Contents

Constraint LTL Satisfiability Checking without Automata

Marcello M. Bersani, Achille Frigeri, Angelo Morzenti, Matteo Pradella,
Matteo Rossi, Pierluigi San Pietro

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milano, Italy

Abstract

This paper introduces a novel technique to decide the satisfiability of formulae written in the language of Linear Temporal Logic with both future and past operators and atomic formulae belonging to constraint system $\mathcal{D}$ (CLTLB($\mathcal{D}$) for short). The technique is based on the concept of *bounded satisfiability*, and hinges on an encoding of CLTLB($\mathcal{D}$) formulae into QF-EU$\mathcal{D}$, the theory of quantifier-free equality and uninterpreted functions combined with $\mathcal{D}$. Similarly to standard LTL, where bounded model-checking and SAT-solvers can be used as an alternative to automata-theoretic approaches to model-checking, our approach allows users to solve the satisfiability problem for CLTLB($\mathcal{D}$) formulae through SMT-solving techniques, rather than by checking the emptiness of the language of a suitable automaton. The technique is effective, and it has been implemented in our ℤot formal verification tool.

Keywords: Satisfiability, Constraint LTL, Bounded Satitisfiability Checking

1. Introduction

Finite-state system verification has attained great successes, both using automata-based and logic-based techniques. Examples of the former are the so-called explicit-state model checkers [1] and symbolic model checkers [2]. However, some of the best results in practice have been obtained by logic-based techniques, such as Bounded Model Checking (BMC) [3]. In BMC, a finite-state machine A (typically, a version of Büchi automata) and a desired property P expressed in Propositional Linear Temporal Logic (PLTL) are translated into a Boolean formula ϕ to be fed to a SAT solver. The translation is made finite by bounding the number of time instants. However, infinite behaviors, which are crucial in proving, e.g., liveness properties, are also considered by using the well-known property that a Büchi automaton accepts an infinite behavior if, and only if, it accepts an infinite periodic behavior. Hence, chosen a bound $k > 0$, a Boolean formula ϕ_k is built, such that ϕ_k is satisfiable if and only if there exists an infinite periodic behavior of the form $\alpha\beta^\omega$, with $|\alpha\beta| \leq k$, that is compatible with system A while violating property P. This procedure allows counterexample detection,

☆This research was partially supported by Programme IDEAS-ERC, Projects 227977-SMScom and PRIN 2010LYA9RH-006.

but it is not complete, since the violations of property P requiring "longer" behaviors, i.e., of the form $\alpha\beta^\omega$ with $|\alpha\beta| > k$, are not detected. However, in many practical cases it is possible to find bounds large enough for representing counterexamples, but small enough so that the SAT solver can actually find them in a reasonable time.

Clearly, the BMC procedure can be used to check satisfiability of a PLTL formula, without considering a finite state system A. This has practical applications, since a PLTL formula can represent both the system and the property to be checked (see, e.g., [4], where the translation into Boolean formulae is made more specific for dealing with satisfiability checking and metric temporal operators). We call this case *Bounded Satisfiability Checking* (BSC), which consists in solving a so-called Bounded Satisfiability Problem: Given a PLTL formula P, and chosen a bound $k > 0$, define a Boolean formula ϕ_k such that ϕ_k is satisfiable if, and only if, there exists an infinite periodic behavior of the form $\alpha\beta^\omega$, with $|\alpha\beta| \leq k$, that satisfies P.

More recently, great attention has been devoted to the automated verification of *infinite*-state systems. In particular, many extensions of temporal logic and automata have been proposed, typically by adding integer variables and arithmetic constraints. For instance, PLTL has been extended to allow formulae with various kinds of arithmetic constraints [5, 6]. This has led to the study of CLTL($\mathcal{D}$), a general framework extending the future-only fragment of PLTL by allowing arithmetic constraints belonging to a generic constraint system $\mathcal{D}$. The resulting logics are expressive and well-suited to define infinite-state systems and their properties, but, even for the bounded case, their satisfiability is typically undecidable [7], since they can simulate general two-counter machines when $\mathcal{D}$ is powerful enough (e.g., Difference Logic).

However, there are some decidability results, which allow in principle for some kind of automatic verification. Most notably, satisfiability of CLTL($\mathcal{D}$) is decidable (in PSPACE) when $\mathcal{D}$ is the class of Integer Periodic Constraints (IPC*) [8], or when it is the structure $(D, <, =)$ with $D \in \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$ [9]. In these cases, decidability is shown by using an automata-based approach similar to the standard case for LTL, by reducing satisfiability checking to the verification of the emptiness of Büchi automata. Given a CLTL($\mathcal{D}$) formula ϕ, with $\mathcal{D}$ as in the above cases, it is possible to define an automaton $\mathcal{A}_\phi$ such that ϕ is satisfiable if, and only if, the language recognized by $\mathcal{A}_\phi$ is not empty.

These results, although of great theoretical interest, are of limited practical relevance for what concerns a possible implementation, since the involved constructions are very inefficient, as they rely on the complementation of Büchi automata.

In this paper, we extend the above results to a more general logic language, called CLTLB($\mathcal{D}$), which is an extension of PLTLB (PLTL with Both future and past operators) with arithmetic constraints in constraint system $\mathcal{D}$, and define a procedure for satisfiability checking that does not rely on automata constructions.

The idea of the procedure is to determine satisfiability by checking a finite number of k-satisfiability problems. Informally, k-satisfiability amounts to looking for ultimately periodic *symbolic* models of the form $\alpha\beta^\omega$, i.e., such that prefix $\alpha\beta$ of length k admits a bounded arithmetic model (up to instant k). Although the k-bounded problem is defined with respect to a bounded arithmetical model, it provides a representation of infinite symbolic models by means of ultimately periodic words. When CLTLB($\mathcal{D}$) has the property that its ultimately periodic symbolic models, of the form $\alpha\beta^\omega$, al-

ways admit an arithmetic model, then the k-satisfiability problem can be reduced to satisfiability of QF-EU$\mathcal{D}$ (the theory of quantifier-free equality and uninterpreted functions combined with $\mathcal{D}$). In this case, k-satisfiability is equivalent to satisfiability over infinite models.

There are important examples of constraint systems $\mathcal{D}$, such as for example IPC*, in which determining the existence of arithmetical models is achieved by complementing a Büchi automaton $\mathcal{A}_C$. In this paper we define a novel condition, tailored to ultimately periodic models of the form $\alpha\beta^\omega$, which is proved to be equivalent to the one captured by automaton $\mathcal{A}_C$. Thanks to this condition, checking for the existence of arithmetical models can be done in a bounded way, without resorting to the construction (and the complementation) of Büchi automata. This is the key result that makes our decision procedure applicable in practice.

Symmetrically to standard LTL, where bounded model-checking and SAT-solvers can be used as an alternative to automata-theoretic approaches to model-checking, reducing satisfiability to k-satisfiability allows us to determine the satisfiability of CLTLB($\mathcal{D}$) formulae through Satisfiability Modulo Theories (SMT) solvers, instead of checking the emptiness of a Büchi automaton. Moreover, when the length of all prefixes $\alpha\beta$ to be tested is bounded by some $K \in \mathbb{N}$, then the number of bounded problems to be solved is finite. Therefore, we also prove that k-satisfiability is *complete* with respect to the satisfiability problem, i.e., by checking at most K bounded problems the satisfiability of CLTLB($\mathcal{D}$) formulae can always be determined.

To the best of our knowledge, our results provide the first effective implementation of a procedure for solving the CLTLB($\mathcal{D}$) satisfiability problem: we show that the encoding into QF-EU$\mathcal{D}$ is linear in the size of the formula to be checked and quadratic in the length k. The procedure is implemented in the $\mathbb{Z}$ot toolkit[1], which relies on standard SMT-solvers, such as Z3 [10].

The paper is organized as follows. Section 2 describes CLTL($\mathcal{D}$) and CLTLB($\mathcal{D}$), and their main known decidability results and techniques. Section 3 defines the k-satisfiability problem, introduces the bounded encoding of CLTLB($\mathcal{D}$) formulae, and shows its correctness. Section 4 introduces a novel, bounded condition for checking the satisfiability of CLTLB($\mathcal{D}$) formulae when $\mathcal{D}$ is IPC*, and discusses some cases under which the encoding can be simplified. Section 5 studies the complexity of the defined encoding and proves that, provided that $\mathcal{D}$ satisfies suitable conditions, there exists a completeness threshold. Section 6 illustrates an application of the CLTLB logic and the $\mathbb{Z}$ot toolkit to specify and verify a system behavior. Section 7 describes relevant related works. Finally, Section 8 concludes the paper highlighting some possible applications of the implemented decision procedure for CLTLB($\mathcal{D}$).

2. Preliminaries

This section presents an extension to Kamp's [11] PLTLB, by allowing formulae over a constraint system. As suggested in [5], and unlike the approach of [12], the atomic formulae of this logic are Boolean atoms or atomic arithmetical constraints.

[1] `http://zot.googlecode.com`

2.1. Language of constraints

Let V be a finite set of variables; a *constraint system* is a pair $\mathcal{D} = (D, \mathcal{R})$ where D is a specific domain of interpretation for variables and constants and $\mathcal{R}$ is a family of relations on D. An *(atomic) $\mathcal{D}$-constraint* is a term of the form $R(x_1, \ldots, x_n)$, where R is an n-ary relation of $\mathcal{R}$ on domain D and $x_1, \ldots, x_n$ are variables. A $\mathcal{D}$-valuation is a mapping $v : V \to D$, i.e., an assignment of a value in D to each variable. A $\mathcal{D}$-constraint is *satisfied* by a D-valuation v, written $v \models_{\mathcal{D}} R(x_1, \ldots, x_n)$, if $(v(x_1), \ldots, v(x_n)) \in R$.

In Section 4, we consider $\mathcal{D}$ to be $(D, <, =)$, where $D \in \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$ and $<$ is a strict total order on D. When domain D is discrete, we can endow $\mathcal{D}$ with the congruence modulo c over D that allows formulae of the form $x \equiv_c d$ and $x \equiv_c y + d$, where $c, d \in D$, to be part of the language of constraints. We call this extension IPC*, by borrowing its name from the original definition in [13]. We consider here the quantifier-free version of the constraint system, which has the same expressive power of the quantified one [13, Lemma 1]. Given a set of $\mathcal{D}$-constraints C, we write $v \models_{\mathcal{D}} C$ when $v \models_{\mathcal{D}} \gamma$ for every $\gamma \in C$.

2.2. Syntax of CLTLB

CLTLB($\mathcal{D}$) is defined as an extension of PLTLB, where atomic formulae are relations from $\mathcal{R}$ over arithmetic temporal terms defined in $\mathcal{D}$. The resulting logic is actually equivalent to the quantifier-free fragment of first-order LTL over signature $\mathcal{R}$. Let x be a variable over D and c be a constant in D; *arithmetic temporal terms* (a.t.t.) are defined as:

$$\alpha := c \mid x \mid \mathrm{X}\alpha \mid \mathrm{Y}\alpha.$$

In CLTLB($\mathcal{D}$), a.t.t.'s may appear in atomic $\mathcal{D}$-constraints. The syntax of (well formed) formulae of CLTLB($\mathcal{D}$) is recursively defined as follows:

$$\phi := R(\alpha_1, \ldots, \alpha_n) \mid \phi \wedge \phi \mid \neg\phi \mid \mathbf{X}\phi \mid \mathbf{Y}\phi \mid \phi\mathbf{U}\phi \mid \phi\mathbf{S}\phi$$

where α_i's are a.t.t.'s, $R \in \mathcal{R}$; $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{U}$, and $\mathbf{S}$ are the usual "next", "previous", "until", and "since" operators from LTL.

Note that X and $\mathbf{X}$ are two distinct operators. Intuitively, if ϕ is a formula, $\mathbf{X}\phi$ has the standard PLTL meaning, while $\mathrm{X}\alpha$ denotes the *value* of a.t.t. α in the next time instant. The same holds for Y and $\mathbf{Y}$, which refer to the previous time instant. Each relation symbol is associated with a natural number denoting its arity. As we will see in Section 3.4, we can treat separately 0-ary relations, i.e., propositional letters, whose set is denoted by $\mathcal{R}_0$. We also write CLTLB($\mathcal{D}, \mathcal{R}_0$) to denote the language CLTLB over the constraint system $\mathcal{D}$ whose 0-ary relations are exactly those in $\mathcal{R}_0$. CLTL($\mathcal{D}$) is the future-only fragment of CLTLB($\mathcal{D}$).

The *depth* $|\alpha|$ is defined for a.t.t.'s over variables as the total amount of temporal shift needed in evaluating α:

$$|x| = 0, \quad |\mathrm{X}\alpha| = |\alpha| + 1, \quad |\mathrm{Y}\alpha| = |\alpha| - 1.$$

The depth of a.t.t.'s over constants is 0.

Let ϕ be a CLTLB($\mathcal{D}, \mathcal{R}_0$) formula, x a variable of V and $\Gamma_x(\phi)$ the set of all a.t.t.'s occurring in ϕ in which x appears. We define the "look-forwards" $\lceil\phi\rceil_x$ and "look-backwards" $\lfloor\phi\rfloor_x$ of ϕ relatively to x as:

$$\lceil\phi\rceil_x = \max_{\alpha_i \in \Gamma_x(\phi)} \{0, |\alpha_i|\}, \quad \lfloor\phi\rfloor_x = \min_{\alpha_i \in \Gamma_x(\phi)} \{0, |\alpha_i|\}.$$

The definitions above naturally extend to V by letting $\lceil\phi\rceil = \max_{x \in V}\{\lceil\phi\rceil_x\}$, $\lfloor\phi\rfloor = \min_{x \in V}\{\lfloor\phi\rfloor_x\}$. Hence, $\lceil\phi\rceil$ ($\lfloor\phi\rfloor$) is the largest (smallest) depth of all the a.t.t.'s of ϕ, representing the length of the future (past) segment needed to evaluate ϕ in the current instant.

2.3. Semantics

The semantics of CLTLB($\mathcal{D}, \mathcal{R}_0$) formulae is defined with respect to a strict linear order $(\mathbb{Z}, <)$ representing time. Truth values of propositions in $\mathcal{R}_0$, and values of variables belonging to V are defined by a pair (π, σ) where $\sigma : \mathbb{Z} \times V \to D$ is a function which defines the value of variables at each position in $\mathbb{Z}$ and $\pi : \mathbb{Z} \to \wp(\mathcal{R}_0)$ is a function associating a subset of the set of propositions with each element of $\mathbb{Z}$. Function σ is extended to terms as follows:

$$\sigma(i, \alpha) = \begin{cases} \sigma(i + |\alpha|, x_\alpha), & x_\alpha \text{ is the variable in } V \text{ occurring in } \alpha \\ c_\alpha & c_\alpha \text{ is the constant in } D \text{ occurring in } \alpha. \end{cases}$$

By definition of $\sigma(i, \alpha)$, it is obvious that $\mathrm{XY}x = \mathrm{YX}x = x$; hence, we may assume, with no loss of generality, that a.t.t.'s do not contain alternated occurrences of the operators X and Y. Moreover, for every constant c, $\mathrm{X}c = \mathrm{Y}c = c$.

The semantics of a CLTLB($\mathcal{D}, \mathcal{R}_0$) formula ϕ at instant $i \geq 0$ over a linear structure (π, σ) is recursively defined by means of a satisfaction relation $\models$ as follows, for every formulae ϕ, ψ and for every a.t.t. α:

$$\begin{aligned}
(\pi, \sigma), i &\models p \text{ iff } p \in \pi(i) \text{ for } p \in \mathcal{R}_0 \\
(\pi, \sigma), i &\models R(\alpha_1, \ldots, \alpha_n) \text{ iff } (\sigma(i, \alpha_1), \ldots, \sigma(i, \alpha_n)) \in R \quad \text{for } R \in \mathcal{R} \setminus \mathcal{R}_0 \\
(\pi, \sigma), i &\models \neg\phi \text{ iff } (\pi, \sigma), i \not\models \phi \\
(\pi, \sigma), i &\models \phi \wedge \psi \text{ iff } (\pi, \sigma), i \models \phi \text{ and } (\pi, \sigma), i \models \psi \\
(\pi, \sigma), i &\models \mathbf{X}\phi \text{ iff } (\pi, \sigma), i + 1 \models \phi \\
(\pi, \sigma), i &\models \mathbf{Y}\phi \text{ iff } (\pi, \sigma), i - 1 \models \phi \text{ and } i > 0 \\
(\pi, \sigma), i &\models \phi \mathbf{U} \psi \text{ iff } \exists\, j \geq i : (\pi, \sigma), j \models \psi \text{ and} \\
&\qquad\qquad (\pi, \sigma), n \models \phi \; \forall n : \; i \leq n < j \\
(\pi, \sigma), i &\models \phi \mathbf{S} \psi \text{ iff } \exists\, 0 \leq j \leq i : (\pi, \sigma), j \models \psi \text{ and} \\
&\qquad\qquad (\pi, \sigma), n \models \phi \; \forall n : \; j < n \leq i.
\end{aligned}$$

A formula $\phi \in$ CLTLB($\mathcal{D}, \mathcal{R}_0$) is *satisfiable* if there exists a pair (π, σ) such that $(\pi, \sigma), 0 \models \phi$; in this case, we say that (π, σ) is a *model* of ϕ, π is a *propositional model* and σ is an *arithmetic model*. By introducing as primitive the connective $\vee$, the dual operators "release" $\mathbf{R}$, "trigger" $\mathbf{T}$ and "previous" $\mathbf{Z}$ are defined as: $\phi \mathbf{R} \psi \equiv$

$\neg(\neg\phi \mathbf{U} \neg\psi)$, $\phi \mathbf{T} \psi \equiv \neg(\neg\phi \mathbf{S} \neg\psi)$ and $\mathbf{Z}\phi \equiv \neg \mathbf{Y} \neg\phi$; by applying De Morgan's rules, we may assume every CLTLB formula to be in *positive normal form*, i.e., negation may only occur in front of atomic propositions and relations.

2.4. CLTLB with automata

The *satisfiability* problem for a CLTLB formula ϕ consists in determining whether there exists a model (π, σ) for ϕ such that $(\pi, \sigma), 0 \models \phi$. In this section, we recall some known results where the propositional part π of (π, σ) is either missing or can be eliminated (hence, with a slight abuse of notation we will write $\sigma, 0 \models \phi$ instead of $(\pi, \sigma), 0 \models \phi$).

Hereafter, we restrict $\mathcal{D}$ to be the structure defined by IPC^*, or by $(D, <, =)$, where $D \in \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$. For such constraint systems a decision procedure based on Büchi automata is studied in [9]. The presented notions are essential to develop our decision procedure without automata construction. We remark that, although for convenience we admit the use of constants in the syntax of CLTLB formulae, they can be replaced by variables associated with suitable constraints, as done in [9].

Let ϕ be a CLTLB($\mathcal{D}$) formula and $terms(\phi)$ be the set of arithmetic terms of the form $\mathrm{X}^i x$ for all $0 \leq i \leq \lceil \phi \rceil$ and of the form $\mathrm{Y}^i x$ for all $1 \leq i \leq -\lfloor \phi \rfloor$ and for all $x \in V$. Let $const'(\phi) = \{m, \ldots, M\}$ be the set of constants occurring in ϕ, where $m, M \in D$ are the minimum and maximum constants. If domain D is discrete, we extend $const'(\phi)$ to the set $const(\phi) = [m, M]$ of all values between m and M, following [13]. If domain D is $\mathbb{Q}$ or $\mathbb{R}$, then we assume that constants are in $D \cap \mathbb{Q}$ and we fix $const(\phi) = const'(\phi)$.

Definition 1. Let A be a set of variables and $fresh : terms(\phi) \to A$ be an injective function mapping each a.t.t of ϕ to a fresh variable in set A. Let C be a set of $\mathcal{D}$-constraints over $terms(\phi)$ and $const(\phi)$. Function $fresh$ is naturally extended to (any) set C, by mapping each a.t.t. $\alpha \in terms(\phi)$ in C to $fresh(\alpha)$ and each constant to itself. A set C of $\mathcal{D}$-constraints over $terms(\phi)$ and $const(\phi)$ is *satisfiable* if there exists a $\mathcal{D}$-valuation $v : A \to D$, such that $v \models_{\mathcal{D}} fresh(C)$, i.e., the satisfiability of C over v considers all a.t.t.'s as fresh variables.

Definition 2. Given a valuation v for elements of A, we call C_v the set of all $\mathcal{D}$-constraints over $terms(\phi)$ and $const(\phi)$ such that $v \models_{\mathcal{D}} fresh(C_v)$. A *symbolic valuation* sv for ϕ is a set of $\mathcal{D}$-constraints over $terms(\phi)$ and $const(\phi)$ for which there is a valuation v such that $sv = C_v$. We indicate by $SV(\phi)$ the set of all symbolic valuations of a formula ϕ.

The satisfiability of a set of $\mathcal{D}$-constraints, for the constraint systems $\mathcal{D}$ considered in this work, is decidable [9]. Given a symbolic valuation sv and a $\mathcal{D}$-constraint ξ over a.t.t.'s, we write $sv \models^{sym} \xi$ if for every $\mathcal{D}$-valuation v' such that $v' \models_{\mathcal{D}} fresh(sv)$ then we have $v' \models_{\mathcal{D}} fresh(\xi)$. We assume that the problem of checking $sv \models^{sym} \xi$ is decidable. The satisfaction relation $\models^{sym}$ can also be extended to infinite sequences $\rho : \mathbb{N} \to SV(\phi)$ (or, equivalently, $\rho \in SV(\phi)^\omega$) of symbolic valuations; it is the same as $\models$ for all temporal operators except for atomic formulae:

$$\rho, i \models^{sym} \xi \text{ iff } \rho(i) \models^{sym} \xi.$$

Then, given a CLTLB($\mathcal{D}$) formula ϕ, we say that a symbolic model ρ *symbolically satisfies* ϕ (or ρ is a *symbolic model* for ϕ) when $\rho, 0 \stackrel{sym}{\models} \phi$.

