S. Demri and R. Gascon

Verification of qualitative Z constraints

Research Report LSV-05-07
June 2005

Laboratoire
Spécification
et
Vérification

CENTRE NATIONAL
DE LA RECHERCHE
SCIENTIFIQUE

Ecole Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

Verification of qualitative $\mathbb{Z}$ constraints

Stéphane Demri and Régis Gascon

LSV/CNRS UMR 8643 & INRIA Futurs projet SECSI & ENS Cachan
61, av. Pdt. Wilson, 94235 Cachan Cedex, France
email: {demri,gascon}@lsv.ens-cachan.fr

Abstract. We introduce an LTL-like logic with atomic formulae built over a constraint language interpreting variables in $\mathbb{Z}$. The constraint language includes periodicity constraints, comparison constraints of the form $x = y$ and $x < y$, it is closed under Boolean operations and it admits a restricted form of existential quantification. This is the largest set of qualitative constraints over $\mathbb{Z}$ known so far, shown to admit a decidable LTL extension. Such constraints are those used for instance in calendar formalisms or in abstractions of counter automata by using congruences modulo some power of two. Indeed, various programming languages perform arithmetic operators modulo some integer. We show that the satisfiability and model-checking problems (with respect to an appropriate class of constraint automata) for this logic are decidable in polynomial space improving significantly known results about its strict fragments. As a by-product, LTL model-checking over integral relational automata is proved complete for polynomial space which contrasts with the known undecidability of its CTL counterpart.

1 Introduction

Model-checking infinite-state systems. The verification of systems with an infinite amount of states has benefited from the numerous decidable model-checking problems for infinite-state systems, including timed automata [AD94], infinite transition graphs [MS85,Cau03], or subclasses of counter systems (see e.g. [CJ98]). Even though decidability can be obtained via numerous proof techniques (finite partition of the infinite domain, well-structured systems, Presburger definable reachability sets, reduction to the second-order theory of the binary tree), showing undecidability of model-checking for some classes of infinite-state systems is often easy. After all, the halting problem for Minsky machines is already undecidable [Min67]. Decidability is more difficult to establish and it can be sometimes regained by naturally restricting the class of models (see e.g. the flatness condition in [CJ98]) or by considering fragments of the specification language (to consider only reachability or repeated reachability for instance). Often, symbolic representations of infinite sets of states are the key argument to get decidability (see e.g. [HMR05]).

Systems with variables interpreted in $\mathbb{Z}$. Structures with a finite set of control states augmented with a finite set of variables interpreted either in $\mathbb{Z}$ or in $\mathbb{N}$

(counters) are operational models of numerous infinite-state systems, including broadcast protocols (see e.g. [EFM99,FL02]). The class of counter machines has numerous undecidable model-checking problems such as the reachability problem but many classes of counter systems have been shown to be decidable:

1. reversal-bounded multicounter machines [Iba78],
2. flat counter systems with affine update functions forming a finite monoid (see e.g. [Boi98,FL02,BFLP03]),
3. flat counter systems [CJ98] (weaker class of Presburger guards but no condition on the monoid),
4. constraint automata with qualitative constraints on $\mathbb{Z}$ [DD03].

Our motivation. Constraint automata with qualitative constraints on $\mathbb{Z}$ are quite attractive operational models since they can be viewed as abstractions of counter automata where incrementations and decrementations are abstracted by operations modulo some power of two. Common programming languages perform arithmetic operators for integer types modulo 2^k [MOS05], typically k is either 32 or 64. For example, $x = y+1$ can be abstracted by $x \equiv_{2^k} y+1 \wedge y < x$. Such an abstraction is well-suited to check safety properties about the original counter system. In the paper, we study a class of constraint automata with a language of qualitative constraints as rich as possible and a companion LTL-like logic to perform model-checking on such operational models. Our framework should be able to deal both with abstractions modulo (see e.g. [CGL94,LS01]) and with integer periodicity constraints used in logical formalisms to deal with calendars [LM01], i.e. constraints of the form $x \equiv_k y + c$. By a qualitative constraint, we mean for instance a constraint that is interpreted as a non-deterministic binary relation, like $x < y$ and $x \equiv_{2^k} y + 5$ (the relationship between x and y is not sharp).

Our contribution. We introduce a version of constraint LTL over the constraint language $\mathrm{IPC}^\star$, whose expressions are Boolean combinations of IPC^{++} constraints from [Dem04] and constraints of the form $x < y$. The language IPC^{++} is already closed under Boolean operators and first-order quantification. No constraint of the form $x < y$ occurs in the scope of a quantifier. Otherwise incrementation is definable and it leads to undecidability of the logic. So, as shown in this paper, adding the single type of constraints $x < y$ leads to many technical complications, but not to undecidability. We call $\mathrm{CLTL}(\mathrm{IPC}^\star)$ the specification language built over $\mathrm{IPC}^\star$ constraints. We also introduce the class of $\mathrm{IPC}^\star$-automata defined as finite-state automata with transitions labelled by $\mathrm{CLTL}(\mathrm{IPC}^\star)$ formula à la Wolper [Wol83]. Such structures can be viewed as labelled transition systems obtained by abstraction of counter automata.

Constraint LTL over IPC^{++} is shown to be in PSPACE in [Dem04] whereas constraint LTL over constraints of the form either $x = y$ or $x < y$ is also shown to be in PSPACE in [DD03]. Both proofs use reductions to the emptiness problem for Büchi automata following the approach in [VW94]. However, the proofs are of different nature: in [Dem04] the complexity upper bound is obtained by a finite model property argument whereas in [DD03] approximations of classes of symbolic models are considered because some formulae can generate non ω-regular classes of symbolic models. We show that model-checking and satisfiability problems for the logic CLTL(IPC*) are decidable (which was open so far) and moreover in PSPACE (PSPACE-hardness is easy). The proof substantially generalizes what is done for constraint LTL over the domain $\langle \mathbb{Z}, <, = \rangle$ by considering both new constraints of the form $x \leq d$, $d \in \mathbb{Z}$ and integer periodicity constraints. The optimal treatment of constants occurring in such constraints is our main technical contribution. As a corollary, we establish that LTL model-checking over integral relational automata [Čer94] is PSPACE-complete. Hence, even though IPC* is a powerful language of qualitative constraints, the PSPACE upper bound is preserved in CLTL(IPC*). To our opinion, we provide a definite complexity characterization of LTL with qualitative constraints over $\mathbb{Z}$.

Related work. Reachability problems for subclasses of counter systems have been addressed for instance in [Iba78,CJ98,FS00,FL02,BFLP03] (see also richer questions in [BEM97,JKMS04]). In our work, we have a full LTL-like language, not restricted to reachability questions, used as a specification language and no restriction on the structure of the models. However, atomic formulae of the specification language are qualitative constraints. If we give up the decidability requirement, LTL over Presburger constraints can be found in [BEH95,CC00,ID01].

Constraint LTL over concrete domains (not only restricted to $\mathbb{Z}$) has been considered in [WZ00,BC02,DD03,GKK$^+$03,Dem04] where often PSPACE-completeness is shown. The idea of building LTL over a language of constraints, although already present in first-order temporal logics, stems from the use of concrete domains for description logics [BH91,Lut04]. The language CLTL(IPC*) extends the different LTL-like fragments from [Čer94,LM01,DD03,Dem04] (past-time operators can be added for free in our formalism thanks to [GK03]). The class of IPC*-automata introduced in the paper generalizes the class of integral relational automata from [Čer94] (see details in Appendix A).

Integer periodicity constraints, a special class of Presburger constraints, have found applications in many logical formalisms such as abstractions with congruences modulo an integer of the form 2^k (see e.g. [CGL94,MOS05]), logical formalisms dealing with calendars (see e.g. [LM01]) and temporal reasoning in database access control [BBFS98]. Periodicity constraints can be also found in real-time logics (see e.g. [AH94]) and our approach of constraint LTL makes explicit the constraints on variables, similarly to the explicit clock approach from [HLP90]. Finally, the concept of symbolic models used in the paper has similarities with untimed languages recognized by some classes of timed machines. In [Bér95], sufficient conditions to get regular untimed languages from timed machines are exhibited.

Plan of the paper. In Section 2, we introduce the logic CLTL(IPC*), the class of IPC*-automata, we present the model-checking and satisfiability problems and discuss expressivity issues. In Section 3, we analyze the computational complexity of the model-checking and satisfiability problems of the underlying constraint language IPC*. Moreover, we provide a symbolic representation of the valuations that is used later in order to build the forthcoming Büchi automata. Section 4 contains a characterization of the sequences of symbolic valuations that admit models. In the case the sequences are ultimately periodic, testing realizability (existence of some model) is an ω-regular property. In Section 5, we show that given a CLTL(IPC*) formula ϕ, one can build a Büchi automaton $\mathcal{A}_\phi$ such that ϕ is CLTL(IPC*) satisfiable iff $\mathrm{L}(\mathcal{A}_\phi)$ is non-empty. Moreover, we establish that emptiness of $\mathrm{L}(\mathcal{A}_\phi)$ can be checked in polynomial space in $|\phi|$. Section 6 contains concluding remarks.

2 The logic CLTL(IPC*)

2.1 Language of constraints

Let $V = \{x_0, x_1, \ldots\}$ be a countably infinite set of variables (in some places for ease of presentation, V will denote a particular finite set of variables). The language of constraints p is defined by the following grammar:

$$p ::= pmod \mid x < y \mid p \wedge p \mid \neg p$$

$$\begin{aligned} pmod ::= \; & x \equiv_k [c_1, c_2] \mid x \equiv_k y + [c_1, c_2] \mid x = y \mid x < d \mid x = d \mid \\ & pmod \wedge pmod \mid \neg pmod \mid \exists x \; pmod \end{aligned}$$

where $x, y \in V$, $k \in \mathbb{N} \setminus \{0\}$, $c_1, c_2 \in \mathbb{N}$ and $d \in \mathbb{Z}$. This language is denoted by IPC*. We write IPC^{++} to denote its restriction to constraints ranged over by $pmod$, and Z$^\mathrm{c}$ its restriction to constraints of the form either $x \sim y$ or $x \sim d$. The symbol $\sim$ is used to mean either $=$ or $<$. The language Z is the restriction of Z$^\mathrm{c}$ to constraints of the form $x \sim y$. We define a valuation v as a map $v : V \rightarrow \mathbb{Z}$ and the satisfaction relation $v \models_\star p$ is defined as follows in the standard way:

- $v \models_\star x \sim y \stackrel{\text{def}}{\Leftrightarrow} v(x) \sim v(y)$; $\quad v \models_\star x \sim d \stackrel{\text{def}}{\Leftrightarrow} v(x) \sim d$;
- $v \models_\star x \equiv_k [c_1, c_2] \stackrel{\text{def}}{\Leftrightarrow} v(x)$ is equal to c modulo k for some $c_1 \leq c \leq c_2$;
- $v \models_\star x \equiv_k y + [c_1, c_2] \stackrel{\text{def}}{\Leftrightarrow} v(x) - v(y)$ is equal to c modulo k for some $c_1 \leq c \leq c_2$;
- $v \models_\star p \wedge p' \stackrel{\text{def}}{\Leftrightarrow} v \models_\star p$ and $v \models_\star p'$; $\qquad v \models_\star \neg p \stackrel{\text{def}}{\Leftrightarrow}$ not $v \models_\star p$;
- $v \models_\star \exists x \; p \stackrel{\text{def}}{\Leftrightarrow}$ there is $z \in \mathbb{Z}$ such that $v[x \leftarrow z] \models_\star p$ where $v[x \leftarrow z](x') = v(x')$ if $x \neq x'$ and $v[x \leftarrow z](x) = z$.

