Counter automata, fixpoints and additive theories *

Hubert Comon and Yan Jurski
LSV, ENS Cachan
61 av. président Wilson
94235 Cachan cedex
France
E-mail {comon,jurski}@lsv.ens-cachan.fr

December 13, 1999

Abstract

We consider finite automata equipped with finitely many registers (or counters). Each transition can be fired only if the associated formula, the *guard* is satisfied by the current register values. Then the transition updates the registers and the control moves to a new state. In our framework, the guards are conjunctions of atomic constraints $y_i \# y_j + c_{i,j}$ where y_i is either x'_i or x_i, the values of the counter i respectively after and before the transition, and $\#$ is any relational symbol in $\{=, \leq, \geq, >, <\}$. We show that the binary relation between the registers values, which is defined by a counter automaton whose control is *flat* (no nested loops), is definable in the additive theory of $\mathbb{N}$ (or $\mathbb{Z}$ or $\mathbb{Q}$ or $\mathbb{R}$ depending on the type of the counters). In particular, we show that adding a fixed point to the above guards, we stay within Presburger arithmetic (resp. additive theory of $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$).

We actually prove a slightly more general result: the number of transition steps can be a free variable of the formula. We also show that whether a given loop can be infinitely iterated or not is decidable.

As a consequence of these results, reachability and emptiness are decidable for flat counter automata.

*This is the extended version of a paper which appeared in Proc. *Computer Aided verification* under the title "Multiple counter automata, safety analysis and Presburger arithmetic"

1 Introduction

Finite state automata provide a nice framework for the verification of reactive systems. Their main advantage is the equivalence between recognizability and definability in some decidable logic (e.g. Monadic Second Order Logic or some of its fragments such as temporal logics). This allows to verify fully automatically that some structure defined by an automaton satisfies a given formula. Automata techniques are even optimal for the model checking of temporal formulas (see [3]). The counterpart of these nice properties is the relatively weak expressiveness of the finite state automata models. Many actual reactive systems require additional data structures in order to be described in an accurate way. Many models extending finite automata have been introduced in the literature. The most well-known one is probably timed automata [1] which allow to consider some "real time" constraints while keeping the nice decidability properties (with a higher complexity).

One of the most important purpose of verification is the so-called *safety analysis* which reduces most of the time to the following question: "is a bad state reachable from the initial configuration ?" For finite state automata, it is not difficult to compute all reachable states. This is however a more delicate question with infinite states systems: the computability of reachable configurations depends on the model under consideration. Here, we aim at contributing to this question by giving some decidability results. The model we consider is a "multiple counters automaton". A configuration is not only described by a state of the system, but also by the values of finitely many *counters* which may take arbitrary (integer or real) values. Such counters may be used to represent e.g. the distance covered by a car or the speed of a train.

In the model we consider, each transition is associated with a *guard*: the transition can be fired only if the actual values of the counters satisfy the associated formula. The guard also relates the current and the next counter values, expressing an update of the counters. This will be expressed formally later on. To have an example in mind, a typical guard can be $x' = x + 1$, which means "increment counter x" or $x < y \wedge x' = y$ which means "if the value of x is smaller than the value of y, then transfer y to x. Such a model is used in several papers, such as in [15].

Minsky machines [17] are a particular subclass of such multiple counter automata. This implies that reachability is undecidable in general. There are two ways to overcome this problem: either we restrict the class of models we consider or else we consider (lower or upper depending on the problem) approximations of the model. Essentially, these two points of view are not different: if we find some appropriate restriction of the models, then this corresponds to an appropriate class of approximations. The question then is to find a class which is as expressive as possible and for which reachability is still decidable. Assume for instance that counters may take integer values. Then we would like to describe sets of (reachable) configurations by Presburger formulas, assuming that the guards of the transitions are also expressed in Presburger arithmetic. This is not always possible because loops in the automaton yield fixed points which correspond to infinite disjunctions of Presburger formulas and actually, even with a single state, the set of reachable counters values can be a non-recursive set of integers (or reals if we consider real-valued counters).

