$

whereas for fragments of type (b) we have

$$\texttt{Marked}_\texttt{P} \wedge \texttt{Marked}_\texttt{Q} \rightarrow \texttt{fire}_\texttt{o}. \tag{32}$$

For all other axioms not explicitly stated here (e.g., to formalize the initial marking and the cases of several ingoing or outgoing arcs from the same place), the formalization perfectly parallels previous axioms stated in terms of the original TRIO language (see e.g. [38]).

For instance, consider the net fragment of Figure 9. It is immediate to verify that the only firing sequence compatible with the above semantics is:

$$u \text{ at } 0 \longrightarrow v \text{ and } w \text{ at } 1 \longrightarrow x \text{ at } 1+\varepsilon \longrightarrow y \text{ at } 1+2\varepsilon \longrightarrow z \text{ at } 3.$$

We emphasize that having imposed any transition to take a nonnull time — though infinitesimal — to fire once enabled is the condition that allowed us to assume that any place is marked at some time instant since its input transition fired (Formulae (19) and (20)), and therefore to define the firing conditions on the basis of the marking duration. This was not possible in approaches, such as [38], where "pure 0-time" firing was allowed and therefore required a considerably more cumbersome and less intuitive formalization.

Remark. We call the TPN semantics formalized through the above axiomatization "syncronizing" because after any finite sequence of micro-steps the next macro-step occurs at a standard time as exemplified in the above sequence referring to the net of of Figure 9. This choice is certainly a reasonable option but not the only possible one; the semantics defined in [20], instead, which also used NSA to formalize 0-time firings, would have produced, for the same net, the sequence

$$u \text{ at } 0 \longrightarrow v \text{ and } w \text{ at } 1 \longrightarrow x \text{ at } 1+\varepsilon \longrightarrow y \text{ at } 1+2\varepsilon \longrightarrow z \text{ at } 3+2\varepsilon$$

thus allowing for the unbounded accumulation of infinitesimal delays. It would be easy to define X-TRIO axioms formalizing the previous semantics (and

various other ones), but in this case the sufficient conditions that guarantee X-TRIO$^{\text{dec}}_{\mathbb{N}}$ decidability would not be satisfied.

Even more generally, if we consider unbounded TPNs – which have the computational power of Turing machines – it would be impossible to formalize their semantics by using any propositional version of X-TRIO, since the state space of such nets is infinite. It would be easy, however, to produce a natural axiomatization of their semantics by means of the general version of syntax (1) through an obvious generalization of $\texttt{Marked}_{\texttt{P}}$ into $\texttt{Marked}_{\texttt{P}}(n)$. As expected, the price for the increased generality is the loss of decidability, but in this case other, possibly semi-automatic, proof techniques could be applied. This shows the usefulness of X-TRIO in its various versions, whether decidable or not.

6.2. X-TRIO for Simulink/Stateflow diagrams

The Stateflow notation [37] is a variation of Statecharts [40]; it describes finite state machines performing discrete transitions between states in a simple and intuitive way. In a nutshell, a Stateflow diagram is composed of: (i) a finite set of typed variables V partitioned into input (V_I), output (V_O), and local (V_L) variables; input and output events are represented, respectively, through Boolean variables of V_I and V_O; (ii) a finite set of states S which can be associated with *entry*, *exit*, and *during* actions, which are executed, respectively, when the state is entered, exited, or throughout the permanence of the system in the state; (iii) a finite set of transitions H that may include guards (i.e., constraints) on the variables of V and actions. An *action* is the assignment of the value of an expression over constants and variables of V to a non-input variable. We assume all variables in V to take value in a finite domain, which we represent by D_V. Figure 10 shows an example of Stateflow diagram capturing the behavior of the controller of the robotic arm that is part of the example used in this section to carry out some verification experiments.

