Bounded Reachability for Temporal Logic over Constraint Systems

Marcello M. Bersani[1], Achille Frigeri[1], Angelo Morzenti[1],
Matteo Pradella[2], Matteo Rossi[1], and Pierluigi San Pietro[1]

[1]Politecnico di Milano, Milano, Italy, {bersani,frigeri,morzenti,rossi,sanpietro}@elet.polimi.it
[2]CNR IEIIT-MI, Milano, Italy, pradella@elet.polimi.it

Abstract

This paper defines CLTLB($\mathcal{D}$), an extension of PLTLB (PLTL with both past and future operators) augmented with atomic formulae built over a constraint system $\mathcal{D}$. The paper introduces suitable restrictions and assumptions that make the satisfiability problem decidable in many cases, although the problem is undecidable in the general case. Decidability is shown for a large class of constraint systems, and an encoding into Boolean logic is defined. This paves the way for applying existing SMT-solvers for checking the Bounded Reachability problem, as shown by various experimental results.

1 Introduction

Many extensions of temporal logic or automata have been proposed with the goal of verifying infinite-state systems. Among the many extensions of Propositional Linear Temporal Logic (PLTL), there have been proposals of allowing formulae which may include arithmetic constraints belonging to a specific constraint system [6, 10] . These logics are well-suited to define properties of infinite-state systems, but, unfortunately for the aim of automatic verification, previous results have also shown the undecidability of the satisfiability problem, at least in the general case [12]. Here we define a more general logic, called CLTLB($\mathcal{D}$), which is an extension of PLTLB (PLTL with Both future and past operators), allowing arithmetic constraints belonging to a generic constraint system. To cope with undecidability, already known for the less general case above, we introduce suitable assumptions concerning the structure of models, but without any syntactic restriction on formulae. Models only consider partial valuations of arithmetic vari-

ables: the satisfiability of CLTLB($\mathcal{D}$) then turns to be decidable, provided that the constraint system $\mathcal{D}$ has a decidable decision procedure. We then define the Bounded Reachability Problem (BRP) for CLTLB($\mathcal{D}$), which can be decided by showing its equivalence to the satisfiability of CLTLB($\mathcal{D}$) over partial valuations. We realized a Bounded Reachability Checker by using SMT-solvers natively implementing decision procedures for Quantifier-Free Integer Difference Logic with Uninterpreted Functions (QF-UFIDL) and Quantifier-Free Linear Integer Arithmetic with Uninterpreted Functions (QF-UFLIA). Experimental results using the Zot toolkit [18, 20] show that, the greater expressiveness of CLTLB($\mathcal{D}$) notwithstanding, the encoding of the propositional part is considerably faster and with smaller memory footprint than existing encodings of PLTL based on SAT.

The paper is structured as follows. Section 2 relates on the state of the art in extending PLTL with constraint systems. Section 3 introduces CLTLB(D), while Section 4 presents various decidability and undecidability results. Section 5 introduces and solves the BRP. Section 6 defines an encoding of CLTLB($\mathcal{D}$) into logics suitable for SMT-based verification. Section 7 relates on the performance of experimental results of the original SAT-based plugins of Zot with the SMT-based ones on a number of examples taken from different application domains. Finally, Section 8 draws a few conclusions and outlines future research.

2 State of the art

Among the various proposals of extension of LTL, CLTL (Counter LTL) has been defined in [6]. CLTL is, essentially, Propositional LTL with future operators (PLTL), with in addition terms that are arithmetic constraints in Integer Difference Logic (DL). However, by reducing the recurrence problem for Minsky machines to the satisfiability of a CLTL

formula, it is shown that the logic is undecidable, hence unsuitable for automatic verification.

A generalization of CLTL is CLTL($\mathcal{D}$) [10], where the arithmetic constraints belong to a general constraint system $\mathcal{D}$. If $\mathcal{D}$ has an intrinsic counting mechanism, i.e., it contains equality and a binary injective relation R such that its graph is a DAG, then CLTL($\mathcal{D}$) is undecidable. Indeed, a relation satisfying the hypothesis of the theorem generalizes the "successor" function and can be used to define constraints of the form $y = z+1$. [12] proves the undecidability of the satisfiability problem of $\text{CLTL}^l_m(\text{DL})$, which is the restriction of CLTL(DL) to formulae with at most m variables and of depth less or equal to l. $\text{CLTL}^l_m(\text{DL})$ is shown to be Σ^1_1-hard for $m > 1$ and $l > 1$, while $\text{CLTL}^1_1(\text{DL})$ is shown to be PSPACE-complete.

For practical model-checking, a large variety of infinite-state systems can be effectively represented by counters systems. In this case, interesting results on verifying safety and reachability properties can be obtained by constraining the control graph of the counters system to be flat [7, 5], i.e., no control state occours in more than one simple cycle. Properties are defined by means of Presburger arithmetic constraints but they are not considered in the framework of any temporal logic, for instance, like CLTL or CLTL($\mathcal{D}$) described above. In [11], authors extend some results about flat systems to more general classes of infinite-state systems in which some first-order extensions of CTL* have decidable model-checking.

To cope with undecidability, [8] describes a reduction of infinite BMC to a satisfiability problem of Boolean constraints formulae. By translating LTL formulae into a corresponding Büchi automaton, a BMC problem is reduced to the satisfiability of a mixed arithmetic-Boolean formula. The authors also give a proof of soundness and completeness for the $\mathbf{U}$-free fragment of the logic. In this case, the BMC problem is solved by means of a loop-free encoding, since $\mathbf{U}$-free formulae can always be translated into an automaton over *finite* words accepting a prefix of all infinite paths which satisfy it. In all other cases, generic LTL formulae are translated into a corresponding Büchi automaton with acceptance conditions involving an implicit periodicity constraint over counters. However, this translation does not work when counters do not behave periodically. For instance, consider a transition system defining a non-periodic, strictly-increasing counter x starting at 0. Property $\top\mathbf{U}(x < 0)$ does not hold for this system, but the Büchi automaton corresponding to its negation imposes a periodic constraint over the sequence of values of x, which cannot be satisfied. Hence, using the translation outlined above, verification of formula $\top\mathbf{U}(x < 0)$ for the strictly-increasing counter improperly yields true.