We recall that x is equal to y modulo k if there is $z \in \mathbb{Z}$ such that $x - y = k \times z$. We write $x \equiv_k c$ instead of $x \equiv_k [c, c]$, $x \equiv_k y + c$ instead of $x \equiv_k y + [c, c]$ and $v \models_\star X$ where X is a set of IPC*-constraints, whenever $v \models_\star p$ for every $p \in X$.

A constraint p is satisfiable iff there is a valuation v such that $v \models_\star p$. Two constraints are equivalent iff they are satisfied by the same valuations.

Lemma 1.

(I) *The satisfiability problem for* IPC* *is* PSPACE*-complete.*

(II) *Every constraint in* IPC* *admits an equivalent quantifier-free constraint in* IPC*.

Proof. (I) Satisfiability for IPC^{++} is PSPACE-complete [Dem04] whereas satisfiability for Z is NLOGSPACE-complete. Since constraints in IPC* are Boolean combinations of IPC^{++} and Z constraints, IPC* satisfiability is in PSPACE by simply adapting the proof of [Dem04, Theorem 3]. PSPACE-hardness is a consequence of the PSPACE-hardness of IPC^{++}.
(II) IPC^{++} admits quantifier elimination [Dem04] and therefore so does IPC* since Z is quantifier-free. □

Hence, IPC* is a quite well understood fragment of Presburger arithmetic.

2.2 Logical language

We consider the linear-time temporal logic CLTL(IPC*) whose atomic formulae are defined from constraints in IPC*. This is an extension of LTL where propositional variables are replaced by IPC* constraints. The atomic formulae are of the form $p[x_1 \leftarrow \mathsf{X}^{i_1}x_{j_1}, \ldots, x_r \leftarrow \mathsf{X}^{i_r}x_{j_r}]$, where p is a constraint of IPC* with free variables $x_1 \ldots x_r$. We substitute each occurrence of the variable x_l with $\mathsf{X}^{i_l}x_{j_l}$, which corresponds to the variable x_{j_l} preceded by i_l next symbols. Each expression of the form $\mathsf{X}^{\beta}x_\alpha$ is called a term and represents the value of the variable x_α at the β^{th} next state. Here are examples of atomic formulae: $\mathsf{X}y \equiv_{2^{32}} x+1$ and $x < \mathsf{X}y$.

The set of CLTL(IPC*) formulae ϕ is defined by

$$\phi ::= p[x_1 \leftarrow \mathsf{X}^{i_1}x_{j_1}, \ldots, x_r \leftarrow \mathsf{X}^{i_r}x_{j_r}] \mid \neg\phi \mid \phi \wedge \phi \mid \mathsf{X}\phi \mid \phi\mathsf{U}\phi,$$

where p belongs to IPC*. The operators next (X) and until (U) are the classical operators used in temporal logics. In the language, all the integers are encoded with a binary representation (this is important for complexity considerations).

A model $\sigma : \mathbb{N} \times V \rightarrow \mathbb{Z}$ for CLTL(IPC*) is an ω-sequence of valuations. The satisfaction relation is defined as follows (we omit the Boolean cases):

- $\sigma, i \models p[x_1 \leftarrow \mathsf{X}^{i_1}x_{j_1}, \ldots, x_r \leftarrow \mathsf{X}^{i_r}x_{j_r}]$ iff $[x_1 \leftarrow \sigma(i+i_1, x_{j_1}), \ldots$ $, x_r \leftarrow \sigma(i+i_r, x_{j_r})] \models_\star p$;
- $\sigma, i \models \mathsf{X}\phi$ iff $\sigma, i+1 \models \phi$;
- $\sigma, i \models \phi\mathsf{U}\phi'$ iff there is $j \geq i$ s.t. $\sigma, j \models \phi'$ and for every $i \leq l < j,\ \sigma, l \models \phi$.

2.3 Satisfiability and model-checking problems

We recall below the problems we are interested in.

Satisfiability problem for CLTL(IPC*):
Given a CLTL(IPC*) formula ϕ, is there a model σ such that $\sigma, 0 \models \phi$?

If we extend IPC* to allow constraints of the form $x < y$ in the scope of $\exists$, then the satisfiability problem for the corresponding constraint LTL-like logic is undecidable since the successor relation is then definable and the halting problem for Minsky machines can be easily encoded.

The model-checking problem rests on IPC*-automata which are constraint automata [Rev02]. An IPC*-automaton $\mathcal{A}$ is defined as a Büchi automaton over the infinite alphabet composed of CLTL(IPC*) formulae. In an IPC*-automaton, letters on transitions may induce constraints between the variables of the current state and the variables of the next state as done in [CC00]. Hence, guards and update functions are expressed in the same formalism. We are however a bit more general since we allow formulae on transitions as done in [Wol83]. As an illustration, we present an IPC*-automaton in Fig. 1 which is an abstraction of the pay-phone controller from [CC00, Example 1] (x is the number of quarters which have been inserted and y measures the total communication time). Incrementation of a variable z is abstracted by $\mathsf{X}z \equiv_{2^{32}} z + 1 \wedge \mathsf{X}z > z$. The formula $\phi_=$ denotes $\mathsf{X}x = x \wedge \mathsf{X}y = y$. Messages are omitted because they are irrelevant here (simplifications are then possible).

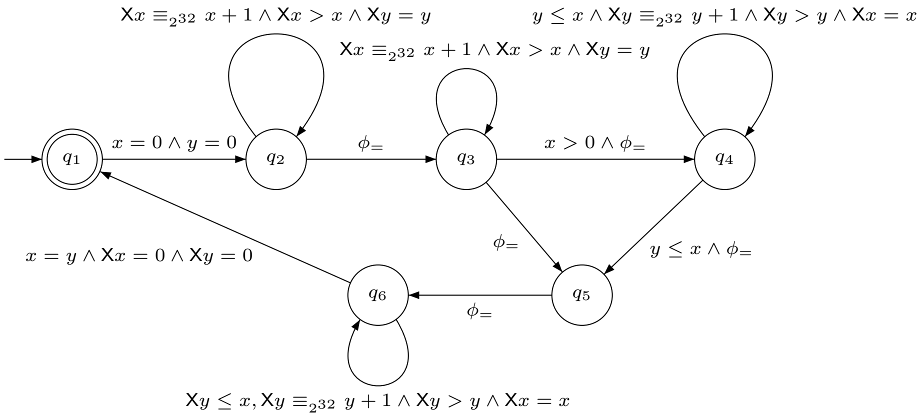

Fig. 1. An IPC*-automaton

Model-checking problem for CLTL(IPC*):
Given an IPC*-automaton $\mathcal{A}$ and a CLTL(IPC*) formula ϕ, are there a symbolic ω-word $v = \phi_0 \cdot \phi_1 \cdot \ldots$ accepted by $\mathcal{A}$ and a model σ (a realization of v) such that $\sigma, 0 \models \phi$ and for every $i \geq 0$, $\sigma, i \models \phi_i$?

The satisfiability problem and the model-checking problem are reducible to each other in logspace following techniques from [SC85], by possibly introducing a new variable. In the following sections, we prove results for the satisfiability problem but they also extend to the model-checking problem.

A restricted IPC*-automaton is defined as an IPC*-automaton such that the labels on transitions are Boolean combinations of atomic formulae with terms of the form x and $\mathsf{X}x$ (see Fig. 1). The logic CCTL*(IPC*) (constraint CTL* over IPC* constraints) is defined as the extension of CLTL(IPC*) with the path quantifiers $\exists$ and $\forall$ but restricted to atomic formulae with no variables in V preceded by X. The models of CCTL*(IPC*) are the configuration graphs of restricted IPC*-automata. The satisfaction relation $\mathcal{A}, \langle q, \overline{x} \rangle \models \phi$ is defined in the usual way. The model-checking problem for $\mathrm{CCTL}^*(\mathrm{IPC}^\star)$ takes as inputs a restricted $\mathrm{IPC}^\star$-automaton $\mathcal{A}$, an initial configuration $\langle q, \overline{0}\rangle$ (q is a control state and $\overline{0}$ is the initial valuation with null values for the variables) and a $\mathrm{CCTL}^*(\mathrm{IPC}^\star)$ formula ϕ and checks whether $\mathcal{A}, \langle q, \overline{0}\rangle \models \phi$.

The equivalence problem for Extended Single-String automata [LM01] can be encoded as a model-checking problem for $\mathrm{CLTL}(\mathrm{IPC}^\star)$ [Dem04]. Furthermore, the model-checking problem for integral relational automata restricted to the LTL fragment of CCTL^* introduced in [Čer94] is a subproblem of the model-checking problem for $\mathrm{CLTL}(\mathrm{IPC}^\star)$ (see details in Appendix A).

Full $\mathrm{CCTL}^*(\mathrm{IPC}^\star)$ model-checking can be shown to be undecidable by using developments in Appendix A and [Čer94] (even its CTL-like fragment) and one can show that its LTL fragment is decidable in polynomial space, a new result not captured by [Čer94].

2.4 Expressive power and conciseness of the language

By definition, $\mathrm{CLTL}(\mathrm{IPC}^\star)$-models interpret variables but not propositional variables. However, it is not difficult to encode propositional variables by using atomic formulae of the form $x = 0$ where x is a new variable introduced for this purpose. Given a set of constraints X included in $\mathrm{IPC}^\star$, we write $\mathrm{CLTL}(X)$ to denote the restriction of $\mathrm{CLTL}(\mathrm{IPC}^\star)$ in which the atomic constraints are built over elements of X. The model-checking problem for $\mathrm{CLTL}(\mathrm{IPC}^{++})$ (resp. CLTL(Z)) is shown to be PSPACE-complete in [Dem04] (resp. in [DD03]). However, the proof for IPC^{++} uses an ω-regular property of the set of models that does not hold when we introduce constraints of the form $x < y$. The problem for $\mathrm{CLTL}(\mathrm{Z}^c)$ is shown to be in EXPSPACE in [DD03] by translation into CLTL(Z) that increases exponentially the size of formulae (with a binary encoding of the integers).

We write $\mathrm{WIPC}^\star$ (weak $\mathrm{IPC}^\star$) to denote the constraints of the form below:

$$x \sim y \ \mid \ x \sim d \ \mid \ x \equiv_k c$$

where $x, y \in V$, $\sim \in \{<, =\}$, $d \in \mathbb{Z}$ and $k, c \in \mathbb{N}$. $\mathrm{WIPC}^\star$ is a fragment of $\mathrm{IPC}^\star$. $\mathrm{CLTL}(\mathrm{WIPC}^\star)$ is as expressive as $\mathrm{CLTL}(\mathrm{IPC}^\star)$ as witnesses by the property below.

Lemma 2. *For every $\phi \in \mathrm{CLTL}(\mathrm{IPC}^\star)$, there is $\psi \in \mathrm{CLTL}(\mathrm{WIPC}^\star)$, such that $\{\sigma : \sigma, 0 \models \phi\} = \{\sigma : \sigma, 0 \models \psi\}$.*

Proof. This a direct consequence of the facts below:

- $\mathrm{IPC}^\star$ admits quantifier-elimination.
- $x \equiv_k [c_1, c_2]$ is equivalent to $\bigvee_{c_1 \leq c \leq c_2} x \equiv_k c$.
- $x \equiv_k y + [c_1, c_2]$ is equivalent to $\bigvee\{x \equiv_k [c'_1, c'_1] \wedge y \equiv_k [c'_2, c'_2] : c'_1 \equiv_k c'_2 \ [c'], \ c_1 \leq c' \leq c_2\}$.

