Verification of Gap-Order Constraint Abstractions of Counter Systems

Laura Bozzelli[1] and Sophie Pinchinat[2]

[1] Technical University of Madrid (UPM), 28660 Boadilla del Monte, Madrid, Spain
[2] IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France

Abstract. We investigate verification problems for *gap-order constraint systems* (GCS), an (infinitely-branching) abstract model of counter machines, in which constraints (over $\mathbb{Z}$) between the variables of the source state and the target state of a transition are *gap-order constraints* (GC) [27]. GCS extend monotonicity constraint systems [5], integral relation automata [12], and constraint automata in [15]. First, we show that checking the existence of infinite runs in GCS satisfying acceptance conditions à la Büchi (fairness problem) is decidable and PSPACE-complete. Next, we consider a constrained branching-time logic, GCCTL*, obtained by enriching CTL* with GC, thus enabling expressive properties and subsuming the setting of [12]. We establish that, while model-checking GCS against the universal fragment of GCCTL* is undecidable, model-checking against the existential fragment, and satisfiability of both the universal and existential fragments are instead decidable and PSPACE-complete (note that the two fragments are not dual since GC are not closed under negation). Moreover, our results imply PSPACE-completeness of the verification problems investigated and shown to be decidable in [12], but for which no elementary upper bounds are known.

1 Introduction

Abstractions of Counter systems. Counter systems are a widely investigated complete computational model, used for instance to model broadcast protocols [19] and programs with pointer variables [7]. Though simple problems like reachability are already undecidable for 2-counter Minsky machines [24], interesting abstractions of counter systems have been studied, for which interesting classes of verification problems have been shown to be decidable. Many of these abstractions are in fact restrictions: examples include Petri nets [25], reversal-bounded counter machines [21], and flat counter systems [6,13]. Genuine abstractions are obtained by approximating counting operations by non-functional fragments of Presburger constraints between the variables of the current state and the variables of the next state. Examples include the class of Monotonicity Constraint Systems (MCS) [5] and its variants, like constraint automata in [15], and integral relation automata (IRA) [12], for which the (monotonicity) constraints (MC) are boolean combinations of inequalities of the form $u < v$ or $u \leq v$, where u and v range over variables or integer constants. MCS and their subclasses (namely, *size-change systems*) have found important applications for automated termination proofs of functional programs (see e.g. [5]). Richer classes of non-functional fragments of Presburger constraints have been investigated, e.g. difference bound constraints [14], and their extension, namely octagon relations [9], where it is shown that the transitive closure of a

V. Kuncak and A. Rybalchenko (Eds.): VMCAI 2012, LNCS 7148, pp. 88–103, 2012.

single constraint is Presburger definable (these results are useful for the verification of safety properties of flat counter systems). Note that difference bound constraints over (real-valued or integer-valued) variables (clocks) are also used as guards of transitions in timed automata [3]. Size-change systems extended with difference bound constraints over the natural number domain have been investigated in [4]: there, the atomic difference constraints are of the form $x - y' \geq c$, where c is an integer constant, and y' (resp., x) range over the variables of the target (resp., source) state. Termination for this class of systems is shown to be undecidable. To regain decidability, the authors consider a restriction, where at most one bound per target variable in each transition is allowed.

Temporal logics with Presburger constraints. An important classification of temporal logics is based on the underlying nature of time. In the *linear-time* setting , formulas are interpreted over linear sequences (corresponding to single computations of the system), and temporal operators are provided for describing the ordering of events along a single computation path. In the *branching-time* setting, formulas are instead interpreted over computation trees, which describe all the possible computations of the system from a designated initial state. Branching-time temporal logics are in general more expressive than linear-time temporal logics since they provide both temporal operators for describing properties of a path in the computation tree, and path quantifiers for describing the branching structure in computation trees.

In order to specify behavioral properties of counter systems, standard propositional linear-time temporal logics (like LTL) and propositional branching-time temporal logics (like CTL*) can be extended by replacing atomic propositions with Presburger constraints, which usually refer to the values of the (counter) variables at two consecutive states along a computation path (run). These enriched temporal logics allow to specify properties of counter systems that go beyond simple reachability. Hence, basic decision problems are generally undecidable. However, decidability has been established for various interesting fragments. We focus on fragments where the constraint language includes MC. For the *linear-time setting*, many decidable fragments of full Presburger LTL have been obtained either by restricting the underlying constraint language, see e.g. [15,17], or by restricting the logical language, see e.g. [8,13]. In particular, satisfiability and model checking (w.r.t. constraint automata) of standard LTL extended with MC are decidable and PSPACE-complete [15] (which matches the complexity of LTL). For the *branching-time setting*, to the best of our knowledge, very few decidability results are known. The extension of standard CTL* with MC, here denoted by MCCTL*, has been introduced in [12], where it is shown that model checking IRA against its existential and universal fragments, E–MCCTL* and A–MCCTL*, is decidable (by contrast, model checking for full MCCTL* is undecidable, even for its CTL-like fragment[1]). As done in [17], adding periodicity constraints and the ability for a fixed $k \geq 1$, to compare the variable values at states of a run at distance at most k, decidability of the above problems is preserved [10]. However, no elementary upper bounds for these problems are known [12,10]. Moreover, it is shown in [16] that model checking a subclass of flat counter machines w.r.t. full Presburger CTL* is decidable. In this subclass of systems, counting acceleration over every cycle in the control graph is Presburger definable. Thus, since the relation between the variables at the current and next state

[1] Quantification over variables can be simulated by the path quantifiers of the logic.

is functional and the control graph is flat, Presburger definability can be extended in a natural way to the set of states satisfying a given formula.

Our contribution. We investigate verification problems for an (infinitely-branching) abstract model of counter machines, we call *gap-order constraint systems* (GCS), in which constraints (over $\mathbb{Z}$) between the variables of the source state and the target state of a transition are (transitional) *gap-order constraints* (GC) [27]. These constraints are positive boolean combinations of inequalities of the form $u - v \geq k$, where u, v range over variables and integer constants and k is a natural number. Thus, GC can express simple relations on variables such as lower and upper bounds on the values of individual variables; and equality, and gaps (minimal differences) between values of pairs of variables. GC have been introduced in the field of constraint query languages (constraint Datalog) for deductive databases [27], and also have found applications in the analysis of safety properties for parameterized systems [1,2] and for determining state invariants in counter systems [20]. As pointed out in [2], using GC for expressing the enabling conditions of transitions allow to handle a large class of protocols, where the behavior depends on the relative ordering of values among variables, rather than the actual values of these variables.

GCS strictly extend IRA (and its variants, namely, MCS and the constraint automata in [15]). This because GC extend MC and, differently from MC, are closed under existential quantification (but not under negation).[2] Moreover, the parameterized systems investigated in [1,2] correspond to the parameterized version of GCS, where a system consists of an arbitrary number of processes which are instances of the same GCS (additionally, transitions of a process can specify global conditions which check the current local states and variables of all, or some of, other active processes). This framework is useful to verify correctness regardless of the number of processes. However, basic decision problems like reachability for the parameterized version of GCS are undecidable [1,2]. Decidability of reachability can be regained for a restricted class of parameterized systems in which processes have at most one integer local variable [1,2].

Note that if we extend the constraint language of GCS by allowing either negation, or constraints of the form $u - v \geq -k$, with $k \in \mathbb{N}$, then the resulting class of systems can trivially emulate Minsky counter machines, leading to undecidable basic decision problems. Moreover, note that GC extended with constraints of the form $u - v \geq -k$, with $k \in \mathbb{N}$, correspond to standard diagonal bound constraints [3,14]. As mentioned above, these constraints are used as guards in timed automata [3], where (integer-valued or real-valued) variables (clocks) record the elapsed time among events. However, guards in timed automata express constraints only over the clocks of the source state, and clocks are synchronized, i.e., they always advance at same speed. Hence, timed automata with integer-valued clocks and GCS are incomparable formalisms.

Our results are as follows. First, we investigate the *fairness problem* for GCS (which is crucial for the verification of liveness properties), that is checking the existence of infinite runs satisfying acceptance conditions à la Büchi. We show that this problem is decidable and PSPACE-complete; moreover, for the given GCS, one can compute a GC representation of the set of states from which there is a 'fair' infinite run. Next, we

[2] Hence, GC are closed under composition which captures the reachability relation in GCS for a fixed path in the control graph.

address verification problems of GCS against a *strict* extension, denoted by GCCTL*, of the logic MCCTL* (given in *complete* positive normal form) [12] obtained by adding transitional GC (we also allow existential quantification over variables in the underlying constraint language). Note that while MCCTL* is closed under negation, its strict extension GCCTL* is not (if we allow negation, the resulting logic would be undecidable also for small fragments). We show that while model-checking GCS against the *universal fragment* A–GCCTL* of GCCTL* is undecidable, model checking GCS against the existential fragment E–GCCTL* of GCCTL*, and satisfiability of both A–GCCTL* and E–GCCTL* are instead decidable and PSPACE-complete (which matches the complexity of model checking and satisfiability for the existential and universal fragments of standard CTL* [23]). Note that E–GCCTL* and A–GCCTL* are not dual. Moreover, for a given GCS $\mathcal{S}$ and E–GCCTL* formula φ, the set of states in $\mathcal{S}$ satisfying φ is *effectively* GC representable.