We define a complementary, purely descriptive, approach which solves this problem. It is also aimed at solving reachability problems for infinite-state systems whose propositional, possibly periodic, behaviors induce a finite prefix of values of variables and satisfying a CLTLB($\mathcal{D}$) specification, instead of LTL properties just over arithmetic constraints.

3 A Temporal Logic over Constraint Systems

This section presents an extension to Kamp's [16] PLTLB, by allowing formulae over a constraint system. As suggested in [6], and unlike the approach of [9], the propositional variables of this logic are Boolean terms or atomic arithmetic constraints.

Let V be a set of variables; a *constraint system* is a pair $\mathcal{D} = \langle D, \Pi \rangle$ where D is a specific domain of interpretation for variables and constants and Π is a family of relations on elements of D. An *atomic $\mathcal{D}$-constraint* is a term of the form $R^n(x_1, \ldots, x_n)$, where R^n is an n-ary relation on 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 constraint is *satisfied* by a D-valuation v, written $v \models R(x_1, \ldots, x_n)$, if $(v(x_1), \ldots, v(x_n)) \in R$.

Let AP be a set of atomic propositions and $\mathcal{D} = \langle D, \Pi \rangle$ a constraint system. CLTLB($\mathcal{D}$) is defined as an extension of PLTLB, by combining Boolean atoms with arithmetic temporal terms defined in $\mathcal{D}$. The resulting logic is actually equivalent to the quantifier-free fragment of FOLTL [13] over signature $\{\Pi, AP\}$. The syntax of CLTLB($\mathcal{D}$) is defined as follows:

$$\phi := \begin{cases} p \mid R(\varphi_1, \ldots, \varphi_n) \mid \phi \wedge \phi \mid \neg\phi \mid \\ \quad \mathbf{X}\phi \mid \mathbf{Y}\phi \mid \phi\mathbf{U}\phi \mid \phi\mathbf{S}\phi \\ \varphi := x \mid \mathbf{X}\varphi \mid \mathbf{Y}\varphi \end{cases}$$

where $p \in AP$, $x \in V$, $\mathbf{X}$ and $\mathbf{Y}$ are the usual "next" and "previous" operators, $\mathbf{U}$ and $\mathbf{S}$ are the usual "until" and "since" operators, $R \in \Pi$, $\mathbf{X}^j$ and $\mathbf{Y}^j$ are shorthands for j applications of $\mathbf{X}$ and $\mathbf{Y}$ (e.g., $\mathbf{X}^2 \equiv \mathbf{XX}$). Each formula φ is called an *arithmetic temporal term* (a.t.t.). Its *depth* $|\varphi|$ is the total amount of temporal shift needed in evaluating φ:

$$|x| = 0,$$
$$|X(\varphi)| = |\varphi| + 1,$$
$$|Y(\varphi)| = |\varphi| - 1.$$

Let ϕ be a CLTLB($\mathcal{D}$) formula, x a variable and Γ_x 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:

$$\lceil\phi\rceil_x = \max_{\varphi_i \in \Gamma_x}\{0, |\varphi_i|\}$$
$$\lfloor\phi\rfloor_x = \min_{\varphi_i \in \Gamma_x}\{0, |\varphi_i|\}$$

The above definitions may naturally be extended to the set V of all variables (by letting $\lceil \phi \rceil = \max_{x \in V}\{\lceil \phi \rceil_x\}$, $\lfloor \phi \rfloor_x = \min_{x \in V}\{\lfloor \phi \rfloor_x\}$). 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.

The semantics of a formula ϕ of CLTLB($\mathcal{D}$) is defined w.r.t. a linear time structure $\pi_\sigma = (S, s_0, I, \pi, \sigma, L)$, where S is a set of states, s_0 is the initial state, $I : \{j \mid \lfloor \phi \rfloor \leq j \leq -1\} \times V \to D$ is an assignment, $\pi \in s_0 S^\omega$ is an *infinite path*, $\sigma : \mathbb{N} \times V \to D$ is a sequence of $\mathcal{D}$-valuations and $L : S \to 2^{AP}$ is a labeling function. From now on, the set of all sequences of $\mathcal{D}$-valuations is denoted by Σ. Function I defines the valuation of variables for each time instant in $\{j \mid \lfloor \phi \rfloor \leq j \leq -1\}$, i.e., for time instants before 0; this way σ can be extended to a.t.t.'s. Indeed, if φ is an a.t.t., x is the variable in φ, $i \in \mathbb{N}$ and $\sigma^i(x)$ is a shorthand for $\sigma(i, x)$, then:

$$\sigma^i(\varphi) = \begin{cases} \sigma^{i+|\varphi|}(x), & \text{if } i + |\varphi| \geq 0; \\ I(i + |\varphi|, x), & \text{if } i + |\varphi| < 0. \end{cases}$$

The semantics of a CLTLB($\mathcal{D}$) formula ϕ at instant $i \in \mathbb{N}$ 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. φ:

$$\pi_\sigma^i \models p \Leftrightarrow p \in L(s_i) \text{ for } p \in AP$$
$$\pi_\sigma^i \models R(\varphi_1, \ldots, \varphi_n) \Leftrightarrow$$
$$(\sigma^{i+|\varphi_1|}(x_{\varphi_1}), \ldots, \sigma^{i+|\varphi_n|}(x_{\varphi_n})) \in R$$
$$\pi_\sigma^i \models \neg p \Leftrightarrow \pi_\sigma^i \not\models p$$
$$\pi_\sigma^i \models \phi \wedge \psi \Leftrightarrow \pi_\sigma^i \models \phi \text{ and } \pi_\sigma^i \models \psi$$
$$\pi_\sigma^i \models \mathbf{X}\phi \Leftrightarrow \pi_\sigma^{i+1} \models \phi$$
$$\pi_\sigma^i \models \mathbf{Y}\phi \Leftrightarrow \pi_\sigma^{i-1} \models \phi \wedge i > 0$$
$$\pi_\sigma^i \models \phi \mathbf{U} \psi \Leftrightarrow \begin{cases} \exists j \geq i : \pi_\sigma^j \models \psi \wedge \\ \pi_\sigma^n \models \phi \ \forall i \leq n < j \end{cases}$$
$$\pi_\sigma^i \models \phi \mathbf{S} \psi \Leftrightarrow \begin{cases} \exists 0 \leq j \leq i : \pi_\sigma^j \models \psi \wedge \\ \pi_\sigma^n \models \phi \ \forall j < n \leq i \end{cases}$$

