Branching-time Temporal Logic Extended with Qualitative Presburger Constraints

Laura Bozzelli and Régis Gascon

LSV, CNRS & ENS Cachan, France
{bozzelli,gascon}@lsv.ens-cachan.fr

Abstract. Recently, *LTL* extended with atomic formulas built over a constraint language interpreting variables in $\mathbb{Z}$ has been shown to have a decidable satisfiability and model-checking problem. This language allows to compare the variables at different states of the model and include periodicity constraints, comparison constraints, and a restricted form of quantification. On the other hand, the *CTL* counterpart of this logic (and hence also its *CTL** counterpart which subsumes both *LTL* and *CTL*) has an undecidable model-checking problem. In this paper, we substantially extend the decidability border, by considering a meaningful fragment of *CTL** extended with such constraints (which subsumes both the universal and existential fragments, as well as the *EF*-like fragment) and show that satisfiability and model-checking over relational automata that are abstraction of counter machines are decidable. The correctness and the termination of our algorithm rely on a suitable well quasi-ordering defined over the set of variable valuations.

1 Introduction

Model-checking of infinite-state counter systems. The formal verification of infinite-state systems has benefited from numerous decidable model-checking problems. This is the case for instance of timed automata [AD94], or subclasses of counter systems, see e.g. [CJ98]. Counter systems are finite state machines operating on a finite set of variables (counters or registers) interpreted as integers. Though simple problems like reachability are already undecidable for 2-counter Minsky machines [Min67], many interesting restrictions of counter systems have been studied, for which reachability and richer temporal properties have been shown to be decidable. For instance, Petri nets represent the subclass of counter systems obtained by removing the ability to test a counter for zero. Other examples include reversal-bounded counter machines [Iba78], flat counter systems [Boi98,BFLP03,LS04] and constraint automata with qualitative constraints on $\mathbb{Z}$ between the states of variables at different steps of the execution [DG05]. "Qualitative" means that the relationship between the constrained variables is not sharp, like $x < y$. This last class of systems can be seen as an abstraction of counter systems where increments and decrements are abstracted by comparisons and congruence relations modulo some integer. For example, $x = y+1$ can be abstracted by $x > y \wedge x \equiv_{2^k} y+1$. This is very common in various programming languages performing arithmetic operations modulo some integer, typically modulo 2^{32} or 2^{64} (see [MOS05]). Periodicity constraints have also found applications in formalisms dealing with calendars [LM01] and temporal reasoning in database access control [BBFS98].

Temporal logics extended with Presburger constraints. Classical problems studied on counter systems often reduce to the reachability of some control state. Recently, richer temporal properties have been investigated and formalized by introducing fragments of Presburger constraints in temporal logics. In this setting, atomic formulas are Presburger arithmetic constraints over variables (counters) taking values in $\mathbb{Z}$. Furthermore, these formalisms involve an hybrid of temporal logic and constraints, with varying degrees of interaction. For instance, one may be allowed to refer to the value of a variable x on the next time instant, leading to constraints of the form $x > \mathsf{O}x$. More generously, one may be permitted to refer to a future value of a variable x a certain number n of steps further. We denote this value by $\mathsf{O} \ldots \mathsf{O}x$ where x is prefixed by n times the symbol O (in the following such an expression is abbreviated by $\mathsf{O}^n x$). For linear-time temporal logics, such extensions can be found in numerous works, see for instance [BEH95,CC00,DD03]. However, full Presburger *LTL* is undecidable, and to regain decidability, one can either restrict the underlying constraint language, see e.g. [DD03,DG05], or restrict the logical language, see e.g. [BEH95,CC00]. In [DG05], full *LTL* extended with a wide set of qualitative constraints, including comparison and periodicity constraints, has been shown to have PSPACE-complete satisfiability and model-checking problems (over constraint automata mentioned above). Similar extensions have also been considered for description logics where models are Kripke structures, see for instance [Lut04]. On the other hand, to the best of our knowledge, very few works deal with decidable fragments of branching-time temporal logics enhanced with Presburger constraints. Actually, we can only refer to the work [Čer93], in which *CTL** extended with only comparison constraints is shown to have an undecidable model checking problem for *Integral Relational Automata* (undecidability already holds for the *CTL*-like fragment). However, model-checking for the existential and universal fragments are shown to be decidable. Note that the logic proposed in [Čer93] does not exhibit any form of interaction between the temporal operators and the comparison constraints (in particular, atomic formulas of the form $x < \mathsf{O}y$ are not considered).

Our contribution. In this paper, we introduce the logic *CCTL** as an extension of the branching–time temporal logic *CTL** with a wide set of qualitative constraints including periodicity constraints of the form $x \equiv_k y + c$, comparison constraints of the form $x < y$ and a restricted form of quantification. This logic is the branching–time counterpart of the constraint *LTL* defined in [DG05] and extends the logic from [Čer93] by introducing richer constraints and the possibility to compare counters at different states of the model. The operational models on which we check temporal properties expressed in this logic are extensions of Integral Relational Automata (*IRA*, for short) [BBK77,Čer93,ACJT96] introduced in [BBK77] as a model for studying possibilities of automated complete test set generation for data processing programs. Our extension is obtained by adding periodicity constraints and makes the new formalism an equivalent variant of the constraint automata with qualitative constrains mentioned above. However, *IRA* provide a representation that is more intuitive and closer to the operational semantics of programs manipulating integers.

Model-checking this extension of *IRA* against full *CCTL** is undecidable (also for the *CTL*-like fragment) as a consequence of [Čer93]. Thus, in this paper we investigate a meaningful fragment, which subsume both the existential and universal fragments as well as the *EF*-like fragment. For instance, the formula $A\Box E\Box(x = \mathsf{O}x)$ is in this fragment and states that for any reachable state, there is a computation starting from it in which the value of counter x remains constant. For this fragment, we show that both satisfiability and model checking of the proposed extension of *IRA* are decidable. The existential and universal fragments of *CCTL** are strictly more expressive than the constraint *LTL* defined in [DG05]. Moreover, the symbolic algorithm we describe builds a finite representation of the set of states satisfying a given formula, a very substantial information compared to the symbolic representation used in [DG05].

IRA belong to the class of well–structured transition systems which have been intensively studied, see e.g. [ACJT96,FS01]. Hence, one can define a decidable well-quasi ordering on the set of states, which is also a simulation. This property is sufficient to guarantee decidability of simple problems such as coverability, but not to decide richer properties like liveness properties[1] which can be expressed in our logical framework. Thus, we need to use a more sophisticated approach, which is a technical non-trivial generalization and refinement of the one used in [Čer93] combining automata-based techniques, theory of well quasi-ordering, and the theory of a specific class of linear inequality systems (used to represent upward closed sets of states). The correctness and the termination of the algorithm rely on a suitable well quasi-ordering defined over these inequality systems. Another major contribution consists in extending to a larger framework the original and difficult proof from [Čer93] and in clarifying all the technical lemmas needed in the last part of the algorithm, which are omitted in [Čer93].

Due to lack of space, many proofs are omitted and can be found in [BG06].

2 Preliminaries

2.1 Language of constraints

Let VAR be a countable set of variables. For $D \subseteq \mathrm{VAR}$, a *valuation* over D is a map $v : D \rightarrow \mathbb{Z}$. For all $x \in D$, we denote by $v.x$ the value assigned to x in v.
The *language of constraints* p, denoted by *IPC** [DG05], is defined as follows: [2]

$$\begin{aligned} p &::= t \mid x \sim y \mid p \wedge p \mid \neg p \\ t &::= \; x \equiv_k [c_1, c_2] \mid x \equiv_k y + [c_1, c_2] \mid x = y \mid x \sim c \mid t \wedge t \mid \neg t \mid \exists x\, t \end{aligned}$$

where $\sim \in \{<, \leq, >, \geq, =\}$, $x, y \in \mathrm{VAR}$, $k \in \mathbb{N} \setminus \{0\}$, and $c_1, c_2, c \in \mathbb{Z}$. For a constraint p and a valuation v over VAR, the satisfaction relation $v \models p$ is defined as follows (we omit the standard clauses for negation, conjunction, and inequalities):

- $v \models x \equiv_k [c_1, c_2] \overset{\text{def}}{\Leftrightarrow} \exists\, c_1 \leq c \leq c_2$ and $m \in \mathbb{Z}.\; v.x = c + m \cdot k$;
- $v \models x \equiv_k y + [c_1, c_2] \overset{\text{def}}{\Leftrightarrow} \exists\, c_1 \leq c \leq c_2$ and $m \in \mathbb{Z}.\; v.x = v.y + c + m \cdot k$;
- $v \models \exists x\, t \overset{\text{def}}{\Leftrightarrow} \exists\, c \in \mathbb{Z}.\; v[x \leftarrow c] \models t$

[1] For instance, liveness properties in lossy channel systems are undecidable [AJ94].