A Simulink graph represents a *component* of the system, which can be *basic* or *composed*. A basic component has a public *interface*, which corresponds to the set of variables $V_{Int} = V_I \cup V_O$ of the module, and a *behavior* description, that is represented by a Stateflow graph. The specification of a composed component is structured as follows: at the lowest level of the system description hierarchy, it is represented by a Simulink graph with two or more basic components. Its interface is the union of the Input and Output variables of its components; the behavior is described by the Stateflow graphs of its modules, plus a network of communication relations between components represented graphically by a set of *links*. Each link corresponds to a flow of messages (signals or data) sent from a component to another one. The communication is realized by the assignment of the value of an Output variable of the sending component to a corresponding Input variable of the receiving component. One or more Simulink graphs can be further composed to obtain a new higher-level component. Figure 11 shows the Simulink graph of a system including the robot controller of Figure 10.

The documentation provided by Mathworks presents the complete, although informal, specification of Stateflow diagrams, but it does not provide a precise definition of their semantics. Our semantics of Stateflow diagrams is inspired

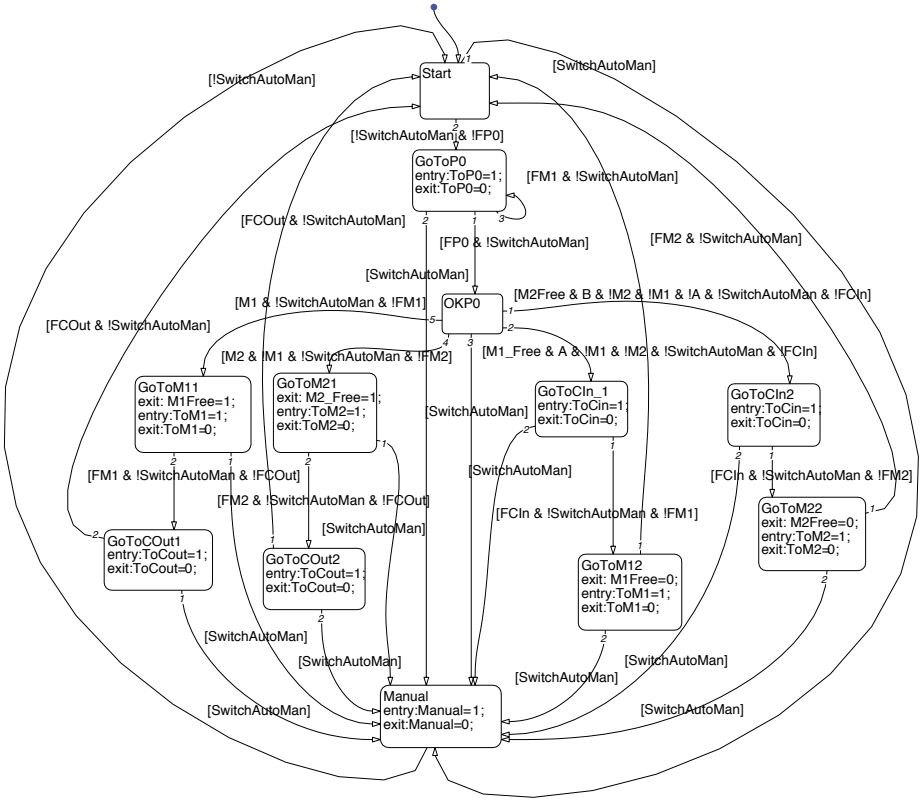


Figure 10: Example of Stateflow diagram: a controller for a robotic arm.

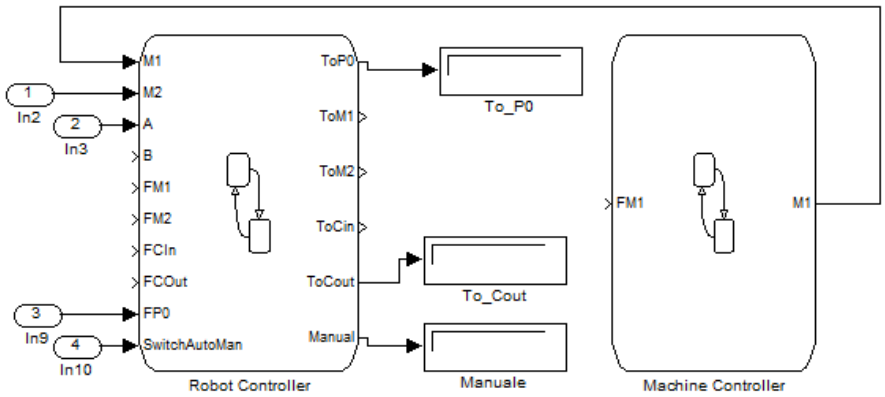


Figure 11: Example of Simulink graph for a robotic cell.

by the STATEMATE semantics of Statecharts [41]. It includes a composition operator for building hierarchical, modular models from simpler ones.