where x_{φ_i} is the variable that appears in φ_i. The semantics of ϕ is well defined, as any valuation σ^i is defined for all $i \geq \lfloor \phi \rfloor$, because of assignment I. A formula $\phi \in$ CLTLB($\mathcal{D}$) is *satisfiable* if there exists a linear time structure $\pi_\sigma = (S, s_0, I, \pi, \sigma, L)$ such that $\pi_\sigma^0 \models \phi$ (in which case π_σ is a *model* of ϕ). Without loss of generality, one may assume that all formulae are in *positive normal form*, where negation may only occur in front of atomic constraints. In fact, by introducing as primitive the connective $\vee$, the dual operators "release" $\mathbf{R}$, "trigger" $\mathbf{T}$ and "previous" $\mathbf{Z}$ 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 $\mathbf{Z}\phi \equiv \neg \mathbf{Y} \neg \phi$, and by applying De

Morgan's rules, every CLTLB formula can be rewritten into positive normal form.

4 (Un)decidability of CLTLB($\mathcal{D}$)

As a first result, by exploiting well-know properties of PLTLB, we prove the equivalence of CLTLB($\mathcal{D}$) to CLTL($\mathcal{D}$) for a quantifier-free constraint system $\mathcal{D}$, w.r.t. *initial* equivalence. Then, as a corollary of results described in Section 2, we obtain the undecidability of CLTLB($\mathcal{D}$) for a large class of constraint systems.

In the following, as customary, we denote with π a structure for a PLTLB formula.

Definition 1. *Two PLTLB formulae ϕ, ψ are* globally *equivalent, written $\phi \equiv_g \psi$, if for all linear-time structures π it is $\pi^i \models \phi \Leftrightarrow \pi^i \models \psi$ for all $i \in \mathbb{N}$. Two PLTLB formulae ϕ, ψ are* initially *equivalent, written $\phi \equiv_i \psi$, when $\pi^0 \models \phi \Leftrightarrow \pi^0 \models \psi$ for all linear-time structures π.*

In [15] it is shown that any PLTLB formula is initially equivalent to a PLTL formula, while the two logics are not globally equivalent (see [22] for details). In order to extend this result to the constrained case, we need to introduce new temporal operators. CLTLB($\mathcal{D}$), as defined in Section 3, includes the "non-strict" until (resp. since) operator, in which formula $\phi \mathbf{U} \psi$ (resp. $\phi \mathbf{S} \psi$) holds in an instant i when ψ holds in i, and only if ϕ holds starting from i. The "strict" version of until $\mathbf{U}^>$, instead, does not require this:

$$\pi_\sigma^i \models \phi \mathbf{U}^> \psi \Leftrightarrow \begin{cases} \exists j > i : \pi_\sigma^j \models \psi \wedge \\ \pi_\sigma^n \models \phi \ \forall i < n < j \end{cases}$$

and similarly for the strict since $\mathbf{S}^>$. It is well known that the following global equivalences hold for any ϕ, ψ:

$$\mathbf{X}\phi \equiv_g \perp \mathbf{U}^> \phi, \quad \phi \mathbf{U} \psi \equiv_g \psi \vee (\phi \wedge \phi \mathbf{U}^> \psi);$$
$$\mathbf{Y}\phi \equiv_g \perp \mathbf{S}^> \phi, \quad \phi \mathbf{S} \psi \equiv_g \psi \vee (\phi \wedge \phi \mathbf{S}^> \psi).$$

Using the previous equivalences, Gabbay [14] proved that any PLTLB formula is globally equivalent to a separated PLTLB formula, i.e. a Boolean combination of formulae containing either $\mathbf{U}^>$ ($\mathbf{U}^>$-formulae) or $\mathbf{S}^>$ ($\mathbf{S}^>$-formulae), but not both. Since this theorem preserves all semantic properties, i.e., it is actually a rewriting syntactic procedure over formulae, it extends also to the case of CLTLB($\mathcal{D}$), provided that each arithmetic constraint is accounted as a propositional letter. In particular, a.t.t.'s $\mathbf{X}x/\mathbf{Y}x$ are not rewritten using strict-until/-since operators, but are considered as is, since their semantics depends on the underlying sequence σ as defined before. Then, we need to show that $\mathbf{S}^>$-formulae can be translated into *initially* equivalent $\mathbf{U}^>$-formulae. More precisely, we prove the following:

Theorem 2. *Any CLTLB(D) formula is initially equivalent to a CLTL(D) formula, while the two logics are not globally equivalent.*

Proof sketch. We first prove that CLTL(D) is not globally equivalent to CLTLB(D) by providing a counterexample. Formula $\top \mathbf{S} A$, where $A \in AP$, was shown in [13] to have no globally equivalent PLTL formula. Now, suppose ϕ is a CLTL(D) formula globally equivalent to CLTLB(D) formula $\top \mathbf{S} A$. Then, for the above reason, it should constrain at least one of its arithmetic variables, by a non trivial arithmetic formula. Since $\top \mathbf{S} A$ does not constrain any arithmetic variables, some of its models cannot be models of ϕ.