In this paper, we consider a fragment (or an approximation) yielding a decidable class. Our main result is the following: assume that the counters values before (unprimed names) and after a transition (primed names) are solutions of conjunctions of atomic formulas of the form $x \# y' + c$ or $x \# y + c$ or $x' \# y' + c$ or $x \# c$ or $x' \# c$ where $c \in \mathbb{Z}$ (resp $c \in \mathbb{R}$) and $\# \in \{\leq, \geq, =, >, <\}$. Then we show that the fixed point of iterating (or composing) such a transition is expressible in Presburger arithmetic (resp. the additive theory of real numbers). For automata with multiple nested iterations, then the same result holds provided that intermediate fixed points are expressible in the adequate fragment.

Even more, we show that, for any *flat counter automaton* A, for every pair of states q, q', the ternary relation $\vec{V} \xrightarrow[q,A,q']{n} \vec{V'}$ meaning "the configuration $(q', \vec{V'})$ is reachable in A from $(q, \vec{V})$ in n transition steps" whose free variables are $\vec{V}, \vec{V'}, n$, is expressible in a (decidable) additive theory. A flat automaton is an automaton with a particular control: we assume that there is at most one loop on each state. This is of course a strong restriction. However, all timed automata fall into this class [8], which shows its relevance.

We also show some side results, for instance the infinite iterability of a loop is decidable (which may be useful in checking liveness properties).

Related works

There are several authors who considered other fragments and other approximations. Let us briefly mention them and compare with our result. N. Halbwachs in [15] considers a similar model. A priori, the fixed point of a loop whose guard is $g(\vec{x}, \vec{x'})$ is the set of counters values which satisfy the

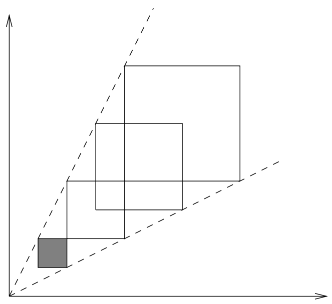


Figure 1: Successive values of the counters

infinite disjunction (n is the number of iterations)

$$\bigvee_{n=0}^{+\infty} (\exists \vec{x_1} \ldots \exists \vec{x_n} . g(\vec{x}, \vec{x_1}) \wedge \ldots g(\vec{x_n}, \vec{x'}))$$

N. Halbwachs, following [9], considers a *widening* operation ∇ and he computes an upper approximation of the above infinite disjunction $\bigvee_{n=0}^{+\infty} P_i$ by considering the limit of $P_0, P_0 \nabla P_1, P_0 \nabla P_1 \nabla P_2 ...$ which is always reached after finitely many steps. Basically, the widening construction removes some of the constraints of either argument, until one of the constraint subsumes the other. Several strategies for computing the polyhedron are given in [15][1]. Consider however the following example:

Example 1 There is only one state, one loop and two counters: initially the counters values satisfy $1 \leq x \leq 2, 1 \leq y \leq 2$. The guard of the transition is given by $x + 1 \leq x' \leq x + 2, y + 1 \leq y' \leq y + 2$. Successive values of the counters after each iteration are represented in figure 1. Following [15], we would get the whole quarter of plan as an upper approximation. The exact computation is however possible, yielding $2y \geq x \wedge 2x \geq y$ (which is depicted on the figure using dashed lines). The guards satisfy our conditions, hence we will get this exact computation using our result.

P. Revesz in [20] also considers similar fixed point computation. The guards are of the form $x'_i \leq x_j + k$ with $k \geq 0$, which disallows for instance some equalities: it is not possible to express e.g. $x' = x + 1$. Moreover, he does not consider the binary (or ternary) reachability relation, but only the

[1]even if the guards in [15] are only linear substitutions, it can be extended to certain kind of linear inequality, and for instance to our example.

computation of the reachable configurations from a given initial one (i.e. the computation of $Post^*$), which is a particular case. On the other hand, he is able to handle several loops. The application of this result to verification is investigated in [14]. It is also extended, allowing periodicity conditions in [22].

B. Boigelot and P. Wolper consider in [6] guards of the form $x'_i = x_i + b$ plus additional guards involving only unprimed variables. They also get an exact fixed point computation in Presburger's arithmetic. Our result is more general in the sense that we may also have relations $x'_i \# x_j + b$ (i.e. relations between different counters) and inequalities as well. On the other hand, the precondition on unprimed variables is more general in [6] than in our result.

L. Fribourg and H. Olsen [12] consider a similar situation as in [6], except that their preconditions are less general. On the other hand, they consider the case of several loops, which is not the case in [6].