Since E–GCCTL* subsumes E–MCCTL*, and E–MCCTL* and A–MCCTL* are dual, our results imply PSPACE-completeness for model-checking (w.r.t. IRA or GCS) of both E–MCCTL* and A–MCCTL*. Hence, in particular, we solve complexity issues left open in [12] (see also [10]). Due to space reasons, many proofs are omitted and can be found in [11].

2 Preliminaries

Let $\mathbb{Z}$ (resp., $\mathbb{N}$) be the set of integers (resp., natural numbers). We fix a finite set $\mathit{Var} = \{x_1, \ldots, x_r\}$ of variables, a finite set of constants $\mathit{Const} \subseteq \mathbb{Z}$ such that $0 \in \mathit{Const}$, and a fresh copy of Var, $\mathit{Var}' = \{x'_1, \ldots, x'_r\}$. For an arbitrary finite set of variables V, an (integer) *valuation* over V is a mapping of the form $\nu : V \to \mathbb{Z}$, assigning to each variable in V an integer value. For $V' \subseteq V$, $\nu_{V'}$ denotes the restriction of ν to V'. For a valuation ν, by convention, we define $\nu(c) = c$ for all $c \in \mathbb{Z}$.

Definition 1. *[27]* A *gap-order constraint* (GC) over V and *Const* is a conjunction ξ of inequalities of the form $u - v \geq k$, where $u, v \in V \cup \mathit{Const}$ and $k \in \mathbb{N}$. W.l.o.g. we assume that for all $u, v \in V \cup \mathit{Const}$, there is at most one conjunct in ξ of the form $u - v \geq k$ for some k. A valuation $\nu : V \to \mathbb{Z}$ *satisfies* ξ if for each conjunct $u - v \geq k$ of ξ, $\nu(u) - \nu(v) \geq k$. We denote by $\mathit{Sat}(\xi)$ the set of such valuations.

Definition 2. *[12]* A (*gap-order*) *monotonicity graph* (MG) over V and *Const* is a directed weighted graph G with set of vertices $V \cup \mathit{Const}$ and edges $u \xrightarrow{k} v$ labeled by natural numbers k, and s.t.: if $u \xrightarrow{k} v$ and $u \xrightarrow{k'} v$ are edges of G, then $k = k'$. The set $\mathit{Sat}(G)$ of *solutions* of G is the set of valuations ν over V s.t. for each $u \xrightarrow{k} v$ in G, $\nu(u) - \nu(v) \geq k$. GC and MG are equivalent formalisms since there is a trivial linear-time computable bijection assigning to each GC ξ an MG $G(\xi)$ such that $\mathit{Sat}(G(\xi)) = \mathit{Sat}(\xi)$.[3]

The notation $G \models u < v$ means that there is an edge in G from v to u with weight $k > 0$. Moreover, $G \models u \leq v$ means that there is an edge of G from v to u, and $G \models u = v$

[3] MG are called Positive Graphose Inequality Systems in [12]. A different constraint graph representation of GC can be found in [27].

means $G \models u \leq v$ and $G \models v \leq u$. Also, we write $G \models u_1 \lhd_1 \ldots \lhd_{n-1} u_n$ to mean that $G \models u_i \lhd_i u_{i+1}$ for each $1 \leq i < n$, where $\lhd_i \in \{<, \leq, =\}$. A *transitional* GC (resp., *transitional* MG) is a GC (resp., MG) over $Var \cup Var'$ and *Const*. For valuations $\nu, \nu' : Var \to \mathbb{Z}$, we denote by $\nu \oplus \nu'$ the valuation over $Var \cup Var'$ defined as follows: $(\nu \oplus \nu')(x_i) = \nu(x_i)$ and $(\nu \oplus \nu')(x'_i) = \nu'(x_i)$ for $i = 1, \ldots, r$.

Definition 3. A *gap-order constraint system* (GCS) over *Var* and *Const* is a finite directed labeled graph $\mathcal{S}$ such that each edge is labeled by a *transitional* GC. $Q(\mathcal{S})$ denotes the set of vertices in $\mathcal{S}$, called *control points*, and $E(\mathcal{S})$ the set of edges.

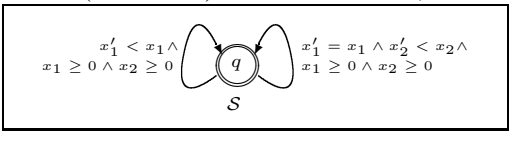
For a finite path $\wp$ of a GCS $\mathcal{S}$, $s(\wp)$ and $t(\wp)$ denote the source and target control points of $\wp$. For a finite path $\wp$ and a path $\wp'$ such that $t(\wp) = s(\wp')$, the composition of $\wp$ and $\wp'$, written $\wp\wp'$, is defined as usual.

The semantics of a GCS $\mathcal{S}$ is given by an infinite directed graph $[\![\mathcal{S}]\!]$ defined as:

- The vertices of $[\![\mathcal{S}]\!]$, called *states* of $\mathcal{S}$, are the pairs of the form (q, ν), where q is a control point of $\mathcal{S}$ and $\nu : Var \to \mathbb{Z}$ is a valuation over *Var*;
- There is an edge in $[\![\mathcal{S}]\!]$ from (q, ν) to (q', ν') iff there is a (labeled) edge in $\mathcal{S}$ of the form $q \xrightarrow{\xi} q'$ such that $\nu \oplus \nu' \in Sat(\xi)$. We say that the edge of $[\![\mathcal{S}]\!]$ from (q, ν) to (q', ν') is an *instance* of the edge $q \xrightarrow{\xi} q'$ of $\mathcal{S}$.

A path of $[\![\mathcal{S}]\!]$ is called a *run* of $\mathcal{S}$. The length $|\wp|$ (resp., $|\pi|$) of a path $\wp$ (resp., run π) of $\mathcal{S}$ is defined in the standard way. A *non-null* path of $\mathcal{S}$ is a path of $\mathcal{S}$ of non-null length. Let $\wp = q_0 \xrightarrow{\xi_0} q_1 \xrightarrow{\xi_1} q_2, \ldots$ be a path of $\mathcal{S}$. A run π of $\mathcal{S}$ is an *instance* of $\wp$ if π is of the form $\pi = (q_0, \nu_0) \to (q_1, \nu_1) \to (q_2, \nu_2), \ldots$ and for each i, $(q_i, \nu_i) \to (q_{i+1}, \nu_{i+1})$ is an instance of $q_i \xrightarrow{\xi_i} q_{i+1}$. Given $F \subseteq Q(\mathcal{S})$, an infinite run $(q_0, \nu_0) \to (q_1, \nu_1) \to \ldots$ of $\mathcal{S}$ is *fair w.r.t* F if for infinitely many $i \geq 0$, $q_i \in F$.

Example 1. The figure depicts a GCS $\mathcal{S}$ consisting of a unique control point q and two self-loops. Note that there is no infinite run since along any run, the pair (x_1, x_2) decreases strictly w.r.t. the lexicographic order (over $\mathbb{N} \times \mathbb{N}$). On the other hand, one can easily check that for each state (q, ν) with $\nu(x_1) > 0$ and $\nu(x_2) \geq 0$, the set of the lengths of the runs from (q, ν) is unbounded.



Convention: since we use MG representations to manipulate GC, we assume that the edge-labels in GCS are transitional MG.

2.1 Properties of Monotonicity Graphs

We recall some basic operations on MG [12] which can be computed in polynomial time. Furthermore, we define a sound and complete (w.r.t. satisfiability) approximation scheme of MG and show that the basic operations preserve soundness and completeness of this approximation. A different approximation scheme for GC can be found in [27].

A MG G is *satisfiable* if $Sat(G) \neq \emptyset$. Let G be a MG over V and *Const*. For $V' \subseteq V$, *the restriction of* G *to* V', written $G_{V'}$, is the MG given by the subgraph of G whose

set of vertices is $V' \cup \mathit{Const}$. For all vertices u, v of G, we denote by $p_G(u,v)$ the least upper bound (possibly ∞) of the weight sums on all paths in G from u to v (we set $p_G(u,v) = -\infty$ if there is no such a path). The MG G is *normalized* iff: (1) for all vertices u, v of G, if $p_G(u,v) > -\infty$, then $p_G(u,v) \neq \infty$ and $u \xrightarrow{p_G(u,v)} v$ is an edge of G, and (2) for all constants $c_1, c_2 \in \mathit{Const}$, $p_G(c_1,c_2) \leq c_1 - c_2$.