In the rest of this section, we consider CLTLB($\mathcal{D}$) formulae that do not include arithmetic temporal operator Y. This is without loss of generality, as Property 3 will show.

Definition 3. A pair of symbolic valuations (sv_1, sv_2) for ϕ is *locally consistent* if, for all R in $\mathcal{D}$, for all $i_i, \ldots, i_n$:

$$R(\mathrm{X}^{i_1} x_1, \ldots, \mathrm{X}^{i_n} x_n) \in sv_1 \text{ iff } R(\mathrm{X}^{i_1 - 1} x_1, \ldots, \mathrm{X}^{i_n - 1} x_n) \in sv_2$$

with $i_j \geq 1$ for all $j \in [1, n]$. A sequence of symbolic valuations $sv_0 sv_1 \ldots$ is *locally consistent* if all pairs (sv_i, sv_{i+1}), $i \geq 0$, are locally consistent.

A locally consistent infinite sequence ρ of symbolic valuations *admits an arithmetic model*, if there exists a $\mathcal{D}$-valuation sequence σ such that $\sigma, i \models \rho(i)$, for all $i \geq 0$. In this case, we write $\sigma, 0 \models \rho$.

We recall some fundamental results of CLTL($\mathcal{D}$), which nonetheless hold also for CLTLB($\mathcal{D}$).

The following proposition draws a link between the satisfiability by sequences of symbolic valuations and by sequences of $\mathcal{D}$-valuations.

Proposition 1 ([9])**.** *A CLTL($\mathcal{D}$) formula ϕ is satisfiable if, and only if, there exists a symbolic model for ϕ which admits an arithmetic model, i.e., there exist ρ and σ such that $\rho, 0 \stackrel{sym}{\models} \phi$ and $\sigma, 0 \models \rho$.*

Following [9], for constraint systems of the form $(D, <, =)$, where $<$ is a strict total ordering on D, it is possible to represent a symbolic valuation sv by its labeled directed graph $G_{sv} = (terms(\phi) \cup const(\phi), \tau)$, $\tau \subseteq terms(\phi) \cup const(\phi) \times \{<, =\} \times terms(\phi) \cup const(\phi)$, such that $(x, \sim, y) \in \tau$ if, and only if, $x \sim y \in sv$. This construction extends also to any locally consistent sequence ρ of symbolic valuations: It is possible to represent ρ via a graph G_ρ, obtained by superimposition of the graphs corresponding to the symbolic evaluations $\rho(i)$. Formally, $G_\rho = ((V \cup const(\phi)) \times \mathbb{N}, \tau_\rho)$, where $((x, i), \sim, (y, j)) \in \tau_\rho$ if, and only if, either $i \leq j$ and $(x \sim \mathrm{X}^{j-i} y) \in \rho(i)$, or $i > j$ and $(\mathrm{X}^{i-j} x \sim y) \in \rho(j)$.

An infinite path $d : \mathbb{N} \to (V \cup const(\phi)) \times \mathbb{N}$ in G_ρ, is called a *forward* (resp. *backward*) path if:

1. for all $i \in \mathbb{N}$, there is an edge from $d(i)$ to $d(i+1)$ (resp., an edge from $d(i+1)$ to $d(i)$);
2. for all $i \in \mathbb{N}$, if $d(i) = (x, j)$ and $d(i+1) = (x', j')$, then $j \leq j'$.

A forward (resp. backward) path is *strict* if there exist infinitely many i for which there is a <-labeled edge from $d(i)$ to $d(i+1)$ (resp., from $d(i+1)$ to $d(i)$). Intuitively, a (strict) forward path represents a sequence of (strict) monotonic increasing values whereas a (strict) backward path represents a sequence of (strict) monotonic decreasing values.

Given a CLTL($\mathcal{D}$) formula ϕ, it is possible [9] to define a Büchi automaton $\mathcal{A}_\phi$ recognizing the symbolic models of ϕ, thus reducing the satisfiability of ϕ to the

non-emptiness of $\mathcal{A}_\phi$. The idea is that automaton $\mathcal{A}_\phi$ accepts the intersection of the following languages, which defines exactly the language of symbolic models of ϕ:

(1) the language of symbolic models ρ for ϕ;

(2) the language of sequences of locally consistent symbolic valuations;

(3) the language of sequences of symbolic valuations which admit an arithmetic model.

Language (1) is accepted by the Vardi-Wolper automaton $\mathcal{A}_s$ of ϕ [14], while language (2) is recognized by the automaton $\mathcal{A}_\ell = (SV(\phi), sv_0, \rightarrow, SV(\phi))$, where the states are $SV(\phi)$, all accepting; sv_0 is the initial state; and the transition relation is such that $sv_i \xrightarrow{sv_i} sv_{i+1}$ if, and only if, all pairs (sv_i, sv_{i+1}) are locally consistent [9].

If the constraint system we are considering has the *completion property* (defined next), then all sequences of locally consistent symbolic valuations admit an arithmetic model, and condition (3) reduces to (2).

2.4.1. Completion property

Each automaton involved in the definition of $\mathcal{A}_\phi$ has the function of "filtering" sequences of symbolic valuations so that: 1) they are locally consistent, 2) they satisfy an LTL property and 3) they admit an arithmetic model. For some constraint systems, admitting an arithmetic model is just a consequence of local consistency. A constraint system $\mathcal{D}$ has the *completion* property if, given:

(i) a symbolic valuation sv over a finite set of terms $terms(\phi) \cup const(\phi)$,

(ii) a subset $A' \subseteq fresh(terms(\phi))$

(iii) a valuation v' over A' such that $v' \models_{\mathcal{D}} fresh(sv')$, where sv' is the subset of constraints in sv which uses only variables in A'

then there exists a valuation v over $fresh(terms(\phi))$ extending v' such that $v \models_{\mathcal{D}} fresh(sv)$. An example of such a relational structure is $(\mathbb{R}, <, =)$.

Let $(D, <, =)$ be a relational structure. We say that D is *dense*, with respect to the order $<$, if for each $d, d' \in D$ such that $d < d'$, there exists $d'' \in D$ such that $d < d'' < d'$, whereas D is said to be *open* when for each $d \in D$, there exist two elements $d', d'' \in D$ such that $d' < d < d''$.

Lemma 1 (Lemma 5.3, [9])**.** *A relational structure $\mathcal{D} = (D, <, =)$, where D is infinite and $<$ is a total order, satisfies the completion property if, and only if, domain D is dense and open.*

The following result relies on the fact that for $\mathcal{D}$ every locally consistent sequence of symbolic valuations admits an arithmetic model.

Proposition 2. *Let $\mathcal{D}$ be a relational structure satisfying the completion property and ϕ be a CLTL($\mathcal{D}$) formula. Then, the language of sequences of symbolic valuations which admit an arithmetic model is ω-regular.*

In this case the automaton $\mathcal{A}_\phi$ that recognizes exactly all the sequences of symbolic valuations which are symbolic models of ϕ is defined by the intersection (*à la* Büchi) $\mathcal{A}_\phi = \mathcal{A}_s \cap \mathcal{A}_\ell$.

In general, however, language (3) may *not* be ω-regular. In some cases, however, it is possible to build an automaton $\mathcal{A}_C$ which captures a sufficient and necessary condition on sequences of symbolic valuations guaranteeing the existence of a sequence σ such that $\sigma, 0 \models \rho$. More precisely, for some constraint systems it is possible, given a formula ϕ, to build an automaton $\mathcal{A}_C$ recognizing sequences of symbolic valuations such that the language of automaton $\mathcal{A}_\phi = \mathcal{A}_s \cap \mathcal{A}_\ell \cap \mathcal{A}_C$ is empty if, and only if, ϕ is unsatisfiable.

For the constraint systems considered in this paper, $\mathcal{A}_C$ can effectively be built. In particular, if the constraint system is of the form $(D, <, =)$, with $D \in \{\mathbb{N}, \mathbb{Z}\}$, automaton $\mathcal{A}_C$ recognizes sequences ρ of symbolic valuations that satisfy the following property:

Property 1. *There do not exist vertices u and v in the same symbolic valuation in G_ρ satisfying all the following conditions:*

1. *there is an infinite forward path d from u;*
2. *there is an infinite backward path e from v;*
3. *d or e are strict;*
4. *for each $i, j \in \mathbb{N}$, whenever $d(i)$ and $e(j)$ belong to the same symbolic valuation, there exists an edge, labeled by $<$, from $d(i)$ to $e(j)$.*

Informally, Property 1 guarantees that in the model, for every pair of an infinite forward path and an infinite backward path, there is a position such that, from that point on, the elements on the forward path are greater than the elements on the backward path.

A fundamental lemma, on which Proposition 3 below relies, shows that, for constraint system $(D, <, =)$, ultimately periodic sequences of symbolic valuations that satisfy Property 1 admit an arithmetic model.

Lemma 2 ([9])**.** *Let $(D, <, =)$, with $D \in \{\mathbb{N}, \mathbb{Z}\}$, be a constraint system and let ρ be a locally consistent, ultimately periodic sequence of symbolic valuations of the form $\rho = \alpha\beta^\omega \in SV(\phi)^\omega$. Then, $\sigma, 0 \models \rho$ (i.e., ρ admits an arithmetic model σ) if, and only if, ρ satisfies Property 1.*

Therefore, the satisfiability problem can be solved by checking the emptiness of the language recognized by the automaton $\mathcal{A}_\phi = \mathcal{A}_s \cap \mathcal{A}_\ell \cap \mathcal{A}_C$, where $\mathcal{A}_C$ recognizes sequences of symbolic valuations satisfying Property 1.

Proposition 3 ([9])**.** *Consider $\mathcal{D} = (D, <, =)$, with $D \in \{\mathbb{N}, \mathbb{Z}\}$. A CLTL($\mathcal{D}$) formula ϕ is satisfiable if, and only if, the language $\mathscr{L}(\mathcal{A}_\phi)$ is not empty.*

In the next section, we provide a way for checking the satisfiability of CLTLB($\mathcal{D}$) formulae that does not require the construction of automata $\mathcal{A}_s$, $\mathcal{A}_\ell$ and $\mathcal{A}_C$. Our approach takes advantage of the semantics of CLTLB($\mathcal{D}$) for building models of formulae through a semi-symbolic construction. We use a reduction to a Satisfiability

Modulo Theories (SMT) problem which extends the one proposed for Bounded Model Checking [15]. In the automata-based construction, the definition of automaton $\mathcal{A}_\phi$ may be prohibitive in practice and requires to devise alternative ways that avoid the exhaustive enumeration of all its states. In fact, the size of $\mathcal{A}_s$ is exponential in the size of the formula; moreover, when the constraint system is $(\mathbb{Z}, <, =)$ (which does not have the completion property) the automaton $\mathcal{A}_C$ is defined by complementing, e.g. through Safra's algorithm, automaton $\mathcal{A}_{\neg C}$ recognizing symbolic sequences satisfying the negation of Property 1 [9]. However, in many cases the complete construction of $\mathcal{A}_\phi$ is useless, since to show the satisfiability of a formula one can just exhibit an ultimately periodic model, whose length may be much smaller than the size of $\mathcal{A}_\phi$. On the other hand, proving unsatisfiability is comparable in complexity building $\mathcal{A}_\phi$, because it requires to verify that no ultimately periodic model $\alpha\beta^\omega$ can be constructed for a size $|\alpha\beta|$ equal to the size of $\mathcal{A}_\phi$.

Motivated by the arguments above, we define the bounded satisfiability problem, which consists in looking for a ultimately periodic symbolic model $\alpha\beta^\omega$ such that its prefix $\alpha\beta$ has fixed length (which is an input of the problem) and admits a *finite* arithmetic model σ_k. Since symbolic valuations partition the space of variable valuations, an assignment of values to terms uniquely identifies a symbolic valuation (see next Lemma 3). For this reason, we do not need to precompute the set $SV(\phi)$ and instead we enforce the periodicity between a pair of sets of relations, those defining the first and last symbolic valuations in β. We show that, when a formula ϕ is boundedly satisfiable, then it is also satisfiable. We provide a (polynomial-space) reduction from the bounded satisfiability problem to the satisfiability of formulae in the quantifier-free theory of equality and uninterpreted functions QF-EUF combined with $\mathcal{D}$.

3. Satisfiability of CLTLB($\mathcal{D}$) without automata

In this section, we introduce our novel technique to solve the satisfiability problem of CLTLB($\mathcal{D}$) formulae without resorting to an automata-theoretic construction.

First, we provide the definition of the k-satisfiability problem for CLTLB($\mathcal{D}$) formulae in terms of the existence of a so-called k-bounded arithmetic model σ_k, which is the basis to provide a finite representation of infinite symbolic models by means of ultimately periodic words. This allows us to prove that k-satisfiability is still representative of the satisfiability problem as defined in Section 2.3. In fact, for some constraint systems, a bounded solution can be used to build the infinite model σ for the formula from the k-bounded one σ_k and from its symbolic model. We show in Section 3.4 that a formula ϕ is satisfiable if, and only if, it is k-satisfiable and its bounded solution σ_k can be used to derive its infinite model σ. In case of negative answer to a k-bounded instance, we cannot immediately deduce the unsatisfiability of the formula. However, we prove in Section 5 that for every formula ϕ there exists an upper bound K, which can effectively be determined, such that if ϕ is not k-satisfiable for all k in $[1, K]$ then ϕ is unsatisfiable.

3.1. Bounded Satisfiability Problem

We first define the Bounded Satisfiability Problem (BSP), by considering bounded symbolic models of CLTLB($\mathcal{D}$) formulae. For simplicity, we consider the set $\mathcal{R}_0$ of

propositional letters to be empty; later, in Section 3.4 (Property 2), we show that this is without loss of generality. Informally, a bounded symbolic model is a finite representation of infinite CLTLB($\mathcal{D}$) models over the alphabet of symbolic valuations $SV(\phi)$. We restrict the analysis to ultimately periodic symbolic models, i.e., of the form $\rho = \alpha\beta^\omega$. Without loss of generality, we consider models where $\alpha = \alpha' s$ and $\beta = \beta' s$ for some symbolic valuation s. BSP is defined with respect to a *k-bounded model* $\sigma_k : \{\lfloor\phi\rfloor, \ldots, k + \lceil\phi\rceil\} \times V \to D$, a finite sequence ρ' (with $|\rho'| = k + 1$) of symbolic valuations and a k-bounded satisfaction relation $\models_k$ defined as follows:

$$\sigma_k, 0 \models_k \rho' \text{ iff } \sigma_k, i \models \rho'(i) \text{ for all } 0 \leq i \leq k.$$

The *k-satisfiability problem* of formula ϕ is defined as follows:

Input A CLTLB($\mathcal{D}$) formula ϕ, a constant $k \in \mathbb{N}$

Problem Is there an ultimately periodic sequence of symbolic valuations $\rho = \alpha\beta^\omega$ with $|\alpha\beta| = k + 1$, $\alpha = \alpha' s$ and $\beta = \beta' s$, such that:

- $\rho, 0 \overset{sym}{\models} \phi$ and
- there is a k-bounded model σ_k for which $\sigma_k, 0 \models_k \alpha\beta$?

Since k is fixed, the procedure for determining the satisfiability of CLTLB($\mathcal{D}$) formulae over bounded models is not complete: even if there is no accepting run of automaton $\mathcal{A}_\phi$ when ρ' as above has length k, there may be accepting runs for a larger ρ'.

Definition 4. Given a CLTLB($\mathcal{D}$) formula ϕ, its *completeness threshold* K_ϕ, if it exists, is the smallest integer such that ϕ is satisfiable if and only if ϕ is K_ϕ-satisfiable.

3.2. Avoiding explicit symbolic valuations

The next, fundamental Lemma 3 and Lemma 4 allow us to avoid the definition of set $SV(\phi)$ and to derive symbolic models for ϕ through σ_k. In particular, Lemma 4 shows how to build a sequence of symbolic valuations from σ_k.

Lemma 3. *Let $\mathcal{D} = (D, \mathcal{R})$ be a constraint system, ϕ be a CLTLB($\mathcal{D}$) formula and v be a $\mathcal{D}$-valuation extended to $terms(\phi)$. Then, there is a unique symbolic valuation sv such that $v \models_\mathcal{D} sv$.*

Proof. By contradiction, suppose there are two symbolic valuations, sv and sv', such that $sv \neq sv'$, $v \models_\mathcal{D} sv$ and $v \models_\mathcal{D} sv'$. Since $sv \neq sv'$, then there exist a relation R of arity $n \geq 0$ and a tuple $(\alpha_1, \ldots, \alpha_n)$ such that $R(\alpha_1, \ldots, \alpha_n) \in sv$ and $R(\alpha_1, \ldots, \alpha_n) \notin sv'$. Since $R(\alpha_1, \ldots, \alpha_n) \in sv$, by definition of symbolic valuation $v \models_\mathcal{D} R(\alpha_1, \ldots, \alpha_n)$. By definition a symbolic valuation built from v contains all $\mathcal{D}$-constraints satisfied by v, hence it must also be that $R(\alpha_1, \ldots, \alpha_n) \in sv'$, a contradiction. □

Corollary 1. *Let ϕ be a CLTLB($\mathcal{D}$) formula, v be a $\mathcal{D}$-valuation extended to $terms(\phi)$ and sv be a symbolic valuation in $SV(\phi)$. If $v \models_\mathcal{D} sv$ then for all relations $R \in \mathcal{R}$*

$$sv \overset{sym}{\models} R(\alpha_1, \ldots, \alpha_n) \textit{ iff } v \models_\mathcal{D} R(\alpha_1, \ldots, \alpha_n).$$

Proof. Suppose that $sv \overset{sym}{\models} R(\alpha_1, \ldots, \alpha_n)$. By definition of $sv \overset{sym}{\models} R(\alpha_1, \ldots, \alpha_n)$, for every $\mathcal{D}$-valuation v' over $terms(\phi)$ such that $v' \models_{\mathcal{D}} sv$, then $v' \models_{\mathcal{D}} R(\alpha_1, \ldots, \alpha_n)$ holds. Therefore, also $v \models_{\mathcal{D}} R(\alpha_1, \ldots, \alpha_n)$. The converse is an immediate consequence of the definition of symbolic valuation. □

Lemma 4. *Let ϕ be a CLTLB($\mathcal{D}$) formula and σ_k be a finite sequence of $\mathcal{D}$-valuations. Then, there exists a unique locally consistent sequence $\rho \in SV(\phi)^{k+1}$ such that $\sigma_k, i \models \rho(i)$, for all $i \in [0, k]$.*

Proof. By Lemma 3 it follows that, for all $i \in [0, k]$, the assignment of variables defined by σ_k is such that $\sigma_k, i \models \rho(i)$ and $\rho(i)$ is unique. By Corollary 1, values in σ_k from position i satisfy a relation R with arguments $(\alpha_1, \ldots, \alpha_n)$ at position i if, and only if, R belongs to symbolic valuation $\rho(i)$, i.e., $\rho(i) \overset{sym}{\models} R(\alpha_1, \ldots, \alpha_n)$ if, and only if, $\sigma_k, i \models_{\mathcal{D}} R(\alpha_1, \ldots, \alpha_n)$. In addition, any two adjacent symbolic valuations $\rho(i)$ and $\rho(i+1)$ are locally consistent, i.e., both $R(\mathrm{X}^{i_1} x_1, \ldots, \mathrm{X}^{i_n} x_n) \in \rho(i)$ and $R(\mathrm{X}^{i_1 - 1} x_1, \ldots, \mathrm{X}^{i_n - 1} x_n) \in \rho(i+1)$. In fact, the evaluation in σ_k of an arithmetic term $\mathrm{X}^{i_j} x_j$ in position i is the same as the evaluation of $\mathrm{X}^{i_j - 1} x_j$ in position $i+1$. □

3.3. An encoding for BSP without automata

We now show how to encode a CLTLB($\mathcal{D}$) formula into a quantifier-free formula in the theory EUF $\cup$ $\mathcal{D}$ (called QF-EU$\mathcal{D}$), where EUF is the theory of Equality and Uninterpreted Functions. This is the basis for reducing the BSP for CLTLB($\mathcal{D}$) to the satisfiability of QF-EU$\mathcal{D}$, as proved in Section 3.4. Satisfiability of QF-EU$\mathcal{D}$ is decidable, provided that $\mathcal{D}$ includes a copy of $\mathbb{N}$ with the successor relation and that EUF$\cup\mathcal{D}$ is consistent, as in our case. The latter condition is easily verified in the case of the union of two consistent, disjoint, stably infinite theories (as is the case for EUF and arithmetic). [16] describes a similar approach for the case of Integer Difference Logic (DL) constraints. It is worth noting that standard LTL can be encoded by a formula in QF-EU$\mathcal{D}$ with $\mathcal{D} = (\mathbb{N}, <)$, rather than in Boolean logic [17], resulting in a more succinct encoding.

The encoding presented below represents ultimately periodic sequences of symbolic valuations ρ of the form $sv_0 sv_1 \ldots sv_{loop-1}(sv_{loop} \ldots sv_k)^\omega$. Therefore, we look for a finite word $\rho' = sv_0 sv_1 \ldots sv_{loop-1}(sv_{loop} \ldots sv_k) sv_{loop}$ of length $k+2$ representing the ultimately periodic model above. Instant $k+1$ in the encoding is used to correctly represent the periodicity of ρ by constraining atomic formulae (propositions and relations) at positions $loop$ and $k+1$. Thanks to the periodicity of suffix $(sv_{loop} \ldots sv_k)$, we can solve the BSP by considering the following decomposition $\alpha\beta^\omega = sv_0 sv_1 \ldots sv_{loop}(sv_{loop+1} \ldots sv_k sv_{loop})^\omega$ where $\alpha = \alpha' sv_{loop}$ and $\beta = \beta' sv_{loop}$ with $\alpha' = sv_0 sv_1 \ldots sv_{loop-1}$ and $\beta' = sv_{loop+1} \ldots sv_k$.