The Stateflow semantics hinges on the concept of *run*, which represents the reaction of the system to a sequence of input events. A run is a sequence of *configurations*; each configuration $\langle s, \nu \rangle$ pairs the current state $s \in S$ with an evaluation function $\nu : V \to D_V$ representing the current values of the variables. Stateflow models are internally deterministic. Input events, however, occur in a nondeterministic manner, so the model overall is nondeterministic.

The semantics of the evolution of time in Statecharts/Stateflow diagrams has proven difficult to pin down precisely, and different solutions have been proposed in the literature (e.g., [42, 43]). The model presented in this section is based on the so-called *run-to-completion* variety. In this model, the system reacts to the input events by performing a sequence of reactions (*macro-steps*). Within every macro-step, a maximal set of enabled transitions (*micro-steps*) is selected and executed based on the events generated in the previous macro-step. Micro-steps are executed infinitely fast, with time advancing only at macro-step boundaries, when the system reaches a *stable* configuration, i.e., in which no transition is enabled. In other words, micro-steps take zero time to execute; when no transition is enabled, time advances and the configuration changes when a new input event is received from the environment. As for STATEMATE, components sense input events and data only at the beginning of macro-steps and communicate output events and data only at their end. In the semantics outlined above each run identifies a sequence of time instants $\{t_i\}_{i \in \mathbb{N}}$, one for each macro-step, hence the time domain is discrete.

Next we report a selection of X-TRIO$^{\text{dec}}_{\mathbb{N}}$ formulae providing the semantics of some crucial aspects of Stateflow diagrams focusing on single Stateflow diagrams; then we informally describe how to deal with the issue of composing diagrams in a hierarchy.

As the domain D_V of Stateflow variables is assumed to be finite, it can be represented through a set of propositional letters: given a variable $v \in V$ and a value $k \in D_V$, when v_k is true this represents that the value of v is k. Similarly for the state space S. For readability, we write $v = k$ instead of v_k.

Given a Stateflow diagram representing the behavior of a module m, for each transition $H_{m,i} : s_{m,i} \overset{g_{m,i}/a_{m,i}}{\longrightarrow} s'_{m,i}$ originating from state $s_{m,i}$ and targeting state $s'_{m,i}$ with guard $g_{m,i}$ and action $a_{m,i}$, we introduce the following formula:

$$\mu_i = \text{AlwF}\Big((\gamma_{m,i} \wedge s_m = s_{m,i}) \to \text{X}_{\text{ns}}\big(s_m = s'_{m,i}\big) \wedge \alpha_{m,i} \wedge \alpha_{ex_{s_{m,i}}} \wedge \alpha_{en_{s'_{m,i}}}\Big) \tag{33}$$

where $\gamma_{m,i}$ is an X-TRIO$^{\text{dec}}_{\mathbb{N}}$ formula encoding guard $g_{m,i}$, and $\alpha_{m,i}$, $\alpha_{ex_{s_{m,i}}}$, and $\alpha_{en_{s'_{m,i}}}$ are X-TRIO$^{\text{dec}}_{\mathbb{N}}$ formulae encoding, respectively, the transition action $a_{m,i}$, the *exit* action of $s_{m,i}$, and the *entry* action of $s'_{m,i}$. Formula (33) formalizes the execution of a micro-step: it asserts that if the current state of module m is $s_{m,i}$ and the transition condition $\gamma_{m,i}$ holds, then in the next micro-step the active state is $s'_{m,i}$ and the *entry* actions of $s'_{m,i}$ and the *exit* actions

of $s_{m,i}$ are executed. Thus, operator $\mathrm{X_{ns}}$ represents a zero-time transition. To guarantee internal determinism only one of the guards can hold at a time.