□

It is not difficult to show that $|\psi|$ is in $2^{\mathcal{O}(|\phi|)}$. In spite of this exponential blow-up, we shall prove that both CLTL(WIPC*) and CLTL(IPC*) have PSPACE-complete model-checking problems.

3 Properties of the constraint language

In this section, we establish results about the constraint language underlying the logic CLTL(IPC*). In order to define automata that recognize symbolic representations of CLTL(IPC*)-models, the valuations v of the form $V \rightarrow \mathbb{Z}$ are represented by symbolic valuations. Given a finite set X of IPC* constraints, typically the set of constraints occurring in a given CLTL(IPC*) formula, we introduce the following notations:

- K is the lcm of $k_1, \ldots, k_n$ where periodicity constraints with relations $\equiv_{k_1}$, $\ldots, \equiv_{k_n}$ occur in X. Observe that $|K|$ is in $\mathcal{O}(|k_1| + \cdots + |k_n|)$.
- C is the set of constants d occurring in constraints of the form $x \sim d$.
- m is the minimal element of C and M is its maximal element.
- C' denotes the set of constants $\{m, m-1, \ldots, M\}$. The cardinality of C' is in $\mathcal{O}(2^{|m|+|M|})$ and each element of C' can be binary encoded in binary representation with $\mathcal{O}(|m| + |M|)$ bits.
- V is the finite set of variables occurring in X.

In the remaining, we assume that the above objects are always defined (possibly by adding dummy valid constraints in order to make the sets non-empty).

A maximally consistent set Y of Z^c constraints with respect to V and C is a set of Z^c constraints using only the variables from V and the constants from C such that there is a valuation $v : V \rightarrow \mathbb{Z}$ verifying $v \models_\star Y$ and for any proper extension Z of Y, there is no valuation $v' : V \rightarrow \mathbb{Z}$ verifying $v' \models_\star Z$. A valuation is abstracted by three disjoint finite sets of IPC* constraints like regions for timed automata.

Definition 1. *Given a finite set X of* IPC* *constraints, a symbolic valuation sv is a triple $\langle Y_1, Y_2, Y_3 \rangle$ such that*

- *Y_1 is a maximally consistent set of* Z^c *constraints wrt V and C.*
- *Y_2 is a set of constraints of the form $x = d$ with $x \in V$ and $d \in C' \setminus C$. Each $x \in V$ occurs at most in one constraint of the form $x = d$ in Y_2. Moreover, for every $x \in V$, $(x = d) \in Y_2$ for some unique $d \in C' \setminus C$ iff for every $d' \in C$, $(x = d') \notin Y_1$ and $\{m < x, x < M\} \subseteq Y_1$.*
- *Y_3 is a set of constraints of the form $x \equiv_K c$ with $x \in V$ and $0 \leq c \leq K-1$. Each $x \in V$ occurs exactly in one constraint of the form $x \equiv_K c$ in Y_3.*

A consequence of Definition 1 is that in a symbolic valuation $sv = \langle Y_1, Y_2, Y_3 \rangle$, no constraint occurs in more than one set. That is why, given an IPC* constraint p, we write $p \in sv$ instead of $p \in Y_1 \cup Y_2 \cup Y_3$. A symbolic valuation is satisfiable iff there is a valuation $v : V \rightarrow \mathbb{Z}$ such that $v \models_\star Y_1 \cup Y_2 \cup Y_3$.

Lemma 3. *Let X be a finite set of* IPC* *constraints and $sv = \langle Y_1, Y_2, Y_3 \rangle$ be a triple composed of* IPC* *constraints such that Y_1 is a set of* Z^c *constraints built over V and C, Y_2 is a set of* Z^c *constraints of cardinality at most $|V|$ built over V and $C' \setminus C$, Y_3 is a set of constraints of the form $x \equiv_K c$ of cardinality $|V|$. Checking whether sv is a satisfiable symbolic valuation can be done in polynomial-time in the sum of the respective size of X and sv.*

Maximal consistency of Y_1 can be checked in polynomial-time by using developments from [Čer94, Lemma 5.5]. Indeed, given a set Y of Z^c constraints built over V and C, a graph G_Y can be built such that Y is maximally consistent wrt V and C iff G_Y satisfies the conditions below. G_Y is a structure $\langle V \cup C, \xrightarrow{=}, \xrightarrow{<} \rangle$ such that $n \xrightarrow{\sim} n' \overset{\text{def}}{\Leftrightarrow} n \sim n'$ belongs to Y. Following [Čer94, Lemma 5.5], Y is maximally consistent iff G_Y satisfies the conditions below:

(MC1) For all n, n', either $n \xrightarrow{\sim} n'$ or $n' \xrightarrow{\sim} n$ for some $\sim \in \{<, =\}$.
(MC2) $\xrightarrow{=}$ is a congruence relation compatible with $\xrightarrow{<}$.
(MC3) There is no path $n_0 \xrightarrow{\sim_0} n_1 \xrightarrow{\sim_1} \ldots \xrightarrow{\sim_{\alpha-1}} n_\alpha$ with $n_0 = n_\alpha$ and $<$ occurs in $\{\sim_0, \sim_1, \ldots, \sim_{\alpha-1}\}$.
(MC4) For all $d_1, d_2 \in C$, $d_1 \sim d_2$ implies $d_1 \xrightarrow{\sim} d_2$.
(MC5) For all d_1, d_2 with $d_1 \leq d_2$, there is no path $n_0 \xrightarrow{\sim_0} n_1 \xrightarrow{\sim_1} \ldots \xrightarrow{\sim_{\alpha-1}} n_\alpha$ with $n_0 = d_1$ and $n_\alpha = d_2$ such that the cardinality of $\{i : \sim_i \text{ equals } <, 1 \leq i \leq \alpha - 1\}$ is strictly more than $d_2 - d_1$.

The symbolic representations of valuations contain the relevant information to evaluate constraints.

Lemma 4. *Let X be a finite set of* IPC* *constraints.*

(I) *For every valuation $v : V \to \mathbb{Z}$ there is a unique symbolic valuation $sv(v) = \langle Y_1, Y_2, Y_3 \rangle$ such that $v \models_\star Y_1 \cup Y_2 \cup Y_3$.*
(II) *For all valuations v, v' such that $sv(v) = sv(v')$ and for every $p \in X$, $v \models_\star p$ iff $v' \models_\star p$.*

Proof. (I) By an easy verification.
(II) By structural induction on p. The proof is similar to the proof of [Dem04, Lemma 1]. □

By Lemma 4, a symbolic valuation is an equivalence class of valuations.

Given a symbolic valuation sv and p a constraint, we write $sv \models_{\text{symb}} p \overset{\text{def}}{\Leftrightarrow}$ for every valuation v such that $sv(v) = sv$, $v \models_\star p$.

Lemma 5. *The problem of checking whether $sv \models_{\text{symb}} p$ is* PSPACE*-complete (given that the syntactic resources used in p are included in those used for the symbolic valuation sv).*

Proof. PSPACE-hardness can be easily obtained by reducing QBF. We can restrict ourselves to the case the constants in X are 0 and 1. The QBF formula $\phi = Q_1\ p_1\ Q_2\ p_2\ \ldots Q_n\ p_n\ \phi'$ with ϕ' a propositional formula in CNF built over the propositional variables $\{p_1, \ldots, p_n\}$ is translated via the map t such that

- $t(\exists\ p_i\ \phi_i) = \exists\ x_i\ (x_i = 0 \vee x_i = 1) \wedge t(\phi_i)$,
- $t(\forall\ p_i\ \phi_i) = \forall\ x_i\ (x_i = 0 \vee x_i = 1) \Rightarrow t(\phi_i)$,
- t is homomorphic for Boolean connectives,
- $t(p_i)$ equals "$x_i = 1$".

One can show that ϕ is QBF satisfiable iff for all symbolic valuations sv, $sv \models_{\rm symb} t(\phi)$ which is equivalent to check that an arbitrary symbolic valuation symbolically satisfies $t(\phi)$, since $t(\phi)$ has no free variable.
In order to show the PSPACE upper bound, the proof is similar to the proof of PSPACE upper bound for first-order model-checking. Indeed, one can define a function $\mathrm{MC}(sv, p', p)$ with

- p' an IPC* constraint occurring in the IPC* constraint p,
- sv is a symbolic valuation over the free variables in p',

that returns 1 iff $sv \models_{\rm symb} p'$. Observe that if a variable occurs in sv but is not free in p' then the satisfaction of $sv \models_{\rm symb} p'$ is independent of its value. The function MC is defined as a case analysis on the form of p'. For instance, $\mathrm{MC}(sv, \exists\ x\ p', p)$ returns 1 iff there is a satisfiable symbolic valuation sv' extending sv by addition of x-constraints, such that $\mathrm{MC}(sv', p', p)$ returns 1. The symbolic valuations $sv' = \langle Y_1', Y_2', Y_3' \rangle$ and $sv = \langle Y_1, Y_2, Y_3 \rangle$ are related as follows:

- sv' is a satisfiable symbolic valuation over the free variables of p'. This can be checked in polynomial-time in $|p|$ (by Lemma 3).
- $Y_1 \subseteq Y_1'$, $Y_2 \subseteq Y_2'$ and $Y_3 \subseteq Y_3'$.
- The only variable in $sv' \setminus sv$ is x.

Even if the number of symbolic valuations over the free variables of p' is exponential in $|p|$, it is possible to enumerate them in polynomial space in order to check the existence of some sv' verifying the above conditions. $\Box$

4 Satisfiable ω-sequences of symbolic valuations

Given a CLTL(IPC*) formula ϕ, we write IPC$^\star(\phi)$ to denote the set of IPC* constraints p such that some atomic formula of the form $p[x_1 \leftarrow \mathsf{X}^{i_1} x_{j_1}, \ldots, x_r \leftarrow \mathsf{X}^{i_r} x_{j_r}]$ occurs in ϕ. To IPC$^\star(\phi)$ we associate the objects relative to any finite set of IPC* constraints. The set V denotes the set of variables occurring in ϕ. We write $|\phi|_{\mathsf{X}}$ to denote the maximal natural number i such that $\mathsf{X}^i x$ occurs in ϕ for some variable x. $|\phi|_{\mathsf{X}}$ is called the X-length of ϕ. Without any loss of generality, we can assume that $|\phi|_{\mathsf{X}} \geq 1$. In the following, we assume that $V = \{x_1, \ldots, x_s\}$ and $|\phi|_{\mathsf{X}} = l$. We write $\mathrm{Terms}(\phi)$ to denote the set of terms of the form $\mathsf{X}^\beta x_\alpha$ with $\beta \in \{0, \ldots, l\}$ and $\alpha \in \{1, \ldots, s\}$.

Let V' be a set of variables of cardinality $|\mathrm{Terms}(\phi)|$ and $f : \mathrm{Terms}(\phi) \to V'$ be an unspecified bijection such that f and f^{-1} can be computed in polynomial time. By extension, for every atomic subformula p of ϕ, $f(p)$ is obtained from p by replacing each occurrence of $\mathsf{X}^\beta x_\alpha$ by $f(\mathsf{X}^\beta x_\alpha)$. The map f^{-1} is used in a similar fashion. A symbolic valuation wrt ϕ is a symbolic valuation built over the set of variables V', C and K.