Encoding terms. Given a term α in $terms(\phi)$, we associate an *arithmetic formula function* $\boldsymbol{\alpha}$ with α, which is a unary function denoted by the same name of the term but written in boldface. Note that if α is a variable $x \in V$, then $\boldsymbol{\alpha}$ is $\boldsymbol{x}$. Function $\boldsymbol{\alpha}$ must obey the following constraints:

α	$0 \leq i < k+1$
$\mathrm{X}\alpha'$	$\boldsymbol{\alpha}(i) = \boldsymbol{\alpha'}(i+1)$

α	$0 < i \leq k+1$
$\mathrm{Y}\alpha'$	$\boldsymbol{\alpha}(i) = \boldsymbol{\alpha'}(i-1)$

The conjunction of the above subformulae gives formula $|ArithConstraints|_k$. Implementing $|ArithConstraints|_k$ is straightforward. In fact, the assignments of values to variables are defined by the interpretation of the symbols of the QF-EU$\mathcal{D}$ formula. The values of a variable x at positions before 0 and after k, i.e. in intervals $[\lfloor\phi\rfloor, -1]$ and $[k+1, k+\lceil\phi\rceil]$, are defined by means of the values of terms $\alpha = \mathrm{X}^i x$ and $\alpha = \mathrm{Y}^i x$. For instance, the value of x at position $0 > i \geq \lfloor\phi\rfloor$ is $\sigma_k(i, x)$, but it is defined by the assignment for term $\alpha = \mathrm{Y}^i x$ at position 0.

Encoding formulae. The truth value of a CLTLB formula is defined with respect to the truth value of its subformulae. Given a subformula θ of ϕ, we introduce a *formula predicate* $\boldsymbol{\theta}$. When the subformula θ holds at instant i then $\boldsymbol{\theta}(i)$ holds.

We first define $\boldsymbol{\theta}$ for atomic formulae and their negations. Let R be an n-ary relation of $\mathcal{R}$ that appears in ϕ, and let $\alpha_1, \ldots \alpha_n$ be a.t.t.'s. Let p be a propositional letter. We define $\boldsymbol{\theta}$ for every subformula θ of ϕ of the form $R, \neg R, p, \neg p$ as follows (where, if α_j is a constant $c \in const(\phi)$, then $\boldsymbol{\alpha_j}$ is simply c):

θ	$\boldsymbol{\theta}(i)$
$R(\alpha_1, \ldots, \alpha_n)$	$R(\boldsymbol{\alpha_1}(i), \ldots, \boldsymbol{\alpha_n}(i))$
$\neg R(\alpha_1, \ldots, \alpha_n)$	$\neg R(\boldsymbol{\alpha_1}(i), \ldots, \boldsymbol{\alpha_n}(i))$
p	$\boldsymbol{p}(i)$
$\neg p$	$\neg \boldsymbol{p}(i)$

When θ is not of the form $R, \neg R, p, \neg p$, then $\boldsymbol{\theta}$ is a unary predicate letter denoted by the same name of the formula but written in boldface. As the last position of a path is fixed to $k+1$ and all paths start from 0, formula predicates are actually subsets of $\{0, \ldots, k+1\}$. We define the constraints on formula predicate $\boldsymbol{\theta}$ recursively as in the following tables. For brevity and ease of reading in each row of the second column the formula predicate associated with the formula in the left column is denoted with $\boldsymbol{\theta}$, rather than with the boldface name of the formula itself.

θ	$0 \leq i \leq k+1$
$\psi_1 \wedge \psi_2$	$\boldsymbol{\theta}(i) \Leftrightarrow \boldsymbol{\psi_1}(i) \wedge \boldsymbol{\psi_2}(i)$
$\psi_1 \vee \psi_2$	$\boldsymbol{\theta}(i) \Leftrightarrow \boldsymbol{\psi_1}(i) \vee \boldsymbol{\psi_2}(i)$

The conjunction of the formulae above is formula $|PropConstraints|_k$. The temporal behavior of future and past operators is encoded in formula $|TempConstraints|_k$ by using their traditional fixpoint characterizations. More precisely, $|TempConstraints|_k$ is the conjunction of the following formulae, for each temporal subformula θ:

θ	$0 \leq i \leq k$
$\mathbf{X}\psi$	$\boldsymbol{\theta}(i) \Leftrightarrow \boldsymbol{\psi}(i+1)$
$\psi_1 \mathbf{U} \psi_2$	$\boldsymbol{\theta}(i) \Leftrightarrow (\boldsymbol{\psi_2}(i) \vee (\boldsymbol{\psi_1}(i) \wedge \boldsymbol{\theta}(i+1)))$
$\psi_1 \mathbf{R} \psi_2$	$\boldsymbol{\theta}(i) \Leftrightarrow (\boldsymbol{\psi_2}(i) \wedge (\boldsymbol{\psi_1}(i) \vee \boldsymbol{\theta}(i+1)))$

θ	$0 < i \leq k+1$	$i = 0$
$\mathbf{Y}\psi$	$\boldsymbol{\theta}(i) \Leftrightarrow \boldsymbol{\psi}(i-1)$	$\boldsymbol{\theta}(0) \Leftrightarrow false$
$\mathbf{Z}\psi$	$\boldsymbol{\theta}(i) \Leftrightarrow \boldsymbol{\psi}(i-1)$	$\boldsymbol{\theta}(0) \Leftrightarrow true$
$\psi_1 \mathbf{S} \psi_2$	$\boldsymbol{\theta}(i) \Leftrightarrow (\boldsymbol{\psi_2}(i) \vee (\boldsymbol{\psi_1}(i) \wedge \boldsymbol{\theta}(i-1)))$	$\boldsymbol{\theta}(0) \Leftrightarrow \boldsymbol{\psi_2}(0)$
$\psi_1 \mathbf{T} \psi_2$	$\boldsymbol{\theta}(i) \Leftrightarrow (\boldsymbol{\psi_2}(i) \wedge (\boldsymbol{\psi_1}(i) \vee \boldsymbol{\theta}(i-1)))$	$\boldsymbol{\theta}(0) \Leftrightarrow \boldsymbol{\psi_2}(0)$

Encoding periodicity. To represent ultimately periodic sequences of symbolic valuations we use a positive integer variable $\boldsymbol{loop} \in [0, k]$ that captures the position in which the loop starts in $sv_0 sv_1 \dots sv_{loop-1}(sv_{loop} \dots sv_k)^\omega$. Informally, if the value of variable $\boldsymbol{loop}$ is i, then there exists a loop which starts at i. To encode the loop we require $sv_{loop} = sv_{k+1}$; this is achieved through the following formula $|LoopConstraints|_k$, which ranges over all relations $R \in \mathcal{R}$ and all terms in $terms(\phi)$, including those that do not appear in ϕ:

$$\bigwedge_{\substack{\theta = R(\alpha_1, \dots, \alpha_n) \\ R \in \mathcal{R}, \alpha_1, \dots, \alpha_n \in terms(\phi)}} \boldsymbol{\theta}(\boldsymbol{loop}) \Leftrightarrow \boldsymbol{\theta}(k+1).$$

Last state constraints (captured by formula $|LastStateConstraints|_k$) define the equivalence between the truth values of the subformulae of ϕ at position $k+1$ and those at the position indicated by the $\boldsymbol{loop}$ variable, since the former position is representative of the latter along periodic paths. These constraints have a similar structure as those in the Boolean encoding of [17]; for brevity, we consider only the case of infinite periodic words, as the case of finite words can easily be defined if needed. Hence, last state constraints are introduced through the following formula (where $sub(\phi)$ indicates the set of subformulae of ϕ) by adding only *one* constraint for each subformula θ of ϕ.

$$\textstyle\bigwedge_{\theta \in sub(\phi)} \boldsymbol{\theta}(k+1) \Leftrightarrow \boldsymbol{\theta}(\boldsymbol{loop}).$$

Eventualities for $\mathbf{U}$ *and* $\mathbf{R}$. To correctly define the semantics of $\mathbf{U}$ and $\mathbf{R}$, their *eventualities* have to be accounted for. Briefly, if $\psi_1 \mathbf{U} \psi_2$ holds at i, then ψ_2 eventually holds in some $j \geq i$; if $\psi_1 \mathbf{R} \psi_2$ does not hold at i, then ψ_2 eventually does not hold in some $j \geq i$. The Boolean encoding of [17] introduces a propositional variable for each subformula of the form $\psi_1 \mathbf{U} \psi_2$ or $\psi_1 \mathbf{R} \psi_2$ and for each position in the finite model, to represent the eventuality of ψ_2 implicit in the formula. Instead, in the QF-EU$\mathcal{D}$ encoding, only *one* variable $\boldsymbol{j}_{\boldsymbol{\psi_1 \mathbf{U} \psi_2}} \in D$ is introduced for each subformula $\psi_1 \mathbf{U} \psi_2$ and only *one* variable $\boldsymbol{j}_{\boldsymbol{\psi_1 \mathbf{R} \psi_2}} \in D$ for each subformula $\psi_1 \mathbf{R} \psi_2$.

θ	
$\psi_1 \mathbf{U} \psi_2$	$\boldsymbol{\theta}(k) \Rightarrow \boldsymbol{loop} \leq \boldsymbol{j}_{\boldsymbol{\psi_1 \mathbf{U} \psi_2}} \leq k \wedge \boldsymbol{\psi_2}(\boldsymbol{j}_{\boldsymbol{\psi_1 \mathbf{U} \psi_2}})$
$\psi_1 \mathbf{R} \psi_2$	$\neg\boldsymbol{\theta}(k) \Rightarrow \boldsymbol{loop} \leq \boldsymbol{j}_{\boldsymbol{\psi_1 \mathbf{R} \psi_2}} \leq k \wedge \neg\boldsymbol{\psi_2}(\boldsymbol{j}_{\boldsymbol{\psi_1 \mathbf{R} \psi_2}})$

The conjunction of the constraints above for all subformulae θ of ϕ constitutes the formula $|Eventually|_k$.

The complete encoding $|\phi|_k$ of ϕ consists of the logical conjunction of all above components, together with $\boldsymbol{\phi}(0)$.

3.4. Correctness of the BSP encoding

To prove the correctness of the encoding defined in Section 3.3, we first introduce two properties, which reduce CLTLB($\mathcal{D}, \mathcal{R}_0$) to CLTLB($\mathcal{D}$) without Y operators. This allows us to base our proof on the automata-based construction for CLTLB($\mathcal{D}$) of [9]. In particular, the two reductions are essential to take advantage of Proposition 2 and Lemma 2 of Section 2, to define a decision procedure for the bounded satisfiability problem of Section 3.1. The properties are almost obvious, hence we only provide the intuition behind their proof (see [18] for full details).

Property 2. *CLTLB($\mathcal{D}, \mathcal{R}_0$) formulae can be equivalently rewritten into CLTLB($\mathcal{D}$) formulae.*

According to the definition given in Section 2.2, CLTLB($\mathcal{D}$) is the language CLTLB where atomic formulae belong to the language of constraints in $\mathcal{D}$, which may contain also 0-ary relations. In this case, atomic formulae are propositions $p \in \mathcal{R}_0$ or relations $R(\alpha_1, \ldots, \alpha_n)$. Any positive occurrence of an atomic proposition $p \in \mathcal{R}_0$ in a CLTLB formula can be replaced by an equality relation of the form $x_p = 1$. Then, a formula of CLTLB($\mathcal{D}, \mathcal{R}_0$) can be easily rewritten into a formula of CLTLB($\mathcal{D}$) preserving their equivalence (modulo the rewriting of propositions in $\mathcal{R}_0$). We define a rewriting function np over formulae such that $(\pi', \sigma'), 0 \models \phi$ if, and only if, $(\pi, \sigma), 0 \models np(\phi) \wedge \psi$ where σ is the same as σ' except for new fresh variables x_p representing atomic propositions, and ψ is a formula restricting the values of variables x_p to $\{0, 1\}$.

For instance, let ϕ be the formula $\mathbf{G}(p \Rightarrow \mathbf{F}(\mathrm{X}x < y \wedge q))$, where the "eventually" ($\mathbf{F}$) and "globally" ($\mathbf{G}$) operators are defined as usual. The formula obtained by means of rewriting np is

$$\mathbf{G}(x_p = 1 \Rightarrow \mathbf{F}(\mathrm{X}x < y \wedge x_q = 1)) \wedge \mathbf{G} \left(\begin{array}{c} (x_p = 1 \vee x_p = 0) \\ \wedge \\ (x_q = 1 \vee x_q = 0) \end{array} \right).$$

Note that formula $np(\phi)$ does not contain any propositional letters, so in a model (π, σ) component π associates with each instant the empty set. From now on we will consider only CLTLB($\mathcal{D}$) formulae without propositional letters; hence, given a propositional letter-free formula ϕ, we will write $\sigma, 0 \models \phi$ instead of $(\pi, \sigma), 0 \models \phi$.

Property 3. *CLTLB($\mathcal{D}$) formulae can be equivalently rewritten into CLTLB($\mathcal{D}$) formulae without* Y *operators.*

Let rw : CLTLB($\mathcal{D}$) $\rightarrow$ CLTLB($\mathcal{D}$) be the following syntactical rewriting, which transforms every formula ϕ into an equisatisfiable formula that does not contain any occurrence of the Y operator. Formula $rw(\phi)$ is identical to ϕ except that, for all $i \geq 0$ all a.t.t.'s of the form $\mathrm{X}^i x$ in ϕ are replaced by $\mathrm{X}^{i - \lfloor \phi \rfloor} x$, while all a.t.t.'s of the form $\mathrm{Y}^i x$ are replaced by $\mathrm{X}^{-i - \lfloor \phi \rfloor} x$ (where x is treated as $\mathrm{X}^0 x$). The latter replacement avoids negative indexes (since if ϕ contains a.t.t.'s of the form $\mathrm{Y}^i x$, then $0 \leq i \leq -\lfloor \phi \rfloor$). The rw function can be naturally extended to symbolic valuations (i.e, sets of atomic constraints) and sequences ρ thereof.

As a consequence, given a CLTLB($\mathcal{D}$) formula ϕ, it is easy to see that Y does not occur in $rw(\phi)$. The equisatisfiability of formulae ϕ and $rw(\phi)$ is guaranteed by

moving the model σ of $\lfloor \phi \rfloor$ instants. We define the sequence of $\mathcal{D}$-valuations $\sigma^{\lfloor \phi \rfloor}$ as follows:

$$\sigma^{\lfloor \phi \rfloor}(i, x) = \sigma(i + \lfloor \phi \rfloor, x),$$

for all $i \geq 0$ and $x \in V$.

Proposition 4. *Let ϕ be a CLTLB($\mathcal{D}$) formula, then $\sigma, 0 \models \phi$ iff $\sigma^{\lfloor \phi \rfloor}, 0 \models rw(\phi)$.*

Corollary 2. *Let $\rho \in SV(\phi)^\omega$ be a sequence of symbolic valuations. Then,*

$$\begin{array}{rcl} \sigma, 0 \models \rho & \textit{iff} & \sigma^{\lfloor \phi \rfloor}, 0 \models rw(\rho) \\ \rho, 0 \overset{sym}{\models} \phi & \textit{iff} & rw(\rho), 0 \overset{sym}{\models} rw(\phi). \end{array}$$

We now have all necessary elements to prove the correctness of our encoding. We first provide the following three equivalences, which are proved by showing the implications depicted in Figure 1, where $\mathcal{A}_s \times \mathcal{A}_\ell$ is the automaton recognizing locally consistent symbolic models of $rw(\phi)$:

1. Satisfiability of $|\phi|_k$ is equivalent to the existence of ultimately periodic runs of automaton $\mathcal{A}_s \times \mathcal{A}_\ell$.
2. k-satisfiability is equivalent to the existence of ultimately periodic runs of automaton $\mathcal{A}_s \times \mathcal{A}_\ell$.
3. k-satisfiability is equivalent to the satisfiability of $|\phi|_k$.

Then we draw, by Proposition 5, the connection between k-satisfiability and satisfiability for formulae over constraint systems satisfying the completion property. In Section 4, thanks to Proposition 6, we extend the result to constraint system IPC*, which does not have the completion property.

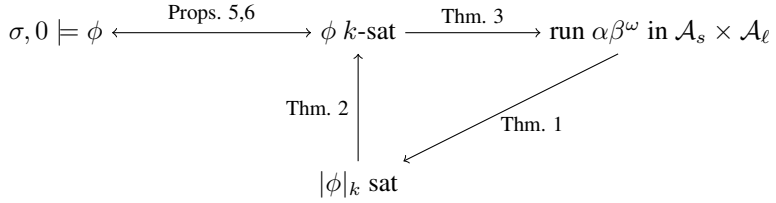

Figure 1: Proof schema.

Before tackling the theorems of Figure 1, we provide the definition of models for QF-EU$\mathcal{D}$ formulae $|\phi|_k$ built according to the encoding of Section 3.3. More precisely, a model $\mathcal{M}$ of $|\phi|_k$ is a pair $(D, \mathcal{I})$ where D is the domain of interpretation of $\mathcal{D}$, and $\mathcal{I}$ maps:

- each function symbol $\boldsymbol{\alpha}$ to a mapping from positions in time into D: $\mathcal{I}(\boldsymbol{\alpha}) : \mathbb{N} \to D$;
- each predicate symbol $\boldsymbol{\theta}$ to a mapping from positions in time into $\{true, false\}$, $\mathcal{I}(\boldsymbol{\theta}) : \mathbb{N} \to \{true, false\}$.

Note that mapping $\mathcal{I}$ trivially induces a finite sequence of $\mathcal{D}$-valuations $\sigma_k : \{\lfloor\phi\rfloor, \ldots, k + \lceil\phi\rceil\} \times V \to D$.

We start by showing that the existence of ultimately periodic runs of automaton $\mathcal{A}_s \times \mathcal{A}_\ell$ implies the satisfiability of $|\phi|_k$.

Theorem 1. *Let $\phi \in$ CLTLB($\mathcal{D}$) with $\mathbb{N}$ definable in $\mathcal{D}$ together with the successor relation. If there exists an ultimately periodic run $\rho = \alpha\beta^\omega$ ($|\alpha\beta| = k+1$) of $\mathcal{A}_s \times \mathcal{A}_\ell$ accepting symbolic models of $rw(\phi)$, then $|\phi|_k$ is satisfiable with respect to $k \in \mathbb{N}$.*

In the following proof, we use the generalized Büchi automaton obtained by the standard construction of [14], in the version of [9]. We slightly modify the construction in [9] to consider formulae of the form $\psi\mathbf{R}\zeta$ and $\psi\mathbf{T}\zeta$. This is useful to show the correspondence between the k-bounded satisfiability and the automata-based approach. Let ϕ' be a CLTLB($\mathcal{D}$) formula (without the Y modality over terms). The closure of ϕ', denoted $cl(\phi')$, is the smallest negation-closed set containing all subformulae of ϕ'. An *atom* $\Gamma \subseteq cl(\phi')$ is a maximally consistent set, i.e., such that for each subformula ψ and ζ of ϕ':

- $\psi \in \Gamma$ iff $\neg\psi \in \Gamma$,
- $\psi \wedge \zeta \in \Gamma$ iff $\psi, \zeta \in \Gamma$,
- $\psi \vee \zeta \in \Gamma$ iff $\psi \in \Gamma$ or $\zeta \in \Gamma$.

A pair (Γ_1, Γ_2) of atoms is *one-step temporally consistent* when:

- for every $\mathbf{X}\psi \in cl(\phi')$, then $\mathbf{X}\psi \in \Gamma_1$ iff $\psi \in \Gamma_2$,
- for every $\mathbf{Y}\psi \in cl(\phi')$, then $\mathbf{Y}\psi \in \Gamma_2$ iff $\psi \in \Gamma_1$,
- for every $\mathbf{Z}\psi \in cl(\phi')$, then $\mathbf{Z}\psi \in \Gamma_2$ iff $\psi \in \Gamma_1$,
- if $\psi\mathbf{U}\zeta \in \Gamma_1$, then $\zeta \in \Gamma_1$ or both $\psi \in \Gamma_1$ and $\psi\mathbf{U}\zeta \in \Gamma_2$,
- if $\psi\mathbf{R}\zeta \in \Gamma_1$, then $\{\psi, \zeta\} \in \Gamma_1$ or both $\zeta \in \Gamma_1$ and $\psi\mathbf{R}\zeta \in \Gamma_2$,
- if $\psi\mathbf{S}\zeta \in \Gamma_2$, then $\zeta \in \Gamma_2$ or both $\psi \in \Gamma_2$ and $\psi\mathbf{S}\zeta \in \Gamma_1$,
- if $\psi\mathbf{T}\zeta \in \Gamma_2$, then $\{\psi, \zeta\} \in \Gamma_2$ or both $\zeta \in \Gamma_2$ and $\psi\mathbf{T}\zeta \in \Gamma_1$.

The automaton $\mathcal{A}_s = (SV(\phi'), Q, Q_0, \eta, F)$ is then defined as follows:

- Q is the set of atoms;
- $Q_0 = \{\Gamma \in Q : \phi' \in \Gamma$ and $\neg\mathbf{Y}\psi \in \Gamma$ for all $\mathbf{Y}\psi \in cl(\phi')$ and $\mathbf{Z}\psi \in \Gamma$ for all $\mathbf{Z}\psi \in cl(\phi')$ and $\psi\mathbf{S}\zeta, \psi\mathbf{T}\zeta \in \Gamma$ iff $\zeta \in \Gamma\}$;
- $\Gamma_1 \xrightarrow{sv} \Gamma_2 \in \eta$ iff
  - $sv \models^{sym} \Gamma_1$
  - (Γ_1, Γ_2) is one-step consistent;

- $F = \{F_1, \ldots, F_p\}$, where $F_i = \{\Gamma \in Q \mid \psi_i \mathbf{U} \zeta_i \notin \Gamma \text{ or } \zeta_i \in \Gamma\}$ and $\{\psi_1 \mathbf{U} \zeta_1, \ldots, \psi_p \mathbf{U} \zeta_p\}$ is the set of Until formulae occurring in $cl(\phi')$.