[2] Note that constraints of the form $\exists x,\ x < y$ are not allowed since they leads to the undecidability already for the corresponding LTL extension (see [DG05]).

where $v[x \leftarrow c].x' = v.x'$ if $x \neq x'$ and $v[x \leftarrow c].x = c$. A constraint p is *atomic* if it has one of the following forms: $x \equiv_k c \mid x \sim y \mid x \sim c$, where $\sim \in \{<, \leq, >, \geq, =\}$ and $x \equiv_k c$ is an abbreviation for $x \equiv_k [c, c]$. Evidently, for a constraint p, whether a valuation v satisfies p depends only on the values of v over the finite set $Vars(p)$ of free variables occurring in p. Thus, in the following as interpretations of a constraint p we consider the set of valuations over finite supersets of $Vars(p)$.

Lemma 1 ([DG05]). *Any* IPC* *constraint can be effectively converted into an equivalent positive boolean combination of* atomic IPC* *constraints.*

The translation implies an exponential blowup of the size of the formula w.r.t the constants used. However, the results in the following do not refer to complexity issues.

2.2 The constrained branching-time temporal logic (*CCTL**)

We introduce the *constrained branching-time temporal logic* (*CCTL**) as an extension of the standard propositional logic *CTL** [EH86] where atomic propositions are replaced by *IPC** constraints between terms representing the value of variables at different sates of the model. We denote these atomic formulae by $p[x_1 \leftarrow \mathsf{O}^{i_1} x_{j_1}, \ldots, x_r \leftarrow \mathsf{O}^{i_r} x_{j_r}]$, where p is an *IPC** constraint with free variables $x_1, \ldots, x_r$ and we substitute each occurrence of variable x_l with $\mathsf{O}^{i_l} x_{j_l}$ (corresponding to variable x_{j_l} preceded by i_l "next" symbols). The expression $\mathsf{O}^i x$ represents the value of the variable x at the i^{th} next state. For example, $\mathsf{O}y \equiv_2 x + 1$ and $x < \mathsf{O}y$ are atomic formulae of *CCTL**.

As for standard *CTL**, there are two types of formulas in *CCTL**: *state formulas* ξ whose satisfaction is related to a specific state, and *path formulas* ψ, whose satisfaction is related to a specific path. Their syntax is inductively defined as follows:

$$\begin{aligned}
&\xi := \top \;\mid \xi \vee \xi \mid \xi \wedge \xi \mid A\, \psi \mid E\, \psi \\
&\psi := \xi \;\mid p[x_1 \leftarrow \mathsf{O}^{i_1} x_{j_1}, \ldots, x_r \leftarrow \mathsf{O}^{i_r} x_{j_r}] \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathsf{O}\psi \mid \Box\psi \mid \psi \mathsf{U} \psi
\end{aligned}$$

where $\top$ denotes "true", E ("for some path") and A ("for all paths") are path quantifiers, and O ("next"), U ("until"), and $\Box$ ("always") are the usual linear temporal operators.[3] The set of state formulas ξ forms the language *CCTL**. For a set X of state formulas, the set of path formulas ψ defined only from state formulas in X is denoted by $PLF(X)$.

For a *CCTL** formula ξ, let $Val(\xi)$ be the set of valuations over finite sets $D \subseteq \mathrm{VAR}$ such that D contains the variables occurring in ξ. The interpretations for the formula ξ are labelled graphs $\mathcal{G} = \langle S, \rightarrow, \mu \rangle$, where S is a (possible infinite) set of vertices (here, called states), $\rightarrow \subseteq S \times S$ is the edge relation, which is total (i.e., for every $s \in S$, $s \rightarrow s'$ for some $s' \in S$), and $\mu : S \rightarrow Val(\xi)$ maps each state $s \in S$ to a valuation in $Val(\xi)$. A path is a sequence of states $\pi = s_0, s_1, \ldots$ such that $s_{i-1} \rightarrow s_i$ for any $1 \leq i < |\pi|$. We denote the suffix $s_i, s_{i+1}, \ldots$ of π by π^i, and the i-th state of π by $\pi(i)$. Let $s \in S$ and π be a infinite path of $\mathcal{G}$. For a state (resp., path) formula ξ (resp. ψ), the satisfaction relation $(\mathcal{G}, s) \models \xi$ (resp., $(\mathcal{G}, \pi) \models \psi$), meaning that ξ (resp., ψ) holds at state s (resp., holds along π) in $\mathcal{G}$, is defined by induction. The clauses for conjunction and disjunction are standard. For the other clauses we have:

[3] We have defined a positive normal form of the logic *CCTL**, i.e. negation is used only in atomic formulae. Moreover, the given syntax is complete since the dual $\widetilde{\mathsf{U}}$ of the until operator can be expressed in terms of the until and always operator: $\psi_1 \widetilde{\mathsf{U}} \psi_2 \equiv \Box\psi_2 \vee \big(\psi_2 \mathsf{U} (\psi_1 \wedge \psi_2)\big)$.

- $(\mathcal{G}, s) \models A\ \psi \overset{\text{def}}{\Leftrightarrow}$ for each infinite path π from s, $(\mathcal{G}, \pi) \models \psi$;
- $(\mathcal{G}, s) \models E\ \psi \overset{\text{def}}{\Leftrightarrow}$ there exists an infinite path π from s such that $(\mathcal{G}, \pi) \models \psi$;
- $(\mathcal{G}, \pi) \models \xi \overset{\text{def}}{\Leftrightarrow} (\mathcal{G}, \pi(0)) \models \xi$;
- $(\mathcal{G}, \pi) \models p[x_1 \leftarrow \mathsf{O}^{i_1} x_{j_1}, \ldots, x_r \leftarrow \mathsf{O}^{i_r} x_{j_r}] \overset{\text{def}}{\Leftrightarrow}$
  $\mu(\pi(0))[x_1 \leftarrow \mu(\pi(i_1)).x_{j_1}, \ldots, x_r \leftarrow \mu(\pi(i_r)).x_{j_r}] \models p$;
- $(\mathcal{G}, \pi) \models \mathsf{O}\psi \overset{\text{def}}{\Leftrightarrow} (\mathcal{G}, \pi^1) \models \psi$;
- $(\mathcal{G}, \pi) \models \Box\psi \overset{\text{def}}{\Leftrightarrow}$ for all $i \geq 0$, $(\mathcal{G}, \pi^i) \models \psi$;
- $(\mathcal{G}, \pi) \models \psi_1 \mathsf{U} \psi_2 \overset{\text{def}}{\Leftrightarrow} \exists i \geq 0.\ (\mathcal{G}, \pi^i) \models \psi_2$ and $\forall j < i.\ (\mathcal{G}, \pi^j) \models \psi_1$.

$\mathcal{G}$ is a *model* of ξ, written $\mathcal{G} \models \xi$ iff $(\mathcal{G}, s) \models \xi$ for some state s. We denote by $[\![\xi]\!]_{SAT}$ the set of valuations v over $Vars(\xi)$ such that $(\mathcal{G}, s) \models \xi$ for some model $\mathcal{G}$ and state s of $\mathcal{G}$ with $\mu(s) = v$. A *CCTL** formula ξ is *satisfiable* iff there exists a model of ξ.

Assumption: By Lemma 1, we can assume w.l.o.g. that the *IPC** constraints p associated with atomic formulas $p[x_1 \leftarrow \mathsf{O}^{i_1} x_{j_1}, \ldots, x_r \leftarrow \mathsf{O}^{i_r} x_{j_r}]$ are atomic.

The existential fragment *E–CCTL** and the dual universal fragment *A–CCTL** of *CCTL** are obtained by disallowing respectively the universal and the existential path quantifier. In order to consider a fragment as large as possible, we also introduce *CEF*$^+$ which subsumes *E–CCTL**, *A–CCTL** and the *IPC**-constrained counterpart of *EF* logic, a well-know fragment of standard *CTL* closed under boolean connectives (see e.g., [May01]). *CEF*$^+$ is defined as follows (where ξ_E is an *E–CCTL** formula):

$$\xi := \xi_E \ \mid \neg\xi \mid \xi \vee \xi \mid E(\xi_E \mathsf{U} \xi) \mid E\mathsf{O}\xi$$

2.3 Integral Relational Automata

In this section we recall the framework of *Integral Relational Automata* (*IRA*) introduced in [BBK77]. An *IRA* consists of a finite-state machine enhanced with a finite number of counters. The operation repertoire of *IRA* includes assignment, input/output operations and guards of the form $x \sim y$ or $x \sim c$ with $\sim \in \{<, \leq, >, \geq, =\}$. We extend this operational model by allowing periodicity constraints as guards. Note that if we also allow guards of the form $x \leq y + c$, then the resulting formalism is Turing-complete (since we can easily simulate unrestricted counter machines). Let OP be the set of operations defined as follows:

$$p \ \mid \ ?x \ \mid \ !x \ \mid \ !c \mid x \leftarrow y \mid x \leftarrow c \mid \mathrm{NOP}$$

where p is an *atomic IPC** constraint, $x, y \in \mathrm{VAR}$ and $c \in \mathbb{Z}$. Informally, $?x$ assigns a new integral value to the variable x, $!x$ (resp $!c$) outputs the value of variable x (resp., constant c), $x \leftarrow y$ (resp. $x \leftarrow c$) assigns the value of variable y (resp., constant c) to x, and NOP is the dummy operation. The atomic *IPC** constraints are used as guards.