B. Boigelot in [4] characterizes precisely the functions from $\mathbb{Z}^n$ to $\mathbb{Z}^n$ of the form $f(x) = Ax + b$ such that the set of iterations of f is definable in weak monadic second order logic. This result is the most accurate one for the guards $x' = Ax + b$. However, there is no inequality here, no guard relating unprimed variables and no guard relating unprimed variables.

The Maryland team in [7] considers guards which can be arbitrary Presburger formulas. This model is more general than ours. However, they do not have any decidability result (the model is too expressive). They provide with approximations computations which yield semi-decision algorithms.

Organization of the paper

In section 2 we define the first-order theories which will be used in the following: Presburger arithmetic, additive theories of real or rational numbers. In section 3, we define multiple counter automata and introduce the "flat" restriction.

Our results can actually be stated in three different ways. In section 4 we show the equivalence of these three formulations. The first one is the expression of the reachability relation (that is what we were talking about so far) and the second one is the expression of fixed points within an additive theory. Third, we use a classical correspondence between constraints and graphs ([18]): n iterations of a loop correspond to n compositions of a graph. This yields another formulation of the result: the relation defined by a parametrized graph is expressible in an additive theory. Once we state these three equivalent formulations, we may use at each time the most convenient one.

Section 5 is dedicated to the proof of our main theorem, as well as some side results. Finally, we derive in section 6 the effective expression of the reachability relation for flat counter automata and we discuss some applications and perspectives in section 7.

2 Additive theories

In what follows D is a domain which can be one of the following: $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{Q}_+$, $\mathbb{R}$, $\mathbb{R}_+$. Variables will be interpreted in D. However, let us emphasize that any constant which will be considered in any syntactic expression (graphs, formulas, automata) will be assumed to belong to $D \cap \mathbb{Q}$: we cannot handle real constants such as $\sqrt{2}$. We will not recall this hypothesis in the following.

We consider the *additive theory of* D: $\mathcal{T}_D$. It is defined as follows.

2.1 Syntax

Terms are expressions $x_1 + \ldots + x_k$ where $x_1, \ldots, x_k$ are either variables or elements of $D \cap \mathbb{Q}$. (We assume the constants in $\mathbb{Q}$ for effectivity reasons).

Atomic formulas are $t \leq t'$, $t < t'$, $\mathsf{Int}(t)$ where t, t' are *terms*.

Formulas are arbitrary first-order formulas built out of the atomic ones.

We may use syntactic facilities. Here are some examples:

- We write $2x$ instead of $x + x$ and, more generally, if n is a fixed positive integer, nx instead of $\underbrace{x + \ldots + x}_{n}$
- $t = t'$ where t, t' are two terms is defined by $t \leq t' \wedge t' \leq t$.
- $\phi[\frac{4}{3}x]$, a formula ϕ in which x is free and containing the term $\frac{4}{3}x$ is an abbreviation for $\exists y.3y = 4x \wedge \phi[y]$. More generally, we may use multiplication and addition by fixed constants in $\mathbb{Q}$.

2.2 Semantics

The interpretation of the symbols are given by their usual definition in D. $\mathsf{Int}(x)$ is interpreted as $\mathbb{N}$ (which is always a subset of D).

$\mathcal{T}_D$ is the set of sentences which hold true in D.

A relation $R \subseteq D^n$ is *definable* in $\mathcal{T}_D$ if there is a formula of the additive theory of D with n free variables and whose models are exactly the assignments which are in R.

2.3 Decidability results

Each of the theories $\mathcal{T}_D$ can be completely axiomatized. In particular $\mathcal{T}_D$ is a decidable theory. For instance, when $D = \mathbb{N}$, we get Presburger arithmetic [19]. When $D = \mathbb{R}_+$, we get a theory which can be embedded in the sequential calculus S1S, identifying real number with their infinite binary development and stacking the integral part and the fractional part (see e.g. [5] for more on this subject). For instance $\frac{4}{3}$ can be written as an infinite word over the alphabet $\{0,1\}^2$ as $\substack{100000\\010101}\cdots$ corresponding to the development $1.010101\ldots$

3 Multiple counters automata

Let D be $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{Q}_+, \mathbb{R}$ or $\mathbb{R}_+$. Let $\vec{x}$ be a vector of variables. Each element $x_i \in \vec{x}$ is associated with another symbol x'_i in such a way that $\vec{x'}$ is a disjoint copy of $\vec{x}$.