Proof. We prove that if there is a run in $\mathcal{A}_s \times \mathcal{A}_\ell$ accepting symbolic models of $rw(\phi)$, then formula $|\phi|_k$ is satisfiable (we assume the rewriting obtained through function np, defined at the beginning of Section 3.4). Suppose there exists an ultimately periodic symbolic model of length $k+1$ which is accepted by $\mathcal{A}_s \times \mathcal{A}_\ell$. It is a locally consistent sequence of symbolic valuations, $\rho = \alpha\beta^\omega$ of the form:

$$\rho = sv_0 \ldots sv_{loop-1}(sv_{loop} \ldots sv_k)^\omega$$

such that $\rho \in \mathscr{L}(\mathcal{A}_s \times \mathcal{A}_\ell)$. Sequence ρ is recognized by a periodic run of $\mathcal{A}_s \times \mathcal{A}_\ell$ of the form[2]:

$$\upsilon = (\Gamma_0, sv_0) \ldots (\Gamma_{loop-1}, sv_{loop-1})((\Gamma_{loop}, sv_{loop}) \ldots (\Gamma_k, sv_k))^\omega.$$

For each subformula $\psi_i \mathbf{U} \zeta_i$ occurring in ϕ, the subrun $(\Gamma_{loop-1}, sv_{loop-1})(\Gamma_{loop}, sv_{loop}) \ldots (\Gamma_k, sv_k)$ visits control states of the set F_i, thus witnessing the acceptance condition of $\mathcal{A}_s$. From υ we build run γ of $\mathcal{A}_s$:

$$\gamma = \Gamma_0 \ldots \Gamma_{loop-1}(\Gamma_{loop} \ldots \Gamma_k)^\omega.$$

In particular, ρ is defined by the projection on the alphabet of $SV(rw(\phi))$ of the subformulae occurring in every Γ_i, for $0 \leq i \leq k$. Sequence ρ and its accepting run γ can be translated by means of rw^{-1} to obtain a symbolic model for ϕ. In particular, because $\rho, 0 \models^{sym} rw(\phi)$ then we obtain, by Corollary 2, $rw^{-1}(\rho), 0 \models^{sym} \phi$. Similarly, by rewriting all formulae in atoms of γ, we obtain an accepting run $rw^{-1}(\gamma)$ for ϕ. The model for $|\phi|_k$ is given by the truth value of all the subformulae in each $rw^{-1}(\Gamma_i)$ and the values of variables occurring in ϕ can be defined as explained later. In particular, we need to complete interpretation $\mathcal{I}$ for uninterpreted predicate and functions formulae: given a position $0 \leq i \leq k$, for all subformulae $\theta \in cl(\phi)$ we define

- $\mathcal{I}(\boldsymbol{\theta})(i) = true$ iff $\theta \in rw^{-1}(\Gamma_i)$,
- $\mathcal{I}(\boldsymbol{\theta})(i) = false$ iff $\neg\theta \in rw^{-1}(\Gamma_i)$.

To complete the interpretation of subformulae at position $k+1$ we can use values from position $loop$: $\mathcal{I}(\boldsymbol{\theta})(k{+}1) = \mathcal{I}(\boldsymbol{\theta})(loop)$. Note that by taking truth values of subformulae $\theta \in cl(\phi)$ from atoms $rw^{-1}(\Gamma_i)$, we obtain all constraints in $|propConstraints|_k$. The sequence ρ of symbolic valuations is consistent and all the a.t.t.'s in the encoding of $|\phi|_k$ can be uniquely defined by considering at each position i a symbolic valuation $rw^{-1}(sv_i)$. Consider the sequence $\rho' = sv_0 \ldots sv_{loop-1}(sv_{loop} \ldots sv_k)sv_{loop}$. Following [9, Lemma 5.2], we can build an edge-respecting assignment of values in D for the finite graph $G_{rw^{-1}(\rho')}$, which associates, for each variable $x \in V$ and for each

[2]For reasons of clarity, we avoid some details of product automaton $\mathcal{A}_s \times \mathcal{A}_\ell$, which are however inessential in the proof.

position $\lfloor \phi \rfloor \leq i \leq k + 1 + \lceil \phi \rceil$, a value $\sigma_k(i, x)$. We exploit assignment $\sigma_k(i, x)$ to define $\mathcal{I}(\boldsymbol{\alpha})$, with $\alpha \in terms(\phi)$, in the following way (where x_α is the variable in α):

$$\mathcal{I}(\boldsymbol{\alpha})(i) = \sigma_k(i + |\alpha|, x_\alpha)$$

for all $0 \leq i \leq k + 1$. Then, formulae $|ArithConstraints|_k$ are satisfied. Since run υ is ultimately periodic, then control state $(\Gamma_{loop}, sv_{loop})$ is visited at position $k + 1$. It witnesses the satisfaction of $|LastStateConstraints|_k$ formulae, which prescribe that θ_{k+1} iff θ_{loop} for all $\theta \in sub(\phi)$. Moreover, by the equality of sv_{k+1} and sv_{loop} in run υ we have that $R(\alpha_1, \ldots, \alpha_n)$ holds at $loop$ if, and only if, it holds at $k + 1$, hence we obtain $|LoopConstraints|_k$. Finally, let us consider $|Eventually|_k$ formulae. By construction, as run υ of $\mathcal{A}_s$ is accepting, if subformula $\psi \mathbf{U} \zeta$ belongs to atom Γ_i, then there exists a position $j \geq i$ such that ζ holds in j. Since the model is periodic, if $\psi \mathbf{U} \zeta$ belongs to atom Γ_k, then $k \leq j \leq k + |\beta|$, i.e., $\boldsymbol{j_{\psi \mathbf{U} \zeta}} = j - |\beta|$ is a position such that $loop \leq \boldsymbol{j_{\psi \mathbf{U} \zeta}} \leq k$ and $\zeta \in \Gamma_{\boldsymbol{j_{\psi \mathbf{U} \zeta}}}$. If $\neg(\psi \mathbf{R} \zeta)$ belongs to Γ_k then there exists a position $j \geq k$ such that $\neg\zeta$ holds in j. Since the model is periodic, if $\neg(\psi \mathbf{R} \zeta)$ belongs to atom Γ_k, then $k \leq j \leq k + |\beta|$, i.e., $\boldsymbol{j_{\psi \mathbf{R} \zeta}} = j - |\beta|$ is a position such that $loop \leq \boldsymbol{j_{\psi \mathbf{R} \zeta}} \leq k$ and $\neg\zeta \in \Gamma_{\boldsymbol{j_{\psi \mathbf{R} \zeta}}}$. Hence, the $|Eventually|_k$ formulae are satisfied. The initial atom Γ_0 is such that $\neg \mathbf{Y} \varphi \in rw^{-1}(\Gamma_0)$, $\mathbf{Z} \varphi \in rw^{-1}(\Gamma_0)$ and $\psi \mathbf{S} \zeta \in rw^{-1}(\Gamma_0)$ iff $\zeta \in rw^{-1}(\Gamma_0)$ and $\psi \mathbf{T} \zeta \in rw^{-1}(\Gamma_0)$ iff $\zeta \in rw^{-1}(\Gamma_0)$ which witnesses the encoding in $|TempConstraints|_k$ for the formulae $\mathbf{Y} \psi$, $\mathbf{Z} \psi$, $\psi \mathbf{S} \zeta$ and $\psi \mathbf{T} \zeta$ which belong to set $cl(\phi)$.

□

We now prove the second implication, which draws the connection between the encoding and the k-satisfiability problem.

Theorem 2. *Let $\phi \in$ CLTLB($\mathcal{D}$) with $\mathbb{N}$ definable in $\mathcal{D}$ together with the successor relation. If $|\phi|_k$ is satisfiable, then formula ϕ is k-satisfiable with respect to $k \in \mathbb{N}$.*

Proof. We prove the theorem by showing that formula $|\phi|_k$ defines ultimately periodic symbolic models $\rho = \alpha \beta^\omega$ for formula ϕ such that $\sigma_k, 0 \models_k \alpha\beta$ and $\rho, 0 \overset{sym}{\models} \phi$. Note that the encoding of $|\phi|_k$ defines precisely the truth value of all subformulae θ of ϕ in instants $i \in [0, k]$. Then, if $|\phi|_k$ is satisfiable, given an $i \in [0, k]$, the set of all subformulae

$$\Gamma_i = \{\varphi \in cl(\phi) \mid \text{ if } \boldsymbol{\theta}(i) \text{ holds then } \varphi = \theta, \text{ else } \varphi = \neg\theta\}$$

is a maximal consistent set of formulae of $cl(\phi)$. We have $\boldsymbol{loop} \in [0, k]$. The sequence of sets Γ_i for $0 \leq i \leq k$ is an ultimately periodic sequence of maximal consistent sets due to formulae $|LastStateConstraints|_k$ and $|LoopConstraints|_k$. We write $\Gamma|_A$ to denote the projection of $\mathcal{D}$-constraints in Γ on symbols of the set A; e.g., if $A = \{R_1, R_2\}$ then $\{R_1(x, y), R_2(\mathrm{X}x, \mathrm{Y}x), \theta_1, \theta_2\}|_A = \{R_1(x, y), R_2(\mathrm{X}x, \mathrm{Y}x)\}$. The sequence of atoms is

$$\gamma = \Gamma_0 \ldots \Gamma_{loop-1} \left(\Gamma_{loop} \ldots \Gamma_k\right)^\omega$$

and such that $\Gamma_{loop}|_{\mathcal{R}}$ is equal to the set of relations of $\Gamma_{k+1}|_{\mathcal{R}}$ by $|LoopConstraints|_k$ formulae. Moreover, by $|LastStateConstraints|_k$ we have $\Gamma_{k+1} = \Gamma_{loop}$.

By Lemma 4, from the bounded sequence σ_k of $\mathcal{D}$-valuations induced by $\mathcal{I}$, we have a unique locally consistent finite sequence of symbolic valuations $\alpha\beta$ such that $\sigma_k, 0 \models_k \alpha\beta$. Formula $|LoopConstraints|_k$ witnesses ultimately periodic sequences of symbolic valuations ρ because it is defined over the set of relations in $\mathcal{R}$ and all terms of the set $terms(\phi)$:

$$\rho = \alpha\beta^\omega = sv_0 \dots sv_{loop-1}(sv_{loop} \dots sv_k)^\omega$$

such that $sv_{loop} = sv_{k+1}$.

By structural induction on ϕ one can prove that for all $0 \leq i \leq k+1$, for all subformulae θ of ϕ, $\boldsymbol{\theta}(i)$ holds (i.e., $\theta \in \Gamma_i$) if, and only if, $\rho, i \overset{sym}{\models} \theta$. Then, since by hypothesis $\boldsymbol{\phi}(0)$ holds, we have that $\rho, 0 \overset{sym}{\models} \phi$.

The base case is the unique fundamental part of the proof because the inductive step over temporal modalities is rather standard. Let us consider a relation formula θ of the form $R(\alpha_1, \dots, \alpha_n)$ where, for all $1 \leq j \leq n$, $\alpha_j \in terms(\phi) \cup const(\phi)$ (the case when θ is $\neg R(\alpha_1, \dots, \alpha_n)$ is similar). We have to show that $\boldsymbol{\theta}(i)$ holds if, and only if, $sv_i \overset{sym}{\models} \theta$. As defined in Section 3.3, $\boldsymbol{\theta}(i)$ is $R(\boldsymbol{\alpha_1}(i), \dots, \boldsymbol{\alpha_n}(i))$ and, by definition of $\mathcal{I}$, we have $\mathcal{I}(\boldsymbol{\alpha_j})(i) = \sigma_k(i + |\alpha_j|, x_{\alpha_j})$. Then, we have that $R(\boldsymbol{\alpha_1}(i), \dots, \boldsymbol{\alpha_n}(i))$ holds if, and only if, $\sigma_k, i \models_k R(\alpha_1, \dots, \alpha_n)$; since, as shown in the proof of Lemma 4, $\sigma_k, i \models R(\alpha_1, \dots, \alpha_n)$ if, and only if, the symbolic valuation sv_i induced by σ_k at i includes $R(\alpha_1, \dots, \alpha_n)$, we have by definition $sv_i \overset{sym}{\models} R(\alpha_1, \dots, \alpha_n)$.

We omit the inductive step, which is standard and is reported in [17] and [4], since we use the same operators with the same encodings. □

Finally, the next theorem links k-satisfiability with the existence of an ultimately periodic run in automaton $\mathcal{A}_s \times \mathcal{A}_\ell$.

Theorem 3. *Let $\phi \in$ CLTLB($\mathcal{D}$) with $\mathbb{N}$ definable in $\mathcal{D}$ together with the successor relation. If formula ϕ is k-satisfiable with respect to $k \in \mathbb{N}$, then there exists an ultimately periodic run $\rho = \alpha\beta^\omega$ of $\mathcal{A}_s \times \mathcal{A}_\ell$, with $|\alpha\beta| = k+1$, accepting symbolic models of $rw(\phi)$.*

Proof. By definition, if ϕ is k-satisfiable so is $rw(\phi)$, and there is an ultimately periodic symbolic model $\rho = \alpha\beta^\omega$ such that $\rho, 0 \overset{sym}{\models} rw(\phi)$. By Lemma 4, ρ is locally consistent because there exists a k-bounded model σ_k such that $\sigma_k \models_k \alpha\beta$. Therefore, $\rho \in \mathscr{L}(\mathcal{A}_s \times \mathcal{A}_\ell)$. □

As explained in Section 2.4, each automaton involved in the definition of $\mathcal{A}_\phi$ has the function of "filtering" sequences of symbolic valuations so that 1) they are locally consistent, 2) they satisfy an LTL property and 3) they admit a (arithmetic) model. As mentioned in Section 2, for constraint systems that have the completion property, local consistency is equivalent to admitting an arithmetic model. For these constraint systems, $\mathcal{A}_\phi$ is exactly automaton $\mathcal{A}_s \times \mathcal{A}_\ell$, and from Proposition 2 and Theorem 2 we obtain the following result.

Proposition 5. *Let $\phi \in$ CLTLB($\mathcal{D}$) with $\mathbb{N}$ definable in $\mathcal{D}$ together with the successor relation and satisfying the completion property. Formula ϕ is k-satisfiable with respect to some $k \in \mathbb{N}$ if, and only if, there exists an arithmetic model σ such that $\sigma, 0 \models \phi$.*

Proof. Suppose formula ϕ is k-satisfiable. Then, by Theorem 3, there is a symbolic model $\rho = \alpha\beta^\omega$ such that $\rho, 0 \overset{sym}{\models} rw(\phi)$. By Proposition 2 ρ admits an arithmetic model $\hat{\sigma}$, i.e., such that $\hat{\sigma}, 0 \models rw(\phi)$. By Corollary 2, we have $\hat{\sigma}^{\lfloor\phi\rfloor}, 0 \models \phi$, so the desired σ is simply $\hat{\sigma}$ translated by $\lfloor\phi\rfloor$.

Conversely, if formula ϕ is satisfiable, then automaton $\mathcal{A}_{rw(\phi)}$ recognizes a non-empty language in $SV(rw(\phi))^\omega$. Hence, there is an ultimately periodic, locally consistent, sequence of symbolic valuations $\rho = \alpha\beta^\omega$, with $|\alpha\beta| = k + 1$, which is accepted by automaton $\mathcal{A}_{rw(\phi)}$. Then, the k-bounded model σ_k that shows the k-satisfiability of ϕ is built considering prefix $\alpha\beta$, by defining an edge-respecting labeling of graph $G_{\alpha\beta}$. □

When constraint systems do not have the completion property, the locally consistent sequence of symbolic models ρ recognized by automaton $\mathcal{A}_s \times \mathcal{A}_\ell$ may not admit arithmetic models σ such that $\sigma \models \rho$. However, as mentioned in Section 2.4.1, for some constraint systems $\mathcal{D}$, it is possible to define a condition over symbolic models which is satisfied by $\rho \in \mathscr{L}(\mathcal{A}_s \times \mathcal{A}_\ell)$ if, and only if ρ admits an arithmetic model. We tackle this issue in the next section.

4. Bounded Satisfiability of CLTLB(IPC*)

When $\mathcal{D}$ is IPC*, Proposition 5 does not apply since, by Lemma 1, $\mathcal{D}$ does not have the completion property. However, in such cases, as shown by Lemma 2, ultimately periodic symbolic models of CLTLB formulae admit arithmetic model if, and only if, they obey the condition captured by Property 1. In this section, we define a simplified condition of (non) existence of arithmetical models for ultimately periodic symbolic models of CLTLB formulae, and we show its equivalence with Property 1. Then, we provide a bounded encoding through QF-EU$\mathcal{D}$ formulae (where $\mathcal{D}$ embeds $\mathbb{N}$ and the successor function) for the new condition, and we define a specialized version of Proposition 5. Finally, we introduce simplifications to the encoding that can be applied in special cases.

Let ρ be a symbolic model for CLTLB(IPC*) formula ϕ and let G_ρ be the graph defined as in Section 2.4. To devise the simplified condition equivalent to Property 1, we associate a set of so-called points with each node of G_ρ: For each node, there are as many points as symbolic valuations including the node. Then, we provide suitable relations over points. Formally, let $P_\rho = (V \cup const(\phi)) \times \mathbb{N} \times [\lfloor\phi\rfloor, \lceil\phi\rceil]$ be called the set of points of ρ. A *point* $p \in P_\rho$ is a triple $p = (x, j, h)$, identifying a variable or a constant $x \in V \cup const(\phi)$ at a position h within symbolic valuation $\rho(j)$, i.e., p refers variable (or constant) x at position $j + h$ of the symbolic model ρ. Denote with $var(p)$ the variable x, with $sv(p)$ the symbolic valuation j (with $sv(p) \geq 0$), and with $shift(p)$ the position h of x within the j-th symbolic valuation (with $shift(p) \in [\lfloor\phi\rfloor, \lceil\phi\rceil]$); Therefore, $x(j + h)$ represents variable x at position h of the j-th symbolic valuation of ρ.

Different triples can refer to the same node. For example, variable x in position 2 of symbolic valuation 4 (i.e., $(x, 4, 2)$) is the same as x in position 1 of adjacent symbolic valuation 5 (i.e., $(x, 5, 1)$), and also of x in position 0 of symbolic valuation 6 (i.e., $(x, 6, 0)$): these points all refer to the node $x(6)$ of G_ρ. Figures 2 and 3 show examples

of equivalent points. Hence, we need to define an equivalence relation on points, called *local equivalence*.

Definition 5. For all points $p_1 = (x, j, h)$, $p_2 = (x, j', h')$ in P_ρ, we say that p_1 is *locally equivalent* to p_2 if $j + h = j' + h'$, with $j, j' \geq 0$ and $h, h' \in [\lfloor\phi\rfloor, \lceil\phi\rceil]$.

Note that the set of equivalence classes induced by local equivalence corresponds to the set of nodes of G_ρ.

Definition 6. We define the relation $\preccurlyeq \subseteq P_\rho \times P_\rho$. Given $p_1 = (x, j, h)$ and $p_2 = (y, i, m)$ of P_ρ, $p_1 \preccurlyeq p_2$ holds if:

1. $i + m - (j + h) < -\lfloor\phi\rfloor + \lceil\phi\rceil + 1$
2. $j + h \leq i + m$
3. in G_ρ there is an edge labeled with $<$ or $=$ from $x(j + h)$ to $y(i + m)$.

Condition 3 symbolically represents the constraint that $x(j+h) \leq y(i+m)$. Relations $\prec, \succcurlyeq, \succ, \approx \subseteq P_\rho \times P_\rho$ are defined as above by replacing "$<$ or $=$" with, respectively, $<$, "$>$ or $=$", $>$, $=$ in Condition 3.

By Condition 1 of Definition 6, for each relation $\sim \in \{\preccurlyeq, \prec, \approx, \succ, \succcurlyeq\}$, $p_1 \sim p_2$ may hold only if the distance of p_1 and p_2 is smaller than the size $-\lfloor\phi\rfloor + \lceil\phi\rceil + 1$ of a symbolic valuation, i.e., p_1 and p_2 are "local", in the sense that they belong either to the same symbolic valuation (i.e., $j = i$) or to the common part of "partially overlapping" symbolic valuations (see Figures 2 and 3 for examples of partially overlapping symbolic valuations). By Condition 2, each relation $\sim$ is a positional precedence, i.e., if $p_1 \sim p_2$ then p_2 cannot positionally precede p_1. Condition 3 is well defined on symbolic valuations, since it corresponds to having, in graph G_ρ, an arc labeled with $\sim$ from p_1 to p_2. The reflexive relations $\preccurlyeq, \succcurlyeq$ have an antisymmetric property, in the sense that if $p_1 \preccurlyeq p_2$ and $p_2 \preccurlyeq p_1$, then $p_1 \approx p_2$ and $p_2 \approx p_1$ (analogously for $\succcurlyeq$): if $p_1 = (x, j, h)$ and $p_2 = (y, i, m)$, then p_1 and p_2 are at the same position $j+h = i+m$ and have the same value $x(j + h) = y(i + m)$.

Notice that the relations $\sim$ are not transitive, because of Condition 1: Each relation $\sim$ is only "locally" transitive, in the sense that if $p_1 \sim p_2$ and $p_2 \sim p_3$, then $p_1 \sim p_3$ if, and only if, Condition 1 holds for p_1 and p_3 (i.e., when also p_1, p_3 are "local", which in general may not be the case).