Proposition 1. [12] *Let G be a* MG *over V and Const. Then:*

1. *If G is* normalized *and $V' \subseteq V$, then G is satisfiable and every solution of $G_{V'}$ can be extended to a whole solution of G.*
2. *G is satisfiable $\Leftrightarrow$ G contains no loop with positive weight sum and for all $c_1, c_2 \in Const$, $p_G(c_1,c_2) \leq c_1 - c_2$ (this can be checked in polynomial time).*
3. *If G is satisfiable, then one can build in polynomial time an equivalent* normalized MG $\overline{G}$ *(i.e., $\mathit{Sat}(\overline{G}) = \mathit{Sat}(G)$), called the* closure *of G.*

According to Proposition 1, for a satisfiable MG G, we denote by $\overline{G}$ the closure of G. Moreover, for all unsatisfiable MG G over V and *Const*, we use a unique closure corresponding to some MG G_{nil} over V and *Const* such that $(G_{nil})_\emptyset$ is unsatisfiable (recall that $(G_{nil})_\emptyset$ denotes the MG given by the subgraph of G_{nil} whose set of vertices is *Const*). Now, we recall some effective operations on MG. Let $\mathit{Var}'' = \{x_1'', \ldots, x_r''\}$ be an additional copy of $\mathit{Var} = \{x_1 \ldots, x_r\}$.

Definition 4. [12] Let G be a MG on V and *Const* and G' be a MG on V' and *Const*.

1. **Projection:** if $V' \subseteq V$, the *projection of G over V'* is the MG given by $(\overline{G})_{V'}$.
2. **Intersection:** the *intersection $G \bigotimes G'$ of G and G'* is the MG over $V \cup V'$ and *Const* defined as: $u \xrightarrow{k} v$ is an edge of $G \bigotimes G'$ iff *either* (1) $u \xrightarrow{k} v$ is an edge of G (resp., G') and there is no edge from u to v in G' (resp., G), *or* (2) $k = \max(\{k', k''\})$, $u \xrightarrow{k'} v$ is an edge of G and $u \xrightarrow{k''} v$ is an edge of G'.
3. **Composition:** assume that G and G' are two transitional MG. Let G'' be obtained from G' by renaming any variable x_i' into x_i'' and x_i into x_i'. The *composition $G \bullet G'$* of G and G' is the transitional MG obtained from the *projection* of $G \bigotimes G''$ over $\mathit{Var} \cup \mathit{Var}''$ by renaming any variable x_i'' into x_i'.

By Definition 4 and Proposition 1, we easily obtain the following known result [12], which essentially asserts that MG (or, equivalently, GC) are closed under intersection and existential quantification.

Proposition 2. *Let G be a* MG *over V and Const and G' be a* MG *over V' and Const.*

1. ***Projection:** if G' is the projection of G over V', then for $\nu' : V' \to \mathbb{Z}$, $\nu' \in \mathit{Sat}(G')$ iff $\nu' = \nu|_{V'}$ for some $\nu \in \mathit{Sat}(G)$.*
2. ***Intersection:** for $\nu : V \cup V' \to \mathbb{Z}$, $\nu \in \mathit{Sat}(G \bigotimes G')$ iff $\nu|_V \in \mathit{Sat}(G)$ and $\nu|_{V'} \in \mathit{Sat}(G')$. Hence, for $V = V'$, $\mathit{Sat}(G \bigotimes G') = \mathit{Sat}(G) \cap \mathit{Sat}(G')$.*
3. ***Composition:** assume that G and G' are transitional* MG*. Then, for all $\nu, \nu' : \mathit{Var} \to \mathbb{Z}$. $\nu \oplus \nu' \in \mathit{Sat}(G \bullet G')$ iff $\nu \oplus \nu'' \in \mathit{Sat}(G)$ and $\nu'' \oplus \nu' \in \mathit{Sat}(G')$ for some $\nu'' : \mathit{Var} \to \mathbb{Z}$. Moreover, the composition operator $\bullet$ is associative.*

Approximation scheme: let K stand for $\max(\{|c_1 - c_2| + 1 \mid c_1, c_2 \in Const\})$. Note that $K > 0$. For each $h \in \mathbb{N}$, let $\lfloor h \rfloor_K = h$ if $h \leq K$, and $\lfloor h \rfloor_K = K$ otherwise.

Definition 5 (K-bounded MG). A MG is *K-bounded* iff for each of its edges $u \xrightarrow{k} v$, $k \leq K$. For a MG G over V and *Const*, $\lfloor G \rfloor_K$ denotes the K-bounded MG over V and *Const* obtained from G by replacing each edge $u \xrightarrow{k} v$ of G with the edge $u \xrightarrow{\lfloor k \rfloor_K} v$.

The proofs of the following propositions are in [11].

Proposition 3. *Let G be a* MG *over V and Const. Then, G is satisfiable iff $\lfloor G \rfloor_K$ is satisfiable. Moreover, $\lfloor \overline{G} \rfloor_K = \lfloor\ \overline{\lfloor G \rfloor_K}\ \rfloor_K$.*

Proposition 4. *For transitional* MG *G_1 and G_2, $\lfloor G_1 \bullet G_2 \rfloor_K = \lfloor \lfloor G_1 \rfloor_K \bullet \lfloor G_2 \rfloor_K \rfloor_K$.*

2.2 Results on the Reachability Relation in GCS

In this subsection, we give constructive results on the reachability relation in GCS.

Definition 6. *A transitional* MG *G is said to be* complete *if:*

- *for all $u, v \in Var \cup Var' \cup Const$, $G \models u \leq v \Rightarrow G \models u \lhd v$ for some $\lhd \in \{<, =\}$;*
- *for all $u, v \in Var \cup Const$, either $G \models u \leq v$ or $G \models v \leq u$;*
- *for all $u, v \in Var' \cup Const$, either $G \models u \leq v$ or $G \models v \leq u$.*

A GCS $\mathcal{S}$ is *complete* iff each MG in $\mathcal{S}$ is complete. Fix a *complete* GCS $\mathcal{S}$. For a finite path $\wp$ of $\mathcal{S}$, the *reachability relation w.r.t. $\wp$*, denoted by $\rightsquigarrow_\wp$, is the binary relation on the set of valuations over *Var* defined as: for all $\nu, \nu' : Var \to \mathbb{Z}$, $\nu \rightsquigarrow_\wp \nu'$ iff there is a run of $\mathcal{S}$ from $(s(\wp), \nu)$ to $(t(\wp), \nu')$ which is an instance of the path $\wp$. For a transitional MG G, *G characterizes the reachability relation $\rightsquigarrow_\wp$* iff $Sat(G) = \{\nu \oplus \nu' \mid \nu \rightsquigarrow_\wp \nu'\}$. We associate to each non-null finite path $\wp$ of $\mathcal{S}$ a transitional MG $G_\wp$ and a transitional K-bounded MG $G^{bd}_\wp$, defined by induction on $\wp$ as follows:

- $\wp = q \xrightarrow{G} q'$: $G_\wp = \overline{G}$ and $G^{bd}_\wp = \lfloor\ \overline{G}\ \rfloor_K$;
- $\wp = \wp' \wp''$, $|\wp'| > 0$, and $\wp'' = q \xrightarrow{G} q'$: $G_\wp = G_{\wp'} \bullet G$ and $G^{bd}_\wp = \lfloor G^{bd}_{\wp'} \bullet \lfloor\ G\ \rfloor_K \rfloor_K$.

Note that the composition operator preserves completeness, and for a transitional MG G, G is complete iff $\lfloor G \rfloor_K$ is complete. Thus, by a straightforward induction on the length of the path $\wp$ and by using Propositions 2 and 4, we obtain the following.

Proposition 5. *For a non-null finite path $\wp$ of $\mathcal{S}$, $G_\wp = \overline{G_\wp}$, and $G_\wp$ is complete and characterizes the reachability relation $\rightsquigarrow_\wp$. Moreover, $G^{bd}_\wp = \lfloor G_\wp \rfloor_K$ and is complete.*

Let $\mathcal{G}^K_\mathcal{S} = \{(\lfloor G_\wp \rfloor_K, s(\wp), t(\wp)) \mid \wp$ is a non-null finite path and $G_\wp$ is satisfiable$\}$. Note that $\mathcal{G}^K_\mathcal{S}$ is finite since the set of transitional K-bounded MG is finite. By Proposition 5, $\mathcal{G}^K_\mathcal{S}$ is exactly the set $\{(G^{bd}_\wp, s(\wp), t(\wp)) \mid \wp$ is a non-null finite path and $G^{bd}_\wp$ is satisfiable$\}$. It follows that we can compute the set $\mathcal{G}^K_\mathcal{S}$ by a simple transitive closure procedure. In particular, we obtain the following result.