If no transition is enabled, the configuration does not change, as captured by the following formula:

$$\mathrm{AlwF}\Big(\bigwedge_{i=1}^{|H_m|} \neg(\gamma_{m,i} \wedge s_m = s_{m,i}) \rightarrow NOCHANGE\Big) \tag{34}$$

where subformula $NOCHANGE$, which is not further detailed for brevity, asserts that in the next micro-step the current state and the values of all output and local variables of module m do not change.

The "real" time advancement of our semantics is modeled through operator $\mathrm{X_{st}}$: every time the system reaches a stable state (where no transition is enabled), time advances to the next standard number. We restrict the distance between two consecutive standard instants (i.e. macro-steps) in a run to be exactly 1. The following formula captures the advancement of the "real" time in a single module:

$$\mathrm{AlwF}\left(\mathrm{X_{st}}(\top) \rightarrow \bigwedge_{i=1}^{|H_m|} \neg(\gamma_{m,i} \wedge s_m = s_{m,i})\right). \tag{35}$$

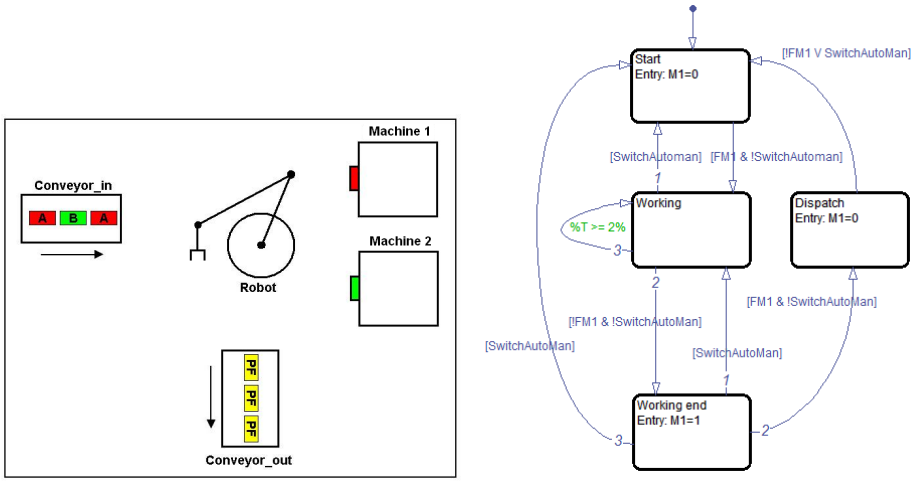
Formula (35) expresses a necessary condition for time advancement. A sufficient condition can be expressed at the level of the single module only after having introduced a pair of additional predicates that are used to coordinate the composition of different modules. This is unsurprising, as time advancement requires *all* modules to have reached a configuration where no further transitions are possible for any of them.

Finally, we introduce a formula asserting that input variables $V_{I,m}$ of module m change their value only at the beginning of a macro-step, i.e., in a standard time instant. In other words, if the next time instant is non-standard, then the values of the input variables must be the same as those in the current instant:

$$\mathrm{AlwF}\Big(\mathrm{X_{ns}}(\top) \rightarrow (\textstyle\bigwedge_{v \in V_{I,m}, x \in D_V} v = x \rightarrow \mathrm{X_{ns}}(v = x))\Big). \tag{36}$$

The formula MOD_m encoding the behavior of a single component m is given by the conjunction of formulae $\bigwedge_{i=1}^{|H_m|} \mu_i$, (34-36), plus others not shown for brevity.

To define the semantics of models composed of basic modules, we employ a hierarchical approach, where basic Simulink graphs are built from Stateflow diagrams, and they can then be in turn composed into Simulink components of a higher level. To achieve this, for each module m, be it a simple Stateflow diagram or a Simulink graph of any level, we introduce two X-TRIO$^{\mathrm{dec}}_{\mathbb{N}}$ predicates – and related X-TRIO$^{\mathrm{dec}}_{\mathbb{N}}$ formulae – that act as the interface of the module for the purpose of coordinating time advancement. All modules included in a composed module first evolve through microsteps of infinitesimal length; then if a component module has reached a stable state where none of its transactions is enabled, it performs a *stutter* step that does not change its state. Only when all component modules are in a stable state then time advances through a macro, non infinitesimal time step.