Definition 7. We say that there is a *local forward* (resp. *local backward*) path from point p_1 to point p_2 if $p_1 \preccurlyeq p_2$ (resp., $p_1 \succcurlyeq p_2$); the path is called *strict* if $p_1 \prec p_2$ (resp., $p_1 \succ p_2$).

Obviously, given two points $p_1 = (x, j, h)$ and $p_2 = (y, i, m)$ of P_ρ such that $|i+m-(j+h)| < -\lfloor\phi\rfloor+\lceil\phi\rceil+1$, it must be at least one of $p_1 \preccurlyeq p_2, p_2 \preccurlyeq p_1, p_1 \succcurlyeq p_2$, $p_2 \succcurlyeq p_1$; if both $p_1 \preccurlyeq p_2$ and $p_1 \succcurlyeq p_2$ hold, then $p_1 \approx p_2$, hence $x(j + h) = y(i + m)$.

It is immediate to notice that the local equivalence is a congruence for all relations, e.g., if p_1 is locally equivalent to p'_1 and p_2 is locally equivalent to p'_2 then $p_1 \preccurlyeq p_2$ iff $p'_1 \preccurlyeq p'_2$. Figures 2 and 3 depict examples of this fact.

We now extend the relations of Definition 7 to cope with non-overlapping symbolic valuations.

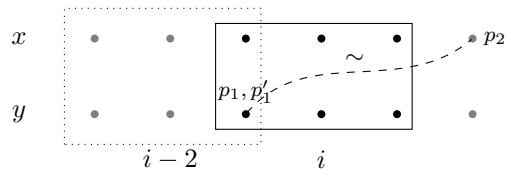

Figure 2: Adjacent and overlapping symbolic valuations $\rho(i)$ (solid line) and $\rho(i-2)$ (dotted line) of length 3 (with $-\lfloor\phi\rfloor = \lceil\phi\rceil = 1$), with $p_1 = (y, i, -1)$ and $p_1' = (y, i-2, 1)$ being locally equivalent. Both $p_1 \overset{\sim}{\frown} p_2$ and $p_1' \overset{\sim}{\frown} p_2$ hold.

Definition 8. Relation $\overset{\sim}{\frown} \subseteq P_\rho \times P_\rho$, for every $\sim \in \{\preccurlyeq, \approx, \succcurlyeq\}$, denotes the transitive closure of $\sim$. Relations $\overset{\prec}{\frown}, \overset{\succ}{\frown} \subseteq P_\rho \times P_\rho$, are defined as follows, for all $p_1, p_2 \in P_\rho$:

$p_1 \overset{\prec}{\frown} p_2$ if there exist $p', p'' \in P_\rho$ such that $p_1 \overset{\preccurlyeq}{\frown} p' \prec p'' \overset{\preccurlyeq}{\frown} p_2$;

$p_1 \overset{\succ}{\frown} p_2$ if there exist $p', p'' \in P_\rho$ such that $p_1 \overset{\succcurlyeq}{\frown} p' \succ p'' \overset{\succcurlyeq}{\frown} p_2$.

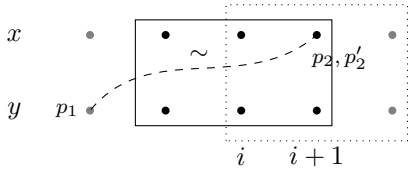

Figure 3: Adjacent and overlapping symbolic valuations $\rho(i)$ (solid line) and $\rho(i+1)$ (dotted line) of length 3 ($-\lfloor\phi\rfloor = \lceil\phi\rceil = 1$), with points $p_2 = (x, i, 1)$ and $p_2' = (x, i+1, 0)$ being locally equivalent. Both $p_1 \overset{\sim}{\frown} p_2$ and $p_1 \overset{\sim}{\frown} p_2'$ hold.

Remark 1. If $p_1 = (x, j, h)$ and $p_2 = (y, i, m)$, then $p_1 \overset{\preccurlyeq}{\frown} p_2$ symbolically represents the constraint $x(j + h) \leq y(i + m)$. The other cases of $\overset{\sim}{\frown}$ are similar. If $\sim$ is, respectively, $\prec, \approx, \succ, \succcurlyeq$, then the relation of $x(j + h)$ with $y(i + m)$ is, respectively, $<, =, >, \geq$. If $p_1 \overset{\preccurlyeq}{\frown} p_2$ holds, but $p_1 \overset{\prec}{\frown} p_2$ does not, then along the path from p_1 to p_2 there are only arcs labeled with $\approx$, i.e. $p_1 \overset{\approx}{\frown} p_2$, which symbolically represents $x(j+h) = y(i+m)$. As a consequence, if $p_1 \overset{\preccurlyeq}{\frown} p_2$ holds, but $p_1 \overset{\prec}{\frown} p_2$ does not, then $p_1 \overset{\succcurlyeq}{\frown} p_2$ also holds. The dual properties hold for $\overset{\succcurlyeq}{\frown}$ and $\overset{\succ}{\frown}$.

Let $\rho = \alpha\beta^\omega \in SV(\phi)^\omega$ be an ultimately periodic symbolic model of ϕ. We need to introduce another equivalence relation, which is useful for capturing properties of points of symbolic valuations in β^ω, though it is defined in general. More precisely, we consider two points $p, p' \in P_\rho$ as equivalent when they correspond to the same variable, in the same position of the symbolic valuation, but in symbolic valuations that are $i|\beta|$ positions apart, for some $i \geq 0$. In fact, points in β^ω that are equivalent according to the definition below have the same properties concerning forward and backward paths.

Definition 9. Two points $p, p' \in P_\rho$ are *equivalent*, written $p \equiv p'$, when $var(p) = var(p')$, $sv(p') = sv(p) + i|\beta|$ and $shift(p) = shift(p')$, for some $i \in \mathbb{Z}$.

The main result of the section is Formula (1) on page 28, which is based on a number of intermediate results that are presented in the following. To test for the condition for the existence of arithmetic models of symbolic model $\rho = \alpha\beta^\omega$, one must represent infinite (possibly strict) forward and backward paths along ρ. To this end, we devise a condition for the existence of infinite paths, resulting from iterating suffix β infinitely many times. Without loss of generality, in the following we consider ultimately periodic models $\rho = \alpha\beta^\omega$ in which $\alpha = \alpha' s$ and $\beta = \beta' s$, i.e., in which the last symbolic valuation of prefix α is the same as the last symbolic valuation of repeated suffix β. We indicate by $k+1$ the length of $\alpha\beta$, and we number the symbolic valuations in $\alpha\beta$ starting from 0, so that the last element in prefix α is in position $|\alpha|-1$, the first element in suffix β is in position $|\alpha|$, and the last element of β is in position k (hence, $\rho(|\alpha|-1) = \rho(k) = s$, with $k = |\alpha\beta| - 1$). An infinite forward (resp. backward) path is represented as a cycle among variables belonging to symbolic valuations $\rho(|\alpha|-1)$ and $\rho(k)$, connected through relations $\overset{\preccurlyeq}{\frown}$ and $\overset{\prec}{\frown}$ (resp. $\overset{\succcurlyeq}{\frown}$ and $\overset{\succ}{\frown}$). Intuitively, in ρ there is an infinite (strict) forward path when there are two points p, p' in $\alpha\beta$ – with $p \neq p'$ – such that $sv(p) = |\alpha| - 1$, $sv(p') = k$, $p \equiv p'$, and $p \overset{\preccurlyeq}{\frown} p'$ ($p \overset{\prec}{\frown} p'$). Now, all results required to obtain Formula (1) equivalent to Property 1 are provided.

We have the following property, which states that if in $\rho = \alpha\beta^\omega$ there is a finite forward path from point p to a point p'' of the suffix β^ω, with $p \equiv p''$, then there is also a finite forward path from p to every point p' between p and p'' and such that $p' \equiv p$.

Lemma 5. *Let $\rho = \alpha\beta^\omega \in SV(\phi)^\omega$ be an ultimately periodic word, and $\beta = \beta' s' \beta''$ for some $\beta', \beta'' \in SV(\phi)^*$, $s' \in SV(\phi)$; let i be the position of s' in $\alpha\beta$ (so $\rho(i) = s'$). Let p_i, p_j be any two points of P_ρ such that $sv(p_i) = i$, $sv(p_j) = j$, $p_i \equiv p_j$ and $j > i + |\beta|$. Let p' be the point such that $p_j \equiv p'$ and $sv(p') = j - |\beta|$. If $p_i \overset{\sim}{\frown} p_j$ (for some $\sim \in \{\preccurlyeq, \prec, \approx, \succcurlyeq, \succ\}$), then also $p_i \overset{\sim}{\frown} p'$.*

Proof. First, since $p_i \equiv p_j$, then $\rho(j - |\beta|) = \rho(j) = s'$ holds.

Let us consider the case $p_i \overset{\preccurlyeq}{\frown} p_j$. Then, as exemplified in Figure 4, along the finite forward path from p_i to p_j, there must be a point p_1 to the right of (or aligned with) p' such that p' and p_1 are locally related (p_1 could be p' itself). More precisely, it must be $0 \leq sv(p_1) + \mathit{shift}(p_1) - (sv(p') + \mathit{shift}(p')) < -\lfloor\phi\rfloor + \lceil\phi\rceil + 1$, or there are two consecutive points along the path from p_i to p_j that are not locally related, which is impossible. Then, we have that:

1. $p_i \overset{\preccurlyeq}{\frown} p_1$
2. $p_1 \overset{\preccurlyeq}{\frown} p_j$
3. either $p' \preccurlyeq p_1$, or $p' \succcurlyeq p_1$

We have two cases. If $p' \preccurlyeq p_1$, then, from condition 2 above and the definition of $\overset{\preccurlyeq}{\frown}$ we have $p' \overset{\preccurlyeq}{\frown} p_j$; since p_i, p' and p_j all belong to β^ω and are such that $p_i \equiv p' \equiv p_j$, then the same forward path from p' to p_j, from which it descends $p' \overset{\preccurlyeq}{\frown} p_j$, can be iterated starting from p_i, because suffix β^ω is periodic. Then, $p_i \overset{\preccurlyeq}{\frown} p'$. If, instead, $p' \succcurlyeq p_1$, then, by condition 1 and the definitions of $\succcurlyeq$ and $\overset{\preccurlyeq}{\frown}$, condition $p_i \overset{\preccurlyeq}{\frown} p'$ also holds.

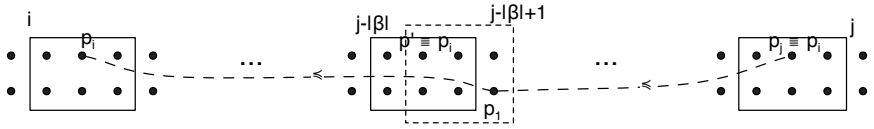

Figure 4: Relations between symbolic valuations i and j.

The case $p_i \overset{\prec}{\frown} p_j$ is similar, when one considers that, in addition to conditions 1-3, it must be $p_i \overset{\prec}{\frown} p_1$ or $p_1 \overset{\prec}{\frown} p_j$. If $p' \preccurlyeq p_1$, then if $p_1 \overset{\prec}{\frown} p_j$ also $p' \overset{\prec}{\frown} p_j$, and the proof is as before. If, instead, $p_1 \overset{\prec}{\frown} p_j$ does not hold, then it must be that $p' \prec p_1$, otherwise from Remark 1 it descends that the value of the variable in p' is equal to the value in p_j, and in turn that the value of the variable in p_i is equal to the value in p_j, thus contradicting $p_i \overset{\prec}{\frown} p_j$. If $p' \succcurlyeq p_1$, then if $p_i \overset{\prec}{\frown} p_1$ we have also $p_i \overset{\prec}{\frown} p'$. Otherwise, if $p_i \overset{\prec}{\frown} p_1$ does not hold, then it must be that $p_1 \overset{\prec}{\frown} p_j$, and in this case $p' \succ p_1$ must also hold (hence also $p_i \overset{\prec}{\frown} p'$), or the arc from p_1 to p' is labeled with $=$, and we have that $p_i \overset{\preccurlyeq}{\frown} p'$, not $p_i \overset{\prec}{\frown} p'$ (hence $p_i \overset{\approx}{\frown} p'$ by Remark 1), and $p' \overset{\prec}{\frown} p_j$, which yields a contradiction.

The proofs for cases $p_i \overset{\succcurlyeq}{\frown} p_j$, $p_i \overset{\succ}{\frown} p_j$, and $p_i \overset{\approx}{\frown} p_j$ are analogous. □

We immediately have the following corollary, which states that a path looping through p_i can be shortened to a single iteration.

Corollary 3. *Let $\rho = \alpha\beta^\omega \in SV(\phi)^\omega$, p_i and p_j as in Lemma 5. Let p' be the point such that $p_j \equiv p'$ and $sv(p') = i + |\beta|$. Then $p_i \overset{\sim}{\frown} p'$ holds.*

The following lemma shows that there is an infinite non-strict (resp. strict) forward path in $\rho = (\alpha' s)(\beta' s)^\omega$ if, and only if, there is an infinite non-strict (resp. strict) forward path that loops through symbolic valuation s.

Lemma 6. *Let $\rho = \alpha\beta^\omega \in SV(\phi)^\omega$ be an ultimately periodic word, with $\alpha = \alpha' s$ and $\beta = \beta' s$. In ρ there is an infinite non-strict (resp. strict) forward path if, and only if, there is an infinite non-strict (resp. strict) forward path that contains a denumerable set of points $\{p_i\}_{i\in\mathbb{N}}$ of P_ρ such that:*

1. *$sv(p_0) = |\alpha| - 1 = |\alpha'|$,*
2. *$p_i \equiv p_j$ and $sv(p_i) < sv(p_j)$ for all $i < j \in \mathbb{N}$,*
3. *$p_i \overset{\preccurlyeq}{\frown} p_{i+1}$ (resp. $p_i \overset{\prec}{\frown} p_{i+1}$) for all $i \in \mathbb{N}$.*

Proof. Let us assume in ρ there is an infinite non-strict forward path, and let $F = \{f_i\}_{i\in\mathbb{N}}$ be the points that it traverses (hence, $f_i \preccurlyeq f_{i+1}$ for all i). Note that $sv(f_0)$ can be any, not necessarily 0 or $|\alpha'|$. Since suffix β^ω is periodic and each arc $\langle f_i, f_{i+1}\rangle$ in F connects two points that, for Condition 1 of Definition 6, have distance at most $-\lfloor\phi\rfloor + \lceil\phi\rceil + 1$ from one another, then there must be a sequence of points $Q = \{q_i\}_{i\in\mathbb{N}}$ such that, for each $q_i \in Q$

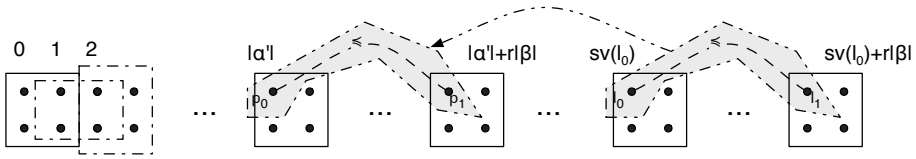

Figure 5: Example of translation by $sv(l_0) - |\alpha'|$.

- $sv(q_{i+1}) > sv(q_i) > |\alpha'|$
- there is a point $f_j \in F$ such that f_j is locally equivalent to q_i
- $\rho(sv(q_i)) = s$.

In other words, Q is made by points of F (or locally equivalent ones) that belong to one of the instances of symbolic valuation s in β^ω. For each $i \in \mathbb{N}$ $q_i \preccurlyeq q_{i+1}$ holds. Since the number of points in symbolic valuation s is finite, there must be an element $q_{\bar{i}} \in Q$ such that an infinite number of points equivalent to $q_{\bar{i}}$ appear in Q. In other words, there is a denumerable sequence $L = \{l_i\}_{i\in\mathbb{N}}$ such that

- $l_0 = q_{\bar{i}}$
- for all i $l_i \equiv q_{\bar{i}}$ holds
- for all i we have that both $l_i \preccurlyeq l_{i+1}$ and $sv(l_i) < sv(l_{i+1})$ hold.

Sequence L is part of an infinite forward path that starts from l_0 and visits all l_i. The desired sequence $\{p_i\}_{i\in\mathbb{N}}$ that satisfies conditions 1-3 is L translated by $sv(l_0) - |\alpha'|$, i.e., for every $i \geq 0$, $sv(p_i) = sv(l_i) - (sv(l_0) - |\alpha'|)$ so that it starts from the symbolic valuation in position $|\alpha'|$; the translation is possible because of the periodicity of β^ω. Figure 5 shows an example of translation.

The proof in case of strict infinite paths is similar. □

A similar lemma holds for backward paths. We have the following result.

Theorem 4. *Let $\rho = \alpha\beta^\omega \in SV(\phi)^\omega$ be an ultimately periodic word, with $\alpha = \alpha's$ and $\beta = \beta's$. Then, there is a non-strict (resp. strict) infinite forward path in ρ if, and only if, there are two points p, p' of P_ρ such that $sv(p) = |\alpha'|$, $sv(p') = k$, $p \equiv p'$, and $p \preccurlyeq p'$ (resp. $p \prec p'$).*

Proof. We consider the case for non-strict forward paths, the case for strict ones being similar.

Assume in ρ there is an infinite non-strict forward path; then, by Lemma 6 there is also an infinite non-strict forward path that contains a denumerable set of points $\{p_i\}_{i\in\mathbb{N}}$ that satisfies conditions 1-3 of the lemma. Then, from Corollary 3 we immediately have $p_0 \preccurlyeq p'$, with $p' \equiv p_0$ and with $sv(p') = |\alpha'| + |\beta| = k$ (recall that $|\alpha\beta| = k + 1$).

Conversely, assume that there are two points p, p' such that $p = (x, |\alpha'|, h)$, $p' = (x, k, h)$, $p \equiv p'$, and $p \preccurlyeq\!\!\!\!\frown\; p'$. By definition of $p \preccurlyeq\!\!\!\!\frown\; p'$, there exists a finite number of points $p^1, p^2, \ldots$ such that $p \preccurlyeq p^1 \preccurlyeq p^2 \ldots \preccurlyeq p'$. This forward path can be iterated infinitely many times, since $p \equiv p'$ and the suffix β is repeated infinitely often. Therefore, point p and all points equivalent to p satisfy conditions 1-3 of Lemma 6. By the same lemma, then, in ρ there is an infinite non-strict forward path. □

Analogously, we can prove the following version of Theorem 4 in case of backward paths.

Theorem 5. *Let $\rho = \alpha\beta^\omega \in SV(\phi)^\omega$ be an ultimately periodic word, with $\alpha = \alpha's$ and $\beta = \beta's$. Then, there is a non-strict (resp. strict) infinite backward path in ρ if, and only if, there are two points p, p' such that $sv(p) = |\alpha'|$, $sv(p') = k$, $p \equiv p'$, and $p \succcurlyeq\!\!\!\!\frown\; p'$ (resp. $p \succ\!\!\!\!\frown\; p'$).*

Our condition for the non existence of an arithmetic model for symbolic model $\rho = \alpha's(\beta's)^\omega$ (with $|\alpha's\beta's| = k+1$) is formalized by Formula (1) below; it captures the negation of Property 1 and takes advantage of the previous Theorems 4 and 5.

$$\exists p_1 p_2 p_1' p_2' \left(\begin{array}{c} p_1 \equiv p_2 \ \wedge\ p_1' \equiv p_2' \wedge \\ sv(p_1) = sv(p_1') = |\alpha'| \ \wedge\ sv(p_2) = sv(p_2') = k \wedge \\ p_1 \preccurlyeq\!\!\!\!\frown\; p_2 \ \wedge\ p_1' \succcurlyeq\!\!\!\!\frown\; p_2' \ \wedge\ (p_1 \prec\!\!\!\!\frown\; p_2 \ \vee\ p_1' \succ\!\!\!\!\frown\; p_2') \wedge \\ (p_1 \prec p_1' \vee p_1' \succ p_1) \end{array} \right). \qquad (1)$$

In Formula (1) four conditions are defined, similar to those of Property 1. Informally, Formula (1) says that:

1. there is an infinite forward path f from p_1 (this derives from the fact that $p_1 \preccurlyeq\!\!\!\!\frown\; p_2$, with $p_1 \equiv p_2$, $sv(p_1) = |\alpha'|$, and $sv(p_2) = k$);
2. there is an infinite backward path b from p_1' (from $p_1' \succcurlyeq\!\!\!\!\frown\; p_2'$, with $p_1' \equiv p_2'$, where $sv(p_1') = |\alpha'|$, and $sv(p_2') = k$);
3. at least one of the paths f and b is strict;
4. there is an edge labeled with $<$ from p_1 to p_1' .

In particular, condition 4 of Property 1 is different from condition 4 of Formula (1). In fact, the former one states that for each $i, j \in \mathbb{N}$, given a forward path d and a backward path e, whenever $d(i)$ and $e(j)$ belong to the same symbolic valuation (i.e., $|i - j| < -\lfloor\phi\rfloor + \lceil\phi\rceil + 1$) there is an edge labeled by $<$ from $d(i)$ to $e(j)$. In other words, this means that point p_d representing $d(i)$ and point p_e representing $e(j)$ are such that either $p_d \prec p_e$ or $p_e \succ p_d$. The next theorem shows that the conditions are nevertheless equivalent when $\rho = \alpha\beta^\omega$. In fact, whereas Property 1 is defined for a general G_ρ, Formula (1) is tailored to the finite representation of ultimately periodic symbolic models $\rho = \alpha\beta^\omega$.

Theorem 6. *Over ultimately periodic symbolic models of the form $\alpha's(\beta's)^\omega$, with $\alpha, \beta \in SV(\phi)^*$ and $s \in SV(\phi)$, the negation of Property 1 is equivalent to Formula* (1).