An *Integral Relational Automaton* (*IRA*) is a tuple $P = \langle V(P), E(P), \ell_V, \ell_E \rangle$, where $V(P)$ is the finite set of *vertices*, $E(P) \subseteq V(P) \times V(P)$ is the set of *edges*, $\ell_V : V(P) \rightarrow OP$ associates an operation to every vertex, and $\ell_E : E(P) \rightarrow \{+, -\}$ is a labelling of the edges (used for tests).

Let $Vars(P)$ be the set of all P variables (used in the operations of P) and $Cons(P)$ $\subseteq \mathbb{Z}$ be the least set containing all the P constants and such that $0 \in Cons(P)$ and for all $c_1, c_2 \in Cons(P)$, $c_1 \leq c \leq c_2$ implies $c \in Cons(P)$. Moreover, let $Mod(P)$ be the set of the *modulo constants* k used in the periodicity constrains $x \equiv_k c$ of P.

Notation: For convenience, we define $v.c = c$ for any valuation v and constant $c \in \mathbb{Z}$.

The semantics of an *IRA* P is described by a labelled graph $\mathcal{G}(P) = \langle \mathcal{S}(P), \rightarrow, \mu \rangle$, where the set of states $\mathcal{S}(P)$ is the set of pairs $\langle n, v \rangle$ such that $n \in V(P)$ is a vertex and v is a valuation over $Vars(P)$, $\mu(\langle n, v \rangle) = v$ for all $\langle n, v \rangle \in \mathcal{S}(P)$, and $\langle n, v \rangle \rightarrow \langle n', v' \rangle$ if and only if $e = (n, n') \in E(P)$ and one of the following conditions holds:

- $\ell_V(n) = ?x$ and $v'.y = v.y$ for every $y \in Vars(P) \setminus \{x\}$,
- $\ell_V(n) = !x$ or $\ell_V(n) = !c$ or $\ell_V(n) = \text{NOP}$ and $v' = v$,
- $\ell_V(n) = x \leftarrow a$, $v'.x = v.a$, and $v'.y = v.y$ for every $y \in Vars(P) \setminus \{x\}$,
- $\ell_V(n) = p$, $v' = v$, and *either* $\ell_E(e) = +$ and $v \models p$, *or* $\ell_E(e) = -$ and $v \not\models p$.

Note that $\mathcal{G}(P)$ is infinitely-branching because of input operations. An *history* of P is a path of $\mathcal{G}(P)$. An infinite history is also called a *computation*. A path $\overline{n}$ of P is a path in the finite–state graph $\langle V(P), E(P) \rangle$. For a finite path $\overline{n}$ of P, two tuples $\mathcal{N} = \langle n_1, \ldots, n_k \rangle$ and $\mathcal{N}' = \langle n'_1, \ldots, n'_h \rangle$ of P-vertices, we say that $\overline{n}$ is a path from $\mathcal{N}$ to $\mathcal{N}'$ iff $|\overline{n}| \geq h+k$ and $n_1, \ldots, n_h$ (resp., $n'_1, \ldots, n'_h$) is a prefix (resp., suffix) of P. The notion of path $\overline{n}$ from a tuple of vertices is similar. These notions can be extended to histories of P in a natural way. Let $\overline{n}_1$ be a P path from $\mathcal{N}_1$ to $\mathcal{N}$ and $\overline{n}_2$ be a P path from $\mathcal{N}$. We denote by $[\overline{n}_1 + \overline{n}_2]_{\mathcal{N}}$ the P path obtained by concatenating $\overline{n}_1$ with the path obtained from $\overline{n}_2$ by eliminating the prefix corresponding to $\mathcal{N}$. This notion of concatenation can be extended to histories in a natural way. In the following, a k-tuple $\langle \langle n_1, v_1 \rangle, \ldots, \langle n_k, v_k \rangle \rangle$ of P states is also denoted by $\langle \langle n_1, \ldots, n_k \rangle, \langle v_1, \ldots, v_k \rangle \rangle$.

We say that an *IRA* P is *complete* if the edge relation $E(P)$ is total and for each vertex n labelled by an *IPC** constraint and each flag $f \in \{+, -\}$, there is an edge labelled by f and having n as source. W.l.o.g. we assume that the *IRA* under our consideration are complete (this implies that the edge relation in $\mathcal{G}(P)$ is total).

Extended Integral Relational Automata: for technical reasons, we introduce *Extended IRA* (*EIRA*). An *EIRA* is a pair $\langle P, \ell_{EXT} \rangle$ where P is an *IRA* and ℓ_{EXT} is an additional P-vertex-labelling, mapping each vertex $n \in V(P)$ to a finite set (interpreted as conjunction) of *CCTL** atomic formulas $p[x_1 \leftarrow \mathsf{O}^{i_1} x_{j_1}, \ldots, x_r \leftarrow \mathsf{O}^{i_r} x_{j_r}]$ (where p is an atomic *IPC** constraint). This labelling induces constraints between the variables of the current state and the variables of succeeding states (along a computation).

For a (finite or infinite) P-history $\pi = \langle n_1, v_1 \rangle, \langle n_2, v_2 \rangle, \ldots$, we say that π is *fair* if π is consistent with the ℓ_{EXT}–labelling. Formally, we require that for all $1 \leq k \leq |\pi|$ and $p[x_1 \leftarrow \mathsf{O}^{i_1} x_{j_1}, \ldots, x_r \leftarrow \mathsf{O}^{i_r} x_{j_r}] \in \ell_{EXT}(n_k)$, the following holds:
if $k + i_p \leq |\pi|$ for all $1 \leq p \leq r$, then $v_k[x_1 \leftarrow v_{k+i_1}.x_{j_1}, \ldots, x_r \leftarrow v_{k+i_r}.x_{j_r}] \models p$.

In this paper we are interested in the following problem:

Model checking problem of *IRA* against *CCTL** : given an *IRA* P, a state s_0 of P, and a *CCTL** formula ξ with $Vars(\xi) \subseteq Vars(P)$, does $(\mathcal{G}(P), s_0) \models \xi$ hold?

In the following, we denote by $[\![\xi]\!]_P$ the set of P states s such that $(\mathcal{G}(P), s) \models \xi$. Model checking *IRA* against full *CCTL** is undecidable (also for the *CTL*-like fragment) as a consequence of [Čer93]. Thus, in the following, we analyze the fragment *CEF*$^+$. consider the satisfiability problem for *CEF*$^+$. We start by giving a symbolic model checking algorithm for *IRA* against *E–CCTL**.

3 Symbolic model checking of *IRA* against *E–CCTL**

In this section we show that given an *IRA* P and an *E–CCTL** formula ξ with $Vars(\xi) \subseteq Vars(P)$, we can compute a finite representation of $[\![\xi]\!]_P$. In the following, we can assume w.l.o.g. that $Cons(\xi) \subseteq Cons(P)$ and $Mod(\xi) \subseteq Mod(P)$, where $Cons(\xi)$ (resp., $Mod(\xi)$) denote the set of constants (resp., modulo constants) occurring in ξ.

First, we recall some basic notions. For a set S, a *quasi-ordering* (*qo*, for short) $\preceq$ over S is a reflexive and transitive (binary) relation on S. Given such a *qo*, we say that $U \subseteq S$ is an *upward closed set* if for all $x \in S$ and $y \in U$, $y \preceq x$ implies $x \in U$. We say that $\preceq$ is a *partial-order* (*po*, for short) iff $x \preceq y$ and $y \preceq x$ imply $x = y$. Finally, we say that the *qo* $\preceq$ is a *well quasi-ordering* (*wqo*, for short) if for every infinite sequence $x_0, x_1, x_2, \ldots$ of elements of S there exist indices $i < j$ such that $x_i \preceq x_j$.

Following [Čer93], we define a *wqo* on the set $\mathcal{S}(P)$ of P states (that is also a *po*). Then, in order to solve the model-checking problem, we will show that: (1) $[\![\xi]\!]_P$ is an upward closed set; (2) we can compute a finite representation $R([\![\xi]\!]_P)$ of this set; (3) we can check whether a given a state s belongs to $R([\![\xi]\!]_P)$.