Theorem 1. *For a* complete GCS *$\mathcal{S}$, each $G \in \mathcal{G}^K_\mathcal{S}$ is* complete*, and the size of $\mathcal{G}^K_\mathcal{S}$ is bounded by $O(|Q(\mathcal{S})|^2 \cdot (K+2)^{(2|Var|+|Const|)^2})$. Moreover, the set $\mathcal{G}^K_\mathcal{S}$ can be computed in time $O(|E(\mathcal{S})| \cdot |Q(\mathcal{S})|^2 \cdot (K+2)^{(2|Var|+|Const|)^2})$.*

Proof. An upper bound on the cardinality of the finite set of K-bounded transitional MG is $(K+2)^{(2|Var|+|Const|)^2}$, as each transitional K-bounded MG has at most $(2|Var|+|Const|)$ vertices and for all vertices u and v, there is at most one edge from u to v, and this edge has the form $u \xrightarrow{k} v$, where $k = 0, 1, \ldots, K$. It follows that the cardinality of $\mathcal{G}_{\mathcal{S}}^K$ is bounded by $|Q(\mathcal{S})|^2 \cdot (K+2)^{(2|Var|+|Const|)^2}$. By Proposition 5, $\mathcal{G}_{\mathcal{S}}^K$ is exactly the set $\{(G_{\wp}^{bd}, s(\wp), t(\wp)) \mid \wp \text{ is a non-null finite path and } G_{\wp}^{bd} \text{ is satisfiable}\}$. It follows that we can compute the set $\mathcal{G}_{\mathcal{S}}^K$ by the following transitive closure procedure: initialize a set B to $\{(\lfloor \overline{G} \rfloor_K, q, q') \mid q \xrightarrow{G} q' \text{ is an edge of } \mathcal{S} \text{ and } \lfloor G \rfloor_K \text{ is satisfiable}\}$ and repeat the following step until no more elements can be added to B (at this point $B = \mathcal{G}_{\mathcal{S}}^K$): for each $(G^{bd}, q, q') \in B$ and edge $q' \xrightarrow{G} q''$ of $\mathcal{S}$ include in B also $\lfloor G^{bd} \bullet \lfloor G \rfloor_K \rfloor_K$, unless it is unsatisfiable. Hence, the result follows. □

By [12] (see also [10]), for a GCS $\mathcal{S}$, the reflexive transitive closure of the transition relation of $[\![\mathcal{S}]\!]$ is effectively GC definable (a similar result can be found in [27], where it is shown that for Datalog queries with GC, there is a closed form evaluation). The GC representation can be computed by a fixpoint iteration whose termination is guaranteed by a suitable decidable well-quasi ordering defined over the set of transitional MG. By an insight in the proof given in [12] (see also [10]), and applying the K-bounded approximation scheme, we easily obtain the following. For details, see [11]. Note that we are not able to give an upper bound on the cardinality of the set $\mathcal{P}_{\mathcal{S}}$.

Theorem 2. *One can compute a* finite *set $\mathcal{P}_{\mathcal{S}}$ of non-null finite paths of $\mathcal{S}$ such that: for each non-null finite path $\wp'$ of $\mathcal{S}$ from q to q', there is a path $\wp \in \mathcal{P}_{\mathcal{S}}$ from q to q' so that $\lfloor G_{\wp'} \rfloor_K = \lfloor G_{\wp} \rfloor_K$, and $\rightsquigarrow_{\wp'}$ implies $\rightsquigarrow_{\wp}$.*

3 Checking Fairness

For a GCS $\mathcal{S}$ and a set F of control points of $\mathcal{S}$, we denote by $Inf_{\mathcal{S},F}$ the set of states of $\mathcal{S}$ from which there is an infinite run that is fair w.r.t. F. In this section, we show that the problem of checking for a given GCS $\mathcal{S}$ and set $F \subseteq Q(\mathcal{S})$, whether $Inf_{\mathcal{S},F} \neq \emptyset$ (*fairness problem*) is decidable and PSPACE-complete.

First, we give additional definitions. Let $\mathcal{S}$ be a GCS. We denote by $\lfloor \mathcal{S} \rfloor_K$ the GCS obtained from $\mathcal{S}$ by replacing each edge $q \xrightarrow{G} q'$ of $\mathcal{S}$ with the edge $q \xrightarrow{\lfloor G \rfloor_K} q'$.

A set U of states of $\mathcal{S}$ is MG *representable* if there is a family $\{\mathcal{G}_q\}_{q \in Q(\mathcal{S})}$ of finite sets of MG over *Var* and *Const* such that $\bigcup_{G \in \mathcal{G}_q} Sat(G) = \{\nu \mid (q, \nu) \in U\}$ for each $q \in Q(\mathcal{S})$. For a set $\mathcal{G}$ of MG, $\lfloor \mathcal{G} \rfloor_K$ denotes the set of K-bounded MG given by $\{\lfloor G \rfloor_K \mid G \in \mathcal{G}\}$. We extend the previous set operation to families of sets of MG in the obvious way. For $F \subseteq Q(\mathcal{S})$ and $q \in Q(\mathcal{S})$, $Inf_{\mathcal{S},F}^q$ denotes the set of states in $Inf_{\mathcal{S},F}$ of the form (q, ν) for some valuation ν. Moreover, $Inf_{\mathcal{S}}$ stands for $Inf_{\mathcal{S},Q(\mathcal{S})}$.

A MG G is *weakly normalized* if for all vertices u, v, $p_G(u, v) \geq 0$ (resp., $p_G(u, v) > 0$) implies $G \models v \leq u$ (resp., $G \models v < u$). Note that G is weakly normalized iff $\lfloor G \rfloor_K$ is weakly normalized. A transitional MG G is *(weakly) idempotent* iff $\lfloor G \bullet G \rfloor_K = \lfloor G \rfloor_K$.

3.1 Checking Fairness for Simple GCS

In this section, we solve the fairness problem for a restricted class of GCS.

Definition 7 (Simple GCS). *A (satisfiable) simple* GCS *is a* GCS *consisting of just two edges of the form* $q_0 \xrightarrow{G_0} q$ *and* $q \xrightarrow{G} q$ *such that* $q_0 \neq q$. *Moreover, we require that* $G_0 \bullet G$ *is satisfiable, and* G *is complete,* weakly *normalized, and idempotent.*

To present our results on simple GCS, we need additional definitions.

Definition 8 (Lower and upper variables). We denote by *MAX* (resp., *MIN*) the maximum (resp., minimum) of *Const*. For a transitional MG G and $y \in Var \cup Var'$, y is a *lower* (resp., *upper*) *variable* of G if $G \models y < MIN$ (resp., $G \models MAX < y$). Moreover, y is a *bounded variable* of G if $G \models MIN \leq y$ and $G \models y \leq MAX$.

Definition 9. A transitional MG is *balanced* iff for all $u, v \in Var \cup Const$ and $\lhd \in \{<, =\}$, $G \models u \lhd v$ iff $G \models u' \lhd v'$ (where for $u \in Var \cup Const$, we write u' to denote the corresponding variable in *Var′* if $u \in Var$, and u itself otherwise).

Fix a simple GCS $\mathcal{S}$ with edges $q_0 \xrightarrow{G_0} q$ and $q \xrightarrow{G} q$. Since G is idempotent, by the associativity of composition $\bullet$ and Proposition 4, we obtain that for each $k \geq 1$, $\lfloor G_0 \bullet \underbrace{G \bullet \ldots \bullet G}_{k \text{ times}} \rfloor_K = \lfloor G_0 \bullet G \rfloor_K$. Hence, $G_0 \bullet \underbrace{G \bullet \ldots \bullet G}_{k \text{ times}}$ and $\underbrace{G \bullet \ldots \bullet G}_{k \text{ times}}$ are satisfiable for each $k \geq 1$. Since G is complete, it follows that G is *balanced* as well. Moreover, since G is satisfiable and complete, a variable $y \in Var \cup Var'$ is either a lower variable, or an upper variable, or a bounded variable of G, where the "or" is exclusive. We denote by $\mathrm{L}_1, \ldots, \mathrm{L}_N$ (resp., $\mathrm{U}_1, \ldots, \mathrm{U}_M$) the lower (resp., the upper) variables of G in *Var*, and by $\mathrm{B}_1, \ldots, \mathrm{B}_H$ the bounded variables of G in *Var*. Hence, we can assume that

$$G \models \mathrm{L}_1 \lhd_2 \ldots \lhd_N \mathrm{L}_N < \mathrm{B}_1 \lhd'_2 \ldots \lhd'_H \mathrm{B}_H < \mathrm{U}_1 \lhd''_2 \ldots \lhd''_M \mathrm{U}_M$$

where $\lhd_2 \ldots \lhd_N, \lhd'_2 \ldots \lhd'_H, \lhd''_2 \ldots \lhd''_M \in \{<, =\}$. Since G is balanced it follows that the lower variables (resp., upper variables) of G in *Var′* are $\mathrm{L}'_1, \ldots, \mathrm{L}'_N$ (resp., $\mathrm{U}'_1, \ldots, \mathrm{U}'_M$), and the bounded variables of G in *Var′* are $\mathrm{B}'_1, \ldots, \mathrm{B}'_H$. Moreover,

$$G \models \mathrm{L}'_1 \lhd_2 \ldots \lhd_N \mathrm{L}'_N < \mathrm{B}'_1 \lhd'_2 \ldots \lhd'_H \mathrm{B}'_H < \mathrm{U}'_1 \lhd''_2 \ldots \lhd''_M \mathrm{U}'_M$$

Now, we define a polynomial-time checkable condition on simple GCS.

Definition 10 (termination condition). We say that G satisfies the termination condition iff one of the following holds:

lower variables: either $G \models \mathrm{L}_i < \mathrm{L}'_i$ *for some* $1 \leq i \leq N$,
or $G \models \mathrm{L}_i = \mathrm{L}'_i$ *and* $G \models \mathrm{L}'_j < \mathrm{L}_j$ *for some* $1 \leq i < j \leq N$.

upper variables: either $G \models \mathrm{U}'_i < \mathrm{U}_i$ *for some* $1 \leq i \leq M$,
or $G \models \mathrm{U}_j = \mathrm{U}'_j$ *and* $G \models \mathrm{U}_i < \mathrm{U}'_i$ *for some* $1 \leq i < j \leq M$.