Proof. Let $\rho = \alpha' s(\beta' s)^\omega$ be an infinite symbolic model and assume that Formula (1) holds in $\alpha' s \beta' s$. Therefore, there exist two pair of points p_1, p_1' and p_2, p_2' satisfying Formula (1), hence, $sv(p_1) = sv(p_1') = |\alpha'|$. By Theorems 4 and 5, p_1, p_1' are visited, respectively, by an infinite forward path and an infinite backward path, where at least one of the two is strict (because $p_1 \precsim p_2 \vee p_1' \succsim p_2'$ holds). Consider any two points $\bar{p}_2$ and $\bar{p}_2'$ such that $p_1 \equiv \bar{p}_2$, $p_1' \equiv \bar{p}_2'$. Since $p_1 \prec p_1' \vee p_1' \succ p_1$ holds, and for both pairs p_1, p_1' and $\bar{p}_2, \bar{p}_2'$ the symbolic valuation is s, then also $\bar{p}_2 \prec \bar{p}_2' \vee \bar{p}_2' \succ \bar{p}_2$ holds. Now, consider any two points q and q' in $\alpha' s(\beta' s)^\omega$, such that $sv(q) = sv(q')$ and q (resp. q') belongs to the infinite strict forward (resp. backward) path from p_1 (resp. p_1'). If $\bar{p}_2$ and $\bar{p}_2'$ are the points in the same iteration of the suffix $\beta' s$ as q, q' such that $\bar{p}_2 \equiv p_2$ and $\bar{p}_2' \equiv p_2'$, then $q \precsim \bar{p}_2$, $q' \succsim \bar{p}_2'$ and $\bar{p}_2 \prec \bar{p}_2' \vee \bar{p}_2' \succ \bar{p}_2$ hold. Hence, there is a path from q to q' along which all edges are labeled with $=$ or with $<$, with at least one edge labeled with $<$. Therefore, $q \prec q'$ or $q' \succ q$, i.e., from q to q' there is an edge labeled with $<$. The vertices u and v that show that Property 1 does not hold are simply p_1 and p_1'.

Conversely, assume Property 1 does not hold; then, by Theorems 4 and 5 there are points p_1, p_1', p_2, p_2' such that $sv(p_1) = sv(p_1') = |\alpha'|$, $sv(p_2) = sv(p_2') = k$, $p_1 \equiv p_2$, $p_1' \equiv p_2'$, $p_1 \precsim p_2$, $p_1' \succsim p_2'$, and $p_1 \precsim p_2 \vee p_1' \succsim p_2'$ hold. From the proof of Theorem 4, point p_1 is equivalent to some point in the original forward path; similarly for point p_1'. Then, since p_1 and p_1' belong to the same symbolic valuation, by condition 4 of Property 1, they are connected through an edge labeled with $<$, i.e., $p_1 \prec p_1'$ or $p_1' \succ p_1$ hold. □

The next theorem extends Proposition 5 to constraint system IPC*, which does not benefit from the completion property.

Proposition 6. *Let $\phi \in$ CLTLB($\mathcal{D}$) and $\mathcal{D}$ be IPC*. Formula ϕ is k-satisfiable for some $k \in \mathbb{N}$ and the induced symbolic model $\rho = \alpha\beta^\omega$ does not satisfy Formula (1) if, and only if, there exists an arithmetic model σ such that $\sigma, 0 \models \phi$.*

Proof. By Theorems 1, 2, and 3, ϕ is k-satisfiable for some $k \in \mathbb{N}$ if, and only if, formula $|\phi|_k$ is satisfiable; in addition, when formula $|\phi|_k$ is satisfiable, it induces a model σ_k and a sequence $\alpha\beta$ of symbolic valuations of length k representing an infinite sequence $\rho = \alpha\beta^\omega$ of symbolic valuations such that $\rho \overset{sym}{\models} \phi$. Since Formula (1) does not hold for ρ, then by Theorem 6 Property 1 holds, hence, by Lemma 2, ρ admits an arithmetic model σ such that $\sigma, 0 \models \phi$.

Conversely, if formula ϕ is satisfiable, then automaton $\mathcal{A}_\phi$ recognizes locally consistent symbolic models of ϕ which satisfy Property 1. Then, a symbolic model $\alpha\beta^\omega \in \mathscr{L}(\mathcal{A}_\phi)$ which satisfies the negation of Formula (1) and a k-bounded model $\sigma_k, 0 \models_k \alpha\beta$ can be obtained as in the proof of Proposition 5. □

Bounded Encoding of Formula (1)

The encoding shown afterwards represents, by means of a finite representation, infinite – strict and non strict – paths over infinite symbolic models. As before, we consider models $\rho = \alpha\beta^\omega$ where $\alpha = \alpha' s$ and $\beta = \beta' s$, and we consider the finite sequence of symbolic valuations $\alpha' s \beta' s$. We indicate by $P_{\alpha\beta} \subset P_\rho$ the set of points

of finite path $\alpha' s \beta' s$ (for all $p \in P_{\alpha\beta}$, $sv(p) \in [0, k+1]$). We use the points of $P_{\alpha\beta}$ to capture properties of P_ρ. To encode the previous formulae into QF-EU$\mathcal{D}$ formulae, where $\mathcal{D}$ is a suitable constraint system embedding $\mathbb{N}$ and having the successor function plus order $<$, we rearrange the formulae above by splitting information, which is now encapsulated in the notion of point, on variables and positions over the model. Predicate $\boldsymbol{f}^{<}_{x,y} : \mathbb{N}^3 \to \{true, false\}$ for all pairs $x, y \in V \cup const(\phi)$ (resp. $\boldsymbol{f}^{\leq}_{x,y}$) encodes relation $p_1 \prec p_2$ (resp. $p_1 \preccurlyeq p_2$) where $p_1 = (x, j, h)$ and $p_2 = (y, j, m)$. For all $h, m \in [\lfloor \phi \rfloor, \lceil \phi \rceil]$ predicates $\boldsymbol{f}^{<}_{x,y}$ and $\boldsymbol{f}^{\leq}_{x,y}$ are defined by the following table, where $\mathrm{O}^h x$ is x if $h = 0$, $\mathrm{X}^h x$ if $h > 0$, and $\mathrm{Y}^h x$ if $h < 0$ (similarly for $\mathrm{O}^m y$). Denote with $\mathbf{O^h}\boldsymbol{x}(j)$ the unary function associated with a.t.t. $\mathrm{O}^h x$ introduced in Section 3.3 and obeying $|ArithConstraints|_k$.

$0 \leq j \leq k+1$ and $h \leq m$	$0 \leq j \leq k+1$ and $h > m$
$\boldsymbol{f}^{<}_{x,y}(j,h,m) \Leftrightarrow \mathbf{O^h}\boldsymbol{x}(j) < \mathbf{O^m}\boldsymbol{y}(j)$	$\neg \boldsymbol{f}^{<}_{x,y}(j,h,m)$
$\boldsymbol{f}^{\leq}_{x,y}(j,h,m) \Leftrightarrow \mathbf{O^h}\boldsymbol{x}(j) \leq \mathbf{O^m}\boldsymbol{y}(j)$	$\neg \boldsymbol{f}^{\leq}_{x,y}(j,h,m)$

Constants are implicitly included in the model. For instance, if $5 \in const(\phi)$ and $x \in V$ we have formulae $\boldsymbol{f}^{<}_{x,5}(j, 0, m) \Leftrightarrow \boldsymbol{x}(j) < 5$ and $\boldsymbol{f}^{<}_{5,x}(j, 0, m) \Leftrightarrow 5 < \boldsymbol{x}(j)$. When $x, y \in const(\phi)$ then $\boldsymbol{f}^{<}_{x,y} \Leftrightarrow x < y$ and $\boldsymbol{f}^{\leq}_{x,y} \Leftrightarrow x \leq y$ for all $0 \leq j \leq k+1$ and $h \leq m$; $\neg \boldsymbol{f}^{<}_{x,y}$ and $\neg \boldsymbol{f}^{\leq}_{x,y}$ for all $0 \leq j \leq k+1$ and $h > m$.

Relation $\overset{\prec}{\frown}$ (resp. relation $\overset{\preccurlyeq}{\frown}$) is encoded by the uninterpreted predicates $\boldsymbol{F}^{<}_{x,y} : \mathbb{N}^4 \to \{true, false\}$ (resp. $\boldsymbol{F}^{\leq}_{x,y} : \mathbb{N}^4 \to \{true, false\}$) for all pairs of variables $x, y \in V \cup const(\phi)$. To build in practice $\overset{\prec}{\frown}$ (resp., $\overset{\preccurlyeq}{\frown}$) through $\boldsymbol{F}^{<}$ (resp. $\boldsymbol{F}^{\leq}$), over points of the symbolic model $\alpha' s \beta' s$, we construct the transitive closure of $\boldsymbol{F}^{<}$ (resp. $\boldsymbol{F}^{\leq}$) explicitly. Starting from $\rho(0)$, we propagate the information about relations $\prec$ and $\preccurlyeq$ that are represented by $\boldsymbol{f}^{<}$ and $\boldsymbol{f}^{\leq}$ among all points representing variables of model ρ. In fact, it is immediate to show that $p_1 \overset{\prec}{\frown} p_2$ holds if, and only if, there is a point p such that either $p_1 \prec p$ and $p \overset{\preccurlyeq}{\frown} p_2$ or $p_1 \preccurlyeq p$ and $p \overset{\prec}{\frown} p_2$ (note that p cannot be locally equivalent to both p_1 and p_2, but it can be locally equivalent to one of them). Similarly for the other relations. Figure 6 provides a graphical representation for $\overset{\prec}{\frown}$. Formulae defining $\boldsymbol{F}^{<}_{x,y}$ and $\boldsymbol{F}^{\leq}_{x,y}$ are the following:

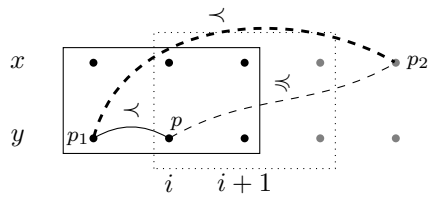

Figure 6: Adjacent symbolic valuations $\rho(i)$ (solid line) and $\rho(i+1)$ (dotted line) not covering both points $p_1 = (y, i, -1)$ and $p_2 = (x, j, h)$ (with $j > i$ and $-1 \leq h \leq 1$) of the model, with $p_1 \prec p$, $p \overset{\preccurlyeq}{\frown} p_2$ and $p_1 \overset{\prec}{\frown} p_2$.

$$\boldsymbol{F}^{<}_{x,y}(j,h,i,m) \Leftrightarrow \begin{cases} \bigvee\limits_{z\in V}\bigvee\limits_{u=\lfloor\phi\rfloor}^{\lceil\phi\rceil} \boldsymbol{f}^{<}_{x,z}(j,h,u) \wedge \boldsymbol{F}^{\leq}_{z,y}(j,u,i,m) \vee \\ \bigvee\limits_{z\in V}\bigvee\limits_{u=\lfloor\phi\rfloor}^{\lceil\phi\rceil} \boldsymbol{f}^{\leq}_{x,z}(j,h,u) \wedge \boldsymbol{F}^{<}_{z,y}(j,u,i,m) \end{cases} \tag{2}$$

$$\boldsymbol{F}^{\leq}_{x,y}(j,h,i,m) \Leftrightarrow \bigvee_{z\in V}\bigvee_{u=\lfloor\phi\rfloor}^{\lceil\phi\rceil} \boldsymbol{f}^{\leq}_{x,z}(j,h,u) \wedge \boldsymbol{F}^{\leq}_{z,y}(j,u,i,m) \tag{3}$$

for all $j, i \in [0, k{+}1]$ with $j < i$ and for all $h, m \in [\lfloor\phi\rfloor, \lceil\phi\rceil]$ such that $i{+}m{-}(j{+}h) > -\lfloor\phi\rfloor + \lceil\phi\rceil$, $(x = z) \Rightarrow (h \neq u)$ and for all pairs $x, y \in V \cup const(\phi)$. When $j = i \in [0, k+1]$ and $h \leq m$, with $h, m \in [\lfloor\phi\rfloor, \lceil\phi\rceil]$:

$$\boldsymbol{F}^{<}_{x,y}(j,h,j,m) \Leftrightarrow \boldsymbol{f}^{<}_{x,y}(j,h,m)$$
$$\boldsymbol{F}^{\leq}_{x,y}(j,h,j,m) \Leftrightarrow \boldsymbol{f}^{\leq}_{x,y}(j,h,m)$$

When $j + h > i + m$:

$$\neg\boldsymbol{F}^{<}_{x,y}(j,h,i,m)$$
$$\neg\boldsymbol{F}^{\leq}_{x,y}(j,h,i,m)$$

Figure 7 shows how predicate $\boldsymbol{F}^{<}_{x,x}(i,0,j,1)$ is defined as conjunction of local relation $\boldsymbol{f}^{<}_{x,y}(i,0,1)$ and of $\boldsymbol{F}^{\leq}_{y,x}(i,1,j,1)$.

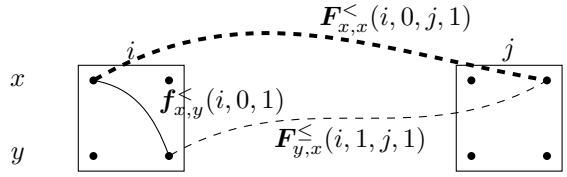

Figure 7: Definition of $\boldsymbol{F}^{<}$ by local relations $\boldsymbol{f}^{<}$.

The following formula $|CongruenceConstraints|_k$ defines congruence classes of locally equivalent points for relations $\overset{\prec}{\sim}, \overset{\preccurlyeq}{\sim}$ captured by predicates $\boldsymbol{F}^{<}$ and $\boldsymbol{F}^{\leq}$. In fact, observe that, since from $p_1 \preccurlyeq p_2$ we obtain $p'_1 \preccurlyeq p'_2$, for all p'_1 (resp. p'_2) that is locally equivalent to p_1 (resp. p'_2), then, in general, the congruence extends to $\overset{\preccurlyeq}{\sim}$; i.e., from $p_1 \overset{\preccurlyeq}{\sim} p_2$ we obtain $p'_1 \overset{\preccurlyeq}{\sim} p'_2$ for all p'_1, p'_2 locally equivalent to p_1, p_2. An analogous argument holds for $\overset{\prec}{\sim}$, $\overset{\succcurlyeq}{\sim}$ and $\overset{\succ}{\sim}$.

Let us consider two points (x, j, h) and (y, i, m) of $P_{\alpha\beta}$ such that $(x, j, h) \sim (y, i, m)$. The first set of formulae enforces congruence for all points (x, j', h') that are locally equivalent to (x, j, h). In particular, each formula defines relation $(x, j', h') \sim (y, i, m)$ when point (x, j', h') is the local equivalent of (x, j, h) belonging to the symbolic valuation at position j' on the right of the one at position j, i.e., $j' = j + 1$. As

the position j' increases by 1, then shift h' is decreased also by 1, i.e., $h' = h - 1$.

$$\frac{i \in [0, k+1] \text{ and } m \in [\lfloor\phi\rfloor, \lceil\phi\rceil] \text{ and } j \in [0, k] \text{ and } h \in [\lfloor\phi\rfloor + 1, \lceil\phi\rceil]}{\boldsymbol{F}^{<}_{x,y}(j, h, i, m) \Leftrightarrow \boldsymbol{F}^{<}_{x,y}(j+1, h-1, i, m)}$$

The second set of formulae is similar to the previous one and considers all points (y, i', m') locally equivalent to (y, i, m).

$$\frac{j \in [0, k+1] \text{ and } h \in [\lfloor\phi\rfloor, \lceil\phi\rceil] \text{ and } i \in [0, k] \text{ and } m \in [\lfloor\phi\rfloor + 1, \lceil\phi\rceil]}{\boldsymbol{F}^{<}_{x,y}(j, h, i, m) \Leftrightarrow \boldsymbol{F}^{<}_{x,y}(j, h, i+1, m-1)}$$

Predicates $\boldsymbol{b}^{>}_{x,y}, \boldsymbol{b}^{\geq}_{x,y}$ for local backward paths $\succ, \succcurlyeq$, predicates $\boldsymbol{B}^{>}_{x,y}, \boldsymbol{B}^{\geq}_{x,y}$ for backward paths $\overset{\succ}{\frown}, \overset{\succcurlyeq}{\frown}$ and congruence among points are defined similarly. For brevity, we only show the definition of $\boldsymbol{b}^{>}_{x,y}$ and $\boldsymbol{b}^{\geq}_{x,y}$, the others are straightforward.

$0 \leq j \leq k+1$ and $h \leq m$	$0 \leq j \leq k+1$ and $h > m$
$\boldsymbol{b}^{>}_{x,y}(j, h, m) \Leftrightarrow \mathbf{O}^{\boldsymbol{h}}\boldsymbol{x}(j) > \mathbf{O}^{\boldsymbol{m}}\boldsymbol{y}(j)$	$\neg\boldsymbol{b}^{>}_{x,y}(j, h, m)$
$\boldsymbol{b}^{\geq}_{x,y}(j, h, m) \Leftrightarrow \mathbf{O}^{\boldsymbol{h}}\boldsymbol{x}(j) \geq \mathbf{O}^{\boldsymbol{m}}\boldsymbol{y}(j)$	$\neg\boldsymbol{b}^{\geq}_{x,y}(j, h, m)$

for all $h, m \in [\lfloor\phi\rfloor, \lceil\phi\rceil]$. When both $x, y \in const(\phi)$ then $\boldsymbol{b}^{>}_{x,y}(j, h, m) \Leftrightarrow x > y$ and $\boldsymbol{b}^{\geq}_{x,y}(j, h, m) \Leftrightarrow x \geq y$ for all $0 \leq j \leq k+1$ and $h \leq m$; $\neg\boldsymbol{b}^{>}_{x,y}(j, h, m)$ and $\neg\boldsymbol{b}^{\geq}_{x,y}(j, h, m)$ for all $0 \leq j \leq k+1$ and $h > m$.

Finally, the condition of existence defined by Formula (1) is encoded by the following QF-EU$\mathcal{D}$ formula. The condition is parametric with respect to a pair of variables $x, x' \in V \cup const(\phi)$. The condition is meaningful only if $x \neq x'$ and if $x \notin const(\phi)$ or $x' \notin const(\phi)$. In fact, a constant value never generates a strict (forward or backward) path; therefore, two constants cannot satisfy the condition of non-existence of an arithmetical model. Formula $C_{x,x'}$ below captures the existence in $\rho(|\alpha'|)$ of a strict relation $<$ between two points, one of a forward and one of backward path, which involve variables x and x'. Variable $\boldsymbol{loop}$ has already been introduced in Section 3.3: it defines the position where, in $\alpha\beta$, suffix β starts (as already explained $|\alpha'| = \boldsymbol{loop}$).

$$C_{x,x'} := \bigvee_{h,h' \in [\lfloor\phi\rfloor, \lceil\phi\rceil]} \left(\begin{array}{c} \left(\begin{array}{c} \boldsymbol{F}^{\leq}_{x,x}(\boldsymbol{loop}, h, k+1, h) \wedge \boldsymbol{B}^{>}_{x',x'}(\boldsymbol{loop}, h', k+1, h') \\ \vee \\ \boldsymbol{F}^{<}_{x,x}(\boldsymbol{loop}, h, k+1, h) \wedge \boldsymbol{B}^{\geq}_{x',x'}(\boldsymbol{loop}, h', k+1, h') \end{array} \right) \\ \wedge \\ \boldsymbol{f}^{<}_{x,x'}(\boldsymbol{loop}, h, h') \vee \boldsymbol{b}^{>}_{x',x}(\boldsymbol{loop}, h', h) \end{array} \right)$$

In Formula $C_{x,x'}$, we use explicitly points that were symbolically represented in Formula (1): $p_1 = (x, loop, h)$, $p'_1 = (x', loop, h')$, $p_2 = (x, k+1, h)$, $p'_2 = (x', k+1, h')$. It is immediate to see that formula $\boldsymbol{f}^{<}_{x,x'}(\boldsymbol{loop}, h, h') \vee \boldsymbol{b}^{>}_{x',x}(\boldsymbol{loop}, h', h)$ encodes $p_1 \prec p'_1 \vee p'_1 \succ p_1$ of Formula (1) and formula $\boldsymbol{F}^{\leq}_{x,x}(\boldsymbol{loop}, h, k+1, h) \wedge \boldsymbol{B}^{>}_{x',x'}(\boldsymbol{loop}, h', k+1, h')$, encodes $p_1 \overset{\preccurlyeq}{\frown} p_2 \wedge p'_1 \overset{\preccurlyeq}{\frown} p'_2 \wedge p_1 \overset{\prec}{\frown} p_2$ (similarly for formula $\boldsymbol{F}^{<}_{x,x}(\boldsymbol{loop}, h, k+1, h) \wedge \boldsymbol{B}^{\geq}_{x',x'}(\boldsymbol{loop}, h', k+1, h')$).

Formula (1) corresponds to $\bigvee_{x,x'} C_{x,x'}$, where x, x' range over all pairs of elements of $V \cup const(\phi)$ such that $x \neq x'$ and at least of x, x' belongs to V. Then, $\neg \bigvee_{x,x'} C_{x,x'}$ captures the existence condition of an arithmetical model, and corresponds to the following formula:

$$\bigwedge_{\substack{x,x' \in V \cup const(\phi) \\ x \neq x', x \notin const(\phi) \vee x' \notin const(\phi)}} \neg C_{x,x'} \tag{4}$$

Finally, the following result is a direct consequence of Proposition 6 and of the fact that Formula (4) captures the negation of Formula (1).

Theorem 7. *Let ϕ be CLTLB(IPC*) formula. ϕ is satisfiable if, and only if, the following QF-EU($\mathcal{D}$) formula is satisfiable with respect to some $k \in \mathbb{N}$:*

$$|\phi|_k \wedge (4). \tag{5}$$

Proof. Suppose ϕ is satisfiable. Then, by Theorems 1–3, $|\phi|_k$ is satisfiable for some $k \in \mathbb{N}$. In addition, by Proposition 6, the induced locally consistent symbolic model $\rho = \alpha\beta^\omega$ satisfies the negation of Formula (1). Since Formula (4) captures the negation of Formula (1), then the model of $|\phi|_k$ also satisfies (4).