We start by defining such a *wqo*. Let κ be the least common multiple of the constants in $Mod(P) \cup \{1\}$. We define a *po* $\preceq$ over tuples of valuations over $Vars(P)$ as follows: $\langle v_1, \ldots, v_h\rangle \preceq \langle v'_1, \ldots, v'_k\rangle$ iff $h = k$ and for all $1 \leq i, j \leq h$ and $a, b \in Cons(P) \cup Vars(P)$, the following holds: (1) $v_i.a \geq v_j.b$ iff $v'_i.a \geq v'_j.b$, (2) $v_i.a \equiv_\kappa v'_i.a$, and (3) if $v_i.a \geq v_j.b$, then $v'_i.a - v'_j.b \geq v_i.a - v_j.b$.[4] We write simply $v_1 \preceq v'_1$ if $h = 1$. Note that $v_i \preceq v'_i$ for all $1 \leq i \leq h$ does not imply that $\langle v_1, \ldots, v_h\rangle \preceq \langle v'_1, \ldots, v'_h\rangle$. Finally, for two h-tuples of states $\langle \mathcal{N}, \mathcal{V}\rangle, \langle \mathcal{N}', \mathcal{V}'\rangle$, we write $\langle \mathcal{N}, \mathcal{V}\rangle \preceq \langle \mathcal{N}', \mathcal{V}'\rangle$ to mean that $\mathcal{N} = \mathcal{N}'$ and $\mathcal{V} \preceq \mathcal{V}'$. The proofs of the following two results are given in [BG06].

Proposition 1. *For every $h \geq 1$, the partial order $\preceq$ is a* wqo *over the set of h-tuples of valuations over $Vars(P)$.*

Lemma 2 (Simulation Lemma).

1. *Let $\pi = \langle n_1, v_1\rangle, \ldots, \langle n_h, v_h\rangle$ be an history and $v'_1 \succeq v_1$. Then, there is an history $\pi' = \langle n_1, v'_1\rangle, \ldots, \langle n_h, v'_h\rangle$ such that $\langle v'_1, \ldots, v'_h\rangle \succeq \langle v_1, \ldots, v_h\rangle$;*
2. *Let $\pi = \langle n_1, v_1\rangle, \langle n_2, v_2\rangle, \ldots$ be a computation and $v'_1 \succeq v_1$. Then, there is a computation $\pi' = \langle n_1, v'_1\rangle, \langle n_2, v'_2\rangle, \ldots$ s.t. for all $h \geq 1$, $\langle v'_1, \ldots, v'_h\rangle \succeq \langle v_1, \ldots, v_h\rangle$.*

Thanks to the Simulation Lemma, we can prove the first important result.

Proposition 2. *$[\![\xi]\!]_P$ is an upward closed set with respect to $\preceq$.*

[4] So, the relation $\preceq$ depends on parameters $Vars(P)$, $Cons(P)$, and κ.

Proof. The proof is by structural induction on ξ. The cases $\xi = \top$, $\xi = \xi_1 \vee \xi_2$, and $\xi = \xi_1 \wedge \xi_2$ are obvious since $[\![\top]\!]_P = \mathcal{S}(P)$, $[\![\xi_1 \vee \xi_2]\!]_P = [\![\xi_1]\!]_P \cup [\![\xi_2]\!]_P$, $[\![\xi_1 \wedge \xi_2]\!]_P = [\![\xi_1]\!]_P \cap [\![\xi_2]\!]_P$, and upward closed sets are closed under union and intersection.

Now, assume that $\xi = E\psi$ for some path formula ψ. Then, there is a set X of state sub-formulas of ξ such that $\psi \in PLF(X)$. Let $s_1 \in [\![E\psi]\!]_P$ and $\overline{s}_1 \succeq s_1$. We claim that $\overline{s}_1 \in [\![E\psi]\!]_P$. Since $s_1 \in [\![E\psi]\!]_P$, there is a computation $\pi = s_1, s_2, \ldots$ such that $(\mathcal{G}(P), \pi) \models \psi$. Since $\overline{s}_1 \succeq s_1$, by Property 2 of Simulation Lemma and definition of $\preceq$, it easily follows that there is a computation $\overline{\pi} = \overline{s}_1, \overline{s}_2, \ldots$ such that for all $i \geq 1$ and atomic formula $\psi_{at} = p[x_1 \leftarrow \mathsf{O}^{i_1} x_{j_1}, \ldots, x_r \leftarrow \mathsf{O}^{i_r} x_{j_r}]$ with constants in $Cons(P)$ and modulo constants in $Mod(P)$: $\overline{s}_i \succeq s_i$ and $(\mathcal{G}(P), \pi^i) \models \psi_{at}$ if and only if $(\mathcal{G}(P), \overline{\pi}^i) \models \psi_{at}$. Moreover, for all $i \geq 1$ and $\xi' \in X$, by the induction hypothesis and the fact that $\overline{s}_i \succeq s_i$, we have that $s_i \in [\![\xi']\!]_P$ implies $\overline{s}_i \in [\![\xi']\!]_P$. These properties evidently imply $(\mathcal{G}(P), \overline{\pi}) \models \psi$, i.e. $\overline{s}_1 \in [\![E\psi]\!]_P$. Therefore, the claim holds. □

In the following subsection, we introduce the framework of *modulo–κ Graphose inequality Systems* (*κ–GS*, for short) as a finite representation of upward closed sets of P states (w.r.t. $\preceq$). In Subsection 3.2, we show some technical results on extended *IRA* and finally, in Subsection 3.3, we describe an algorithm to compute a *κ-GS* representation of the upward closed set $[\![\xi]\!]_P$.

3.1 Modulo–κ Graphose Inequality Systems

κ-GS extend *Graphose inequality Systems* introduced in [Čer93] by allowing to specify periodicity constraints on the set of solutions. Formally, for $\kappa \geq 1$, a *κ–GS* is a tuple $G = \langle D, C, w, mod \rangle$, where $D \subseteq \mathrm{VAR}$ is a finite set of variables, $C \subseteq \mathbb{Z}$ is a finite set of integral constants, $w : A \times A \rightarrow \mathbb{Z}^-$ for $A = D \cup C$ and $\mathbb{Z}^- = \mathbb{Z} \cup \{-\infty\}$ is a *weight function*, and mod is a map $mod : A \rightarrow \{0, \ldots, \kappa - 1\}$.

The semantics of a *κ–GS* G is given by specifying the set $Sol(G)$ of its *solutions*. A valuation v over D is said to be a solution of G iff for all $a, b \in A$,

$$v.a - v.b \geq w(a, b) \;\text{ and }\; v.a \equiv_\kappa mod(a)$$

where by definition for $c \in C$, $mod(c) \equiv_\kappa c$. The *κ–GS* G can be interpreted as a graph with set of vertices A and such that there is an edge from $a \in A$ to $b \in A$ with the *weight* $w(a, b)$ whenever $w(a, b) \neq -\infty$. Finding a solution of G means assigning integral values to the variable vertices so that the constraints imposed by mod are satisfied and for every edge in G, the difference between its source and target vertex values is at least the weight associated with the edge.

A *κ-GS* $G = \langle D, C, w, mod \rangle$ is called *consistent* if it has a solution. Furthermore, we say that G is *positive* if for all $a, b \in D \cup C$, either $w(a, b) = -\infty$ or $w(a, b) \geq 0$. A positive *κ-GS* is also denoted by *κ-PGS*. A *κ-GS* $G = \langle D, C, w, mod \rangle$ is *normalized* iff for all $a, b, c \in D \cup C$, (1) $w(a, b) \geq w(a, c) + w(c, b)$ and (2) $w(a, b) \neq -\infty$ implies $w(a, b) \equiv_\kappa mod(a) - mod(b)$.

Proposition 3 (Effectiveness of the *κ-GS* representation). *We can decide whether a* κ-GS $G = \langle D, C, w, mod \rangle$ *is consistent. In this case we can build effectively an equivalent* normalized κ-GS $|G| = \langle D, C, |w|, mod \rangle$, *called* normal form *of* G, *such* *that: (1) $Sol(|G|) = Sol(G)$, (2) $|G|$ is positive if G is positive, (3) every solution of the restriction of $|G|$ to a subset of D can be extended to a complete solution of $|G|$.*

Given an *IRA* P, let κ be the least common multiple of the integers in $Mod(P) \cup \{1\}$. A κ*-GS* $H = \langle D, C, w, mod \rangle$ is called *local* for P iff $D = Vars(P)$ and $C = Cons(P)$. A set of states $Y \subseteq \mathcal{S}(P)$ is said to be κ*-GS-represented* by a family of finite sets $(\mathcal{H}_n)_{n \in V(P)}$ of local κ*-GS* if for every state $\langle n, v \rangle \in Y$ we have $v \in Sol(H)$ for some $H \in \mathcal{H}_n$. By definition of *wqo* $\preceq$, it easily follows that local *positive* κ*-GS* constitute an effective representation of upward closed sets of states in $\mathcal{S}(P)$ (see details in [BG06]).