Intuitively, the above condition asserts that either there is a lower (resp., upper) variable of G_{Var} whose value strictly increases (resp., decreases) along each run of $\mathcal{S}$, or there are two lower (resp., upper) variables of G_{Var} such that their distance strictly decreases along each run of $\mathcal{S}$. Let $\mathcal{TC}$ be the class of simple GCS satisfying the termination condition. By Definition 10, we easily obtain the following.

Proposition 6. *If $\mathcal{S} \in \mathcal{TC}$, then $Inf_{\mathcal{S}} = \emptyset$.*

It remains to consider the case when $\mathcal{S} \notin \mathcal{TC}$. We define two integers L and U as follows: L is the smallest $1 \leq i \leq N$ such that $G \models \mathrm{L}_i = \mathrm{L}'_i$ (if such an i does not exist, we set $L = N + 1$). Finally, U is the greatest $1 \leq i \leq M$ such that $G \models \mathrm{U}_i = \mathrm{U}'_i$ (if such an i does not exist, we set $U = 0$). Note that $1 \leq L \leq N + 1$ and $0 \leq U \leq M$. The set of *unconstrained variables* in Var, written *Unc*, consists of the lower variables L_i such that $1 \leq i < L$ and the upper variables U_j such that $U < j \leq M$. We denote by *Unc*′ the corresponding subset in *Var*′. Evidently, the following holds.

Lemma 1. *For a valuation $\nu_0 : Var \to \mathbb{Z}$, the set of valuations $\{\nu_{(Var \setminus Unc)} \mid (q, \nu)$ is reachable from (q, ν_0) in $[\![\mathcal{S}]\!]\}$ is finite.*

The proof of the following lemma is in [11]. Essentially, the result follows from Lemma 1 and the following property (which is a consequence of the idempotence of G): if $\mathcal{S} \notin \mathcal{TC}$, then $G \not\models \mathrm{U}'_i \leq \mathrm{U}_j$ and $G \not\models \mathrm{L}_h \leq \mathrm{L}'_k$ for all upper variables $\mathrm{U}'_i, \mathrm{U}_j$ and lower variables $\mathrm{L}_h, \mathrm{L}'_k$ in $Unc \cup Unc'$. In other terms, along a run of $\mathcal{S}$, the unconstrained upper (resp., lower) variables can increase (resp., decrease) arbitrarily.

Lemma 2. *Let $\mathcal{S} \notin \mathcal{TC}$. Then, $(q, \nu_0) \in Inf_{\mathcal{S}}$* iff *there is a finite run π of $\mathcal{S}$ from (q, ν_0) of the form $\pi = (q, \nu_0) \ldots (q, \nu) \ldots (q, \nu')(q, \nu'')$ such that $\nu''_{(Var \setminus Unc)} = \nu_{(Var \setminus Unc)}$.*

Now, we can prove the main result of this subsection.

Theorem 3. *Let $\mathcal{S} \notin \mathcal{TC}$. Then, $Inf_{\mathcal{S}}$ is* MG *representable and one can construct a* MG *representation of $Inf_{\mathcal{S}}$, written $\sigma(\mathcal{S})$, such that: (1) $\lfloor \sigma(\mathcal{S}) \rfloor_K$ can be computed in polynomial time, and (2) $\lfloor \sigma(\mathcal{S}) \rfloor_K = \lfloor \sigma(\lfloor \mathcal{S} \rfloor_K) \rfloor_K$ ($\lfloor \mathcal{S} \rfloor_K$ is simple and $\lfloor \mathcal{S} \rfloor_K \notin \mathcal{TC}$).*

Proof. By Theorem 2, one can compute a *finite* set $\mathcal{P}$ of non-null finite paths of $\mathcal{S}$ from q to q such that for each non-null finite path $\wp'$ of $\mathcal{S}$ from q to q, there is a path $\wp \in \mathcal{P}$ so that $\rightsquigarrow_{\wp'}$ implies $\rightsquigarrow_{\wp}$. Note that given $\wp \in \mathcal{P}$, the transitional MG $G_\wp$ (which characterizes the reachability relation $\rightsquigarrow_\wp$) has the form $\underbrace{G \bullet \ldots \bullet G}_{k \text{ times}}$ for some $k \geq 1$.

Let $G_=$ be the transitional MG corresponding to the GC given by $\bigwedge_{x \in Var \setminus Unc} x' = x$, and $\mathcal{G} = \{G_\wp \bullet (G_{\wp'} \bigotimes G_=) \mid \wp, \wp' \in \mathcal{P}\} \cup \{G_\wp \bigotimes G_= \mid \wp \in \mathcal{P}\}$. Then, $\sigma(\mathcal{S}) = \{\mathcal{G}^q, \mathcal{G}^{q_0}\}$, where $\mathcal{G}^q$ and $\mathcal{G}^{q_0}$ are defined as follows:

$\mathcal{G}^q = \{G' \mid G'$ is the projection of G'' over *Var* for some $G'' \in \mathcal{G}\}$

$\mathcal{G}^{q_0} = \{G' \mid G'$ is the projection of $G_0 \bullet G''$ over *Var* for some $G'' \in \mathcal{G}\}$

Correctness of the construction easily follows from Lemma 2. The second part of the theorem follows from Propositions 3–4, and the fact that for each $\wp \in \mathcal{P}$, $\lfloor G_\wp \rfloor_K = \lfloor G \rfloor_K$ (G is idempotent) and $\lfloor G_\wp \bigotimes G_= \rfloor_K = \lfloor \lfloor G_\wp \rfloor_K \bigotimes \lfloor G_= \rfloor_K \rfloor_K$. For details, see [11]. □

3.2 Checking Fairness for Unrestricted GCS

Fix a GCS $\mathcal{S}$. For a non-null finite path $\wp$ of $\mathcal{S}$ such that $s(\wp) = t(\wp)$ (i.e., $\wp$ is cyclic), $(\wp)^\omega$ denotes the infinite path $\wp\wp\ldots$. A infinite path $\wp$ of $\mathcal{S}$ of the form $\wp = \wp'(\wp'')^\omega$ is said to be *ultimately periodic*. By using Theorem 2 and Ramsey's Theorem (in its infinite version) [26], we show the following result.

Theorem 4 (Characterization Theorem). *Let $\mathcal{S}$ be a complete* GCS*, $F \subseteq Q(\mathcal{S})$, and $\mathcal{P}_\mathcal{S}$ be the finite set of non-null finite paths of $\mathcal{S}$ satisfying Theorem 2. Then, for each state s, $s \in Inf_{\mathcal{S},F}$* iff *there is an infinite run of $\mathcal{S}$ starting from s which is an instance of an* ultimately periodic *path $\wp_0 \cdot (\wp)^\omega$ such that $\wp_0, \wp \in \mathcal{P}_\mathcal{S}$, $s(\wp) \in F$, $G_{\wp_0} \bullet G_\wp$ is satisfiable, $G_\wp$ is idempotent, and $G_{\wp_0}$ and $G_\wp$ are complete and normalized.*

Proof. The left implication $\Leftarrow$ is obvious. For the right implication $\Rightarrow$, assume that $s \in Inf_{\mathcal{S},F}$. Then, there is an infinite run π of $\mathcal{S}$ starting from s which visits infinitely often states whose control points are in F. Moreover, there is an infinite path $\wp_\infty$ of $\mathcal{S}$ such that π is an instance of $\wp_\infty$.

Let us consider the finite set $\mathcal{P}_\mathcal{S}$ of non-null finite paths of $\mathcal{S}$ satisfying the statement of Theorem 2. For each $\wp \in \mathcal{P}_\mathcal{S}$ from q to q', we denote by $[\wp]$ the set of non-null finite paths $\wp'$ of $\mathcal{S}$ from q to q' such that $\rightsquigarrow_{\wp'}$ implies $\rightsquigarrow_\wp$, and $\lfloor G_{\wp'} \rfloor_K = \lfloor G_\wp \rfloor_K$. Let H be the *finite* set given by $H = \{[\wp] \mid \wp \in \mathcal{P}_\mathcal{S}\}$. For each non-null finite path $\wp'$ of $\mathcal{S}$, we associate to $\wp'$ a *color* given by some element $[\wp] \in H$ such that $\wp' \in [\wp]$ (note that such an element of H must exist). Let us consider the infinite path $\wp_\infty$. Then, there is a control point $q \in F$ such that $\wp_\infty$ is of the form $\wp_\infty = \wp_0\wp_1\wp_2 \ldots$, where for each $i \geq 1$, $\wp_i$ is a non-null (cyclic) path from q to q. Let us consider the set of positive natural numbers, and label each pair (i, j) of its elements with $i < j$ with the color of the subpath $\wp_i \ldots \wp_j$ of $\wp_\infty$. By Ramsey's Theorem (in its infinite version)[26], there is an infinite set I of positive natural numbers such that all the pairs (i, j) with $i, j \in I$ (and $i < j$) carry the same label in H, say $[\wp]$. It follows that $\wp_\infty$ can be written in the form $\wp_\infty = \wp'_0\wp'_1\wp'_2 \ldots$ such that $|\wp'_0| > 0$ and for all $i \geq 1$, $\wp'_i \in [\wp]$ and $\wp'_i\wp'_{i+1} \in [\wp]$. Hence, in particular, $\lfloor G_{\wp'_i} \rfloor_K = \lfloor G_\wp \rfloor_K$ and $\lfloor G_{\wp'_i\wp'_{i+1}} \rfloor_K = \lfloor G_\wp \rfloor_K$. By Proposition 4 and associativity of $\bullet$, we obtain that $\lfloor G_\wp \rfloor_K = \lfloor G_\wp \bullet G_\wp \rfloor_K$. Hence, $G_\wp$ is idempotent.