We say that a pair $\langle\langle Y_1, Y_2, Y_3\rangle, \langle Y'_1, Y'_2, Y'_3\rangle\rangle$ of symbolic valuations wrt ϕ is one-step consistent $\stackrel{\text{def}}{\Leftrightarrow}$

1. $f(\mathsf{X}^j x_i) \sim f(\mathsf{X}^{j'} x_{i'}) \in Y_1$ and $j, j' \geq 1$ imply $f(\mathsf{X}^{j-1} x_i) \sim f(\mathsf{X}^{j'-1} x_{i'}) \in Y'_1$,
2. $f(\mathsf{X}^j x_i) \sim d \in Y_1 \cup Y_2$ and $j \geq 1$ imply $f(\mathsf{X}^{j-1} x_i) \sim d \in Y'_1 \cup Y'_2$,
3. $f(\mathsf{X}^j x_i) \equiv_K c \in Y_3$ and $j \geq 1$ imply $f(\mathsf{X}^{j-1} x_i) \equiv_K c \in Y'_3$.

An ω-sequence ρ of satisfiable symbolic valuations wrt ϕ is one-step consistent $\stackrel{\text{def}}{\Leftrightarrow}$ for every $j \in \mathbb{N}$, $\langle \rho(j), \rho(j+1)\rangle$ is one-step consistent. A model for ρ is defined as a CLTL(IPC*)-model σ such that for all $j \in \mathbb{N}$ and $p \in \rho(j)$, $\sigma, j \models f^{-1}(p)$. In order to simplify the future developments, we write ρ_f to denote the ω-sequence obtained from ρ by substituting each occurrence of some variable x by $f^{-1}(x)$.

One-step consistent ω-sequences of symbolic valuations wrt ϕ define abstractions of models for ϕ. We represent a one-step consistent sequence ρ as an infinite labeled structure $G_\rho = \langle (V \cup C') \times \mathbb{N}, \stackrel{=}{\rightarrow}, \stackrel{<}{\rightarrow}, mod\rangle$ where $mod : (V \cup C') \times \mathbb{N} \rightarrow \{0, \ldots, K-1\}$:

$$
\begin{array}{ll}
\langle x, i\rangle \stackrel{\sim}{\rightarrow} \langle y, j\rangle & \text{iff either } i \leq j \text{ and } x \sim \mathsf{X}^{j-i} y \in \rho_f(i) \\
& \quad\ \text{or } i > j \text{ and } \mathsf{X}^{i-j} x \sim y \in \rho_f(j), \\
\langle x, i\rangle \stackrel{=}{\rightarrow} \langle d, j\rangle & \text{iff } x = d \ \in \rho_f(i), \\
\langle d, i\rangle \stackrel{=}{\rightarrow} \langle x, j\rangle & \text{iff } x = d \ \in \rho_f(j), \\
\langle x, i\rangle \stackrel{<}{\rightarrow} \langle d, j\rangle & \text{iff there is } d' \sim d \text{ such that } x \sim' d' \ \in \rho_f(i) \text{ and } {<} \in \{\sim, \sim'\}, \\
\langle d, i\rangle \stackrel{<}{\rightarrow} \langle x, j\rangle & \text{iff there is } d \sim d' \text{ such that } d' \sim' x \ \in \rho_f(j) \text{ and } {<} \in \{\sim, \sim'\}, \\
\langle d_1, i\rangle \stackrel{\sim}{\rightarrow} \langle d_2, j\rangle & \text{iff } d_1 \sim d_2, \\
mod(\langle x, i\rangle) = c & \text{iff } x \equiv_K c \in \rho_f(i) \text{ and } mod(\langle d, i\rangle) = c \text{ iff } d \equiv_K c.
\end{array}
$$

for all $x, y \in V$, $d_1, d_2 \in C$ and $i, j \in \mathbb{N}$ such that $|i - j| \leq l$. By construction of G_ρ, the variables and constants are treated in a similar fashion. It is worth observing that G_ρ is well-defined because ρ is one-step consistent. The construction ensures that the "local" representation of every $\rho(i)$ verifies the conditions (MC1) to (MC5) of Sect. 3.

In the following, we say that a vertex represents the constant d if it is of the form $\langle d, i\rangle$ for some i. The level of a node $n = \langle a, t\rangle$ in G_ρ is t, and is denoted by lev(n). There is some redundancy in G_ρ for the nodes of the form $\langle d, i\rangle$. However, this is useful to establish strict relationships between ρ and G_ρ.

Example 1. Assuming that $C = \{2, 4\}$, $K = 2$, $V' = \{x, x'\}$ ($f(x) = x$ and $f(\mathsf{X}x) = x'$) and $l = 1$, let us define the sequence $\rho = sv^0 \cdot (sv^1 \cdot sv^2)^\omega$ where $sv^0 = \langle Y_1^0, Y_2^0, Y_3^0\rangle$ such that $Y_1^0 = \{x = x, x' = x', x < x', 2 < x, x < 4, x' = 4, x' > 2\}$, $Y_2^0 = \{x = 3\}$ and $Y_3^0 = \{x \equiv_2 1, x' \equiv_2 0\}$. The symbolic valuation $sv^1 = \langle Y_1^1, Y_2^1, Y_3^1\rangle$ satisfies: $Y_1^1 = (Y_1^0 \setminus \{2 < x, x < 4, x' = 4, x' > 2\}) \cup \{4 < x, 4 < x'\}$, $Y_2^1 = \emptyset$ and $Y_3^1 = \{x \equiv_2 0, x' \equiv_2 1\}$. The symbolic valuation $sv^2 = \langle Y_1^2, Y_2^2, Y_3^2\rangle$ verifies $Y_1^2 = Y_1^1$, $Y_2^2 = Y_1^2$ and $Y_3^2 = Y_3^0$. The graph G_ρ is presented in Fig. 2. In order to simplify the representation, closure by transitivity for $\xrightarrow{<}$ and the fact that $\xrightarrow{=}$ is a congruence are omitted. The function *mod* is directly encoded in the node label.

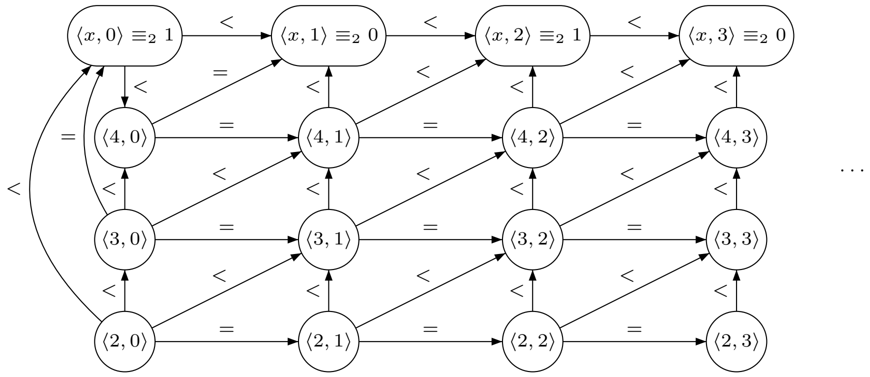

Fig. 2. A graph G_ρ

A path in G_ρ is a sequence (possible infinite) of the form $n_0 \xrightarrow{\sim_0} n_1 \xrightarrow{\sim_1} n_2 \xrightarrow{\sim_2} \ldots$. For any finite path $w = n_0 \xrightarrow{\sim_0} n_1 \xrightarrow{\sim_1} n_2 \xrightarrow{\sim_2} \ldots \xrightarrow{\sim_{\alpha-1}} n_\alpha$, its strict length $slen(w)$ is the cardinality of $\{i : 0 \leq i \leq \alpha-1,\ \sim_i \text{ equals } <\}$. When w has a strict length greater than 1, we say that w is strict. A finite path w such that $n_0 = n_\alpha$ is called a cycle. The strict length between two nodes n_1 and n_2, written $slen(n_1, n_2)$, is the least upper bound of the strict lengths of finite paths between n_1 and n_2. By convention, if there is no path between n_1 and n_2, $slen(n_1, n_2)$ takes the value $-\infty$. In Fig. 2, $slen(\langle 2, 2\rangle, \langle x, 3\rangle) = 4$.

In Lemma 6 below, the one-step consistency of ρ implies global constraints on its graph representation that already hold true locally. By a global constraint, we mean a constraint on the whole graph and not only on the local representation of a single symbolic valuation or on two successive satisfiable symbolic valuations.

Lemma 6. *Let ρ be a one-step consistent sequence.*

(I) *G_ρ has no strict cycle.*
(II) *If there is a finite path w starting at $\langle d, i\rangle$ and ending at the node n of level j, then: if w is strict then $\langle d, j\rangle \xrightarrow{<} n$, otherwise $\langle d, j\rangle \xrightarrow{=} n$.*
(III) *If there is a finite path w starting at the node n of level j and ending at $\langle d, i\rangle$, then: if w is strict then $n \xrightarrow{<} \langle d, j\rangle$, otherwise $n \xrightarrow{=} \langle d, j\rangle$.*

Proof. (I) We show that for every path $w = n_0 \xrightarrow{\sim_0} n_1 \xrightarrow{\sim_1} n_2 \xrightarrow{\sim_2} \ldots \xrightarrow{\sim_{\alpha-1}} n_\alpha$, $n_0 = n_\alpha$ implies $slen(w) = 0$. The proof is by induction on α. The base case with either $\alpha = 1$ or $\alpha = 2$ is by an easy verification since $\rho(min\{\mathrm{lev}(n_i) : 1 \leq i \leq \alpha\})$ satisfies (MC3). In the induction step, suppose that w is a cycle with $\alpha > 2$. Let n_β be a node such that $\mathrm{lev}(n_\beta) = max\{\mathrm{lev}(n_i) : 1 \leq i \leq \alpha\}$. Without any loss of generality, we can assume that $0 < \beta < \alpha$ since $\alpha > 2$ and in a cycle the nodes can be easily permuted. One can always choose the first node in the cycle. Since $\mathrm{lev}(n_\beta)$ is maximal, $|\mathrm{lev}(n_{\beta-1}) - \mathrm{lev}(n_{\beta+1})| \leq l$. The symbolic valuation $\rho(min\{\mathrm{lev}(n_{\beta-1}), \mathrm{lev}(n_{\beta+1})\})$ satisfies (MC2) which implies that $n_{\beta-1} \xrightarrow{\leq} n_{\beta+1}$ iff $<$ belongs to $\{\sim_{\beta-1}, \sim_\beta\}$. As a consequence, we have $slen(w) = 0$ iff $slen(w') = 0$ with the path w' obtained from w by replacing $n_{\beta-1} \xrightarrow{\sim_{\beta-1}} n_\beta \xrightarrow{\sim_\beta} n_{\beta+1}$ by $n_{\beta-1} \xrightarrow{\sim} n_{\beta+1}$. By induction hypothesis, $n_0 = n_\alpha$ implies $slen(w') = 0$ and therefore $slen(w) = 0$.