Proposition 4. κ-GS *representations are effectively closed under complementation.*

Proposition 5. *For every set of states $U \subseteq \mathcal{S}(P)$, U is* κ-PGS*-representable iff U is an upward closed set.*

Definition 1 (Intersection of κ-*GS*). *Given two* κ-GS $G_1 = \langle D_1, C_1, w_1, mod_1 \rangle$ *and* $G_2 = \langle D_2, C_2, w_2, mod_2 \rangle$*, their* intersection $G_1 \bigotimes G_2 = \langle D_1 \cup D_2, C_1 \cup C_2, w, mod \rangle$ *is defined by:*

- $G_1 \bigotimes G_2 = \mathsf{nil}$[5] *if there is $a \in D_1 \cap D_2$ such that $mod_1(a) \neq mod_2(a)$;*
- *otherwise for all $a, b \in D_1 \cup D_2 \cup C_1 \cup C_2$, $mod(a) = max\{mod'_1(a), mod'_2(a)\}$ and $w(a, b) = max\{w'_1(a, b), w'_2(a, b)\}$ where (for $i = 1, 2$)*
  - *if $a \in D_i \cup C_i$ then $mod'_i(a) = mod_i(a)$, else $mod'_i(a) = -\infty$*
  - *if $a, b \in D_i \cup C_i$ then $w'_i(a, b) = w_i(a, b)$, else $w'_i(a, b) = -\infty$.*

Note that intersection of κ*-GS* preserves positiveness. Moreover, the following holds.

Proposition 6. *Let $G = \langle D, C, w, mod \rangle$ and $G' = \langle D', C', w', mod' \rangle$ be two* κ-GS. *Then, for $v : D \cup D' \to \mathbb{Z}$, $v \in Sol(G \bigotimes G')$* iff *$v|_D \in Sol(G)$ and $v|_{D'} \in Sol(G')$. In particular, for $D = D'$, $Sol(G \bigotimes G') = Sol(G) \cap Sol(G')$.*

3.2 Symbolic Characterization of Fair Computations in *EIRA*

In this section, we essentially show that given an *EIRA*, we can compute a *PGS*-representation of the set of states s such that there is a *fair* computation starting from s. This technical result non-trivially generalizes [Čer93, Lemma 5.11] and is used in the following to solve model-checking of *IRA* against *E–CCTL**.

Let $\langle P, \ell_{EXT} \rangle$ be an *EIRA* and $\mathcal{K}$ the maximal natural number i such that a term of the form $\mathsf{O}^i x$ occurs in $\langle P, \ell_{EXT} \rangle$ for some variable x. W.l.o.g., we can assume that $\mathcal{K} \geq 1$ and all the constants (resp., modulo constants) occurring in the atomic formulas of $\langle P, \ell_{EXT} \rangle$ are in $Cons(P)$ (resp. $Mod(P)$). We denote by κ the least common multiple of the integers in $Mod(P) \cup \{1\}$ and $\mathcal{S}(P)^\dagger$ be the set of tuples of P states. In the following we consider only κ*-PGS* or κ*-GS* but we write simply *PGS* or *GS*.

Assume that $U \subseteq \mathcal{S}(P)$ is an *upward closed set* given by a *PGS*-representation. For a set $F \subseteq V(P)$ of P vertices, we denote by $[\![E\Box^F U]\!]_P$ the set of P states s such that there is a *fair* computation from s that only visits states of U and contains infinite occurrences of states $\langle n, v \rangle$ with $n \in F$. The main result of this subsection is the following:

[5] nil denotes some inconsistent κ*-PGS* over $D_1 \cup D_2$ and $C_1 \cup C_2$

Theorem 1. *Given a set $F \subseteq V(P)$ of P vertices, one can build a* PGS *representation of the set $[\![E\Box^F U]\!]_P$.*

To prove this result, we show two important preliminary results (Theorems 2 and 3).

For two tuples $\langle \mathcal{N}, \mathcal{V} \rangle$ and $\langle \mathcal{N}', \mathcal{V}' \rangle$ of P states, a P path $\overline{n}$ from $\mathcal{N}$ to $\mathcal{N}'$, we write:

- $\langle \mathcal{N}, \mathcal{V} \rangle \overset{\mathcal{K}}{\rightsquigarrow}{}^U \langle \mathcal{N}', \mathcal{V}' \rangle$ to mean that there is a *fair* history π from $\langle \mathcal{N}, \mathcal{V} \rangle$ to $\langle \mathcal{N}', \mathcal{V}' \rangle$ visiting only states in U, where $|\pi| = m \cdot \mathcal{K}$ with $m \geq 2$ (*fair reachability relation*);
- $\langle \mathcal{N}, \mathcal{V} \rangle \rightsquigarrow^U_{\overline{n}} \langle \mathcal{N}', \mathcal{V}' \rangle$ to mean that $\langle \mathcal{N}, \mathcal{V} \rangle \overset{\mathcal{K}}{\rightsquigarrow}{}^U \langle \mathcal{N}', \mathcal{V}' \rangle$ by a fair history π whose projection on $V(P)$ is the path $\overline{n}$.

For all $i \geq 1$, let $Vars_i$ be a fresh copy of $Vars(P)$ (we need this notation to formalize access to several copies of P variables), $\mathcal{K}Vars = \bigcup_{i=1}^{i=\mathcal{K}} Vars_i = \{y_1 \ldots, y_p\}$ and $\mathcal{K}Vars' = \{y'_1, \ldots, y'_p\}$. Given a $\mathcal{K}$–tuple $\mathcal{V} = \langle v_1, \ldots, v_\mathcal{K} \rangle$ of valuations over $Vars(P)$, for all $x \in \mathcal{K}Vars$ such that $x \in Vars_i$ (for some $1 \leq i \leq \mathcal{K}$) is a copy of variable $y \in Vars(P)$, $\mathcal{V}.x$ denotes the value of the component y of v_i.

A *GS* $G = \langle D, C, w, mod \rangle$ is called $\mathcal{K}$*–local* for P iff $D = \mathcal{K}Vars$ and $C = Cons(P)$. We denote by $Sat(G)$ the set of $\mathcal{K}$–tuples $\mathcal{V}$ of valuations over $Vars(P)$ that *satisfy* G, where $\mathcal{V}$ satisfies G iff the mapping $v : D \to \mathbb{Z}$ defined as $v.x = \mathcal{V}.x$ is a solution of G. We use $\mathcal{K}$*–local GS* to represent sets of $\mathcal{K}$-tuples of P states. Intuitively, a $\mathcal{K}$*–local GS* contains all the informations needed to evaluate an atomic constraint where all the terms of the form $\mathsf{O}^i x$ are such that $i \leq \mathcal{K}$. A set $X \subseteq \mathcal{S}(P)^\mathcal{K}$ of $\mathcal{K}$-tuples of P states is *GS-represented* by a family of finite sets $(\mathcal{G}_\mathcal{N})_{\mathcal{N} \in V(P)^\mathcal{K}}$ of $\mathcal{K}$–local *GS* if $\langle \mathcal{N}, \mathcal{V} \rangle \in X$ iff $\mathcal{V} \in Sat(G)$ for some *GS* $G \in \mathcal{G}_\mathcal{N}$.

A *PGS* $G = \langle D, C, w, mod \rangle$ is called $\mathcal{K}$*–transitional* for P iff $D = \mathcal{K}Vars \cup \mathcal{K}Vars'$ and $C = Cons(P)$. A pair $\langle \mathcal{V}, \mathcal{V}' \rangle$ of $\mathcal{K}$–tuples of valuations over $Vars(P)$ *satisfies* G iff the mapping $v : D \to \mathbb{Z}$ defined as $v.x = \mathcal{V}.x$ and $v.x' = \mathcal{V}'.x$, for each $x \in \mathcal{K}Vars$, is a solution of G. We denote by $Sat(G)$ the set of pairs of $\mathcal{K}$–tuples of valuations over $Vars(P)$ that satisfy G. We also extend the operator Sat to sets of $\mathcal{K}$–transitional *PGS* as follows: for a set $\mathcal{G}$ of $\mathcal{K}$–transitional *PGS*, $Sat(\mathcal{G}) = \bigcup_{G \in \mathcal{G}} Sat(G)$. Given a relation $\rightsquigarrow_0 \subseteq \mathcal{S}(P)^\dagger \times \mathcal{S}(P)^\dagger$, a pair $\langle \mathcal{N}, \mathcal{N}' \rangle$ of $\mathcal{K}$-tuples of P vertices and a finite set $\mathcal{G}$ of $\mathcal{K}$–transitional *PGS*, we say that $\mathcal{G}$ *characterizes* $\rightsquigarrow_0$ with respect to the pair $\langle \mathcal{N}, \mathcal{N}' \rangle$ iff $Sat(\mathcal{G}) = \{\langle \mathcal{V}, \mathcal{V}' \rangle \mid \langle \mathcal{N}, \mathcal{V} \rangle \ \rightsquigarrow_0 \ \langle \mathcal{N}', \mathcal{V}' \rangle\}$.