Let $\wp''_0 \in \mathcal{P}_\mathcal{S}$ such that $\wp'_0 \in [\wp''_0]$. Since π is an instance of $\wp_\infty = \wp'_0\wp'_1 \ldots$, and $\wp'_i \in [\wp]$ for each $i \geq 1$, it follows that there is an infinite run π' starting from s which is an instance of the *ultimately periodic* path $\wp''_0(\wp)^\omega$. Moreover, $s(\wp) = q \in F$, $\wp''_0, \wp \in \mathcal{P}_\mathcal{S}$, $G_{\wp''_0} \cdot G_\wp$ is satisfiable, $G_\wp$ is idempotent, and by Proposition 5, $G_{\wp''_0}$ and $G_\wp$ are complete and normalized, which concludes. □

Theorem 5. *Let $\mathcal{S}$ be a* GCS *and $F \subseteq Q(\mathcal{S})$. Then, $Inf_{\mathcal{S},F}$ is* MG *representable and one can construct a* MG *representation of $Inf_{\mathcal{S},F}$, written $\sigma_F(\mathcal{S})$, such that:*

1. *$\lfloor \sigma_F(\mathcal{S}) \rfloor_K$ can be computed in time $O(|E(\mathcal{S})| \cdot |Q(\mathcal{S})|^2 \cdot (K+2)^{(2|Var|+|Const|)^2})$;*
2. *$\lfloor \sigma_F(\mathcal{S}) \rfloor_K = \lfloor \sigma_F(\lfloor \mathcal{S} \rfloor_K) \rfloor_K$;*
3. *given $q \in Q(\mathcal{S})$ and a K-bounded* MG *G over Var, checking whether G is in the q-component of $\lfloor \sigma_F(\mathcal{S}) \rfloor_K$ can be done in polynomial space.*

Sketched proof. (A detailed proof is in [11]). We assume that $\mathcal{S}$ is complete (the general case easily follows). Let $\mathcal{P}_\mathcal{S}$ be the computable finite set of non-null finite paths of $\mathcal{S}$ satisfying the statement of Theorem 2, and let $\mathcal{F}_\mathcal{S}$ be the finite set of *simple* GCS constructed as: $\mathcal{S}' \in \mathcal{F}_\mathcal{S}$ iff $\mathcal{S}' \notin \mathcal{TC}$ and $\mathcal{S}'$ is a simple GCS consisting of two edges of the form $(\natural, s(\wp_0)) \xrightarrow{G_{\wp_0}} t(\wp_0)$ and $s(\wp) \xrightarrow{G_\wp} t(\wp)$ such that $\wp_0, \wp \in \mathcal{P}_\mathcal{S}$ and $s(\wp) = t(\wp) \in F$. By Theorem 3, for each $\mathcal{S}' \in \mathcal{F}_\mathcal{S}$ one can compute a MG representation $\mathcal{G}_{\mathcal{S}', in(\mathcal{S}')}$ (resp., $\mathcal{G}_{\lfloor \mathcal{S}' \rfloor_K, in(\mathcal{S}')}$) of $Inf_{\mathcal{S}'}^{(\natural, in(\mathcal{S}'))}$ (resp., $Inf_{\lfloor \mathcal{S}' \rfloor_K}^{(\natural, in(\mathcal{S}'))}$), where $(\natural, in(\mathcal{S}'))$

is the initial control point of $\mathcal{S}'$. Moreover, $\lfloor \mathcal{G}_{\mathcal{S}',in(\mathcal{S}')} \rfloor_K = \lfloor \mathcal{G}_{\lfloor \mathcal{S}' \rfloor_K,in(\mathcal{S}')} \rfloor_K$. Then, $\sigma_F(\mathcal{S})$ is given by

$$\sigma_F(\mathcal{S}) = \{ \bigcup_{\{\mathcal{S}' \in \mathcal{F}_\mathcal{S} | in(\mathcal{S}')=q\}} \mathcal{G}_{\mathcal{S}',in(\mathcal{S}')} \}_{q \in Q(\mathcal{S})}.$$

By Theorems 2 and 4, and Proposition 6, $\sigma_F(\mathcal{S})$ is a MG representation of $Inf_{\mathcal{S},F}$. Thus, the first part of the theorem holds. Now, let us consider Properties 1–3. Here, we focus on Property 1. Let $\mathcal{F}_{\mathcal{S},K}$ be the set of simple GCS $\mathcal{S}'$ such that $\mathcal{S}' = \lfloor \mathcal{S}'' \rfloor_K$ for some $\mathcal{S}'' \in \mathcal{F}_\mathcal{S}$. Since $\lfloor \mathcal{G}_{\mathcal{S}',in(\mathcal{S}')} \rfloor_K = \lfloor \mathcal{G}_{\lfloor \mathcal{S}' \rfloor_K,in(\mathcal{S}')} \rfloor_K$ for each $\mathcal{S}' \in \mathcal{F}_\mathcal{S}$, we obtain

$$\lfloor \sigma_F(\mathcal{S}) \rfloor_K = \{ \bigcup_{\{\mathcal{S}' \in \mathcal{F}_{\mathcal{S},K} | in(\mathcal{S}')=q\}} \lfloor \mathcal{G}_{\mathcal{S}',in(\mathcal{S}')} \rfloor_K \}_{q \in Q(\mathcal{S})}$$

Since for each $\mathcal{S}' \in \mathcal{F}_{\mathcal{S},K}$, $\lfloor \mathcal{G}_{\mathcal{S}',in(\mathcal{S}')} \rfloor_K$ can be computed in polynomial time in the size of $\mathcal{S}'$ (Theorem 3), it suffices to show that $\mathcal{F}_{\mathcal{S},K}$ can be computed in time $O(|E(\mathcal{S})| \cdot |Q(\mathcal{S})|^2 \cdot (K+2)^{(2|Var|+|Const|)^2})$. This last condition holds since: (i) for a GCS $\mathcal{S}'$, $\mathcal{S}'$ is simple iff $\lfloor \mathcal{S}' \rfloor_K$ is simple, (ii) for a simple GCS $\mathcal{S}''$, $\mathcal{S}'' \notin \mathcal{TC}$ iff $\lfloor \mathcal{S}'' \rfloor_K \notin \mathcal{TC}$, (iii) by Theorem 2, the set $\{(\lfloor G_\wp \rfloor_K, s(\wp), t(\wp)) \mid \wp \in \mathcal{P}_\mathcal{S}$ and $\lfloor G_\wp \rfloor_K$ is satisfiable$\}$ coincides with the set $\mathcal{G}^K_\mathcal{S} = \{(\lfloor G_\wp \rfloor_K, s(\wp), t(\wp)) \mid \wp$ is a non-null finite path of $\mathcal{S}$ and $\lfloor G_\wp \rfloor_K$ is satisfiable$\}$, and (iv) by Theorem 1, the set $\mathcal{G}^K_\mathcal{S}$ is computable in time $O(|E(\mathcal{S})| \cdot |Q(\mathcal{S})|^2 \cdot (K+2)^{(2|Var|+|Const|)^2})$. Thus, Property 1 holds. □

Corollary 1. *The fairness problem is* PSPACE*-complete.*

Proof. The upper bound easily follows from Property 3 in Theorem 5, and the fact that for each set $\mathcal{G}$ of MG, $\mathcal{G}$ contains a satisfiable MG iff $\lfloor \mathcal{G} \rfloor_K$ contains a satisfiable MG. Moreover, PSPACE-hardness follows from PSPACE-hardness of non-termination for Boolean Programs [22] and the fact that GCS subsume Boolean Programs. □

4 The Constrained Branching–Time Temporal Logic (GCCTL*)

We introduce the *constrained branching–time temporal logic* (GCCTL*) and investigate the related satisfiability and model checking problems. The logic GCCTL* is an extension of standard logic CTL* [18], where the set of atomic propositions is replaced with a subclass of Presburger constraints whose atomic formulas correspond to transitional GC. Formally, for a set of variables V and a set of constants *Const*, the *language of constraints* η, denoted by $\exists$GC, over V and *Const* is defined as follows:

$$\eta := u - v \geq k \mid \eta \vee \eta \mid \eta \wedge \eta \mid \exists x.\ \eta$$

where $u, v \in V \cup Const$, $k \in \mathbb{N}$, and $x \in V$. For a $\exists$GC constraint η and a valuation $\nu : V \to \mathbb{Z}$ over V, the satisfaction relation $v \models \eta$ is defined as follows (we omit the standard clauses for conjunction and disjunction):

- $\nu \models u - v \geq k \overset{\text{def}}{\Leftrightarrow} \nu(u) - \nu(v) \geq k$;
- $\nu \models \exists x.\ \eta \overset{\text{def}}{\Leftrightarrow}$ there is $c \in \mathbb{Z}$ such that $\nu[x \leftarrow c] \models \eta$.

where $\nu[x \leftarrow c](y) = \nu(y)$ if $y \neq x$, and $\nu[x \leftarrow c](y) = c$ otherwise. Note that $\exists$GC constraints are not closed under negation. Moreover, by Proposition 1(3) and Proposition 2(1) (see also [27]), GC are closed under existential quantification and quantification elimination can be done in polynomial time.