(II) We show that for every path $w = n_0 \xrightarrow{\sim_0} n_1 \xrightarrow{\sim_1} n_2 \xrightarrow{\sim_2} \ldots \xrightarrow{\sim_{\alpha-1}} n_\alpha$ such that $n_0 = \langle d, i\rangle$, $\langle d, \mathrm{lev}(n_\alpha)\rangle \xrightarrow{\sim} n_\alpha$ and $\sim$ equals $<$ iff w is strict. The proof is by induction on α. The base case with $\alpha = 1$ is obvious. In the induction step, let w be a path with $n_0 = \langle d, i\rangle$ and $\alpha > 1$. By induction hypothesis, $\langle d, \mathrm{lev}(n_{\alpha-1})\rangle \xrightarrow{\sim'} n_{\alpha-1}$ and $\sim'$ equals $<$ iff $w' = n_0 \xrightarrow{\sim_0} n_1 \xrightarrow{\sim_1} n_2 \xrightarrow{\sim_2} \ldots \xrightarrow{\sim_{\alpha-2}} n_{\alpha-1}$ is strict. We treat the case $\mathrm{lev}(n_{\alpha-1}) \geq \mathrm{lev}(n_\alpha)$, the other case is similar. Since $\mathrm{lev}(n_{\alpha-1}) - \mathrm{lev}(n_\alpha) \leq l$ and $\rho(\mathrm{lev}(n_\alpha))$ satisfies (MC2), $\langle d, \mathrm{lev}(n_{\alpha-1})\rangle \xrightarrow{\sim} n_\alpha$ and $\sim$ equals $<$ iff either $\sim'$ or $\sim_{\alpha-1}$ equals $<$. Hence, $\langle d, \mathrm{lev}(n_{\alpha-1})\rangle \xrightarrow{\sim} n_\alpha$ and $\sim$ equals $<$ iff w is strict. Since $\langle d, \mathrm{lev}(n_{\alpha-1})\rangle \xrightarrow{=} \langle d, \mathrm{lev}(n_\alpha)\rangle$ and $\rho(\mathrm{lev}(n_\alpha))$ satisfies (MC2), we get $\langle d, \mathrm{lev}(n_\alpha)\rangle \xrightarrow{\sim} n_\alpha$ and $\sim$ equals $<$ iff w is strict.

(III) Similar to (II). □

Corollary 1. *Let ρ be a one-step consistent sequence and G_ρ its graph representation. Then, for all nodes $\langle d_1, i\rangle$ and $\langle d_2, j\rangle$ in G_ρ representing constants such that $d_1 \leq d_2$, $slen(\langle d_1, i\rangle, \langle d_2, j\rangle) = d_2 - d_1$.*

Proof. Let $\langle d_1, i\rangle$ and $\langle d_2, j\rangle$ be vertices of G_ρ representing respectively the constants d_1 and d_2. Without any loss of generality, we can assume that $i \leq j$ (the case $j > i$ has a similar treatment). The path below is of strict length $d_2 - d_1$:

$$\langle d_1, i\rangle \xrightarrow{=} \langle d_1, i+1\rangle \xrightarrow{=} \langle d_1, i+2\rangle \ldots \langle d_1, j\rangle \xrightarrow{\leq} \langle d_1 + 1, j\rangle \xrightarrow{\leq} \langle d_1 + 2, j\rangle \ldots \langle d_2, j\rangle.$$

So $slen(\langle d_1, i\rangle, \langle d_2, j\rangle) \geq d_2 - d_1$.

Now suppose that there is a path w between $\langle d_1, i\rangle$ and $\langle d_2, j\rangle$ such that $slen(w) > d_2 - d_1$. Let R be the restriction of the transitive closure of $\xrightarrow{=}$ to the nodes appearing in w. R is an equivalence relation with exactly $slen(w) + 1$ equivalence classes, say $X_0, \ldots, X_{slen(w)}$. By Lemma 6(II,III), for every $i \in \{0, \ldots, slen(w)\}$, there is $d_1 \leq d'_i \leq d_2 \in C'$ such that every node n of level j in X_i satisfies $n \xrightarrow{=} \langle d'_i, j\rangle$ in G_ρ. By satisfiability of every symbolic valuation $\rho(\mathrm{lev}(n))$ with n in w, the constants $d'_1, \ldots, d'_{slen(w)+1}$ are mutually distinct, which leads to a contradiction. □

So far, we have stated properties about the graph G_ρ. Below, we establish simple conditions on G_ρ equivalent to the existence of a model for ρ. An edge-respecting labeling for G_ρ is a map $lab : (V \cup C') \times \mathbb{N} \to \mathbb{Z}$ such that

- for all nodes n_1, n_2, $n_1 \xrightarrow{\sim} n_2$ implies $lab(n_1) \sim lab(n_2)$,

– for every node n, $lab(n) \equiv_K mod(n)$.

Additionally, lab is said to be strict if for every $\langle d, i\rangle$ in G_ρ, $lab(\langle d, i\rangle) = d$.

Lemma 7. *A one-step consistent sequence ρ has a model iff G_ρ has a strict edge-respecting labeling.*

Proof. Let σ be a model for ρ and $lab : (V \cup C') \times \mathbb{N} \to \mathbb{Z}$ be the map defined by:

$$lab(\langle x, i\rangle) = \sigma(i, x); \quad lab(\langle d, i\rangle) = d \quad \text{for all } x \in V, d \in C' \text{ and } i \in \mathbb{N}.$$

It is not difficult to show that lab is a strict edge-respecting labeling for G_ρ. For instance, we have implications between the propositions below :

– $\langle x, i\rangle \xrightarrow{\sim} \langle y, j\rangle$ and $i \leq j$,
– $x \sim \mathsf{X}^{j-i} y \in \rho_f(i)$ (by definition of G_ρ),
– $f(x \sim \mathsf{X}^{j-i} y) \in \rho(i)$ (by definition of ρ_f),
– $\sigma, i \models f^{-1}(f(x \sim \mathsf{X}^{j-i} y))$ (σ be a model for ρ),
– $\sigma, i \models x \sim \mathsf{X}^{j-i} y$ (f is a bijection),
– $\sigma(i, x) \sim \sigma(j, y)$ (by definition of $\models$),
– $lab(\langle x, i\rangle) \sim lab(\langle y, j\rangle)$ (by definition of lab).

Hence, $\langle x, i\rangle \xrightarrow{\sim} \langle y, j\rangle$ and $i \leq j$ implies $lab(\langle x, i\rangle) \sim lab(\langle y, j\rangle)$ Satisfaction of periodicity constraints is based on the implications between the propositions below:

– $mod(\langle x, i\rangle) = c$,
– $x \equiv_K c \in \rho_f(i)$ (by definition of G_ρ),
– $\sigma, i \models x \equiv_K c$ (arguments as above),
– $\sigma(i, x) \equiv_K c$ (by definition of $\models$),
– $lab(\langle x, i\rangle) \equiv_K c$ (by definition of lab).

Conversely, let lab be a strict edge-respecting labeling of G_ρ. We show that the model σ defined by $\sigma(i, x) = lab(\langle x, i\rangle)$ for all $x \in V$ and $i \in \mathbb{N}$ satisfies ρ. By way of example, we show that $\mathsf{X}^j x \sim \mathsf{X}^k y \in \rho_f(i)$ and $j \leq k$ implies $\sigma, i \models \mathsf{X}^j x \sim \mathsf{X}^k y$. We have implications between the propositions below:

– $\mathsf{X}^j x \sim \mathsf{X}^k y \in \rho_f(i)$ and $j \leq k$,
– $\langle x, i+j\rangle \xrightarrow{\sim} \langle y, i+k\rangle$ in G_ρ (by definition of G_ρ),
– $lab(\langle x, i+j\rangle) \sim lab(\langle y, i+k\rangle)$ (lab is edge-respecting),
– $\sigma(i+j)(x) \sim \sigma(i+k)(y)$ (by definition of σ),
– $\sigma, i \models \mathsf{X}^j x \sim \mathsf{X}^k y$ (by definition of $\models$).

□

A refinement is possible.

Lemma 8. *A one-step consistent sequence ρ has a model iff G_ρ has an edge-respecting labeling (not necessarily strict).*

Proof. By Lemma 7, if ρ has a model then G_ρ has a strict edge-respecting labeling.

Conversely, let lab be an edge-respecting labeling of G_ρ. We build from it, lab', a strict edge-respecting labeling of G_ρ. The values greater than M are divided in consecutive blocks of K consecutive values such that if $lab(n) - lab(\langle M, \mathrm{lev}(n)\rangle) = \beta \geq 0$ then $lab'(n)$ takes its value in the βth block. Then the constraint $mod(n)$ insures the unicity of $lab'(n)$ such that $lab'(n) \equiv_K mod(n)$. A similar division is performed for the values smaller than m.

- For every $\langle x, i\rangle$ such that $\langle x, i\rangle \xrightarrow{\sim} \langle m, i\rangle$, then $lab'(\langle x, i\rangle) \stackrel{\text{def}}{=} \alpha$ with
  - $\alpha \equiv_K mod(\langle x, i\rangle)$,
  - $m - (lab(\langle m, i\rangle) - lab(\langle x, i\rangle)) \times K \leq \alpha \leq m - ((lab(\langle m, i\rangle) - lab(\langle x, i\rangle) - 1) \times K - 1$.
- For every $\langle x, i\rangle$ such that $\langle M, i\rangle \xrightarrow{\sim} \langle x, i\rangle$, then $lab'(\langle x, i\rangle) \stackrel{\text{def}}{=} \alpha$ with
  - $\alpha \equiv_K mod(\langle x, i\rangle)$,
  - $M + ((lab(\langle x, i\rangle) - lab(\langle M, i\rangle) - 1) \times K + 1 \leq \alpha \leq M + (lab(\langle x, i\rangle) - lab(\langle M, i\rangle)) \times K$.

  In both above cases, α is unique since it belongs to an interval of length K with a periodicity constraint that forces a unique value in this interval.
- For every $\langle x, i\rangle$ such that $\langle x, i\rangle \xrightarrow{=} \langle d, i\rangle$ for some $d \in C'$, $lab'(\langle x, i\rangle) = d$.
- For every $\langle d, i\rangle$, $lab'(\langle d, i\rangle) = d$.

lab' is well-defined because ρ is a sequence of satisfiable symbolic valuations wrt ϕ. Moreover, lab' is a strict edge-respecting labeling. By way of example, suppose that $n \xrightarrow{\leq} n'$, $\mathrm{lev}(n) \leq \mathrm{lev}(n')$, $\langle M, \mathrm{lev}(n)\rangle \xrightarrow{\leq} n$ and $\langle M, \mathrm{lev}(n')\rangle \xrightarrow{\leq} n'$. Since lab is edge-respecting, $lab(\langle M, \mathrm{lev}(n)\rangle) < lab(n) < lab(n')$. Since the values of the $(lab(n) - lab(\langle M, \mathrm{lev}(n)\rangle)$th block after M are greater than the values of $(lab(n') - lab(\langle M, \mathrm{lev}(n)\rangle)$th block, then $lab'(n) < lab'(n')$. By Lemma 7, ρ has a model. $\Box$

Lemmas 7 and 8 state correspondences between ρ and its graphical representation G_ρ. However, we need a more abstract characterization of the one-step consistent sequences admitting a model (see Lemma 9 below).

Lemma 9. *Let ρ be a one-step consistent sequence. The graph G_ρ has an edge-respecting labeling iff for all nodes n_1, n_2 in G_ρ, $slen(n_1, n_2) < \omega$.*

By construction of G_ρ, for all nodes $\langle d_1, i\rangle$ and $\langle d_2, j\rangle$ representing constants such that $d_1 \leq d_2$, $slen(\langle d_1, i\rangle, \langle d_2, j\rangle) = d_2 - d_1$ (see Corollary 1). That is why, in Lemma 9, there is no additional constraint for nodes of the graph representing constants.

Proof. If G_ρ has an edge-respecting labeling lab, then one can show that for all nodes n_1, n_2 in G_ρ such that $slen(n_1, n_2) \neq -\infty$, $slen(n_1, n_2) \leq lab(n_2) - lab(n_1)$.