Remark 1. Let π_1 be a *fair* history from $\langle \mathcal{N}_1, \mathcal{V}_1 \rangle$ to $\langle \mathcal{N}, \mathcal{V} \rangle$ and π_2 be a *fair* history from $\langle \mathcal{N}, \mathcal{V} \rangle$ with $\langle \mathcal{N}, \mathcal{V} \rangle \in \mathcal{S}(P)^\mathcal{K}$. Then, $[\pi_1 + \pi_2]_{\langle \mathcal{N}, \mathcal{V} \rangle}$ is a *fair* history.

As first result, we show that the fair reachability relation $\overset{\mathcal{K}}{\rightsquigarrow}{}^U$ can be *PGS*-characterized.

Theorem 2. *For each pair $\langle \mathcal{N}, \mathcal{N}' \rangle$ of $\mathcal{K}$–tuples of Pvertices, one can build effectively a finite set $\mathcal{G}^U(\mathcal{N}, \mathcal{N}')$ of $\mathcal{K}$–transitional* PGS *that characterizes the fair reachability relation $\overset{\mathcal{K}}{\rightsquigarrow}{}^U$ w.r.t. the pair $\langle \mathcal{N}, \mathcal{N}' \rangle$. Moreover, for each $G \in \mathcal{G}^U(\mathcal{N}, \mathcal{N}')$, $\{G\}$ characterizes the fair reachability relation $\rightsquigarrow^U_{\overline{n}}$ for some path $\overline{n}$ from $\mathcal{N}$ to $\mathcal{N}'$.*

The algorithm we propose relies on Remark 1, properties of normalized *PGS* (see Proposition 3), and its termination is guaranteed by a suitable decidable *wqo*, which is defined over the set of *PGS* [6] (for a fixed set of variables and constants). More details are given in [BG06].

For a set $X \subseteq \mathcal{S}(P)^{\mathcal{K}}$, let us define $re^U(X) = \{\langle n, v\rangle \in \mathcal{S}(P) \mid \exists \langle \mathcal{N}, \mathcal{V}\rangle \in X.\ \langle n, v\rangle \overset{\mathcal{K}}{\rightsquigarrow}^U \langle \mathcal{N}, \mathcal{V}\rangle\}$. By Proposition 3 and Theorem 2, we easily obtain the following important corollary.

Corollary 1. *Given a family of $\mathcal{K}$–local* GS *(resp.,* PGS *) representing a set $X \subseteq \mathcal{S}(P)^{\mathcal{K}}$, we can construct a* GS *(resp.,* PGS *) representation of $re^U(X)$.*

The second preliminary result for the proof of Theorem 1 essentially relies on properties of *PGS*. Its proof is highly technical (full details are given in [BG06]). For a $\mathcal{K}$-tuple $\mathcal{N}$ of vertices and a P path h from $\mathcal{N}$ to $\mathcal{N}$, we define the following sets of $\mathcal{K}$-tuples of valuations over $Vars(P)$:

- $Sp(h) := \{\mathcal{V} \mid \exists \mathcal{V}'.\ \langle \mathcal{N}, \mathcal{V}\rangle \rightsquigarrow_h^U \langle \mathcal{N}, \mathcal{V}'\rangle$ and $\mathcal{V}' \succeq \mathcal{V}\}$;
- $Rea^{\infty}(h) := \{\mathcal{V} \mid \langle \mathcal{N}, \mathcal{V}\rangle \rightsquigarrow_{h^{\infty}}^U\}$.

where $\langle \mathcal{N}, \mathcal{V}\rangle \rightsquigarrow_{h^{\infty}}^U$ means that there is a fair U-computation π starting from $\langle \mathcal{N}, \mathcal{V}\rangle$ whose projection on $V(P)$ is the path h^{∞} (h^{∞} is the infinite path $h, h_1, h_1, \ldots$, where h_1 is obtained from h by eliminating the prefix corresponding to $\mathcal{N}$). By Simulation Lemma $Sp(h) \subseteq Rea^{\infty}(h)$.

Theorem 3. *Let $\mathcal{N}$ be a $\mathcal{K}$-tuple of vertices, h be a path from $\mathcal{N}$ to $\mathcal{N}$, and $G(h)$ be a $\mathcal{K}$–transitional* PGS *such that $\{G(h)\}$ characterizes the fair reachability relation $\rightsquigarrow_h^U$. Then, we can construct a $\mathcal{K}$–local* PGS *H^h such that $Sp(h) \subseteq Sat(H^h) \subseteq Rea^{\infty}(h)$.*

The idea behind this result is that we can build a *PGS*-representation H^h having the properties needed to prove Theorem 1 instead of considering $Sp(h)$ and $Rea^{\infty}(h)$ (see the following proof). Now, we can prove the main result of this subsection.

Proof of Theorem 1. Let $F_{\mathcal{K}}$ be the set of $\mathcal{K}$–tuples $\mathcal{N}$ of vertices such that some component of $\mathcal{N}$ is in F. By Theorem 2, for each $\mathcal{N} \in F_{\mathcal{K}}$, we can construct a finite set $\mathcal{G}^U(\mathcal{N}, \mathcal{N})$ of $\mathcal{K}$-transitional *PGS* characterizing $\overset{\mathcal{K}}{\rightsquigarrow}^U$ w.r.t. the pair $(\mathcal{N}, \mathcal{N})$. Moreover, there is *finite* set of *representative* paths h from $\mathcal{N}$ to $\mathcal{N}$, denoted by $Repr(\mathcal{N})$, such that $\mathcal{G}^U(\mathcal{N}, \mathcal{N}) = \bigcup_{h \in Repr(\mathcal{N})} \{G(h)\}$, where $\{G(h)\}$ characterizes $\rightsquigarrow_h^U$. By Theorem 3, for every $\mathcal{N} \in F_{\mathcal{K}}$, we can compute a family $\{H_{\mathcal{N}}^h\}_{h \in Repr(\mathcal{N})}$ of $\mathcal{K}$–local *PGS* such that $Sp(h) \subseteq Sat(H_{\mathcal{N}}^h) \subseteq Rea^{\infty}(h)$ for every $h \in Repr(\mathcal{N})$. Let us consider the set X of $\mathcal{K}$-tuples of P states defined as follows:

$$X = \{\langle \mathcal{N}, \mathcal{V}\rangle \mid \mathcal{N} \in F_{\mathcal{K}}, \exists h \in Repr(\mathcal{N}) :\ \mathcal{V} \in Sat(H_{\mathcal{N}}^h)\}$$

Note that X is *PGS*-represented by the family $\{\mathcal{H}_{\mathcal{N}}\}_{\mathcal{N} \in F_{\mathcal{K}}}$ of $\mathcal{K}$-local *PGS*, where $\mathcal{H}_{\mathcal{N}} = \bigcup_{h \in Repr(\mathcal{N})} \{H_{\mathcal{N}}^h\}$. Therefore, by Corollary 1, Theorem 1 directly follows from the following claim: $re^U(X) = [\![E\Box^F U]\!]_P$. It remains to prove this claim.

$re^U(X) \subseteq [\![E\Box^F U]\!]_P$: let $\langle n, v\rangle \in re^U(X)$. Then, there is $\mathcal{N} \in F_{\mathcal{K}}$, $h \in Repr(\mathcal{N})$, and $\mathcal{V} \in Sat(H_{\mathcal{N}}^h)$ such that $\langle n, v\rangle \overset{\mathcal{K}}{\rightsquigarrow}^U \langle \mathcal{N}, \mathcal{V}\rangle$. Since $Sat(H_{\mathcal{N}}^h) \subseteq Rea^{\infty}(h)$, it holds

[6] the hypothesis of positiveness is crucial.