To prove the initial equivalence we suppose each formula is written using only $\mathbf{U}^{>}$ and $\mathbf{S}^{>}$ operators, using the equivalences above. From Gabbay's Separation Theorem such a formula can be rewritten to a separated CLTLB(D) formula which is a Boolean combination of $\mathbf{S}^{>}$- and $\mathbf{U}^{>}$-formulae. The proof is concluded by noticing that any $\mathbf{S}^{>}$-formula is trivially initially equivalent to false. □

Corollary 3. *Let $\mathcal{D} = \langle D, \Pi \rangle$ be a constraint system where Π contains equality and a binary relation R such that (D, R) is a DAG; then, satisfiability of CLTLB(D) is undecidable.*

In the following, in the case of a decidable constraint system $\mathcal{D}$, we prove the decidability of the satisfiability and the model checking problems for CLTLB(D) formulae for partial $\mathcal{D}$-valuations, in which that for all computations the value of counters will be considered only for a fixed number of steps. The counting mechanism of $\mathcal{D}$ is not altered along finite paths by means of constraints imposing periodicity of values of variables and all relations are still considered over infinite, possibly periodic, paths. This allows us to define a complementary approach to the one of [8], aimed at bounded satisfiability checking [19] and BMC of infinite-state systems. With this assumption, any periodic behavior which induces a finite, even periodic, prefix of values of variables ruled by the counting mechanism and satisfying a CLTLB(D) formula, can be represented. An arithmetic variable varying over a bounded set may still be represented by its Boolean representation and be part of the propositional infinite paths. It is worth noticing that, since we limit the counting mechanism along finite paths, the partial model is an under-approximation, due to the intrinsic undecidability of the general problem.

Definition 4. *Let ϕ be a CLTLB(D) formula and $k \in \mathbb{N}$, then a k-partial $\mathcal{D}$-valuation σ_k for ϕ is a relation in $\{i \in \mathbb{Z} \mid i \geq \lfloor \phi \rfloor\} \times V \times D$ with the condition that for each variable x occurring in ϕ, its restriction over $\{i \in \mathbb{Z} \mid \lfloor \phi \rfloor_x \leq i \leq k + \lceil \phi \rceil_x\} \times \{x\} \times D$ is a function from $\{i \in \mathbb{Z} \mid \lfloor \phi \rfloor_x \leq i \leq k + \lceil \phi \rceil_x\} \times \{x\}$ to D. Then, Σ_k is the set of all k-partial $\mathcal{D}$-valuations for ϕ.*

Informally, σ_k defines a unique value for each counter x from 0 up to the bound k by means of boundaries conditions in the intervals $\{i \in \mathbb{Z} \mid \lfloor \phi \rfloor_x \leq i < 0\}$ and $\{i \in \mathbb{Z} \mid k < i \leq k + \lceil \phi \rceil_x\}$, and it accounts for relations over infinite, even periodic, paths, after k. For the case of k-partial $\mathcal{D}$-valuation one can define a semantics of CLTLB(D) formulae. It coincides with the semantics of the (full) $\mathcal{D}$-valuations except for the case of arithmetic relations R; namely:

$$\pi_{\sigma_k}^i \models R(\varphi_1, \ldots, \varphi_n) \Leftrightarrow$$
$$\forall y_1, \ldots, y_n \in D \text{ s.t. } \forall 1 \leq j \leq n, (i + |\varphi_j|, x_{\varphi_j}, y_j) \in \sigma_k$$
$$\text{then } (y_1, \ldots, y_n) \in R, \tag{1}$$

where x_{φ_j} is the variable that appears in φ_j. If σ_k is a function, this semantics reduces exactly to the previous one. The satisfiability problem for a CLTLB(D) formula ϕ over k-*partial* $\mathcal{D}$-valuations is that of looking for a (partial) linear time structure $\pi_{\sigma_k} = (S, s_0, \pi, \sigma_k, L)$ such that $\pi_{\sigma_k}^0 \models \phi$. It is worth noticing that the initialization function I is implicit in the definition of σ_k.

Theorem 5. *The satisfiability of a CLTLB(D) formula ϕ over k-partial $\mathcal{D}$-valuations is decidable when $\mathcal{D}$ is decidable.*

Proof sketch. Thanks to the initial equivalence of CLTLB(D) and CLTLF(D) formulae (Theorem 2), we assume without loss of generality that $\phi \in$ CLTLF(D); moreover, we assume that a.t.t.'s do not appear negated (i.e., negated a.t.t.'s are transformed into the positive form of the complement relation) and that constraints in ϕ are in disjunctive normal form (i.e., disjunction of conjunction of propositions and a.t.t.'s). Let $\mathcal{C}$ be the set containing all conjunctions of such terms, and let $\mathcal{A}_\phi$ be the corresponding Büchi automaton whose alphabet is $A = \mathcal{P}(\mathcal{C})$. The satisfiability of ϕ is reduced to the emptiness of $L(\mathcal{A}_\phi)$. In fact, if $L(\mathcal{A}_\phi)$ is empty, then ϕ is unsatisfiable. If $L(\mathcal{A}_\phi)$ is not empty, then $\mathcal{A}_\phi$ has one or more strongly connected components that are reachable from an initial state and contain a final state. Hence, it is enough to check if there exists a path of length k from the initial state (which also considers the initial values of the variables) that can be extended to one of the above components and which satisfies each constraint. This is decidable, because the consistency problem of $\mathcal{D}$ is decidable. Finally, it can be shown that the finite sequence of variable assignments appearing in such a path of length k can be extended to a k-partial $\mathcal{D}$-valuation on which ϕ is satisfied, for example by using the empty relation outside those instants in which the valuation is required to be a function. □

Section 6 computes an estimation of the complexity of problem for a large class of constraint system.
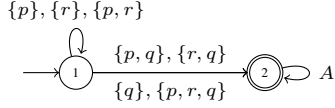
Figure 1. Büchi automaton for $(p \vee r)\mathbf{U}q$**, with** $p := x = \mathbf{Y}y + 1$**,** $r := y = x + 2$ **and** $q := y \leq \mathbf{X}^2 x \wedge x < \mathbf{X}x$**.**
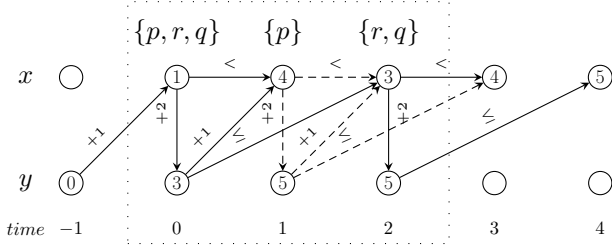
Figure 2. Constraint graph of $\{p, r, q\}\{p\}\{r, q\}$**.**