Conversely, if $|\phi|_k \wedge (4)$ is satisfiable, then by Theorems 1–3 there is $\rho = \alpha\beta^\omega$ such that $\rho \overset{sym}{\models} \phi$, and since Formula (4) captures the negation of Formula (1), by Proposition 6 ϕ is satisfiable. □

4.1. Simplifying the condition of existence of arithmetical models

In this section, we relax the condition of existence of an arithmetical model σ for sequences of symbolic valuations of CLTLB(IPC*) formulae. In fact, Property 1 is stronger than necessary in those cases in which not all variables appearing in a formula ϕ are compared against each other. Consider for example the following formula

$$\mathbf{G}(x < \mathrm{X}x \wedge \neg(y < \mathrm{X}y)) \tag{6}$$

which enforces strict increasing monotonicity for variable x and decreasing monotonicity for variable y. Figure 8 shows a symbolic model for Formula (6) which does not admit an arithmetic model, as it does not satisfy Property 1 (in fact, the strict forward path that visits all points $\{(x, i, 0)\}_{i \in \mathbb{N}}$ and the strict backward path that visits all points $\{(y, i, 0)\}_{i \in \mathbb{N}}$ are such that, for all i, $(x, i, 0) \prec (y, i, 0)$). However, in Formula

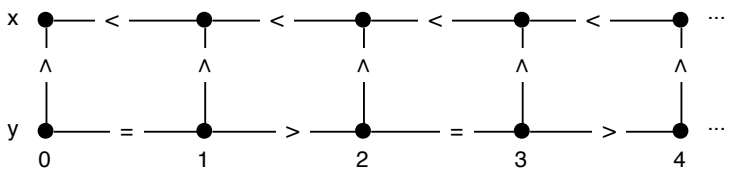

Figure 8: A symbolic model for Formula (6) that does not admit an arithmetic model.

(6) x and y are not compared, neither directly, nor indirectly: we can still obtain an

arithmetic model for Formula (6) if we disregard the relations between x and y in the symbolic model of Figure 8, and produce an assignment of the variables that only respects the relations between variables that are actually compared in the formula (i.e., x with itself, and y with itself). Figure 9 shows a "weaker" version of the symbolic model of Figure 8, one that is more concise to encode into QF-EU($\mathcal{D}$) formulae than the maximally consistent one, as it does not contain any comparison between unrelated terms.

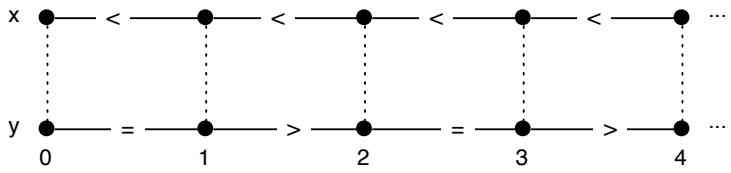

Figure 9: A weak symbolic model for Formula (6).

To characterize sequences of symbolic valuations which do not take into account relations among variables that are not compared with each other in a formula ϕ, we first introduce a binary relation $\asymp$ on variables of V. We say that, for a pair of variables $x, y \in V$, $x \asymp y$ if, and only if, there is an IPC* constraint $R(\mathrm{O}^i x, \mathrm{O}^j y)$ occurring in ϕ, for some $i, j \in \mathbb{Z}$ (we recall that $\mathrm{O}^n x$ stands for $\mathrm{Y}^{-n} x$ when $n < 0$, for $\mathrm{X}^n x$ when $n > 0$, and for x when $n = 0$). The equivalence relation obtained by considering the reflexive, symmetric and transitive closure of $\asymp$ induces a finite partition $\{V_1, \ldots, V_h\}$ of set V. Then, we introduce the notions of *weak symbolic valuation* and of *sequence of weak symbolic valuations*.

Definition 10. Given a symbolic valuation $sv \in SV(\phi)$, its *weak* version $\overline{sv}$ is obtained by removing from sv all relations $R(\mathrm{X}^i x, \mathrm{X}^j y)$ where $x \in V_l$ and $y \in V_t$ with $l \neq t$. We similarly define the weak version $\overline{\rho}$ of a sequence ρ of symbolic valuations.

Given a CLTLB(IPC*) formula ϕ, we indicate with $SV_w(\phi)$ the set of all its weak symbolic valuations. A weak symbolic model $\overline{\rho} \in SV_w(\phi)^\omega$ of ϕ is a sequence of weak symbolic valuations such that $\overline{\rho}, 0 \models^{sym} \phi$. Given $\rho \in SV(\phi)^\omega$ and its weak version $\overline{\rho}$, $G_{\overline{\rho}}$ is the subgraph of G_ρ obtained by removing all arcs between points $p = (x, j, h)$, $p' = (y, i, m)$ such that $x \in V_l$, $y \in V_t$, and $l \neq t$.

The next lemma shows that focusing on weak symbolic valuations is enough to determine whether symbolic models for ϕ exist or not.

Lemma 7. *Let ϕ be a CLTLB(IPC*) formula. Given $\rho \in SV(\phi)^\omega$ such that $\rho, 0 \models^{sym} \phi$, we have that $\overline{\rho}, 0 \models^{sym} \phi$. Conversely, given a sequence $\nu \in SV_w(\phi)$ of weak symbolic valuations, if $\nu, 0 \models^{sym} \phi$, then for any $\rho \in SV(\phi)$ such that $\overline{\rho} = \nu$ we also have that $\rho, 0 \models^{sym} \phi$.*

Proof. Assume that $\rho \models^{sym} \phi$. We only need to focus on the base case, as the inductive one is trivial. For all $i \geq 0$ and all $R(\alpha_1, \alpha_2)$ occurring in ϕ, $\rho, i \models^{sym} R(\alpha_1, \alpha_2)$ if,

and only if, $R(\alpha_1, \alpha_2) \in \rho(i)$. Since $R(\alpha_1, \alpha_2)$ occurs in ϕ then, by Definition 10, we have that $R(\alpha_1, \alpha_2) \in \overline{\rho}(i)$, hence $\overline{\rho}, i \models^{sym} R(\alpha_1, \alpha_2)$.

The converse case is similar. If $\nu \in SV_w(\phi)$ is such that $\nu, 0 \models^{sym} \phi$, then for all i and $R(\alpha_1, \alpha_2)$ that occurs in ϕ we have that $\nu, i \models^{sym} R(\alpha_1, \alpha_2)$ if, and only if, $R(\alpha_1, \alpha_2) \in \nu(i)$; in addition, for any ρ such that $\overline{\rho} = \nu$ we have $R(\alpha_1, \alpha_2) \in \rho(i)$ if, and only if, $R(\alpha_1, \alpha_2) \in \nu(i)$. Finally, $\nu, i \models^{sym} R(\alpha_1, \alpha_2)$ implies $\rho, i \models^{sym} R(\alpha_1, \alpha_2)$. □

We have the following variant of Lemma 2, which defines a condition of existence of arithmetical models for symbolic models defined on weak symbolic valuations.

Lemma 8. *Let ϕ be a CLTLB(IPC*) formula. Given an ultimately periodic, locally consistent sequence $\rho \in SV(\phi)^\omega$ of symbolic valuations, if there is $\sigma : \mathbb{Z} \times V \to D$ such that $\sigma, 0 \models \rho$, then Property 1 holds for graph $G_{\overline{\rho}}$. Conversely, if $\nu \in SV_w(\phi)^\omega$ is an ultimately periodic, locally consistent sequence of weak symbolic valuations such that Property 1 holds for graph G_ν, then there are σ, ρ such that $\overline{\rho} = \nu$ and $\sigma, 0 \models \rho$.*

Proof. If there is σ such that $\sigma, 0 \models \rho$ then, by Lemma 2, Property 1 holds for G_ρ. Since $G_{\overline{\rho}}$ is a subgraph of G_ρ, a fortiori Property 1 holds for $G_{\overline{\rho}}$.

Conversely, if Property 1 holds for G_ν, then each set of variables V_i, with $i \in \{1..h\}$, in which V is partitioned induces an ultimately periodic sequence ν_{V_i} of symbolic valuations that only include constraints on V_i, such that its graph $G_{\nu_{V_i}}$ is not connected to any other graph $G_{\nu_{V_j}}$, for $j \neq i$. Then, Lemma 2 can be applied to ν_{V_i}, which then admits an arithmetic model $\sigma_{V_i} : \mathbb{Z} \times V_i \to D$. By definition, each σ_{V_i} assigns a different set of variables, so the complete arithmetic model σ is simply the union of all σ_{V_i}. By Lemma 3, σ induces a sequence of symbolic valuations ρ, and $\sigma, 0 \models \rho$, $\overline{\rho} = \nu$ by construction. □

Thanks to Lemmata 7 and 8, in Formula (1) and in the corresponding QF-EU($\mathcal{D}$) encoding of Formula (4) we can focus only on relations between points that belong to the same set V_i.

5. Complexity and Completeness

Complexity

In the following we provide an estimation of the size of the formulae constituting the encoding of Section 3.3, including, where they are needed, the constraints of Section 4.

The encoding of Section 3.3 is linear in the size of the formula ϕ (and of the bound k). In fact, if m is the total number of subformulae and n is the total number of temporal operators $\mathbf{U}$ and $\mathbf{R}$ occurring in ϕ, the QF-EU$\mathcal{D}$ encoding requires $n + 1$ integer variables (one each for $\boldsymbol{loop}$ and the $\boldsymbol{j_\psi}$'s) and m unary predicates (one for each subformula in $cl(\phi)$).

The total size of the formulae in Section 4 is polynomial in the bound k, in the cardinality of the set of variables and constants, and in the size of symbolic valuations. In fact, the encoding of the condition for the existence of an arithmetical model requires

a QF-EU($\mathbb{N}, <, =$) formula of size quadratic in the length k, cubic in the number $|V|$ of variables, and double quadratic in the size of symbolic valuations.

Let λ be the size $\lambda = \lceil\phi\rceil - \lfloor\phi\rfloor + 1$ of symbolic valuations and V' be the set $V \cup const(\phi)$. The total number of non-trivial predicates $\boldsymbol{f}^{\leq}_{x,y}, \boldsymbol{f}^{<}_{x,y}$ (resp. $\boldsymbol{b}^{\geq}_{x,y}, \boldsymbol{b}^{>}_{x,y}$), i.e., those where $h \leq m$, is defined by the following parametric formula (where a, b are the sets to which x, y belong, respectively):

$$\begin{aligned} N(a,b) &= (k+1)\sum_{i=1}^{\lambda} |a| \cdot ((\lambda - i) + (|b| - 1) \cdot (\lambda - i + 1)) \\ &= (k+1)\left(|a||b|\frac{\lambda(\lambda+1)}{2} - |a|\lambda\right). \end{aligned}$$

Each predicate has fixed dimension and the number of non-trivial ones results from the sum of the following three cases:

- $x, y \in V$, which is $N(V, V)$
- $x \in V$, $y \in const(\phi)$, which is $N(V, const(\phi))$
- $x \in const(\phi)$, $y \in V$, which is $N(const(\phi), V)$.

that is bounded by $N_{local} = N(V', V') \leq (k+1)|V'|^2\lambda^2$.

To compute the size of formulae defining $\boldsymbol{F}^{\leq}_{x,y}, \boldsymbol{F}^{<}_{x,y}$ (resp. $\boldsymbol{B}^{\geq}_{x,y}, \boldsymbol{B}^{>}_{x,y}$) we first determine the number of pairs of points for which $\boldsymbol{F}^{\leq}_{x,y}(j,h,i,m)$ is not trivially false. The following function $N_{p,p'}$

$$\begin{aligned} N_{p,p'} &= |V'| \sum_{i=\lfloor\phi\rfloor}^{k+\lceil\phi\rceil} |V'|(k + \lceil\phi\rceil - i) = |V'|^2 \sum_{i=0}^{k+\lambda-1} i = |V'|^2 \frac{(k+\lambda-1)(k+\lambda)}{2} \\ &\leq |V'|^2(k+\lambda)^2 \end{aligned}$$

corresponds to the number of pairs of points p, p' that generate non-trivial predicates $\boldsymbol{F}^{\leq}_{x,y}, \boldsymbol{F}^{<}_{x,y}$ (resp. $\boldsymbol{B}^{\geq}_{x,y}, \boldsymbol{B}^{>}_{x,y}$) because their position is such that $sv(p_1) + shift(p_1) \leq sv(p_2) + shift(p_2)$ (resp. $sv(p_1) + shift(p_1) \geq sv(p_2) + shift(p_2)$). We compute the size of (non-trivial) formulae (2)-(3) defining $\boldsymbol{F}^{<}_{x,y}, \boldsymbol{F}^{\leq}_{x,y}$ (and $\boldsymbol{B}^{>}_{x,y}, \boldsymbol{B}^{\geq}_{x,y}$) by counting the number of subformulae involved in their definition. We consider only the case for $\boldsymbol{F}^{<}_{x,y}$ because the others have the same (worst) complexity. Each Formula (2) involves, in the worst case (i.e., for points that do not belong to the same symbolic valuation), $|V|-1$ variables $z \in V$ with respect to λ different positions u. Then, an instance of (2) requires at most $(|V|-1)\lambda$ disjuncts. The upper bound for the total size of all formulae defining predicates $\boldsymbol{F}^{\leq}_{x,y}, \boldsymbol{F}^{<}_{x,y}$ (resp. $\boldsymbol{B}^{\geq}_{x,y}, \boldsymbol{B}^{>}_{x,y}$) is

$$N_{far} = N_{p,p'} 2(|V| - 1)\lambda \leq \lambda|V||V'|^2(k+\lambda)^2 \leq \lambda|V'|^3(k+\lambda)^2.$$

The analysis of formulae $|CongruenceConstraints|_k$ shows that each point belongs to λ symbolic valuations (e.g., if $\lceil\phi\rceil = 0$, $\lfloor\phi\rfloor = -1$, then $\lambda = 2$, and points $(x, 4, 1)$ and $(x, 5, 0)$ correspond to the same element), and for all pairs p_1, p_2 we define

the consistency of the definition of predicate $\boldsymbol{F}^{<}_{x,y}$ among the λ points corresponding to p_1 and the λ points corresponding to p_2. Therefore, we need at most

$$N_{CC} = 4\lambda^2|V'|^2k^2$$

constraints $|CongruenceConstraints|_k$, where each constraint has fixed dimension.

Finally, predicate $C_{x,x'}$ appears in Formula (4) once for each of the $|V'||V|\lambda^2$ pairs of points x, x'. In addition, each instance of $C_{x,x'}$ has λ^2 disjuncts, one for each possible pair $h, h' \in [\lfloor\phi\rfloor, \lceil\phi\rceil]$. Therefore, the total size of Formula (4) is $N_C = |V||V'|\lambda^4$.

Finally, the complete set of formulae that we require to capture the existence condition of arithmetical models over discrete domains has the following total size:

$$4N_{local} + 4N_{far} + 4N_{CC} + N_C \leq$$
$$4(k+1)|V'|^2\lambda^2 + 4\lambda|V'|^3(k+\lambda)^2 + 16\lambda^2|V'|^2k^2 + |V||V'|\lambda^4.$$

In conclusion, for a given formula ϕ, the parameters λ and $|V'|$ are fixed, hence the size is $\mathcal{O}(k^2)$.

Completeness

Completeness has been studied in depth for Bounded Model Checking. Given a state-transition system M, a temporal logic property ϕ and a bound $k > 0$, BMC looks for a witness of length k for $\neg\phi$. If no witness exists then length k may be increased and BMC may be reapplied. In principle, the process terminates when a witness is found or when k reaches a value, the *completeness threshold* (see Definition 4), which guarantees that if no counterexample has been found so far, then no counterexample disproving property ϕ exists in the model. LTL always has a completeness threshold; [19] shows a procedure to estimate an over-approximation of the value, by satisfying a formula representing the existence of an accepting run of the product automaton $M \times B_{\neg\phi}$, where $B_{\neg\phi}$ is the Büchi automaton for $\neg\phi$ and M is the system to be verified.

In [20] we have already given a positive answer to the problem of whether there exists a completeness threshold for the satisfiability problem of CLTLB($\mathcal{D}$), provided that ultimately periodic symbolic models of the form $\alpha\beta^\omega$ of CLTLB($\mathcal{D}$) formulae admit an arithmetic model. By the results of Section 2.4.1 this occurs when the constraint system $\mathcal{D}$ has the completion property, or when it is possible to define an automaton $\mathcal{A}_C$. In [20] we used a mixed automata- and logic-based approach to prove the existence of a completeness threshold. In that approach, automata $\mathcal{A}_C$ and $\mathcal{A}_\ell$ described in Section 2.4 are represented by means of two CLTLB($\mathcal{D}$) formulae $\phi_{\mathcal{A}_C}$ and $\phi_{\mathcal{A}_\ell}$. Formulae $\phi_{\mathcal{A}_C}$ and $\phi_{\mathcal{A}_\ell}$ capture the runs of automata $\mathcal{A}_C$ and $\mathcal{A}_\ell$, respectively. Then, checking the satisfiability for ϕ is reduced to studying a finite amount of k-satisfiability problems of formula $\phi \wedge \phi_{\mathcal{A}_C} \wedge \phi_{\mathcal{A}_\ell}$, for increasing values of k. Automaton $\mathcal{A}_\ell$ recognizes sequences of locally consistent symbolic valuations, so its runs are the models of formula $\phi_{\mathcal{A}_\ell} := \mathbf{G}(\bigvee_1^m sv_i)$. Since the bounded representation of formulae (see Section 3.3) is not contradictory (i.e., two consecutive symbolic valuations are satisfiable when they are locally consistent), the previous formula exactly represents words of $\mathscr{L}(\mathcal{A}_\ell)$.

Formula $\phi_{\mathcal{A}_C}$, instead, is derived from automaton $\mathcal{A}_C$, by means of the translation in [21]. Automaton $\mathcal{A}_C$ is built by complementing automaton $\mathcal{A}_{\neg C}$ [22], recognizing the complement language of $\mathscr{L}(\mathcal{A}_C)$, which is obtained according to the procedure proposed in [9]. Finally, to check the satisfiability of ϕ we verify whether formula $\phi \wedge \phi_{\mathcal{A}_C} \wedge \phi_{\mathcal{A}_\ell}$ is k-satisfiable, with $k \in \mathbb{N}$. The existence of a finite completeness threshold for the procedure above is a consequence of the existence of automaton $\mathcal{A}_\phi$ (see Section 2.4) recognizing symbolic models of ϕ, and of Lemma 2 and Proposition 2. Let $rd(\mathcal{A}_\phi)$ be the recurrence diameter of $\mathcal{A}_\phi$, i.e., the longest loop-free path in the automaton that starts from an initial state [23]. Then, if formula $\phi \wedge \phi_{\mathcal{A}_C} \wedge \phi_{\mathcal{A}_\ell}$ is not k-satisfiable for all $k \in [1, rd(\mathcal{A}_\phi) + 1]$, then there is no ultimately periodic symbolic model ρ such that both $\rho, 0 \overset{sym}{\models} \phi$ and there exists an arithmetic model σ with $\sigma, 0 \models \rho$. Hence, formula ϕ is unsatisfiable. Otherwise, we have found an ultimately periodic symbolic model ρ of length $k > 0$ which admits an arithmetic model σ. From the k-bounded solution, we have a symbolic model $\rho = \alpha\beta^\omega$ and its bounded arithmetic model σ_k. The infinite model σ is built from σ_k by iterating infinitely many times the sequence of symbolic valuations in β. Therefore, the completeness bound for BSP of CLTLB($\mathcal{D}$) formulae is defined by the recurrence diameter of $\mathcal{A}_\phi$.

Thanks to the results of the previous sections, we can simplify the method presented in [20]. We avoid the construction of automaton $\mathcal{A}_{\neg C}$ through Safra's method and the construction of set $SV(\phi)$. In particular, we take advantage of the definition of k-bounded models of ϕ. By Lemma 4, a finite sequence σ_k of $\mathcal{D}$-valuations induces a unique locally consistent sequence of symbolic valuations ρ, such that $\sigma_k, i \models \rho(i)$, for all $i \in [0, k]$. Therefore, we do not need to precompute set $SV(\phi)$ of symbolic valuations and formula $\phi_{\mathcal{A}_\ell}$ is no longer needed to obtain a finite locally consistent sequence of symbolic valuations. If ϕ is a formula of CLTLB($\mathcal{D}$) and $\mathcal{D}$ has the completion property, we can simply solve k-satisfiability problems for ϕ instead of $\phi \wedge \phi_{\mathcal{A}_\ell}$; when $\mathcal{D}$ does not have the completion property, Formula (1) allows us to avoid the construction of $\mathcal{A}_C$. In the first case, by Theorems 1 – 3 and Proposition 5 $|\phi|_k$ is satisfiable if, and only if, there is an ultimately periodic run $\alpha\beta^\omega$ which is recognized by automaton $\mathcal{A}_s \times \mathcal{A}_\ell$. In the second case, Proposition 6 guarantees that $|\phi|_k$ is satisfiable and Formula (1) does not hold if, and only if, ϕ is satisfiable. Therefore, model $\alpha\beta^\omega$ obtained by solving the k-satisfiability problem belongs to the language recognized by automaton $\mathcal{A}_s \times \mathcal{A}_\ell$ and also to the one recognized by $\mathcal{A}_C$.