Syntax and semantics of GCCTL*: for the fixed set of variables *Var* and set of constants *Const*, the *state formulas* φ and *path formulas* ψ of GCCTL* are defined as:

$$\varphi := \top \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid A\,\varphi \mid E\,\varphi$$
$$\psi := \varphi \mid \eta \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathsf{O}\psi \mid \Box\psi \mid \psi \mathsf{U} \psi$$

where $\top$ denotes "true", E ("for some path") and A ("for all paths") are path quantifiers, η is a $\exists$GC constraint over *Var* $\cup$ *Var'* and *Const*, and O ("next"), U ("until"), and $\Box$ ("always") are the usual linear temporal operators. Since $\exists$GC constraints are not closed under negation, the logic is not closed under negation as well.[4] The set of state formulas φ forms the language GCCTL*. We also consider the existential and universal fragments E–GCCTL* and A–GCCTL* of GCCTL*, obtained by disallowing the universal and existential path quantifiers, respectively. GCCTL* formulas are interpreted over directed graphs $\mathcal{G} = \langle S, \rightarrow, \mu \rangle$ augmented with a mapping μ assigning to each vertex (or state) a valuation over *Var*. For an infinite path $\pi = s_0, s_1, \ldots$ of $\mathcal{G}$, 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 φ (resp. ψ), the satisfaction relation $(\mathcal{G}, s) \models \varphi$ (resp., $(\mathcal{G}, \pi) \models \psi$), meaning that φ (resp., ψ) holds at state s (resp., holds along π) in $\mathcal{G}$, is defined as (we omit the clauses for conjunction and disjunction):

- $(\mathcal{G}, s) \models A\,\psi \stackrel{\text{def}}{\Leftrightarrow}$ for each infinite path π from s, $(\mathcal{G}, \pi) \models \psi$;
- $(\mathcal{G}, s) \models E\,\psi \stackrel{\text{def}}{\Leftrightarrow}$ there exists an infinite path π from s such that $(\mathcal{G}, \pi) \models \psi$;
- $(\mathcal{G}, \pi) \models \varphi \stackrel{\text{def}}{\Leftrightarrow} (\mathcal{G}, \pi(0)) \models \varphi$;
- $(\mathcal{G}, \pi) \models \eta \stackrel{\text{def}}{\Leftrightarrow} \mu(\pi(0)) \oplus \mu(\pi(1)) \models \eta$;
- $(\mathcal{G}, \pi) \models \mathsf{O}\psi \stackrel{\text{def}}{\Leftrightarrow} (\mathcal{G}, \pi^1) \models \psi$;
- $(\mathcal{G}, \pi) \models \Box\psi \stackrel{\text{def}}{\Leftrightarrow}$ for all $i \geq 0$, $(\mathcal{G}, \pi^i) \models \psi$;
- $(\mathcal{G}, \pi) \models \psi_1 \mathsf{U} \psi_2 \stackrel{\text{def}}{\Leftrightarrow}$ there is $i \geq 0$. $(\mathcal{G}, \pi^i) \models \psi_2$ and for all $j < i$. $(\mathcal{G}, \pi^j) \models \psi_1$.

Note that the *dual* until operator $\widetilde{\mathsf{U}}$ can be expressed in the logic since: $\psi_1 \widetilde{\mathsf{U}} \psi_2 \equiv \Box\psi_2 \vee \big(\psi_2 \mathsf{U}(\psi_1 \wedge \psi_2)\big)$. A GCCTL* formula ξ is *satisfiable* if $(\mathcal{G}, s) \models \varphi$ for some labeled graph $\mathcal{G}$ and $\mathcal{G}$-state s. The *model checking problem of* GCS *against* GCCTL* is checking for a given GCS $\mathcal{S}$, state s of $\mathcal{S}$, and GCCTL* formula φ, whether $(\mathcal{G}(\mathcal{S}), s) \models \varphi$, where $\mathcal{G}(\mathcal{S})$ is obtained from $[\![\mathcal{S}]\!]$ by adding the mapping which assigns to each state of $\mathcal{S}$ the associated valuation over *Var*. We denote by $[\![\varphi]\!]_{\mathcal{S}}$ the set of states s of $\mathcal{S}$ such that $(\mathcal{G}(\mathcal{S}), s) \models \varphi$.

Example 2. Let us consider the requirement: "there is an infinite run from the given state such that variables x and y behave like clocks with rates at least k and k', respectively". This can be expressed by the E–GCCTL* formula

$$E\Box[((x' = 0) \vee (x' - x) \geq k) \wedge ((y' = 0) \vee (y' - y) \geq k')$$

[4] If we allow negation, then the successor relation is definable and by [17], basic decision problems become undecidable.

We can also use our framework to solve verification of non-local constraints (between variables at states arbitrarily far away from each other), which are not directly expressible in GCCTL*. As a relevant example, we consider *unboundedness requirements* on the values of a given variable along an infinite run. For each $x \in Var$, let us denote by ξ_x a special atomic formula (*unboundedness constraint*) that hold along an infinite run π iff the set of x-values along π is unbounded. Let E–GCCTL$^*_{Unb}$ be the extension of E–GCCTL* with these constraints. By the following result (whose proof is in [11]), it follows that model checking GCS against E–GCCTL$^*_{Unb}$ can be reduced in polynomial time to model checking GCS against E–GCCTL*.

Theorem 6. *Let $\mathcal{S}$ be a* GCS *over Var and φ be a* E–GCCTL$^*_{Unb}$ *formula over Var. Then, one can construct in polynomial-time an extension Var_{ext} of Var, a* GCS $\mathcal{S}_{ext}$ *over Var_{ext}, and a* E–GCCTL* *formula $f(\varphi)$ over Var_{ext} such that: for each state s of $\mathcal{S}$, one can compute in linear-time a state s_{ext} of $\mathcal{S}_{ext}$ so that*

$$(\mathcal{G}(\mathcal{S}), s) \models \varphi \text{ if and only if } (\mathcal{G}(\mathcal{S}_{ext}), s_{ext}) \models f(\varphi)$$

Decision procedures. By [12], model checking GCS against GCCTL* is undecidable. It is straightforward to extend this negative result to model checking GCS against A–GCCTL* (see [11]). In the following, we show that model checking GCS against E–GCCTL*, and satisfiability for E–GCCTL* and A–GCCTL* are instead decidable and PSPACE-complete.

Theorem 7. *Given a* GCS $\mathcal{S}$ *and a* E–GCCTL* *formula φ, $[\![\varphi]\!]_{\mathcal{S}}$ is* MG *representable and one can construct a* MG *representation of $[\![\varphi]\!]_{\mathcal{S}}$, written $\pi(\mathcal{S}, \varphi)$, such that: (1) $\lfloor \pi(\mathcal{S}, \varphi) \rfloor_K$ can be built in time $O(|E(\mathcal{S})| \cdot |Q(\mathcal{S})|^2 \cdot 2^{O(|\varphi|)} \cdot (K+2)^{O((2|Var|+|Const|)^2)})$, and (2) for a K-bounded* MG *G on Var and $q \in Q(\mathcal{S})$, checking whether G is in the q-component of $\lfloor \pi(\mathcal{S}, \varphi) \rfloor_K$ can be done in space polynomial in the sizes of $\mathcal{S}$ and φ.*

Sketched proof. (A detailed proof is in [11]). Fix a GCS $\mathcal{S}$. For a (state) E–GCCTL* formula φ, we construct $\pi(\mathcal{S}, \varphi)$ and prove Properties 1–2 by induction on the structure of φ. Note that we can assume that each $\exists$GC constraint occurring in φ is a disjunction of transitional GC. The non-trivial case is when $\varphi = E\,\psi$ for some path formula ψ. Let X be the set of state formulas θ such that there is an occurrence of θ in ψ which is not in the scope of E. By induction hypothesis, we can assume that the result holds for each formula in X. By a generalization of the standard construction for LTL model-checking, we show the following: one can build two GCS $\mathcal{S}_\varphi$ and $\mathcal{S}^{bd}_\varphi$ with set of control points $Q(\mathcal{S}) \times Q_\varphi$, where $Q_\varphi = O(2^{|\varphi|})$, and two subsets $Q^0_\varphi \subseteq Q_\varphi$ and $F \subseteq Q(\mathcal{S}_\varphi)$ such that the following holds:

Claim 1: $(q, \nu) \in [\![\varphi]\!]_{\mathcal{S}}$ *iff* $((q, q_0), \nu) \in Inf_{\mathcal{S}_\varphi, F}$ for some $q_0 \in Q^0_\varphi$.