Conversely, if for all nodes n_1, n_2 in G_ρ, $slen(n_1, n_2) < \omega$, we define the following map $lab : (V \cup C') \times \mathbb{N} \to \mathbb{Z}$:

- $lab(\langle d, i\rangle) \stackrel{\text{def}}{=} d$.
- If $\langle x, i\rangle \xrightarrow{=} \langle d, i\rangle$ then $lab(\langle x, i\rangle) \stackrel{\text{def}}{=} d$.
- Otherwise,
  - If $\langle x, i\rangle \xrightarrow{<} \langle m, i\rangle$ then $lab(\langle x, i\rangle) \stackrel{\text{def}}{=} \alpha$ with
    - $\alpha \equiv_K mod(\langle x, i\rangle)$,
    - $m - slen(\langle x, i\rangle, \langle m, i\rangle) \times K \leq \alpha \leq m - (slen(\langle x, i\rangle, \langle m, i\rangle) - 1) \times K - 1$.
  - If $\langle M, i\rangle \xrightarrow{<} \langle x, i\rangle$ then $lab(\langle x, i\rangle) \stackrel{\text{def}}{=} \alpha$ with
    - $\alpha \equiv_K mod(\langle x, i\rangle)$,
    - $M + (slen(\langle M, i\rangle, \langle x, i\rangle) - 1) \times K + 1 \leq \alpha \leq M + slen(\langle M, i\rangle, \langle x, i\rangle) \times K$.

Similarly to the proof of Lemma 8, α is unique in both above cases since it belongs to an interval of length K and the periodicity constraint forces a unique value in this interval. We now show that lab is a strict edge-respecting labeling of G_ρ. If the labeling is not edge-respecting, then one of the following cases arises:

- Suppose $n \xrightarrow{=} n'$. We treat the case $\text{lev}(n) < \text{lev}(n')$ (the symmetrical case has an analogous treatment).

  Case 1: $\langle M, \text{lev}(n)\rangle \xrightarrow{<} n$ and $\langle M, \text{lev}(n')\rangle \xrightarrow{<} n'$.
  Since $n \xrightarrow{=} n'$ and $\langle M, \text{lev}(n)\rangle \xrightarrow{=} \langle M, \text{lev}(n')\rangle$, we have

  $$slen(\langle M, \text{lev}(n)\rangle, n) = slen(\langle M, \text{lev}(n')\rangle, n').$$

  Hence $lab(n)$ and $lab(n')$ belong to the same block of size K after M. Moreover, $\rho(\text{lev}(n))$ is satisfiable which entails $mod(n) = mod(n')$ and $lab(n) = lab(n')$.

  Case 2: $n \xrightarrow{<} \langle m, \text{lev}(n)\rangle$ and $n' \xrightarrow{<} \langle m, \text{lev}(n')\rangle$.
  Similar to Case 1.

  Case 3: $\langle M, \text{lev}(n)\rangle \xrightarrow{<} n$ and $n' \xrightarrow{\sim} \langle M, \text{lev}(n')\rangle$.
  Since $\rho(\text{lev}(n))$ is satisfiable, by (MC2), $n \xrightarrow{\sim} \langle M, \text{lev}(n')\rangle$.

  $$\langle M, \text{lev}(n)\rangle \xrightarrow{<} n \xrightarrow{\sim} \langle M, \text{lev}(n')\rangle \xrightarrow{=} \langle M, \text{lev}(n)\rangle,$$

  which leads to a contradiction by (MC3).

  Case 4: $n \xrightarrow{<} \langle m, \text{lev}(n)\rangle$ and $n' \xrightarrow{\sim} \langle m, \text{lev}(n')\rangle$.
  Similar to Case 3.

  Case 5: $n = \langle d, \cdot\rangle$ and $n' = \langle d', \cdot\rangle$ with $d < d'$.
  Since $\rho(\text{lev}(n))$ is satisfiable and by (MC2),(MC3) $d = d'$, which leads to a contradiction.
- The case $n \xrightarrow{<} n'$ is treated in a similar fashion.

□

Lemma 9 characterizes the set of sequences having a model but what we really need is to recognize them with automata. The main difficulty rests on the fact that the set of satisfiable one-step consistent ω-sequences of satisfiable symbolic valuations is not ω-regular, a consequence of [DD03] for the fragment CLTL(Z). In order to approximate this class of sequences, we define below a condition $(\mathcal{C})$ shown to be ω-regular such that for every one-step consistent ω-sequence ρ of satisfiable symbolic valuations that is ultimately periodic, ρ has a model iff G_ρ satisfies $(\mathcal{C})$.

An infinite forward (resp. backward) path in G_ρ is defined as a sequence $w : \mathbb{N} \to (V \cup C') \times \mathbb{N}$ such that:

- for every $i \in \mathbb{N}$, there is an edge $w(i) \xrightarrow{\sim} w(i+1)$ (resp. $w(i+1) \xrightarrow{\sim} w(i)$) in G_ρ,
- for every $i \in \mathbb{N}$, if $\mathrm{lev}(w(i)) = j$, then $\mathrm{lev}(w(i+1)) \geq j+1$.

The path w is infinitely often strict $\stackrel{\text{def}}{\Leftrightarrow}$ for every $i \geq 0$, there is $j \geq i$ such that $w(j) \xrightarrow{<} w(j+1)$ (resp. $w(j+1) \xrightarrow{<} w(j)$). The condition $(\mathcal{C})$ on the graph G_ρ is: there *do not* exist vertices n_1 and n_2 in G_ρ with $|\mathrm{lev}(n_1) - \mathrm{lev}(n_2)| \leq l$ satisfying

(AP1) there is an infinite forward path w_{for} from n_1,
(AP2) there is an infinite backward path w_{back} from n_2,
(AP3) either w_{for} or w_{back} is infinitely often strict, and
(AP4) for all $i, j \in \mathbb{N}$, whenever $|\mathrm{lev}(w_{\mathrm{for}}(i)) - \mathrm{lev}(w_{\mathrm{back}}(j))| \leq l$, $w_{\mathrm{for}}(i) \xrightarrow{<} w_{\mathrm{back}}(j)$ in G_ρ.

If ρ admits a model, then necessarily G_ρ satisfies $(\mathcal{C})$. Indeed, if G_ρ does not satisfy $(\mathcal{C})$, then $slen(n_1, n_2) = \omega$ which entails that ρ has no model by Lemma 9. The converse does not hold in general. However, when ρ is ultimately periodic, the condition $(\mathcal{C})$ is indeed sufficient. We say an infinite word is ultimately periodic if it is of the form $\tau \cdot \delta^\omega$ for some finite words τ and δ.

Lemma 10. *Let ρ be one-step consistent ω-sequence of satisfiable symbolic valuations that is ultimately periodic. Then ρ admits a model iff G_ρ satisfies $(\mathcal{C})$.*

Thanks to the way G_ρ is built from ρ, $(\mathcal{C})$ does not explicitly mention the constants in C' and the constraints of the form $x \equiv_K c$. Hence, Lemma 10 can be proved as [DD03, Lemma 6.2]: the map *mod* in G_ρ is ignored and a uniform treatment for all nodes in $(V \cup C') \times \mathbb{N}$ is provided. In [DD03, Lemma 6.2], there are no nodes of the form $C' \times \mathbb{N}$ but we take into account their specificity in our construction of G_ρ. If ρ admits a model then by Lemma 9 it satisfies the condition $(\mathcal{C})$. Conversely, let $\rho = \tau \cdot \delta^\omega$ be an ultimately periodic one-step consistent ω-sequence. We can show that if ρ has no model then it does not satisfy $(\mathcal{C})$. By Lemma 9, if ρ has no model, then there exist two vertices n_1 and n_2 such that $slen(n_1, n_2) = \omega$. One can construct a finite path w between n_1 and n_2 long enough so that paths w_{for} and w_{back} satisfying the conditions (AP1)–(AP4) can be constructed, witnessing that G_ρ does not satisfy $(\mathcal{C})$. The construction of w_{for} and w_{back} from w uses the fact that ρ is ultimately periodic by repeating infinitely finite subpaths. The construction of such infinite paths can be done smoothly by using the properties established in this section (see e.g. Lemma 6). As the proof is not essentially different from [DD03, Lemma 6.2] modulo slight changes mentioned above, we omit it here.

5 Büchi automata and PSPACE upper bound

Based on the previous results and following the approach in [VW94], we show that given a CLTL(IPC*) formula ϕ, one can build a standard Büchi automaton $\mathcal{A}_\phi$ such that ϕ is CLTL(IPC*) satisfiable iff $\mathrm{L}(\mathcal{A}_\phi)$ is non-empty. Moreover, we establish that emptiness of $\mathrm{L}(\mathcal{A}_\phi)$ can be checked in polynomial space in $|\phi|$. From the technical viewpoint, the construction of $\mathcal{A}_\phi$ as the intersection of three Büchi automata can be done quite smoothly thanks to the previous results. In the following, V, V', C and C' are the sets of variables and constants associated to ϕ as defined in Sect. 4. Moreover, K, m and M are constants with their usual meaning and we use the map $f : \mathrm{Terms}(\phi) \rightarrow V'$ as previously.

Unlike LTL, the language recognized by the Büchi automaton $\mathcal{A}_\phi$ is not a set of models but rather a set of symbolic models. We write Σ to denote the set of satisfiable symbolic valuations wrt ϕ. A symbolic model for ϕ is an ω-sequence $\rho : \mathbb{N} \rightarrow \Sigma$. We write $\rho \models' \phi$ where the symbolic satisfaction relation $\models'$ is defined as $\models$ except at the atomic level: $\rho, i \models' p \stackrel{\text{def}}{\Leftrightarrow} \rho(i) \models_{\text{symb}} f(p)$ where $\models_{\text{symb}}$ is the satisfaction relation between symbolic valuations and constraints.

By Lemma 5 and by using standard techniques for LTL [VW94], checking whether there is a symbolic model ρ satisfying $\rho \models' \phi$ can be done in PSPACE (see more details below). Since every model for ϕ generates a unique symbolic model for ϕ, we obtain the result below.

Lemma 11. *A* CLTL(IPC*) *formula ϕ is satisfiable iff there is a one-step consistent symbolic valuation ρ such that $\rho \models' \phi$ and ρ has a model.*

In order to prove the above lemma one can use the basic fact below:

Lemma 12. *Let ϕ be a* CLTL(IPC*) *formula. Let σ be a model and ρ a a one-step consistent symbolic valuation such that σ is a model for ρ. Then $\sigma \models \phi$ iff $\rho \models_{\text{symb}} \phi$.*

All the following automata are built over the alphabet Σ which is of exponential size in $|\phi|$. The automaton $\mathcal{A}_\phi$ is formally defined as the intersection $\mathcal{A}_{\text{LTL}} \cap \mathcal{A}_{1\text{cons}} \cap \mathcal{A}_{\mathcal{C}}$ of Büchi automata where

- $\mathrm{L}(\mathcal{A}_{\text{LTL}})$ is the set of symbolic models satisfying ϕ,
- $\mathrm{L}(\mathcal{A}_{1\text{cons}})$ is the set of one-step consistent sequences of satisfiable symbolic valuations,
- $\mathrm{L}(\mathcal{A}_{\mathcal{C}})$ is the set of sequences of symbolic valuations verifying $(\mathcal{C})$.