As an illustrative example, consider the satisfiability of the formula $\phi := (p \vee r)\mathbf{U}q$ where $p := x = \mathbf{Y}y + 1$, $r := y = x + 2$ and $q := y \leq \mathbf{X}^2 x \wedge x < \mathbf{X}x$ and let be $k = 3$. The emptiness problem reduces to finding a consistent assignment to x and y along a path of length 3 over the Büchi automaton $\mathcal{A}_\phi$ on the alphabet $A = \{\varnothing, \{p\}, \{q\}, \{r\}, \{p, r\}, \{p, q\}, \{r, q\}, \{p, q, r\}\}$ shown in Fig. 1. Actually, we need to check the consistency for at least one prefix of length 3 of $L(\mathcal{A}_\phi)$. In Fig. 2 we show the corresponding graph of the constraints to be solved for the word $\{p, r, q\}\{p\}\{p, q\}$. A dashed line means that the constraint in the label does not hold, numbers in the circles are possible assignments to the variables, while a blank means that the corresponding value is irrelevant, and can be left undefined.

So far, we neglected any initialization condition, solving a *general* satisfiability problem. If a formula is shown to be unsatisfiable, then there is no prefix of an infinite model π_σ, of length equal to k, satisfying the formula.

5 Bounded Reachability Problem

This section studies the bounded satisfiability of CLTLB($\mathcal{D}$) formulae by using a finite representation of infinite models. It is then shown that this entails the satisfiability of the same formula with respect to *k-partial $\mathcal{D}$-valuations*. Finally, the section introduces the Bounded (existential) Reachability Problem (BRP) for Kripke structures, showing that BRP also admits a complete procedure.

First, we need to define a bounded semantics, i.e., a semantics of a formula on finite structures. Let $k > 0$, let ϕ

be a CLTLB($\mathcal{D}$) formula and let $\widehat{\sigma}_k : \{i \in \mathbb{Z} \mid \lfloor \phi \rfloor_x \leq i \leq k + \lceil \phi \rceil_x\} \times \{x\} \to \mathcal{D}$, for each $x \in V$, called a *local sequence*, be a finite sequence of assignments to variables in V. Informally, sequence $\widehat{\sigma}_k$ is not only defined between instants 0 and k, but it is bordered by two segments defining variable values before 0 and after k, as shown also in Fig. 2. This is necessary to correctly define the value of all a.t.t's in the interval from 0 to k; in fact, the evaluation of an a.t.t. may involve also a bounded number of instants before instant 0 or after instant k. Let $\pi \in S^+$, called a finite path. A finite path is *cyclic* if it is of the form $usvs$, for some $s \in S$, $u, v \in S^*$. A cyclic finite path can be considered a finite representation of an infinite one, e.g., $u(sv)^\omega$. If π is a cyclic path $usvs$, then a bounded semantics for ϕ over π and local assignment $\widehat{\sigma}_k$ is defined as in the case of a k-partial $\mathcal{D}$-valuation of Section 4, by replacing σ_k with $\widehat{\sigma}_k$ and π with $u(sv)^\omega$ in (1). If π is not cyclic, instead, the semantics of each relation R is, for $0 \leq i \leq k$:

$$
\begin{aligned}
\pi^i_{\widehat{\sigma}_k} \models_k R(\varphi_1, \ldots, \varphi_n) \Leftrightarrow \\
(\widehat{\sigma}_k^{i+|\varphi_1|}(x_{\varphi_1}), \ldots, \widehat{\sigma}_k^{i+|\varphi_n|}(x_{\varphi_n})) \in R
\end{aligned}
$$

The bounded semantics of temporal operators is the same as the one in [3], e.g.:

$$
\pi^i_{\widehat{\sigma}_k} \models_k \phi\mathbf{U}\psi \Leftrightarrow \begin{cases} \exists i \leq j \leq k : \pi^j_{\widehat{\sigma}_k} \models \psi \wedge \\ \pi^n_{\widehat{\sigma}_k} \models \phi \quad \forall i \leq n < j \end{cases}
$$

$$
\pi^i_{\widehat{\sigma}_k} \models_k \phi\mathbf{R}\psi \Leftrightarrow \begin{cases} \exists i \leq j \leq k : \pi^j_{\widehat{\sigma}_k} \models_k \phi \wedge \\ \pi^n_{\widehat{\sigma}_k} \models_k \psi \quad \forall i \leq n \leq j \end{cases}
$$

$$
\pi^i_{\widehat{\sigma}_k} \models_k \mathbf{X}\phi \Leftrightarrow 0 \leq i + 1 \leq k \wedge \pi^{i+1}_{\widehat{\sigma}_k} \models \phi
$$

By using the bounded semantics, the following theorem holds:

Theorem 6. *For every* CLTLB(D) *formula ϕ, if, there exist $k > 0$, a finite path π of length k and a local assignment $\widehat{\sigma}_k$ such that $\pi_{\widehat{\sigma}_k} \models_k \phi$ then ϕ is satisfiable over k-partial $\mathcal{D}$-valuations.*

Proof sketch. The statement is proven by means of a completion of the sequence $\widehat{\sigma}_k$ satisfying property (1). A legal completion may also involve undefined values: constraints encompassed in the loop of $\pi_{\widehat{\sigma}_k}$ can be suitably bordered. In particular, if $\pi = uv^\omega$ and l is the length of v, for each variable x such that $\lceil \phi \rceil_x > 0$, $\forall c \in D$, then $\forall h \geq 0$, $(k + 1 + hl, x, c) \notin \widehat{\sigma}_k$. By exploiting the results in [3] and a syntactic rewriting of each $\mathcal{D}$ constraint with a propositional letter, which results in a formula ϕ', from ϕ, satisfied by a propositional model π', then $\pi' \models_k \phi'$ implies $\pi' \models \phi'$. $\square$

The above concepts can be generalized and extended in the case of $\mathcal{D}$-Kripke structures, as suggested in [10].