(a) (b)

Figure 12: Robotic Cell 12(a) and Stateflow graph of machine M_1 12(b).

System properties verification and experimental results. The formalization outlined above has been implemented in the Zot tool to perform the verification of some typical real-time properties of an example system.

Zot [11] is a bounded satisfiability checker which supports the verification of PLTLB models. It solves satisfiability (and validity) problems for PLTLB formulae by exploiting Satisfiability Modulo Theories (SMT) [44] solvers. Through Zot one can check whether stated properties hold for the system being analyzed (or parts thereof) or not; if a property does not hold, Zot produces a counterexample that violates it.

The analyzed system consists of a robotic cell composed of a robot arm that loads and unloads various parts on two machines, M_1 and M_2. The cell, as shown in Figure 12(a), is served by a conveyor belt, which provides pallets to be processed. There are two types of pallets, A and B, which are processed, respectively, by machine M_1 and by machine M_2. After processing, the finished parts are discharged from the cell by means of the conveyor out belt. The M_1 component is presented in Figure 12(b), while Figure 10 shows a Stateflow diagram describing the behavior of the robot arm. At any time, the robot arm can switch from automatic to manual mode or from manual to automatic mode upon a suitable command from the operator. For example, in the graph of Figure 10, the transition from state `GoToP0` to state `OKP0` is enabled when a photocell signals that the robot arm has reached the central position `P0`, setting the input variable `FP0`. Figure 11 shows a Simulink graph representing a part of the robotic cell.

A first, fundamental property that we checked is that the modeled system does not have Zeno runs, which would make it unrealizable. The system shows a

Zeno behavior if, from a certain point on, "real" time does not advance, i.e., no macro-steps are performed. The *presence* of Zeno runs is formalized as follows:

$$\mathrm{SomF}(\mathrm{AlwF}(\mathrm{X}_{\mathrm{ns}}(\top))) \tag{37}$$

Formula (37) states that, from a certain instant on, the clock does not tick any more, i.e., the trace presents an infinite sequence of non-standard instants. We checked through the Zot tool that formula $SYS \wedge \mathrm{SomF}(\mathrm{AlwF}(\mathrm{X}_{\mathrm{ns}}(\top)))$ is *unsatisfiable* (where SYS is the formalization of the whole system as explained above), hence no runs of the system show property (37), and the system is devoid of Zeno runs.

In [2] we formalized and automatically checked other liveness and safety properties of the robotic cell, such as its ability to produce and deliver one processed workpiece of any kind within given time bounds, or the absence of deadlocks. All properties were checked in a time ranging from a few tens of seconds to a few hours, depending on the portion of the system state space explored by the checker. Considering that the sole Stateflow diagram of the controller of the robot arm of Figure 10 has $12 \cdot 2^{18}$ possible configurations, i.e., $|S| \cdot 2^{|D_V|}$, the first verification experiments are encouraging, and show the feasibility of the approach. In fact, we were able to detect deadlocks in an early specification of the FMS that stemmed from an incorrect communication protocol between the robot and machine M_1.

7. Conclusions and Future Work

We introduced a novel approach to the modeling and analysis of systems that evolve through a sequence of micro- and macro-steps occurring at different time scales, such that the duration of the micro-steps is negligible with respect to that of the macro-steps. In some sense, we can position our approach in between the "time granularity approach" [26] where different but positive standard and comparable time scales are adopted at different levels of abstraction, and the "zero-time transition" approach [41], [12] which instead collapses the duration of some action to a full zero. By introducing the notion of infinitesimal duration for micro-steps and by borrowing the elegant terminology of NSA to formalize them, we overcome the limitations of the two other cases and generalize them: on the one side, unlike traditional mappings of different but positive standard time granularities, infinitesimal steps may accumulate in unbounded ways, thus allowing for the analysis of usually pathological cases such as Zeno behaviors; on the other side, by imposing that the effect of an event strictly follows in time its cause, we are closer to the traditional view of dynamical system theory, and we can reason explicitly about possible synchronizations between different components, even at the level of micro-steps.