that $\langle \mathcal{N}, \mathcal{V} \rangle \rightsquigarrow^{U}_{h^{\infty}}$. Since h^{∞} contains infinite occurrences of accepting vertices, by Remark 1 we deduce that $\langle n, v \rangle \in [\![E\Box^{F} U]\!]_{P}$.
$[\![E\Box^{F} U]\!]_{P} \subseteq re^{U}(X)$: let $\langle n, v \rangle \in [\![E\Box^{F} U]\!]_{P}$. Then, there is a fair U-computation π starting from $\langle n, v \rangle$ that visits some vertex in F infinitely many times. Hence, we deduce the existence of an infinite sequence

$$\langle n, v \rangle \overset{\mathcal{K}}{\rightsquigarrow}^{U} \langle \mathcal{N}_0, \mathcal{V}_0 \rangle \overset{\mathcal{K}}{\rightsquigarrow}^{U} \langle \mathcal{N}_1, \mathcal{V}_1 \rangle \overset{\mathcal{K}}{\rightsquigarrow}^{U} \langle \mathcal{N}_2, \mathcal{V}_2 \rangle \ldots$$

with $\mathcal{N}_i \in F_{\mathcal{K}}$ for each $i \geq 0$. Due to well quasi-ordering of $\preceq$, there are $i < j$ such that $\mathcal{N}_i = \mathcal{N}_j$ and $\mathcal{V}_i \preceq \mathcal{V}_j$. Since $\langle \mathcal{N}_i, \mathcal{V}_i \rangle \overset{\mathcal{K}}{\rightsquigarrow}^{U} \langle \mathcal{N}_i, \mathcal{V}_j \rangle$, by Properties of $\mathcal{G}^{U}(\mathcal{N}_i, \mathcal{N}_i)$, there is $h \in Repr(\mathcal{N}_i)$ such that $\langle \mathcal{N}_i, \mathcal{V}_i \rangle \rightsquigarrow^{U}_{h} \langle \mathcal{N}_i, \mathcal{V}_j \rangle$. Since $\mathcal{V}_i \preceq \mathcal{V}_j$, it follows that $\mathcal{V}_i \in Sp(h) \subseteq Sat(H^{h}_{\mathcal{N}})$. As $\langle n, v \rangle \overset{\mathcal{K}}{\rightsquigarrow}^{U} \langle \mathcal{N}_i, \mathcal{V}_i \rangle$, it holds that $\langle n, v \rangle \in re^{U}(X)$. □

3.3 Symbolic Model-checking Algorithm

We fix an *IRA* P and an *E–CCTL** formula ξ such that $Vars(\xi) \subseteq Vars(P)$, $Cons(\xi) \subseteq Cons(P)$, and $Mod(\xi) \subseteq Mod(P)$. Let κ be the least common multiple of the constants in $Mod(P) \cup \{1\}$. In the following, by using Theorem 1 and a generalization of the standard tableau-based construction for *LTL* model-checking, we show that we can construct a κ-*PGS* representation of the set of states $[\![\xi]\!]_{P}$. Hence, model-checking *IRA* against *E–CCTL** is decidable (note that the membership problem for κ-*PGS* representations is trivially decidable).

Let $\psi \in PLF(X)$ be a path *E–CCTL** formula with $X = \{\xi_1, \ldots, \xi_k\}$. The *closure* of ψ, denoted by $cl(\psi)$, is the smallest set containing $\xi_1, \ldots, \xi_k$, each *subformula* of ψ (considering $\xi_1, \ldots, \xi_k$ as atomic propositions), and satisfying: (1) if $\psi_1 \mathsf{U} \psi_2 \in cl(\psi)$, then $\mathsf{O}(\psi_1 \mathsf{U} \psi_2) \in cl(\psi)$, (2) if $\Box\psi_1 \in cl(\psi)$, then $\mathsf{O}\Box\psi_1 \in cl(\psi)$. An *LTL-atom* of ψ is a set $A \subseteq cl(\psi)$ satisfying the following properties:

- for $\psi_1 \vee \psi_2 \in cl(\psi)$, $\psi_1 \vee \psi_2 \in A$ iff either $\psi_1 \in A$ or $\psi_2 \in A$;
- for $\psi_1 \wedge \psi_2 \in cl(\psi)$, $\psi_1 \wedge \psi_2 \in A$ iff $\psi_1 \in A$ and $\psi_2 \in A$;
- for $\psi_1 \mathsf{U} \psi_2 \in cl(\psi)$, $\psi_1 \mathsf{U} \psi_2 \in A$ iff either $\psi_2 \in A$ or $\{\psi_1, \mathsf{O}(\psi_1 \mathsf{U} \psi_2)\} \subseteq A$;
- for $\Box\psi_1 \in cl(\psi)$, $\Box\psi_1 \in A$ iff $\{\psi_1, \mathsf{O}\Box\psi_1\} \subseteq A$.

Let $Atoms(\psi)$ be the set of *LTL*-atoms of ψ. When an until-formula $\psi_1 \mathsf{U} \psi_2$ is asserted at a state along a computation, we must make sure that the liveness requirement ψ_2 is eventually satisfied. This is done (as for *LTL*) using a generalized Büchi condition, one for each until formula. Formally, we denote by $\mathcal{F}(\psi)$ the family of subsets of $Atoms(\psi)$ defined as: for any until formula $\psi_1 \mathsf{U} \psi_2 \in cl(\psi)$, there is a component $F \in \mathcal{F}(\psi)$ that contains all and only the *LTL*-atoms A such that either $\psi_2 \in A$ or $\psi_1 \mathsf{U} \psi_2 \notin A$.

The main step of the proposed algorithm is represented by the following result.

Lemma 3. *Let $\psi \in PLF(X)$ be a path sub-formula of ξ such that $X = \{\xi_1, \ldots, \xi_k\}$ and for each $1 \leq i \leq k$, $[\![\xi_i]\!]_{P}$ is given by a family $(\mathcal{H}^{\xi_i}_{n})_{n \in V(P)}$ of local κ-*PGS*. Then, we can construct a κ-*PGS* representation of $[\![E\psi]\!]_{P}$.*

Proof. We build an *EIRA* $\langle P', \ell_{EXT} \rangle$, a set $F \subseteq V(P')$, and a family $\mathcal{H} = (\mathcal{H}_{n'})_{n' \in V(P')}$ of sets of local κ-*PGS* (w.r.t. P') such that $V(P') = V(P) \times Atoms(\psi) \times \{0, \ldots, |\mathcal{F}(\psi)|\}$, $Vars(P') = Vars(P)$, $Cons(P') = Cons(P)$, $Mod(P') = Mod(P)$, and

Claim 1. for all $\langle n, v\rangle \in \mathcal{S}(P)$, $\langle n, v\rangle \in \llbracket E\psi\rrbracket_P$ if and only if $\langle\langle n, A, 0\rangle, v\rangle \in \llbracket E\Box^F \ulcorner\mathcal{H}\urcorner\rrbracket_{P'}$ for some *LTL*-atom $A \in Atoms(\psi)$ such that $\psi \in A$ (where $\ulcorner\mathcal{H}\urcorner$ denotes the upward closed subset of $\mathcal{S}(P')$ that is κ–*PGS* represented by $\mathcal{H}$).

Evidently, the current Lemma directly follows from the claim above and Theorem 1. The *EIRA* $\langle P', \ell_{EXT}\rangle$ and $F \subseteq V(P')$ are defined as (where $\mathcal{F}(\psi) = \{F_1, \ldots, F_m\}$):

- $V(P') = V(P) \times Atoms(\psi) \times \{0, \ldots, m\}$. A P' vertex is a triple $\langle n, A, i\rangle$, where n is a P vertex, A is an atom that intuitively represents the set of formulas that hold at n (along the current computation), and i is a finite counter used to check the fulfillment of the generalized Büchi condition $\mathcal{F}(\psi)$;
- $\langle\langle n, A, i\rangle, \langle n', A', j\rangle\rangle \in E(P')$ if and only if (1) $\langle n, n'\rangle \in E(P)$, (2) for all $\mathsf{O}\psi' \in cl(\psi)$, $\mathsf{O}\psi' \in A$ iff $\psi' \in A'$ (i.e., the next-requirements in A are met in A'), and (3) $j = i$ if $i < m$ and $A' \notin F_{i+1}$, and $j = (i+1)\ mod\ (m+1)$ otherwise;
- the labelling ℓ'_V and ℓ'_E of P' are consistent with those of P, i.e. $\ell'_V(\langle n, A, i\rangle) = \ell_V(n)$ and $\ell'_E(\langle\langle n, A, i\rangle, \langle n', A', i'\rangle\rangle) = \ell_E(\langle n, n'\rangle)$; $\ell_{EXT}(\langle n, A, i\rangle)$ is the set of atomic formulas $p[x_1 \leftarrow \mathsf{O}^{i_1} x_{j_1}, \ldots, x_r \leftarrow \mathsf{O}^{i_q} x_{j_q}]$ in A;
- $F = \{\langle n, A, m\rangle \in V(P')\}$.

It remains to define the family $\mathcal{H} = (\mathcal{H}_{n'})_{n' \in V(P')}$ of sets of local κ-*PGS*. Let $n' = \langle n, A, i\rangle$ with $A \cap X = \{\xi_{j_1}, \ldots, \xi_{j_r}\}$. Intuitively, $A \cap X$ represents the set of "atomic" state formulas asserted at n along the current computation. Thus, we have to require that $Sat(\mathcal{H}_{n'}) = \{v \mid \langle n, v\rangle \in \bigcap_{i=1}^{i=r} \llbracket\xi_{j_i}\rrbracket_P\}$. Formally, $\mathcal{H}_{n'} = \{H \mid H = H^1 \otimes \ldots \otimes H^r$ with $H^h \in \mathcal{H}_n^{\xi_{j_h}}$ for all $1 \leq h \leq r\}$. A full proof of Claim 1 is given in [BG06]. □

Now, we can prove the desired result.