Definition 7. *A $\mathcal{D}$-Kripke structure is a tuple $M = \langle S, T, C, \lambda \rangle$ with a finite set of states S, a transition relation $T \subseteq S \times S$ between states, a set C of $\mathcal{D}$ relations on a.t.t.'s and a labeling function $\lambda : S \rightarrow 2^{AP} \times C$.*

Given a $\mathcal{D}$-Kripke structure M, a CLTLB($\mathcal{D}$) formula ϕ and an initial state s_0, the *existential* model checking (MC) problem amounts to checking if there exists a linear structure π_σ such that $\pi_\sigma \models \phi$. Because of the undecidability results of Section 2, the existential MC problem must be redefined for k-*partial* $\mathcal{D}$-valuations in order to have a decidable under-approximation. Thanks to the well-known representation of Kripke structures through LTL formulae, and by considering a.t.t.'s in C as atomic elements, it is possible to obtain a CLTLB($\mathcal{D}$) formula χ_M defining the "propositional" description of the language of $\mathcal{D}$-Kripke structure M. The k-*partial* $\mathcal{D}$-*evaluation* model checking problem is defined as the satisfiability of $\chi_M \wedge \phi$ over k-partial $\mathcal{D}$-evaluations.

Theorem 6 may be strengthened for $\mathcal{D}$-Kripke structures when ϕ is a reachability formula. Formula ϕ is a *reachability* formula when it is of the form $\mathbf{F}\psi$, where ψ is a CLTLB($\mathcal{D}$) formula without temporal operators (which are allowed only in a.t.t.). Then, the ***Bounded Reachability Problem*** (BRP) for M and ϕ is defined as the existence of $k > 0$, a finite path π of length k and a local assignment $\widehat{\sigma}_k$ such that $\pi_{\widehat{\sigma}_k} \models_k \chi_M \wedge \phi$.

Corollary 8. *For every reachability formula ϕ in CLTLB($\mathcal{D}$) and for every $\mathcal{D}$-Kripke structure M, the BRP is equivalent to the k-partial $\mathcal{D}$-evaluation MC problem.*

6 Encoding of the Bounded Reachability Problem

In this section the BRP is encoded as the satisfiability of a quantifier-free formula in the theory EUF $\cup \mathcal{D}$ (QF-UF$\mathcal{D}$), where EUF is the theory of Equality and Uninterpreted Functions, provided that the set D includes a copy of $\mathbb{N}$ and that EUF $\cup \mathcal{D}$ is consistent. The last condition is easily verified in the case of a union of two consistent, disjoint, stably infinite theories (as is the case for EUF and arithmetic). In [1] a similar encoding is described for the case of Integer Difference Logic (DL) constraints: in that case it results to be more succinct and expressive than the Boolean one: lengthy propositional constraints are substituted by more concise DL constraints and arithmetic (infinite) domains do not require an explicit finite representation. These facts, considering also that the satisfiability problem for the quantifier-free fragment of EUF $\cup$ DL (QF-UFIDL) has the same complexity of SAT, make this approach particularly efficient, as demonstrated by the tests outlined in Section 7.

Under the above assumption, the proposed encoding is an effective proof of the decidability of the BRP over k-

partial $\mathcal{D}$-valuations. In the general case an estimation of the complexity of the satisfiability problem (for quantifier-free formulae) can be performed via the Nelson-Oppen Theorem [17] as shown in Corollary 9.

As discussed before, the BMC problem amounts to looking for a finite representation of infinite (possibly periodic) paths. The Boolean approach [3] encodes finite paths by means of $2k+3$ propositional variables, while the same temporal behavior can be defined by means of *one* QF-UF$\mathcal{D}$ formula involving only *one loop-selecting* variable $\boldsymbol{loop} \in D$:

$$\bigwedge_{i=1}^{k} \left((\boldsymbol{loop} = i) \Rightarrow L(s_{i-1}) = L(s_k) \right).$$

If the value i of variable $\boldsymbol{loop}$ is between 1 and k, then there exists a loop, and it starts at i; notice that the formula $\boldsymbol{loop} = i$ is well defined since D contains a copy of $\mathbb{N}$.

To encode a.t.t.'s, an *arithmetic formula function*, i.e., an uninterpreted function $\boldsymbol{\tau} : D \rightarrow D$, is associated with each arithmetic temporal subterm of Φ. Let τ be such a subterm, then the arithmetic formula function associated with it (denoted by the same name but written in boldface), is recursively defined w.r.t. the sequence of valuations σ as:

τ	$0 \le i \le k$
x	$\boldsymbol{x}(i) = \sigma^i(x)$
$\mathbf{X}\alpha$	$\boldsymbol{\tau}(i) = \boldsymbol{\alpha}(i+1)$
$\mathbf{Y}\alpha$	$\boldsymbol{\tau}(i) = \boldsymbol{\alpha}(i-1)$

If D includes a copy of $\mathbb{Z}$, this semantics is well-defined between 0 and k thanks to the initialization function I, otherwise we need to consider a *shifted* function $\overline{\sigma}$ such that $\overline{\sigma}(i, \cdot) = \sigma(i - \lfloor \phi \rfloor, \cdot)$.