Claim 2: $\mathcal{S}^{bd}_\varphi$ can be built in time $O(|E(\mathcal{S})| \cdot |Q(\mathcal{S})|^2 \cdot 2^{O(|\varphi|)} \cdot (K+2)^{O((2|Var|+|Const|)^2)})$ starting from $\mathcal{S}$ and $\{\lfloor \pi(\mathcal{S}, \theta) \rfloor_K \mid \theta \in X\}$. Moreover, $E(\mathcal{S}^{bd}_\varphi)$ has cardinality bounded by $|E(\mathcal{S})| \cdot 2^{O(|\varphi|)} \cdot (K+2)^{(2|Var|+|Const|)^2}$, and $\mathcal{S}^{bd}_\varphi = \lfloor \mathcal{S}_\varphi \rfloor_K$.

Let $\sigma_F(\mathcal{S}_\varphi)$ be the *computable* MG representation of $Inf_{\mathcal{S}_\varphi, F}$ satisfying the statement of Theorem 5. Then, for each $q \in Q(\mathcal{S})$, the q-component of $\pi(\mathcal{S}, \varphi)$ is the union of the (q, q_0)-components of $\sigma_F(\mathcal{S}_\varphi)$ such that $q_0 \in Q^0_\varphi$. By Claim 1, it follows that $\pi(\mathcal{S}, \varphi)$

is a *computable* MG representation of $[\![\varphi]\!]_{\mathcal{S}}$. For the remaining part of the theorem, here, we focus on Property 1. By Claim 2, $\mathcal{S}^{bd}_{\varphi} = \lfloor \mathcal{S}_{\varphi} \rfloor_K$, hence, by Property 2 of Theorem 5, $\lfloor \sigma_F(\mathcal{S}_{\varphi}) \rfloor_K = \lfloor \sigma_F(\mathcal{S}^{bd}_{\varphi}) \rfloor_K$. Thus, since $Q(\mathcal{S}^{bd}_{\varphi})$ has cardinality bounded by $|Q(\mathcal{S})| \cdot 2^{O(|\varphi|)}$, by Property 1 of Theorem 5, and Claim 2, Property 1 follows. □

Theorem 8. *The model checking problem of* GCS *against* E–GCCTL* *and satisfiability of* E–GCCTL* *and* A–GCCTL* *are* PSPACE*-complete.*

Sketched proof. By Theorem 7, checking for a GCS $\mathcal{S}$, control point q, and E–GCCTL* formula φ, whether $(\mathcal{G}(\mathcal{S}), (q, \nu)) \models \varphi$ for some valuation ν, is in PSPACE. By an easy linear-time reduction to this last problem, the upper bound for model checking GCS against E–GCCTL* follows. The upper bounds for satisfiability of E–GCCTL* and A–GCCTL* easily follow by a linear-time reduction to the considered model checking problem. For details, see [11]. Finally, the lower bounds directly follow from PSPACE-hardness of model checking and satisfiability for the existential and universal fragments of standard CTL* (see, e.g., [23]). □

5 Concluding Remarks

We focus on the logic GCCTL*. An intriguing question left open is the decidability status for satisfiability of full GCCTL*. Moreover, it would be interesting to investigate extensions of GCCTL* which allow to compare variables at states arbitrarily far away from each other. A possibility would be to permit atomic formulas of the form $x - \Diamond y \geq k$, or $\Diamond y - x \geq k$, or $x - \Box y \geq k$, or $\Box y - x \geq k$ ($k \in \mathbb{N}$), where $\Diamond y$ means "for some future value of y" and $\Box y$ means "for each future value of y". Thus, for example, $x - \Box y \geq 1$ asserts that the future values of y remain below the current value of x. We conjecture that with this extension, Theorem 8 still holds.

References

1. Abdulla, P.A., Delzanno, G.: On the coverability problem for constrained multiset rewriting. In: Proc. 5th AVIS (2006)
2. Abdulla, P.A., Delzanno, G., Rezine, A.: Approximated parameterized verification of infinite-state processes with global conditions. Formal Methods in System Design 34(2), 126–156 (2009)
3. Alur, R., Dill, D.L.: Automata For Modeling Real-Time Systems. In: Paterson, M. (ed.) ICALP 1990. LNCS, vol. 443, pp. 322–335. Springer, Heidelberg (1990)
4. Ben-Amram, A.M.: Size-change termination with difference constraints. ACM Transactions on Programming Languages and Systems 30(3) (2008)
5. Ben-Amram, A.M.: Size-change termination, monotonicity constraints and ranking functions. Logical Methods in Computer Science 6(3) (2010)
6. Boigelot, B.: Symbolic methods for exploring infinite state spaces. PhD thesis, Université de Liège (1998)
7. Bouajjani, A., Bozga, M., Habermehl, P., Iosif, R., Moro, P., Vojnar, T.: Programs with Lists Are Counter Automata. In: Ball, T., Jones, R.B. (eds.) CAV 2006. LNCS, vol. 4144, pp. 517–531. Springer, Heidelberg (2006)

8. Bouajjani, A., Echahed, R., Habermehl, P.: On the verification problem of nonregular properties for nonregular processes. In: LICS 1995, pp. 123–133. IEEE Computer Society Press (1995)
9. Bozga, M., Gîrlea, C., Iosif, R.: Iterating Octagons. In: Kowalewski, S., Philippou, A. (eds.) TACAS 2009. LNCS, vol. 5505, pp. 337–351. Springer, Heidelberg (2009)
10. Bozzelli, L., Gascon, R.: Branching-Time Temporal Logic Extended with Qualitative Presburger Constraints. In: Hermann, M., Voronkov, A. (eds.) LPAR 2006. LNCS (LNAI), vol. 4246, pp. 197–211. Springer, Heidelberg (2006)
11. Bozzelli, L., Pinchinat, S.: Verification of gap-order constraint abstractions of counter systems. Technical report (2011), http://clip.dia.fi.upm.es/~lbozzelli
12. Cerans, K.: Deciding Properties of Integral Relational Automata (Extended Abstract). In: Shamir, E., Abiteboul, S. (eds.) ICALP 1994. LNCS, vol. 820, pp. 35–46. Springer, Heidelberg (1994)
13. Comon, H., Cortier, V.: Flatness Is Not a Weakness. In: Clote, P.G., Schwichtenberg, H. (eds.) CSL 2000. LNCS, vol. 1862, pp. 262–276. Springer, Heidelberg (2000)
14. Comon, H., Jurski, Y.: Multiple Counters Automata, Safety Analysis and Presburger Arithmetic. In: Vardi, M.Y. (ed.) CAV 1998. LNCS, vol. 1427, pp. 268–279. Springer, Heidelberg (1998)
15. Demri, S., D'Souza, D.: An automata-theoretic approach to constraint LTL. Information and Computation 205(3), 380–415 (2007)
16. Demri, S., Finkel, A., Goranko, V., van Drimmelen, G.: Towards a Model-Checker for Counter Systems. In: Graf, S., Zhang, W. (eds.) ATVA 2006. LNCS, vol. 4218, pp. 493–507. Springer, Heidelberg (2006)
17. Demri, S., Gascon, R.: Verification of qualitative Z constraints. Theoretical Computer Science 409(1), 24–40 (2008)
18. Emerson, E.A., Halpern, J.Y.: Sometimes and not never revisited: On branching versus linear time. Journal of the ACM 33(1), 151–178 (1986)
19. Finkel, A., Leroux, J.: How to Compose Presburger-Accelerations: Applications to Broadcast Protocols. In: Agrawal, M., Seth, A.K. (eds.) FSTTCS 2002. LNCS, vol. 2556, pp. 145–156. Springer, Heidelberg (2002)
20. Fribourg, L., Richardson, J.: Symbolic Verification with Gap-Order Constraints. In: Gallagher, J.P. (ed.) LOPSTR 1996. LNCS, vol. 1207, pp. 20–37. Springer, Heidelberg (1997)
21. Ibarra, O.: Reversal-bounded multicounter machines and their decision problems. Journal of ACM 25(1), 116–133 (1978)
22. Jonson, N.D.: Computability and Complexity from a Programming Perspective. Foundations of Computing Series. MIT Press (1997)
23. Kupferman, O., Vardi, M.Y.: An automata-theoretic approach to modular model checking. ACM Trans. Program. Lang. Syst. 22(1), 87–128 (2000)
24. Minsky, M.: Computation: Finite and Infinite Machines. Prentice Hall (1967)
25. Peterson, J.L.: Petri Net Theory and the Modelling of Systems. Prentice-Hall (1981)
26. Ramsey, F.: On a problem of formal logic. Proceedings of the London Mathematical Society 30, 264–286 (1930)
27. Revesz, P.Z.: A Closed-Form Evaluation for Datalog Queries with Integer (Gap)-Order Constraints. Theoretical Computer Science 116(1-2), 117–149 (1993)