We pursued our approach through the novel language X-TRIO, which includes both metric operators on continuous time and the next-time operator to refer to the next discrete step in the computation. Under simple and realistic conditions, X-TRIO can be encoded into an equivalent PLTLB formulation,

which makes it amenable to automated verification, still retaining, however, the distinguishing TRIO feature (the Dist temporal operator) that allows us to assert and prove properties about the system state even in time instants when no step occurs.

We emphasize the generality and flexibility of our approach. In previous work [2] we applied X-TRIO to formalize (one of the many possible semantics of) the Stateflow notation: we developed a complete industrial case study in the field of FMS, we analyzed some relevant system properties by means of a tool that implements (a minor variation of) the translation schema presented in this paper; [2] also describes a method to formalize the cooperation and synchronization of several modules specified as Stateflow diagrams. In the present work (Section 6.2) we have reported some explicative examples of the Stateflow formalization and verification by means of X-TRIO.

As a complement, we also applied X-TRIO to formalize the semantics of timed Petri nets allowing for 0-time transitions, whose firings are naturally formalized as micro-steps, while firings of transitions with standard positive time bounds correspond to macro-steps. Thanks to the generality of our approach it is quite easy to formalize any possible variation of the semantics of the original notation: this claim has been verified both in the case of Stateflow and in that of TPN, and we are quite confident that it holds for most other traditional operational formalisms. Finally, the NSA-based approach used in this work for dealing with infinitesimal time advances need not be confined to the TRIO metric temporal logic: mechanisms and operators similar to those introduced in this paper for X-TRIO could be provided for other logics such as MTL [9].

We plan to further pursue such a generality along several dimensions. The present choice of a constant duration for micro-steps is good enough for several application fields including, e.g., FMS, but it is not a necessary restriction. Different, fixed or even variable durations for micro-steps could be used to model different components of a system and their synchronization at the micro-level. Also, non-zero infinitesimal durations for micro-steps are well-suited to investigate –the risk of– dangerous behaviors such as Zeno ones. However, once these have been excluded, one could revert to a finite metric of micro-steps, perhaps exploiting different time granularities. Something similar occurs during hardware design where, in various design phases, the designer analyzes the risk of critical races and the duration of precise finite sequences of micro-steps, or "collapses" all such sequences in an "abstract zero-time". Our approach allows the designer to manage all such phases in a uniform and general way.

We envisage that our approach can be used to formalize and analyze issues related to the "robustness" of specifications, along the lines of [45, 46, 47], which deal with infinitesimal perturbations in time measurements.

Furthermore, we plan to exploit the flexibility of our approach in decidability issues. The trade-off between expressiveness and decidability (efficiency) offers many opportunities. On the one side, other fragments of X-TRIO, more general than X-TRIO$_{\mathbb{N}}$, and decision procedures different from –or complementary to– the translation into PLTLB are under investigation. On the other side, however, there are clearly systems whose models and properties are necessarily

undecidable; for instance the full generality of basic X-TRIO is required by the Bekeley behavior expressed by Formula (4), as well as by the axiomatization of unbounded TPNs. In such cases system analysis and property verification should be supported by semi-automatic tools other than "pure model checkers": For instance, the axiomatization approach outlined in Section 6.1 is well suited not only for a translation into a decidable logic such as PLTLB, but also for translation into more powerful ones for which (semi)automatic theorem provers are available, such as PVS [48]; [49] reports on previous experiences in proving TRIO theorems by exploiting a translation into PVS.

Acknowledgments. We thank Emanuele Carpanzano and Mauro Mazzolini formerly of CNR-ITIA for providing expertise, insight, and examples of design of FMS.

References

[1] L. Ferrucci, D. Mandrioli, A. Morzenti, M. Rossi, A metric temporal logic for dealing with zero-time transitions, International Syposium on Temporal Representation and Reasoning (TIME) (2012) 81–88.

[2] L. Ferrucci, D. Mandrioli, A. Morzenti, M. Rossi, Modular automated verification of flexible manufacturing systems with metric temporal logic and non-standard analysis, in: Formal Methods for Industrial Critical Systems, Vol. 7437 of Lecture Notes in Computer Science, 2012, pp. 162–176.

[3] E. A. Lee, S. A. Seshia, Introduction to Embedded Systems, A Cyber-Physical Systems Approach, 2011, `http://LeeSeshia.org`.