The completeness property still holds without the explicit representation of automata $\mathcal{A}_\ell$ and $\mathcal{A}_C$ in the formula we check for satisfiability. Since the role of Formula (1) is to filter, by eliminating edges in the automaton, some of the symbolic models of ϕ which, in turn, by Theorems 1 – 3 correspond to the runs of automaton $\mathcal{A}_s \times \mathcal{A}_\ell$, the completeness threshold for our decision procedure can be over-approximated by the recurrence diameter of $\mathcal{A}_s \times \mathcal{A}_\ell$, which is at most exponential in the size of ϕ. Since the number of control states of automaton $\mathcal{A}_s$ is at most $\mathcal{O}(2^{|\phi|})$, a rough estimation for the completeness threshold is given by the value $|SV(\phi)| \cdot 2^{|\phi|}$. The number of symbolic valuations $|SV(\phi)|$ is, in the worst case, exponential in the size of formula ϕ [9].

6. Applications of k-bounded satisfiability

The decision procedure defined in this paper has been implemented in our bounded satisfiability checker $\mathbb{Z}$ot (available at `http://zot.googlecode.com`). The ae^2Zot plug-in of $\mathbb{Z}$ot solves k-satisfiability for CLTLB over Quantifier-Free Presburger arithmetic (QFP), of which IPC* is a fragment, but it also supports the constraint system $(\mathbb{R}, <, =)$. Even if constraint systems like IPC*, or fragments thereof, do not provide a counting mechanism (provided, for instance, through the addition of functions, such as $+$ in QFP), they can still be used to represent an abstraction of a richer transition system. In fact, functions like addition, or in general relations over unbounded variables which embed a counting mechanism, make the satisfiability problem of CLTLB undecidable (see [9, Section 9.3]).

We next exemplify the use of the CLTLB logic to specify and verify systems behavior, thus highlighting the applicability of the approach.

We use CLTLB over $(D, <, =)$ to specify a sorting process of a sequence of fixed length N of values in D. Though for reasons of conciseness we do not present all details and formulae of the example, we provide its salient points. Let $\mathbf{v} \in D^N$ be the (initial) vector that we want to sort and $\mathbf{a} \in D^N$ be the vector during each step of sorting. We write $\mathbf{v}(i)$ for the i-th component of $\mathbf{v}$, $1 \leq i \leq N$. Notice that we will use the notation $\mathbf{a}(i)$, which, strictly speaking, is not a CLTLB term; however, since the length of the array is fixed, we can use N variables a_i to represent the elements of $\mathbf{a}$, one for each $\mathbf{a}(i)$. Then, in the following, if $\mathbf{a}(i)$ is replaced with a_i, one obtains CLTLB$(D, <, =)$ formulae. We define a set of formulae representing a sorting process which swaps unsorted pairs of values at some nondeterministically chosen position in the vector (we report here only the most relevant formulae). A variable $p \in [0, N-1]$ stores the position of elements which are a candidate pair for swapping; i.e., $p = i$ means that $\mathbf{a}(i)$ is swapped with $\mathbf{a}(i+1)$, while $p = 0$ means that no elements are swapped (0 is not a position of the vector). A nondeterministic algorithm can swap two arbitrary elements in $[1, N]$; then, the only constraint on variable p is that it holds that $0 \leq p < N$, i.e.: $\mathbf{G}(p < N \wedge p \geq 0)$. An unsorted pair of values is indexed by a nonzero value of p:

$$\mathbf{G}\left(\bigwedge_{i\in[1,N-1]} p = i \Rightarrow \mathbf{a}(i) > \mathbf{a}(i+1)\right).$$

A swap between two adjacent positions of $\mathbf{a}$ is formalized by the following formula:

$$\mathbf{G}\left(\bigwedge_{i\in[1,N-1]} p = i \Rightarrow \mathrm{X}\mathbf{a}(i) = \mathbf{a}(i+1) \wedge \mathrm{X}\mathbf{a}(i+1) = \mathbf{a}(i)\right).$$

Vector $\mathbf{a}$ is unchanged when no pairs are candidate for swapping: $\mathbf{G}(p = 0 \Rightarrow \bigwedge_{i\in[1,N]}(\mathbf{a}(i) = \mathrm{X}\mathbf{a}(i)))$. For brevity, we omit the formula defining the initial configuration of vectors, which imposes that, at instant 0, vectors $\mathbf{a}$ and $\mathbf{v}$ are equal (i.e., $\mathbf{a}(i) = \mathbf{v}(i)$ for all $1 \leq i \leq N$), and that $\mathbf{v}$ does not contain duplicates. Various properties of the algorithm have been verified through the ae^2Zot plugin of the $\mathbb{Z}$ot tool,

e.g., whether there exists a way to sort array $\mathbf{a}$ within k steps (with k the verification bound), which is formalized by the following formula:

$$\mathbf{F}\left(\bigwedge_{i\in[1,N-1]}(\mathbf{a}(i)\leq\mathbf{a}(i+1))\wedge\bigwedge_{i\in[1,N]}\bigvee_{j\in[1,N]}(\mathbf{a}(i)=\mathbf{v}(j))\right).$$

7. Related works

For some constraint system $\mathcal{D}$ more expressive than IPC*, the future fragment CLTL($\mathcal{D}$) can encode runs of two-counter (Minsky) machines. For example, to represent increment and decrement instructions the grammar of formulae ξ of IPC* can be enriched with formulae of the form $x < y + d$, where $d \in D$ and x, y are variables (these correspond to difference logic – DL – constraints). Hereafter, we write $\mathrm{CLTL}^{\mathrm{b}}_{\mathrm{a}}(\mathcal{D})$ to denote the language of CLTL formulae such that the cardinality of V is a and $\lceil\phi\rceil$ is b (while $\lfloor\phi\rfloor$ is of course 0).

The first undecidability result for the satisfiability of CLTL is given by Comon and Cortier [5, Theorem 3], showing that halting runs of a Minsky machine can be encoded into $\mathrm{CLTL}^1_3(\mathrm{DL})$ formulae, where one auxiliary counter encodes control states. Therefore, the satisfiability problem for $\mathrm{CLTL}^1_3(\mathrm{DL})$ is Σ^1_1-hard. The authors suggest a way to regain decidability by means of a syntactic restriction on formulae including the $\mathbf{U}$ temporal operator. The "flat" fragment of $\mathrm{CLTL}^1_\omega(\mathrm{DL})$ consists of CLTL formulae such that subformula ϕ of $\phi\mathbf{U}\psi$ is $\top$, $\bot$ or a conjunction $\zeta_1 \wedge \cdots \wedge \zeta_m$ where $\zeta_i \in \mathrm{DL}$. The fragment has a nice correspondence with a special class of counter system (flat relational counter system) with Büchi acceptance condition, for which the emptiness problem is decidable. Satisfiability is undecidable also in the case of $\mathrm{CLTL}^2_1(\mathrm{DL})$ and $\mathrm{CLTL}^1_2(\mathrm{DL})$. In fact, even though $\mathrm{CLTL}^2_1(\mathrm{DL})$ has only one variable, it is expressive enough to encode runs of Minsky machines: models of $\mathrm{CLTL}^2_1(\mathrm{DL})$ formulae can represent counter c_1 at even positions and counter c_2 at odd positions. The recurrence problem for nondeterministic Minsky machines, which is Σ^1_1-hard [24], can be reduced to the satisfiability problem for $\mathrm{CLTL}^2_1(\mathrm{DL})$, which then results Σ^1_1-hard. It also follows that the satisfiability problem of CLTL with two integer variables, $\mathrm{CLTL}^1_2(\mathrm{DL})$ is Σ^1_1-hard. In fact, formulae of $\mathrm{CLTL}^2_1(\mathrm{DL})$ can be syntactically translated to formulae of $\mathrm{CLTL}^1_2(\mathrm{DL})$ by means of a map f such that ϕ belonging to $\mathrm{CLTL}^2_1(\mathrm{DL})$ is satisfiable if, and only if, $f(\phi)$ belonging to $\mathrm{CLTL}^1_2(\mathrm{DL})$ is satisfiable. Both the languages $\mathrm{CLTL}^2_1(\mathrm{DL})$ and $\mathrm{CLTL}^1_2(\mathrm{DL})$ are indeed Σ^1_1-complete, by using a reduction from the Σ^1_1-hard model-checking problem to their satisfiability.

The satisfiability (and model-checking) problem for CLTL over structure $(D, <, =)$ with $D \in \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$ is studied in [9], and for IPC* in [8]. Decidability of the satisfiability problem for the above cases is shown by means of an automata-based approach similar to the standard case for LTL. Satisfiability for $\mathrm{CLTL}^\omega_\omega(\mathrm{IPC}^*)$ and $\mathrm{CLTL}^\omega_\omega(<,=)$ over $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$ is obtained by Demri and Gascon in [13] by reducing it to the emptiness of Büchi automata. Given a CLTL formula ϕ, it is possible to define an automaton $\mathcal{A}_\phi$ such that ϕ is satisfiable if, and only if, $\mathscr{L}(\mathcal{A}_\phi)$ is not empty. Since the emptiness of $\mathscr{L}(\mathcal{A}_\phi)$ in the considered structures is decidable with a PSPACE upper bound (in the dimension of ϕ), then the satisfiability problem is also decidable with

the same complexity. Since the procedure is purely symbolic, constraints representing equality relation $x = d$ and constraints of the form $x \equiv_c d$, with $d, c \in D$, are explicitly considered, as no arithmetical model σ is available. A symbolic valuation is defined there as a triple $\langle S_1, S_2, S_3 \rangle$ where S_1 is a maximally consistent set of $\mathcal{D}$-constraints over $terms(\phi)$ and $const(\phi)$; S_2 is a set of constraints of the form $x = d$, and S_3 is a set of constraints $x \equiv_K c$, where constant K is the least common multiple of constants occurring in constraints $x \equiv_c y$ and $x \equiv_c y + d$.

Schüle and Schneider [25] provide a general algorithm to decide bounded $\mathrm{LTL}(L)$ model-checking problems of infinite state systems where L is a general underlying logic. An $\mathrm{LTL}(L)$ formula ϕ is translated into an equivalent Büchi automaton $\mathcal{A}_\phi$ which is symbolically represented by means of a structure defining its transition relation and acceptance condition. Then, the $\mathrm{LTL}(L)$ model-checking problem is reduced to the μ-calculus model-checking problem modulo L, i.e., a verification of a fixpoint problem for a given Kripke structure with respect to symbolic representations of $\mathcal{A}_\phi$ and the underlying language L. Whenever properties are neither proved nor disproved over finite computations, their truth value cannot be defined. For this reason, the authors adopt a three-valued logic to evaluate formulae whose components may have undefined value. Bounded model-checking is performed essentially by computing approximate fixpoint sets of the desired formula and by checking whether the initial condition is a subset of such set of states. The work of [25] is based on previous results presented in [26], which defines a hierarchy of Büchi automata (and, therefore, temporal formulae) for which infinite state bounded model-checking is complete. The specification language of [26] is the quantifier-free fragment of Presburger LTL, $\mathrm{LTL}(\mathrm{PA})$, with past-time temporal modalities. The bounded model-checking problem is defined with respect to Kripke structures (S, I, R) and it is solved by means of a reduction to the satisfiability of Presburger formulae. In general, acceptance conditions of Büchi automata, requiring that some states are visited infinitely often, cannot be handled immediately by bounded approaches which do not consider ultimately periodic models used, for instance, in the bounded model-checking approach of Biere et al. [3] or in the encoding of Büchi automata of de Moura et al. [27]. Therefore, Schüle and Schneider follow a different approach, tailored to bounded verification, and focus on the analysis of some classes of LTL formulae, denoted $\mathrm{TL}_{\mathbf{F}}$ and $\mathrm{TL}_{\mathbf{G}}$, such that the corresponding Büchi automaton has a simpler accepting condition which does not involve infinite computations. $\mathrm{TL}_{\mathbf{F}}$ and $\mathrm{TL}_{\mathbf{G}}$ are the sets of LTL formulae such that each occurrence of a weak/strong temporal operator is negative/positive and positive/negative, respectively. LTL formulae are then represented symbolically by an automaton which is built using the method proposed by Clarke et al. in [28] rather than using the Vardi-Wolper construction [14].

Reducing the model-checking problem to Presburger satisfiability is a rather standard approach when dealing with infinite-state systems. Demri et al. in [29] show how to solve the $\mathrm{LTL}(\mathrm{PA})$ model-checking problem for the class of *admissible* counter systems, which are finite state automata endowed with variables over $\mathbb{Z}$ whose transitions are labeled by Presburger formulae. In [29] the authors study the decidability of the model-checking problem for admissible counter systems with respect to the first-order CTL^* language over Presburger formulae.

Hodkinson et al. study decidable fragments of first-order temporal logic in [30].

Although some axiomatizations of first-order temporal logic are known, various incompleteness results induce the authors to study useful fragments with expressiveness between that of propositional and of first-order temporal logic. Hodkinson et al. are interested in studying the satisfiability problem and they do not consider the model-checking problem, which requires a formalism defining the interpretation of first-order variables over time. In other words, variables do not vary over time and their temporal behavior is not relevant. The languages investigated by the authors are obtained by restricting both the first-order part and the temporal part.

Bultan et al. present a symbolic model checker for analyzing programs with unbounded integer domains [31]. Programs are defined by an event-action language where atomic events are expressed by Presburger formulae over programs variables V. Semantics of programs is defined in terms of infinite transition systems where the states are determined by the values of variables. The specification language is a CTL-like temporal logic enriched with Presburger-definable constraints over V. Solving the CTL model-checking problem involves the computation of least fixpoints over sets of programs states: the abstract interpretation of Cousot and Cousot [32] provides a method to compute approximation of fixpoints. Model-checking is done conservatively: the approximation technique admits false negatives, i.e., the solver may indicate that a property does not hold when it actually does. Programs are analyzed symbolically by means of symbolic execution techniques and they are represented by means of Presburger-definable transition systems where Presburger formulae represent symbolically the transition relation and the set of program states. Then, the state space is partitioned to reduce the complexity of verification and to obtain decidability for some classes of temporal properties, such as reachability ones. Experimental results, based on the standard Bakery algorithm and the Ticket mutual-exclusion algorithm, show the effectiveness of the method when verification involves a mutual exclusion requirement.

8. Conclusions and further developments

In this paper, we provide a procedure for deciding the satisfiability problem for CLTLB over some suitable constraint systems. The main advantage of our approach is that it allowed us to implement the first effective tool based on SMT-solvers for those logics. On one side, this method illustrates a new way to solve verification problems of formalisms dealing with variables ranging over infinite domains and having an inherent notion of discrete time as that of LTL. Instead of building an automaton for proving the satisfiability of a formula (which would be unfeasible in practice), we devise a direct method to construct one of its accepting runs which define a model for the formula. On the other hand, our framework constitutes a foundation for defining extensions to handle different temporal formalisms. In [33] we use the same approach presented in this paper to allow for the use of variables whose behavior is restricted to clocks [34] into CLTLB$(\mathbb{R}, <, =)$. A clock is a nonnegative variable accumulating the time elapsed since the position it was reset to 0; hence, a clock can be used to measure time between two discrete positions. Typically, all clocks proceed with the same rate. In [33] we prove the decidability and the complexity of the satisfiability problem for a version of CLTLB endowed with a finite set of clocks, and we provide a working implementation by means of SMT-solvers, which extends the one presented in this work.

In [35], we devise a reduction of MITL formulae, interpreted over continuous time, into equisatisfiable CLTLB formulae with clocks. Therefore, we were able to provide the first actual implementation of a satisfiability solver for MITL.

Acknowledgments

The authors gratefully thank the reviewers for their suggestions, which have greatly helped in improving the paper.

References

[1] G. Holzmann, The model checker SPIN, IEEE Transactions on Software Engineering 23 (5) (1997) 279 –295.

[2] E. Clarke, K. McMillan, S. Campos, V. Hartonas-Garmhausen, Symbolic model checking, in: Computer Aided Verification, Vol. 1102 of Lecture Notes in Computer Science, 1996, pp. 419–422.

[3] A. Biere, A. Cimatti, E. Clarke, Y. Zhu, Symbolic model checking without BDDs, in: Tools and Algorithms for the Construction and Analysis of Systems, Vol. 1579 of Lecture Notes in Computer Science, 1999, pp. 193–207.

[4] M. Pradella, A. Morzenti, P. San Pietro, Bounded satisfiability checking of metric temporal logic specifications, ACM Transactions on Software Engineering and Methodology (TOSEM) 22 (3) (2013) 20:1–20:54.

[5] H. Comon, V. Cortier, Flatness is not a weakness, in: Computer Science Logic, Vol. 1862 of Lecture Notes in Computer Science, 2000, pp. 262–276.

[6] S. Demri, D. D'Souza, An automata-theoretic approach to constraint LTL, in: FST TCS 2002: Foundations of Software Technology and Theoretical Computer Science, Vol. 2556 of Lecture Notes in Computer Science, 2002, pp. 121–132.

[7] S. Demri, R. Gascon, The effects of bounding syntactic resources on Presburger LTL, Tech. Rep. LSV-06-5, LSV (2006).

[8] S. Demri, R. Gascon, The effects of bounding syntactic resources on Presburger LTL, in: International Syposium on Temporal Representation and Reasoning (TIME), IEEE Computer Society, 2007, pp. 94–104.

[9] S. Demri, D. D'Souza, An automata-theoretic approach to constraint LTL, Information and Computation 205 (3) (2007) 380–415.

[10] Microsoft Research, Z3: An efficient SMT solver, http://research.microsoft.com/en-us/um/redmond/projects/z3/ (2009).

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

[12] S. Demri, LTL over integer periodicity constraints, in: Foundations of Software Science and Computation Structures, Vol. 2987 of Lecture Notes in Computer Science, 2004, pp. 121–135.

[13] S. Demri, R. Gascon, Verification of qualitative $\mathbb{Z}$ constraints, in: CONCUR 2005 - Concurrency Theory, Vol. 3653 of Lecture Notes in Computer Science, 2005, pp. 518–532.

[14] M. Y. Vardi, P. Wolper, An automata-theoretic approach to automatic program verification, in: Proceedings, Symposium on Logic in Computer Science, IEEE Computer Society, 1986, pp. 332–344.

[15] A. Biere, A. Cimatti, E. M. Clarke, O. Strichman, Y. Zhu, Bounded model checking, Advances in Computers 58 (2003) 118–149.

[16] M. M. Bersani, L. Cavallaro, A. Frigeri, M. Pradella, M. Rossi, SMT-based verification of LTL specification with integer constraints and its application to runtime checking of service substitutability, in: IEEE International Conference on Software Engineering and Formal Methods, 2010, pp. 244–254.

[17] A. Biere, K. Heljanko, T. A. Junttila, T. Latvala, V. Schuppan, Linear encodings of bounded LTL model checking, Logical Methods in Computer Science 2 (5) (2006) 1–64.

[18] M. M. Bersani, A. Frigeri, A. Morzenti, M. Pradella, M. Rossi, P. San Pietro, Constraint ltl satisfiability checking without automata, CoRR abs/1205.0946.

[19] E. Clarke, D. Kroening, J. Ouaknine, O. Strichman, Completeness and complexity of bounded model checking, in: Verification, Model Checking, and Abstract Interpretation, Vol. 2937 of Lecture Notes in Computer Science, 2004, pp. 85–96.

[20] M. M. Bersani, A. Frigeri, M. Rossi, P. San Pietro, Completeness of the bounded satisfiability problem for constraint LTL, in: Reachability Problems, Vol. 6945 of Lecture Notes in Computer Science, Springer, 2011, pp. 58–71.

[21] A. P. Sistla, E. M. Clarke, The complexity of propositional linear temporal logics, Journal of the ACM 32 (3) (1985) 733–749.

[22] S. Safra, On the complexity of omega -automata, in: IEEE Annual Symposium on Foundations of Computer Science, IEEE Computer Society, 1988, pp. 319–327.

[23] D. Kroening, O. Strichman, Efficient computation of recurrence diameters, in: Verification, Model Checking, and Abstract Interpretation, Vol. 2575 of Lecture Notes in Computer Science, 2003, pp. 298–309.

[24] R. Alur, T. A. Henzinger, A really temporal logic, Journal of the ACM 41 (1) (1994) 181–204.

[25] T. Schüle, K. Schneider, Bounded model checking of infinite state systems, Formal Methods in System Design 30 (2007) 51–81.

[26] T. Schüle, K. Schneider, Bounded model checking of infinite state systems: exploiting the automata hierarchy, in: Proceedings of the ACM and IEEE International Conference on Formal Methods and Models for Co-Design, 2004, pp. 17 – 26.

[27] L. M. de Moura, H. Rueß, M. Sorea, Lazy theorem proving for bounded model checking over infinite domains, in: Automated Deduction-CADE-18, Vol. 2392 of Lecture Notes in Computer Science, 2002, pp. 438–455.

[28] E. Clarke, O. Grumberg, K. Hamaguchi, Another look at LTL model checking, in: Formal Methods in System Design, Springer-Verlag, 1994, pp. 415–427.

[29] S. Demri, A. Finkel, V. Goranko, G. van Drimmelen, Model-checking CTL* over flat Presburger counter systems, Journal of Applied Non-Classical Logics 20 (4) (2010) 313–344.

[30] I. M. Hodkinson, F. Wolter, M. Zakharyaschev, Decidable fragment of first-order temporal logics, Annals of Pure and Applied Logic 106 (1–3) (2000) 85–134.

[31] T. Bultan, R. Gerber, W. Pugh, Model-checking concurrent systems with unbounded integer variables: symbolic representations, approximations, and experimental results, ACM Transactions on Programming Languages and Systems 21 (1999) 747–789.

[32] P. Cousot, R. Cousot, Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints, in: Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages, POPL '77, 1977, pp. 238–252.

[33] M. M. Bersani, M. Rossi, P. San Pietro, A tool for deciding the satisfiability problem of continuous-time metric temporal logic, in: TIME 2013, 2013.

[34] R. Alur, D. L. Dill, A theory of timed automata, Theoretical Computer Science 126 (2) (1994) 183–235.

[35] M. M. Bersani, M. Rossi, P. San Pietro, Deciding the satisfiability of mitl specifications, CoRR abs/1307.4469.