We briefly explain below how these automata are built. The automaton $\mathcal{A}_{\text{LTL}}$ is obtained from [VW94] with a difference for atomic formulae. We define $cl(\phi)$ the closure of ϕ as usual, and an atom of ϕ is a maximally consistent subset of $cl(\phi)$. We define $\mathcal{A}_{\text{LTL}} = (Q, Q_0, \rightarrow, F)$ as the generalized Büchi automaton below:

- Q is the set of atoms of ϕ and $Q_0 = \{X \in Q : \phi \in X\}$,
- $X \stackrel{sv}{\rightarrow} Y$ iff

(atomic constraints) for every atomic formula p in X, $sv \models_{\rm symb} f(p)$,
(one step) for every $\mathsf{X}\psi \in cl(\phi)$, $\mathsf{X}\psi \in X$ iff $\psi \in Y$,

– let $\{\psi_1 \mathsf{U} \varphi_1, \ldots, \psi_r \mathsf{U} \varphi_r\}$ be the set of until formulas in $cl(\phi)$. We pose F equal to $\{F_1, \ldots, F_r\}$ with for every $i \in \{1, \ldots, r\}$, $F_i = \{X \in Q : \psi_i \mathsf{U} \varphi_i \notin X \text{or } \varphi_i \in X\}$.

By Lemma 5, the condition about atomic formulae can be checked in PSPACE. Hence, the transition relation can be computed in PSPACE.

We define $\mathcal{A}_{1\rm cons} = \langle Q, Q_0, \rightarrow, F\rangle$ as a Büchi automaton such that $Q = Q_0 = F = Q = \Sigma$ and the transition relation satisfies: $sv \xrightarrow{sv''} sv' \overset{\rm def}{\Leftrightarrow} \langle sv, sv'\rangle$ is one-step consistent and $sv' = sv''$. Since checking whether a symbolic valuation is satisfiable can be done in P (Lemma 3) and checking whether a pair of symbolic valuations is one-step consistent can be also done in P, the transition relation of $\mathcal{A}_{1\rm cons}$ can be computed in P.

It remains to define $\mathcal{A}_{\mathcal{C}}$ that recognizes ω-sequences of symbolic valuations satisfying $(\mathcal{C})$. As done in [DD03], instead of building $\mathcal{A}_{\mathcal{C}}$, it is easier to construct the Büchi automaton $\mathcal{A}_{\mathcal{C}}^{\neg}$ that recognizes the complement language of $\mathrm{L}(\mathcal{A}_{\mathcal{C}})$. The automaton $\mathcal{A}_{\mathcal{C}}^{\neg}$ is essentially the automaton $\mathcal{B}$ defined in [DD03, page 20] except that we work with a different type of alphabet. We need to consider vertices in the graph that represent constants in C and equality between constants does not need to be explicitly present in the symbolic valuations. The variables in V and the constants in C have a uniform treatment in the definition of $\mathcal{A}_{\mathcal{C}}^{\neg}$ except that equality between constants does not need to be explicitly present in the symbolic valuations.

The automaton $\mathcal{A}_{\mathcal{C}}^{\neg}$ non-deterministically guesses the vertices n_1, n_2 and which path among $w_{\rm for}$ and $w_{\rm back}$ is infinitely often strict. Then it checks that the sequence fails to meet $(\mathcal{C})$. The Büchi acceptance condition guarantees that $\xrightarrow{<}$-edges are infinitely often visited. The automaton $\mathcal{A}_{\mathcal{C}}^{\neg} = \langle Q, Q_0, \rightarrow, F\rangle$ is defined as follows:

– $Q = \{q_0\} \cup \{(V \cup C) \times \{0, \ldots, l\} \times (V \cup C) \times \{0, \ldots, l\} \times \{\mathit{for}, \mathit{back}\} \times \{0, 1\}\}$ with $l = |\phi|_{\mathsf{X}}$,
– $Q_0 = \{q_0\}$,
– The transition relation $\rightarrow$ is defined as follows.
  - $q_0 \xrightarrow{sv} q_0$ for every $sv \in \Sigma$.
  - $q_0 \xrightarrow{sv} \langle a, i, b, j, \mathit{for}, 0\rangle$ and $q_0 \xrightarrow{sv} \langle a, i, b, j, \mathit{back}, 0\rangle$ for all $a, b \in V \cup C$, $i, j \in \{0, \ldots, l\}$ and $sv \in \Sigma$.
  - $\langle a, i, b, j, p, \mathit{bin}\rangle \xrightarrow{sv} \langle a, i-1, b, j-1, p, \mathit{bin}\rangle$ for all $p \in \{\mathit{for}, \mathit{back}\}$, $\mathit{bin} \in \{0, 1\}$, $i, j \geq 2$ and $sv \in \Sigma$.
  - $\langle a, 1, b, j, \mathit{for}, \mathit{bin}\rangle \xrightarrow{sv} \langle a', i, b, j-1, \mathit{for}, \mathit{bin}'\rangle$ with $a' \notin C$ such that
    * $j > 1$,
    * $f(a < \mathsf{X}^i a') \in sv$ and $\mathit{bin}' = 1$ or $f(a = \mathsf{X}^i a') \in sv$ and $\mathit{bin}' = 0$,
    * $f(\mathsf{X}^i a' < \mathsf{X}^{j-1} b) \in sv$.
  - $\langle a, 1, b, j, \mathit{for}, \mathit{bin}\rangle \xrightarrow{sv} \langle a', i, b, j-1, \mathit{for}, \mathit{bin}'\rangle$ with $a' \in C$ such that $j > 1$, $a = a'$ and $\mathit{bin}' = 0$.
  - similarly for $\langle b, 1\rangle$, and for $\langle a, 1\rangle$ and $\langle b, 1\rangle$ simultaneously.

- • similar clauses for w_{back}.
- F is the union of all states of the form $\langle a, i, b, j, p, 1\rangle$.

It is worth observing that we do not need to consider the set of constants C' instead of C.

Lemma 13. *A* CLTL(IPC*) *formula* ϕ *is satisfiable iff* $\mathrm{L}(\mathcal{A}_\phi)$ *is non-empty.*

The proof of this lemma is similar to [DD03, Lemma 6.3]. The main trick is to observe that if $\mathrm{L}(\mathcal{A}_\phi)$ is non-empty then $\mathcal{A}_\phi$ accepts an ultimately periodic ω-sequence so that Lemma 10 can be applied. Since given a formula ϕ we can effectively construct $\mathcal{A}_\phi$ and check whether $\mathrm{L}(\mathcal{A}_\phi)$ is empty, the model-checking and satisfiability problems for CLTL(IPC*) are decidable. We also have all the arguments to establish the PSPACE upper bound by using subtle arguments from complexity theory and [Saf89].

Theorem 1. *The satisfiability problem for* CLTL(IPC*) *is* PSPACE*-complete.*

PSPACE-hardness is a consequence of the PSPACE-hardness of LTL [SC85]. As far as the PSPACE upper bound is concerned, the automata $\mathcal{A}_{\mathrm{LTL}}$, $\mathcal{A}_{1\mathrm{cons}}$ and $\mathcal{A}_{\mathcal{C}}^{\neg}$ are of exponential size in $|\phi|$ and can be built in polynomial space in $|\phi|$. The size of $\mathcal{A}_{\mathcal{C}}^{\neg}$ is even polynomial in $|\phi|$ and $\mathcal{A}_{\mathcal{C}}^{\neg}$ can be built in logarithmic space in $|\phi|$. In particular, their respective transition relations can be checked in polynomial space. Following [Saf89], building $\mathcal{A}^{\neg}$ from $\mathcal{A}$ such that $\mathrm{L}(\mathcal{A}^{\neg}) = \Sigma^\omega \setminus \mathrm{L}(\mathcal{A})$ can be done in polynomial space in $|\mathcal{A}|$ (but the result automaton is of exponential size). By [BDG88, Lemma 3.35], the composition of a polynomial space function with a logarithmic space function is a polynomial space function. Hence, $\mathcal{A}_{\mathcal{C}}$ can be built in polynomial space in ϕ. So, computing the intersection automaton $\mathcal{A}_\phi = \mathcal{A}_{\mathrm{LTL}} \cap \mathcal{A}_{1\mathrm{cons}} \cap \mathcal{A}_{\mathcal{C}}$ can be done in polynomial space in $|\phi|$. Since the emptiness problem for Büchi automata is NLOGSPACE-complete, by [BDG88, Corollary 3.36], we finally get that testing emptiness of $\mathrm{L}(\mathcal{A}_\phi)$ can be done non-deterministically in polynomial space in $|\phi|$. As usual, by Savitch's theorem we get the PSPACE upper bound. $\square$

All the temporal operators in CLTL(IPC*) are definable in monadic second order logic (MSO) and by using [GK03], it is immediate that any extension of CLTL(IPC*) obtained by adding a finite amount of MSO-definable temporal operators remains in PSPACE. Only the automaton $\mathcal{A}_{\mathrm{LTL}}$ needs to be updated.

Corollary 2. *The model-checking problem for integral relational automata restricted to the LTL fragment of CCTL* introduced in [Čer94] is in* PSPACE.

6 Conclusion

In the paper, we have introduced the logic CLTL(IPC*) extending formalisms in [Čer94,LM01,DD03,Dem04] and we have shown that both model-checking over IPC*-automata and satisfiability are decidable in polynomial space. The proof heavily relies on a translation into the emptiness problem for standard Büchi automata and on the approximation of non ω-regular sets of symbolic models. As a by-product, the model checking problem over the integral relational automata defined in [Čer94] is also PSPACE-complete when restricted to its LTL fragment. The logic CLTL(IPC*) supports a rich class of constraints including those of the form $x < y$ unlike periodicity constraints from [Dem04] (which are quite useful to compare absolute dates) and comparison with constants unlike logics shown in PSPACE in [DD03]. Abstraction of counter automata by performing reasoning modulo can be encoded in CLTL(IPC*) thanks to the presence of integer periodicity constraints.

To conclude, we mention a few open problems that are worth investigating.

- CTL* for integral relational automata is undecidable [Čer94] whereas we have shown that its LTL fragment is PSPACE-complete. It is interesting to design other decidable fragments of CTL* strictly more expressive than Boolean combinations of LTL formulae.
- The decidability status of constraint LTL over the domain $\langle \{0,1\}^*, \subseteq \rangle$ is open either with the subword relation or with the prefix relation. Constraint LTL over the domain $\langle \{0\}^*, \subseteq \rangle$ is already equivalent to constraint LTL over $\langle \mathbb{N}, <, = \rangle$ that is a strict fragment of CLTL(IPC*).
- The decidability status of CLTL(IPC*) extended with constraints of the form $3x + 2\mathsf{X}y \equiv_5 3$ is open. They are considered in [MOS05] but not integrated in any LTL-like language.

References

[AD94] R. Alur and D. Dill. A theory of timed automata. *TCS*, 126:183–235, 1994.

[AH94] R. Alur and Th. Henzinger. A really temporal logic. *JACM*, 41(1):181–204, 1994.

[BBFS98] E. Bertino, C. Bettini, E. Ferrari, and P. Samarati. An access control model supporting periodicity constraints and temporal reasoning. *ACM Transactions on Databases Systems*, 23(3):231–285, 1998.

[BC02] Ph. Balbiani and J.F. Condotta. Computational complexity of propositional linear temporal logics based on qualitative spatial or temporal reasoning. In *FroCoS'02*, volume 2309 of *LNAI*, pages 162–173. Springer, 2002.

[BDG88] J. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I*. Springer, Berlin, 2nd edition, 1988.

[BEH95] A. Bouajjani, R. Echahed, and P. Habermehl. On the verification problem of nonregular properties for nonregular processes. In *LICS'95*, pages 123–133, 1995.