The propositional encoding is based on the one presented in [4], which is modified to take also into account relations over a.t.t.'s. In the case of the Boolean encoding, the truth value of a PLTLB formula Φ is defined w.r.t. the truth value of its subformulae. For each subformula t, a set of Boolean variables $\{t_i\}_{0 \le i \le k+1}$ is associated with it: if t_i holds, then subformula t holds at instant i. Instant $k+1$ is introduced to more easily represent the instant in which the periodic behavior starts. The truth value of a CLTLB($\mathcal{D}$) formula Φ is defined in a similar way. The QF-UF$\mathcal{D}$ encoding, however, associates with each subformula θ a *formula predicate* that is a unary uninterpreted predicate (denoted by the same name but written in boldface) $\boldsymbol{\theta} \in \mathcal{P}(D)$. When the subformula θ holds at instant i then $\boldsymbol{\theta}(i)$ holds. As the length of paths is fixed to $k+1$ and all paths start from 0, formula predicates are actually subsets of $\{0, \ldots, k+1\}$. Let θ be a subformula of Φ, $\alpha_1, \ldots \alpha_n$ be a.t.t.'s and R be an n-ary

relation in $\mathcal{D}$; formula predicate $\boldsymbol{\theta}$ is recursively defined as:

θ	$0 \leq i \leq k+1$
p	$\boldsymbol{\theta}(i) \Leftrightarrow p \in L(s_i)$
$R(\alpha_1, \ldots, \alpha_n)$	$\boldsymbol{\theta}(i) \Leftrightarrow R(\boldsymbol{\alpha_1}(i), \ldots, \boldsymbol{\alpha_n}(i))$
$\neg \phi$	$\boldsymbol{\theta}(i) \Leftrightarrow \neg\boldsymbol{\phi}(i)$
$\phi \wedge \psi$	$\boldsymbol{\theta}(i) \Leftrightarrow \boldsymbol{\phi}(i) \wedge \boldsymbol{\psi}(i)$

Temporal subformulae constraints define the basic temporal behavior of future and past operators, by using their traditional fixpoint characterizations:

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

The encoding for the past operators is analogous to that for future operators except for the instant 0, which must be treated separately (see [1]).

Last state constraints define an equivalence between truth at point $k+1$ and that at the point indicated by the *loop* variable, since the instant $k+1$ is representative of the instant *loop* along periodic paths. Otherwise, for non-periodic paths, truth values in $k+1$ are trivially false. These constraints have a similar structure to the corresponding Boolean ones, but here they are defined by only *one* constraint, for each subformula θ of Φ, w.r.t. the variable *loop*:

$$\left(\bigwedge_{i=1}^{k} (\boldsymbol{loop} = i) \Rightarrow (\boldsymbol{\theta}(k+1) \Leftrightarrow \boldsymbol{\theta}(i))) \right) \wedge$$
$$\left(\left(\bigwedge_{i=1}^{k} \neg(\boldsymbol{loop} = i) \right) \Rightarrow (\neg\boldsymbol{\theta}(k+1)) \right).$$

Note that if a loop does not exist then the fixpoint semantics of $\mathbf{R}$ is exactly the one defined over finite acyclic paths in Section 5. To correctly define the semantics of $\mathbf{U}$ and $\mathbf{R}$, their *eventualities* have to be accounted for. Briefly, if $\phi\mathbf{U}\psi$ holds at i, then ψ eventually holds in some $j \geq i$; if $\phi\mathbf{R}\psi$ does not hold at i, then ψ eventually does not hold in some $j \geq i$. Along finite paths of length k, eventualities must hold between 0 and k. If a loop exists, an eventuality may hold within the loop. The original Boolean encoding introduces k propositional variables for each subformula θ of Φ of the form $\phi\mathbf{U}\psi$ or $\phi\mathbf{R}\psi$ (one for each $1 \leq i \leq k$), which represent the eventuality of ψ implicit in the formula, as first defined in [4]. Instead, in the QF-UF$\mathcal{D}$ encoding, only *one* variable $\boldsymbol{j_\psi} \in D$ is introduced for each ψ occurring in a subformula $\phi\mathbf{U}\psi$ or $\phi\mathbf{R}\psi$.

θ	Base
$\phi\mathbf{U}\psi$	$\left(\bigvee_{i=1}^{k} \boldsymbol{loop} = i \right) \Rightarrow$ $(\boldsymbol{\theta}(k) \Rightarrow \boldsymbol{loop} \leq \boldsymbol{j_\psi} \leq k \wedge \boldsymbol{\psi}(\boldsymbol{j_\psi}))$
$\phi\mathbf{R}\psi$	$\left(\bigvee_{i=1}^{k} \boldsymbol{loop} = i \right) \Rightarrow$ $(\neg\boldsymbol{\theta}(k) \Rightarrow \boldsymbol{loop} \leq \boldsymbol{j_\psi} \leq k \wedge \neg\boldsymbol{\psi}(\boldsymbol{j_\psi}))$

The complete encoding of Φ consists of the logical conjunction of all constraints above, together with Φ evaluated at the first instant along the time structure.

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, then the Boolean encoding requires $(2k+3) + (k+2)m + (k+1)n = O(k(m+n))$ fresh propositional variables. The QF-UF$\mathcal{D}$ encoding requires only $n+1$ integer variables (*loop* and $\boldsymbol{j_\psi}$) and m unary predicates (one for each subformula).

As previously anticipated, if $\mathcal{D}$ is a consistent, stably infinite theory, ϕ is a formula of length n and $T(n)$ is the complexity of the satisfiability problem in $\mathcal{D}$ then, by the Nelson-Oppen Theorem, the satisfiability of a CLTLB($\mathcal{D}$) formula ϕ over k-partial $\mathcal{D}$-valuations can be solved in $O(2^{n^2}(nk \log(nk) + T(nk)))$; moreover, if $\mathcal{D}$ is convex it can be solved in $O(n^3(nk \log(nk) + T(nk)))$.

Corollary 9. *The satisfiability of a CLTLB($\mathcal{D}$) formula over k-partial $\mathcal{D}$-valuations is* NP-*complete when $\mathcal{D}$ is DL,* P *when $\mathcal{D}$ is RDL (Real DL) and* 4-EXPTIME *when $\mathcal{D}$ is LIA (Linear Integer Arithmetic).*

7 Experimental Results

The encoding presented in Section 6 for CLTLB(DL) has been implemented as a plugin of the Zot tool[1]. This implementation exploits SMT solvers as verification engines, and in particular it is based on the SMT-LIB [21] to achieve independence from the particular SMT solver used[2]. The Zot plugin has been used to carry out a number of experiments on a variety of examples, old and new. For the sake of brevity, we do not report here the full experimental data[3], and we only briefly summarize them in an informal way.