[4] M.-A. Esteve, J.-P. Katoen, V. Y. Nguyen, B. Postma, Y. Yushtein, Formal correctness, safety, dependability, and performance analysis of a satellite, in: Proceedings of the Int. Conf. on Software Engineering (ICSE), 2012, pp. 1022–1031.

[5] C. A. Furia, D. Mandrioli, A. Morzenti, M. Rossi, Modeling Time in Computing, EATCS Monographs in Theoretical Computer Science, Springer, 2012.

[6] R. Alur, T. A. Henzinger, Real-time logics: Complexity and expressiveness, Information and Computation 104 (1) (1993) 35–77.

[7] J. Ouaknine, J. Worrell, Some recent results in metric temporal logic, in: Formal Modeling and Analysis of Timed Systems, Vol. 5215 of Lecture Notes in Computer Science, 2008, pp. 1–13.

[8] A. Morzenti, D. Mandrioli, C. Ghezzi, A model parametric real-time logic, ACM Transactions on Programming Languages and Systems 14 (4) (1992) 521–573.

[9] R. Koymans, Specifying real-time properties with metric temporal logic, Real-Time Systems 2 (4) (1990) 255–299.

[10] A. Robinson, Non-standard analysis, Princeton University Press, 1996.

[11] M. Pradella, A. Morzenti, P. San Pietro, Bounded satisfiability checking of metric temporal logic specifications, ACM Transactions on Software Engineering and MethodologyTo appear.

[12] J. S. Ostroff, Temporal Logic for Real Time Sytems, Advanced Software Development Series, John Wiley & Sons, 1989.

[13] J. Ouaknine, J. Worrell, On metric temporal logic and faulty turing machines, in: Foundations of Software Science and Computation Structures, Vol. 3921 of Lecture Notes in Computer Science, 2006, pp. 217–230.

[14] Y. Hirshfeld, A. M. Rabinovich, Logics for real time: Decidability and complexity, Fundamenta Informaticae 62 (1) (2004) 1–28.

[15] J. Bengtsson, W. Yi, Timed automata: Semantics, algorithms and tools, in: Lect. on Concurrency and Petri Nets, Vol. 3098 of LNCS, Springer, 2004, pp. 87–124.

[16] C. Bayer, J. P. Katoen, Principles of Model Checking, The MIT Press, 2008.

[17] X. Liu, E. Matsikoudis, E. A. Lee, Modeling timed concurrent systems, in: CONCUR 2006 – Concurrency Theory, Vol. 4137 of Lecture Notes in Computer Science, 2006, pp. 1–15.

[18] B. Marinković, Z. Ognjanović, D. Doder, A. Perović, A propositional linear time logic with time flow isomorphic to ω^2, Journal of Applied Logic 12 (2) (2014) 208 – 229.

[19] D. P. Guelev, D. V. Hung, Prefix and projection onto state in duration calculus, Electronic Notes in Theoretical Computer Science 65 (6) (2002) 101–119.

[20] A. Gargantini, D. Mandrioli, A. Morzenti, Dealing with zero-time transitions in axiom systems, Information and Computation 150 (2) (1999) 119–131.

[21] A. Benveniste, T. Bourke, B. Caillaud, M. Pouzet, Non-standard semantics of hybrid systems modelers, Journal of Computer and System Sciences 78 (3) (2012) 877 – 910.

[22] S. Bliudze, D. Krob, Modelling of complex systems: Systems as dataflow machines, Fundamenta Informaticae 91 (2009) 251–274.

[23] T. A. Henzinger, Z. Manna, A. Pnueli, What good are digital clocks?, in: Automata, Languages and Programming, Vol. 623 of Lecture Notes in Computer Science, 1992, pp. 545–558.

[24] C. A. Furia, M. Rossi, A theory of sampling for continuous-time metric temporal logic, ACM Transactions on Computational Logic 12 (1) (2010) 1–40, article 8.

[25] A. Burns, I. J. Hayes, A timeband framework for modelling real-time systems, Real-Time Systems 45 (1–2) (2010) 106–142.