[BEM97] A. Bouajjani, J. Esparza, and O. Maler. Reachability analysis of pushdown automata: application to model-checking. In *CONCUR'97*, volume 1243 of *LNCS*, pages 135–150. Springer, 1997.

[Bér95] B. Bérard. Untiming timed languages. *IPL*, 55:129–135, 1995.

[BFLP03] S. Bardin, A. Finkel, J. Leroux, and L. Petrucci. FAST: Fast Acceleration of Symbolic Transition systems. In *CAV'03*, volume 2725 of *LNCS*, pages 118–121. Springer, 2003.

[BH91] F. Baader and P. Hanschke. A scheme for integrating concrete domains into concept languages. In *IJCAI'91*, pages 452–457, 1991.

[Boi98] B. Boigelot. *Symbolic methods for exploring infinite state spaces.* PhD thesis, Université de Liège, 1998.

[Cau03] D. Caucal. On infinite transition graphs having a decidable monadic theory. *TCS*, 290:79–115, 2003.

[CC00] H. Comon and V. Cortier. Flatness is not a weakness. In *CSL'00*, volume 1862 of *LNCS*, pages 262–276. Springer, 2000.

[Čer94] K. Čerāns. Deciding properties of integral relational automata. In *ICALP*, volume 820 of *LNCS*, pages 35–46. Springer, 1994.

[CGL94] E. Clarke, O. Grumberg, and D. Long. Model checking and abstraction. *ACM Transactions on Programming Languages and Systems*, 16(5):1512–1542, 1994.

[CJ98] H. Comon and Y. Jurski. Multiple counters automata, safety analysis and Presburger arithmetic. In *CAV'98*, volume 1427 of *LNCS*, pages 268–279. Springer, 1998.

[DD03] S. Demri and D. D'Souza. An automata-theoretic approach to constraint LTL. Technical Report LSV-03-11, LSV, August 2003. 40 pages. An extended abstract appeared in Proc. of FSTTCS'02.

[Dem04] S. Demri. LTL over integer periodicity constraints. Technical Report LSV-04-6, LSV, February 2004. 35 pages. An extended abstract appeared in Proc. of FOSSACS'04.

[EFM99] J. Esparza, A. Finkel, and R. Mayr. On the verification of broadcast protocols. In *LICS'99*, pages 352–359, 1999.

[FL02] A. Finkel and J. Leroux. How to compose Presburger accelerations: Applications to broadcast protocols. In *FST&TCS'02*, volume 2256 of *LNCS*, pages 145–156. Springer, 2002.

[FS00] A. Finkel and G. Sutre. Decidability of reachability problems for classes of two counters automata. In *STACS'00*, volume 2256 of *LNCS*, pages 346–357. Springer, 2000.

[GK03] P. Gastin and D. Kuske. Satisfiability and model checking for MSO-definable temporal logics are in PSPACE. In *CONCUR'03*, volume 2761 of *LNCS*, pages 222–236. Springer, 2003.

[GKK+03] D. Gabelaia, R. Kontchakov, A. Kurucz, F. Wolter, and M. Zakharyaschev. On the computational complexity of spatio-temporal logics. In *FLAIRS'03*, pages 460–464, 2003.

[HLP90] E. Harel, O. Lichtenstein, and A. Pnueli. Explicit clock temporal logic. In *Proc. 5th IEEE Symp. Logic in Computer Science (LICS '90), Philadelphia, PA, USA, June 1990*, pages 400–413. IEEE Computer Society Press, 1990.

[HMR05] Th. Henzinger, R. Majumdar, and J.F. Raskin. A classification of symbolic transitions systems. *ACM Transactions on Computational Logic*, 6(1), 2005. to appear.

[Iba78] O. Ibarra. Reversal-bounded multicounter machines and their decision problems. *JACM*, 25(1):116–133, 1978.

[ID01] O. Ibarra and Z. Dang. On removing the stack from reachability constructions. In *ISAAC'01*, volume 2223 of *LNCS*, pages 244–256. Springer, 2001.

[JKMS04] P. Jančar, A. Kučera, F. Moller, and Z. Sawa. DP lower bounds for equivalence-checking and model-checking of one-counter automata. *I & C*, (188):1–19, 2004.

[LM01] U. Dal Lago and A. Montanari. Calendars, time granularities, and automata. In *Int. Symposium on Spatial and Temporal Databases*, volume 2121 of *LNCS*, pages 279–298. Springer, Berlin, 2001.

[LS01] G. Logothetis and K. Schneider. Abstraction from counters: an application on real-time systems. In *TIME'01*, pages 214–223. IEEE, 2001.

[Lut04] C. Lutz. NEXPTIME-complete description logics with concrete domains. *ACM Transactions on Computational Logic*, 5(4):669–705, 2004.

[Min67] M. Minsky. *Computation, Finite and Infinite Machines*. Prentice Hall, 1967.

[MOS05] M. Müller-Olm and H. Seidl. Analysis of modular arithmetic. In *ESOP'05*, LNCS. Springer, 2005.

[MS85] D. Muller and P. Schupp. The theory of ends, pushdown automata , and second-order logic. *TCS*, 37:51–75, 1985.

[Rev02] P. Revesz. *Introduction to Constraint Databases*. Springer, New York, 2002.

[Saf89] S. Safra. *Complexity of Automata on Infinite Objects*. PhD thesis, The Weizmann Institute of Science, 1989.

[SC85] A. Sistla and E. Clarke. The complexity of propositional linear temporal logic. *JACM*, 32(3):733–749, 1985.

[VW94] M. Vardi and P. Wolper. Reasoning about infinite computations. *I & C*, 115:1–37, 1994.

[Wol83] P. Wolper. Temporal logic can be more expressive. *I & C*, 56:72–99, 1983.

[WZ00] F. Wolter and M. Zakharyaschev. Spatio-temporal representation and reasoning based on RCC-8. In *KR'00*, pages 3–14, 2000.

A Integral relational automata as restricted IPC*-automata

An integral relational automata (RA) is defined in [Čer94] as a program with a finite amount of variables interpreted in $\mathbb{Z}$. The set OP of operations is composed of the following instructions and guards:

- comparisons of the form $x < y$, $x < d$, $d < y$,
- assignments of the form $x \leftarrow y$, $x \leftarrow d$,
- input value $?x$,
- output value $!x$ or $!d$,
- dummy operation NOP,

where x, y are variables and d is a constant in $\mathbb{Z}$. A relational automaton $\mathcal{A} = \langle Q, \delta, op, g \rangle$ [Čer94] is a finite directed graph where

- Q is a finite set of control states,
- $\delta \subseteq Q \times Q$,
- $op : Q \to \mathrm{OP}$,
- $g : \delta \to \{+, -\}$.

Let $\mathit{Var}(\mathcal{A})$ and $\mathit{Cons}(\mathcal{A})$ be respectively the sets of variables and constants occuring in *op*. A configuration of $\mathcal{A}$ is a pair $\langle n, v \rangle$ such that $n \in Q$ and v is a map $v : \mathit{Var}(\mathcal{A}) \cup \mathit{Cons}(\mathcal{A}) \to \mathbb{Z}$ equals to the identity for its restriction to $\mathit{Cons}(\mathcal{A})$. The configuration graph of $\mathcal{A}$ is defined as the pair $\langle Q^{cg}, \to \rangle$ where Q^{cg} is the set of configurations of $\mathcal{A}$ and $\langle n, v \rangle \to \langle n', v' \rangle$ iff there exists $e \in \delta$ such that

1. $e = \langle n, n' \rangle$,
2. v and v' are related depending on the nature of $op(n)$:
   - if $op(n)$ is $?x$, then for every $y \in \mathit{Var}(\mathcal{A}) \setminus \{x\}$, $v'(y) = v(y)$,
   - if $op(n)$ is an output value or the dummy operation, then $v' = v$,
   - if $op(n)$ is $x \leftarrow a$, then $v' = v[x \leftarrow v(a)]$,
   - if $op(n)$ is $a < b$, then $v = v'$ and
     either $g(e) = +$ and $v(a) < v(b)$,
     or $g(e) = -$ and $v(a) \geq v(b)$,

   where $a, b \in \mathit{Var}(\mathcal{A}) \cup \mathit{Cons}(\mathcal{A})$, $v(x)$ denotes the value of x in v and $v[x \leftarrow a]$ is such that $v[x \leftarrow z](y) = z$ if $x = y$ and $v[x \leftarrow z](y) = v(y)$ otherwise.

Observe that equality between variables can be tested by performing two negative tests on $a < b$ and $b < a$.

From a relational automaton $\mathcal{A} = \langle Q, \delta, op, g \rangle$, we build a restricted IPC*-automaton $\mathcal{A}'$, having an isomorphic configuration graph in a sense to be precised below. Given $\mathcal{A} = \langle Q, \delta, op, g \rangle$, the restricted IPC*-automaton $\mathcal{A}' = \langle Q', Q'_0, \to, F' \rangle$ is defined as follows:

1. $Q' = Q'_0 = F' = Q$. Without any loss of generality, we can assume that Q is a finite set of $\mathbb{Z}$ constants not occurring in *op*.

2. To each $e = \langle n, n' \rangle$ in $\mathcal{A}$, we associate $n \xrightarrow{\phi_e} n'$ in $\mathcal{A}'$ where ϕ_e is a conjunction of constraints defined as follows:
   - $\mathsf{X}ic = n'$ is a conjunct of ϕ_e where ic is a new variable taking care of the instruction counter,
   - if $op(n) = ?x$ then $\bigwedge_{y \in \mathrm{VAR}(\mathcal{A}) \setminus \{x\}} y = \mathsf{X}y$ belongs to ϕ_e,
   - if $op(n)$ is an output or a dummy operation, then $\bigwedge_{y \in \mathrm{VAR}(\mathcal{A})} y = \mathsf{X}y$ belongs to ϕ_e,
   - if $op(n) = x \leftarrow a$ then $\bigwedge_{y \in \mathrm{VAR}(\mathcal{A}) \setminus \{x\}} y = \mathsf{X}y \wedge \mathsf{X}x = a$ belongs to ϕ_e,
   - if $op(n) = a < b$ then
     if $g(e) = +$ then $a < b \wedge \bigwedge_{y \in \mathrm{VAR}(\mathcal{A})} y = \mathsf{X}y$ belongs to ϕ_e,
     $g(e) = -$ then $a \geq b \wedge \bigwedge_{y \in \mathrm{VAR}(\mathcal{A})} y = \mathsf{X}y$ belongs to ϕ_e.

The configuration graphs of $\mathcal{A}$ and $\mathcal{A}'$ are isomorphic with the following property. The transition $\langle n, v \rangle \to \langle n', v' \rangle$ belongs to the configuration graph of $\mathcal{A}$ iff $\langle n, \overline{v} \rangle \to \langle n', \overline{v'} \rangle$ belongs to the configuration graph of $\mathcal{A}'$ where $\overline{u} : Var(\mathcal{A}) \cup Cons(\mathcal{A}) \cup \{ic\} \to \mathbb{Z}$ is a conservative extension of $u : Var(\mathcal{A}) \cup Cons(\mathcal{A}) \to \mathbb{Z}$ and $\overline{u}(ic) = n$. As a corollary, the LTL fragment of CLTL* defined in [Čer94] where the atomic formulae are of the form n, $x < y$ and $x = y$, can be reduced to the model-checking problem for restricted IPC*-automata (just replace n by $ic = n$ to obtain CLTL(IPC*) formulae).