Theorem 4. *We can construct a* κ-PGS *representation* $(\mathcal{H}_n^\xi)_{n \in V(P)}$ *of* $\llbracket\xi\rrbracket_P$.

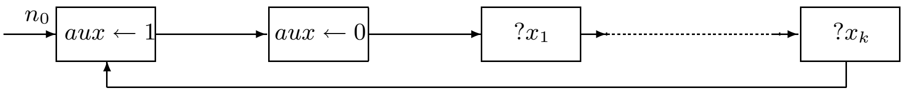
Proof. By structural induction on ξ. The case $\xi = \top$ is obvious. If $\xi = \xi_1 \vee \xi_2$ (resp., $\xi = \xi_1 \wedge \xi_2$), then for all $n \in V(P)$, $\mathcal{H}_n^\xi = \mathcal{H}_n^{\xi_1} \cup \mathcal{H}_n^{\xi_2}$ (resp., $\mathcal{H}_n^\xi = \{H_1 \otimes H_2 \mid H_i \in \mathcal{H}_n^{\xi_i},\ i = 1, 2\}$), where $(\mathcal{H}_n^{\xi_i})_{n \in V(P)}$ is the κ-*PGS* representation of ξ_i with $i = 1, 2$. Finally, the case $\xi = E\psi$ follows from the induction hypothesis and Lemma 3. □

4 Satisfiability and model-checking for *CEF*$^+$

In this section we show the main result of this paper, i.e. satisfiability and model-checking for *CEF*$^+$ are decidable. We need the following preliminary result.

Lemma 4. *For an* E–CCTL* *formula* ξ*, we can construct in polynomial time an* IRA P *with a distinguished vertex* n_0 *and a new* E–CCTL* *formula* ξ' *such that* $\llbracket\xi\rrbracket_{SAT} = \{v \mid \langle n_0, v'\rangle \in \llbracket\xi'\rrbracket_P$ where $v'.x = v.x$ for every $x \in Vars(\xi)\}$.

Proof. Let $Var(\xi) = \{x_1, \ldots, x_k\}$. The *IRA* P is defined as follows:



This *IRA* essentially consists of a sequence of inputs operations and we use an auxiliary variable aux to distinguish the state where all the values of the variables have been updated, which correspond to a new valuation.

Now consider the map f over *E–CCTL** formulas defined as: $f(\mathsf{O}^i x \sim \mathsf{O}^j y) = \mathsf{O}^{i(k+2)} x \sim \mathsf{O}^{j(k+2)} y$, $f(\mathsf{O}^i x \equiv_k c) = f(\mathsf{O}^{i(k+2)} x) \equiv_k c$, f is homomorphic w.r.t. the positive boolean operators, $f(\mathsf{O}\psi) = \mathsf{O}^{(k+2)} f(\psi)$, $f(\psi \mathsf{U} \psi') = \big((aux = 1) \Rightarrow f(\psi)\big)\mathsf{U}\big((aux = 1) \wedge f(\psi')\big)$, $f(\Box\psi) = \Box\big((aux = 1) \Rightarrow f(\psi)\big)$, $f(E\psi) = Ef(\psi)$. We can check that $v \in [\![\xi]\!]_{\mathrm{SAT}}$ iff there is a valuation v' over $Vars(\xi) \cup \{aux\}$ such that $v'.x = v.x$ for every $x \in Vars(\xi)$ and $\langle n_0, v' \rangle \in [\![f(\xi)]\!]_P$. □

Theorem 5. *(1) The model checking problem of* IRA *against* CEF$^+$ *is decidable.*
(2) Satisfiability of CEF$^+$ *is decidable.*

Proof. (1) For given *IRA* P and *CEF*$^+$ formula ξ, we prove by structural induction on ξ that we can build a κ-*GS* representation of $[\![\xi]\!]_P$ (where κ is defined as in Section 3). The cases in which ξ is a *E–CCTL** formula or a disjunction of formulas directly follow from Theorem 4, while the case $\xi = \neg\xi'$ follows from Proposition 4. For the case $\xi = E(\xi_1 \mathsf{U} \xi_2)$ where ξ_1 is an *E–CCTL** formula, we observe that $[\![\xi]\!]_P = re^{[\![\xi_1]\!]_P}([\![\xi_2]\!]_P)$, and the result follows from Theorem 4 and Corollary 1 (setting $\mathcal{K} = 1$). Finally, the case $\xi = E\mathsf{O}\xi'$ follows from a simple variant of Corollary 1.
(2) For a *CEF*$^+$ formula ξ, we construct a κ-*GS* representation of $[\![\xi]\!]_{SAT}$ (where κ and κ-*GS* representation have an obvious meaning). For the boolean connectives we proceed as above. The case in which ξ is an *E–CCTL** formula easily follows from Proposition 3, Lemma 4, and Theorem 4. Finally, we observe that (1) $[\![E(\xi_1 \mathsf{U} \xi_2)]\!]_{SAT} = \emptyset$ if $[\![\xi_2]\!]_{SAT} = \emptyset$, and $[\![E(\xi_1 \mathsf{U} \xi_2)]\!]_{SAT} = [\![\xi_1]\!]_{SAT} \cup [\![\xi_2]\!]_{SAT}$ otherwise, (2) $[\![E\mathsf{O}\xi]\!]_{SAT}$ contains all valuations over $Vars(\xi)$ if $[\![\xi]\!]_{SAT} \neq \emptyset$, and $[\![E\mathsf{O}\xi]\!]_{SAT} = \emptyset$ otherwise.
□

5 Conclusion

We have considered an extension of standard *CTL**, called *CCTL**, whose atomic formulas are constraints from *IPC** with comparison of variables at different states. For this logic, we have addressed two problems: satisfiability and model checking of Integral Relational Automata [BBK77,Čer93] extended with periodicity constraints. Since model checking *IRA* against full *CCTL** is undecidable (also for the *CTL*-like fragment), we have considered a meaningful fragment of *CCTL**, namely *CEF*$^+$ (which subsumes both the existential and universal fragment of *CCTL** and the *EF*-like fragment) showing that for this fragment both satisfiability and model checking of *IRA* are decidable. Furthermore, using a symbolic approach based on theory of κ-*GS*, the theory of well quasi-ordering, and automata-theoretic techniques, we have shown that it is possible to compute a finite representation of the set of states of the given *IRA* that satisfy a given formula. There are still interesting and non-trivial open questions such as the decidability status of satisfiability of full *CCTL** and the complexity for the considered decidable fragment (termination of our algorithm (see Theorem 2) is guaranteed by a *wqo* defined over the set of κ-*PGS* for a fixed set of variables and constants).

References

[ACJT96] P. A. Abdulla, K. Cerans, B. Jonsson, and Yih-Kuen Tsay. General decidability theorems for infinite-state systems. In *LICS'96*, pages 313–321. IEEE Computer Society Press, 1996.

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

[AJ94] P.A. Abdulla and B. Jonsson. Undecidable verification problems for programs with unreliable channels. In *ICALP'04*, volume 820 of *LNCS*. Springer, 1994.

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

[BBK77] J. Bardzin, J. Bicevskis, and A. Kalninsh. Automatic construction of complete sample systems for program testing. In *IFIP Congress*, pages 57–62, 1977.

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

[BG06] L. Bozzelli and R. Gascon. Branching-time temporal logic extended with Presburger constraints. Technical Report LSV-06-10, LSV, May 2006.

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

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

[Čer93] K. Čerāns. Deciding properties of integral relational automata. Technical Report No. 73, Dept. of Computer Sciences, Chalmers University of Technology, Göteborg, Sweden, 1993. An extended abstract appeared in Proc. of ICALP'04, LNCS 820.

[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, 2003. An extended abstract appeared in Proc. of FSTTCS'02.

[DG05] S. Demri and R. Gascon. Verification of qualitative $\mathbb{Z}$-constraints. In *CONCUR'05*, volume 3653 of *LNCS*, pages 518–532. Springer, 2005.

[EH86] E.A. Emerson and J.Y. Halpern. Sometimes and not never revisited: On branching versus linear time. *Journal of ACM*, 33(1):151–178, 1986.

[FS01] A. Finkel and Ph. Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1-2):63–92, 2001.

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

[LM01] U. Dal Lago and A. Montanari. Calendars, time granularities, and automata. In *SSTD'01*, volume 2121 of *LNCS*, pages 279–298. Springer, 2001.

[LS04] J. Leroux and G. Sutre. On flatness for 2-dimensional vector addition systems with states. In *CONCUR'04*, volume 3170 of *LNCS*, pages 402–416. Springer, 2004.

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

[May01] Richard Mayr. Decidability of model checking with the temporal logic EF. *Theoretical Computer Science*, 256:31–62, 2001.

[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*, volume 3444 of *LNCS*, pages 46–60. Springer, 2005.