[26] E. Corsetti, E. Crivelli, D. Mandrioli, A. Morzenti, A. Montanari, P. San Pietro, E. Ratto, Dealing with different time scales in formal specifications, in: Proc. of the 6th Int. Work. on Software Specification and Design, 1991, pp. 92–101.

[27] A. Montanari, G. Puppis, Decidability of the theory of the totally unbounded omega-layered structure, in: 11th International Symposium on Temporal Representation and Reasoning (TIME), 2004, pp. 156–160.

[28] T. A. Henzinger, S. Qadeer, S. K. Rajamani, Assume-guarantee refinement between different time scales, in: Computer Aided Verification, Vol. 1633 of Lecture Notes in Computer Science, 1999, pp. 208–221.

[29] C. Ghezzi, D. Mandrioli, A. Morzenti, TRIO: A logic language for executable specifications of real-time systems, The Journal of Systems and Software 12 (2) (1990) 107–123.

[30] E. Ciapessoni, A. Coen-Porisini, E. Crivelli, D. Mandrioli, P. Mirandola, A. Morzenti, From formal models to formally-based methods: an industrial experience, ACM Transactions on Software Engineering and Methodology 8 (1) (1999) 79–113.

[31] P. Wolper, Temporal logic can be more expressive, in: Proceedings of the 22nd Annual Symposium on Foundations of Computer Science, 1981, pp. 340–348.

[32] R. Alur, T. Feder, T. A. Henzinger, The benefits of relaxing punctuality, Journal of the ACM 43 (1) (1996) 116–146.

[33] R. Goldblatt, Lectures on the Hyperreals: An Introduction to Nonstandard Analysis, Springer, 1998.

[34] R. Alur, T. A. Henzinger, A really temporal logic, Journal of the ACM 41 (1994) 181 – 203.

[35] J. A. W. Kamp, Tense logic and the theory of linear order, Ph.D. thesis, University of California at Los Angeles (1968).

[36] P. Schnoebelen, The complexity of temporal logic model checking, in: Advances in Modal Logic, 2002, pp. 393–436.

[37] Mathworks, Stateflow online documentation, `http://www.mathworks.it/help/toolbox/stateflow/` (2011).

[38] M. Felder, D. Mandrioli, A. Morzenti, Proving properties of real-time systems through logical specifications and Petri net models, IEEE Transactions on Software Engineering 20 (2) (1994) 127–141.

[39] P. M. Merlin, D. J. Farber, Recoverability and communication protocols: Implications of a theoretical study, IEEE Transactions on Communications 24 (9) (1976) 1036–1043.

[40] D. Harel, Statecharts: A visual formalism for complex systems, Science of Computer Programming 8 (3) (1987) 231–274.

[41] D. Harel, A. Naamad, The STATEMATE semantics of statecharts, ACM Transactions on Software Engineering and Methodology 5 (4) (1996) 293–333.

[42] F. Levi, Compositional verification of quantitative properties of statecharts, Journal of Logic and Computation 11 (6) (2001) 829–878.

[43] R. Alur, T. A. Henzinger, Reactive modules, Formal Methods in System Design (1999) 15:7–48.

[44] M. M. Bersani, A. Frigeri, A. Morzenti, M. Pradella, M. Rossi, P. San Pietro, Constraint LTL satisfiability checking without automata, Journal of Applied Logic 12 (4) (2014) 522–557.

[45] P. Bouyer, N. Markey, O. Sankur, Robustness in timed automata, in: Proceedings of the International Workshop on Reachability Problems (RP), 2013, pp. 1–18.

[46] A. Puri, Dynamical properties of timed automata, in: Proceedings of the International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT), 1998, pp. 210–227.

[47] S. Akshay, L. Hélouët, C. Jard, P.-A. Reynier, Robustness of time petri nets under guard enlargement, in: Proceedings of the International Workshop on Reachability Problems (RP), 2012, pp. 92–106.

[48] S. Owre, J. M. Rushby, N. Shankar, PVS: A prototype verification system, in: Proceedings of CADE-11, Vol. 607 of LNCS, 1992, pp. 748–752.

[49] A. Gargantini, A. Morzenti, Automated deductive requirements analysis of critical systems, ACM Trans. Softw. Eng. Methodol. 10 (3) (2001) 255–307.