We carried out two kinds of experiments. First, we used the new encoding to perform BMC on a set of previously defined PLTLB specifications, to compare the performances of the new Zot plugin w.r.t. the existing SAT-based one presented in [20]. The SMT-based encoding showed considerable improvements in the vast majority of experiments, for both of the SMT solvers used. The recorded speedup (computed as the ratio T_{SAT}/T_{SMT}) was always substantial, and in many cases it was more than tenfold (often considerably more than that). For example, we repeated the experiments of [2] with the new encoding, and the average speedup in the overall verification time was around 2.4 with Z3, and 21.4 with Yices; we point out that the gains in performance were particularly significant for the most complex specifications.

In the second set of experiments we exploited also the new features of CLTLB(DL) w.r.t. PLTLTB, and we used

---

[1]Zot is available at *home.dei.polimi.it/pradella*.

[2]As SMT solvers we used both Yices (*yices.csl.sri.com*) and Z3 (*research.microsoft.com/en-us/um/redmond/projects/z3*).

[3]The data are available at *home.dei.polimi.it/bersani*.

the bounded reachability results presented in Section 5 to analyze some relevant aspects of non-trivial applications based on the Service-Oriented paradigm [1]. On examples that fall in the range of properties expressible through both CLTLB(DL) and PLTLB (e.g., those that involve only bounded domains), the performances of the SMT-based verification are, again, an order of magnitude better than the SAT-based one (the average performance speedup over such properties was 55 with Z3 and 7.4 with Yices).

8 Conclusions and Future Work

In this paper, we introduced the logic CLTLB($\mathcal{D}$), an extension of PLTLB allowing as subformulae arithmetic constraints belonging to a generic constraint system $\mathcal{D}$. We introduced suitable assumptions concerning the structure of models, to make satisfiability of CLTLB($\mathcal{D}$) decidable, provided that $\mathcal{D}$ has, in turn, a decidable decision procedure. In this case, the Bounded Reachability Problem (BRP) for CLTLB($\mathcal{D}$) formulae can be solved by means of automatic software verification tools. We built a Bounded Reachability Checker by using SMT-solvers which natively implement decision procedures for QF-UF$\mathcal{D}$ when $\mathcal{D}$ is DL or LIA, with very encouraging experimental results.

Future work will compare the new arithmetic-based encoding with existing Boolean ones by means of a comprehensive set of tests; we also intend to define new extensions representing infinite behaviors of variables and search for suitable classes of formulae inducing actual ω-periodic models.

Acknowledgments

References

[1] M. M. Bersani, L. Cavallaro, A. Frigeri, M. Pradella, and M. Rossi. SMT-based Verification of LTL Specifications with Integer Constraints and its Applications to Runtime Checking of Service Substitutability. Technical report, arXiv:1004.2873v1, 2010.

[2] M. M. Bersani, C. A. Furia, M. Pradella, and M. Rossi. Integrated modeling and verification of real-time systems through multiple paradigms. In *Proc. of SEFM*, pages 13–22, 2009.

[3] A. Biere, A. Cimatti, E. M. Clarke, and Y. Zhu. Symbolic model checking without BDDs. In *Proc. of TACAS*, pages 193–207, 1999.

[4] A. Biere, K. Heljanko, T. A. Junttila, T. Latvala, and V. Schuppan. Linear Encodings of Bounded LTL Model Checking. *Log. Meth. in Comp. Sci.*, 2(5), 2006.

[5] B. Boigelot. *Symbolic Methods for Exploring Infinite State Spaces*. PhD thesis, Université de Liège, 1998.

[6] H. Comon and V. Cortier. Flatness Is Not a Weakness. In *CSL*, pages 262–276, 2000.

[7] H. Comon and Y. Jurski. Multiple Counters Automata, Safety Analysis and Presburger Arithmetic. In *CAV*, pages 268–279, 1998.

[8] L. M. de Moura, H. Rueß, and M. Sorea. Lazy theorem proving for bounded model checking over infinite domains. In *CADE*, pages 438–455, 2002.

[9] S. Demri. LTL over Integer Periodicity Constraints: (Extended Abstract). In *FoSSaCS*, pages 121–135, 2004.

[10] S. Demri and D. D'Souza. An automata-theoretic approach to constraint LTL. In *FSTTCS*, pages 121–132, 2002.

[11] S. Demri, A. Finkel, V. Goranko, and G. van Drimmelen. Towards a Model-Checker for Counter Systems. In *ATVA*, pages 493–507, 2006.

[12] S. Demri and R. Gascon. The Effects of Bounding Syntactic Resources on Presburger LTL. Technical Report LSV-06-5, LSV, 2006.

[13] E. A. Emerson. Temporal and modal logic. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, pages 995–1072. 1990.

[14] D. M. Gabbay. The declarative past and imperative future: Executable temporal logic for interactive systems. In *Temporal Logic in Specification*, pages 409–448, 1987.

[15] D. M. Gabbay, A. Pnueli, S. Shelah, and J. Stavi. On the temporal basis of fairness. In *POPL*, pages 163–173, 1980.

[16] J. A. W. Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California at Los Angeles, 1968.

[17] D. C. Oppen. Complexity, convexity and combinations of theories. *Theor. Comput. Sci.*, 12:291–302, 1980.

[18] M. Pradella, A. Morzenti, and P. San Pietro. The symmetry of the past and of the future: bi-infinite time in the verification of temporal properties. In *ESEC/SIGSOFT FSE*, pages 312–320, 2007.

[19] M. Pradella, A. Morzenti, and P. San Pietro. Refining real-time system specifications through bounded model- and satisfiability-checking. In *ASE*, pages 119–127, 2008.

[20] M. Pradella, A. Morzenti, and P. San Pietro. A metric encoding for bounded model checking. In A. Cavalcanti and D. Dams, editors, *FM 2009: Formal Methods*, volume 5850 of *LNCS*, pages 741–756. Springer, 2009.

[21] S. Ranise and C. Tinelli. The SMT-LIB standard: Version 1.2. Technical report, 2006. http://combination.cs.uiowa.edu/smtlib/.

[22] P. Schnoebelen. The complexity of temporal logic model checking. In *Adv. in Modal Logic*, pages 393–436, 2002.