The class $\mathcal{C}(\vec{x})$ of constraints consists of formulas whose free variables are in $\vec{x} \cup \vec{x'}$ and which are conjunctions of atomic formulas of one of the following forms: $x < y + c$, $x \leq y + c$, $x + c < y$, $x + c \leq y$, $x + c = y$, $x < c$, $c < x$, $x \leq c$, $c \leq x$, $c = x$ where x, y are variables and c is a constant in D. Such formulas are interpreted in D with the usual meaning of relational and functional symbols.

Definition 1 *A* multiple counter automaton *is a tuple $(Q, A, \vec{x}, T)$ where Q is a finite set of states, A is a finite set of labels, $\vec{x}$ is a finite set of counters and T is a finite set of transitions: $T \subseteq Q \times A \times \mathcal{C}(\vec{x}) \times Q$.*

A configuration *of the automaton is a pair $(q, \vec{v}) \in Q \times D^N$ where $N = |\vec{x}|$. The automaton may* move *from a configuration $(q, \vec{v})$ to a configuration $(q', \vec{v'})$ iff there is a transition $(q, a, c, q') \in T$ such that $\vec{v}, \vec{v'} \models c$.*

If γ_1, γ_2 are two configurations, we write $\gamma_1 \xrightarrow[\mathcal{A}]{a} \gamma_2$ (resp. $\gamma_1 \xrightarrow[\mathcal{A}]{t} \gamma_2$) if the automaton $\mathcal{A}$ may move from γ_1 to γ_2 using a transition whose label is a (resp. using transition t). The subscript $\mathcal{A}$ will be omitted most of the time, as well as the label a which plays no role in the present work. We will however use other sub- or super- scripts as we will see later.

Example 2 We show in figure 2 a toy example of a pay phone controller. There are two counters: x, y. x is the available credit (number of quarters which have been put in the machine) and y counts the number of units of time spent while the user is connected. (The actual time duration of a unit

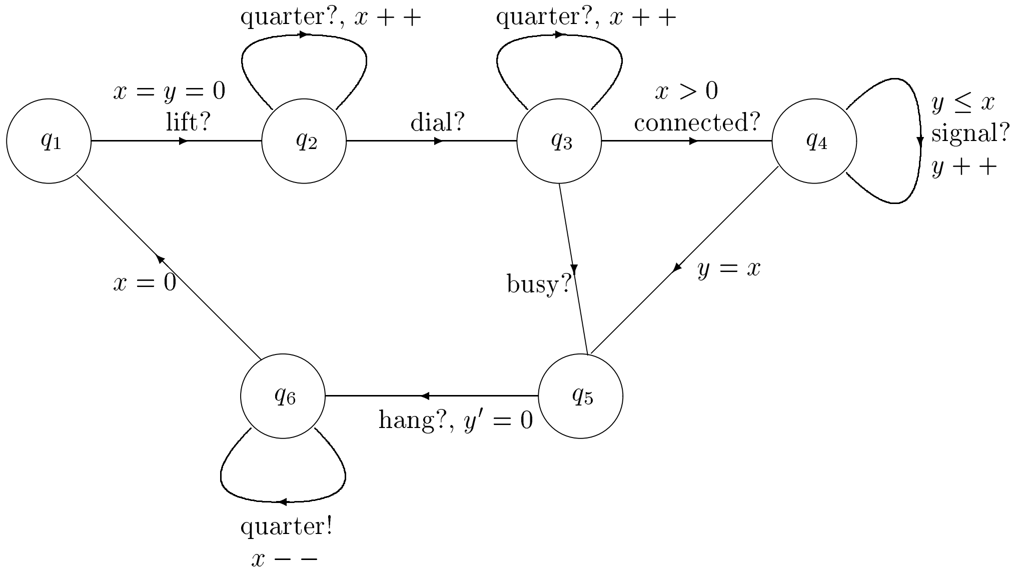


Figure 2: A pay phone

may depend on the distance of the call). The automaton may receive signals such as "quarter?" meaning that a coin has been put in the machine, or "connected?" which means that the connection has been established. The automaton also emits some signals e.g. "quarter!" meaning that a quarter should be given back. The guards of the transitions are e.g. $x = y$ or $y \leq x \wedge y++$. The notation $x++$ is an abbreviation for $x' = x + 1$ (similarly for $y++$ and $x--$).

The following is a valid transition sequence of the automaton:

$$
\begin{aligned}
(q_1, \begin{pmatrix} 0 \\ 0 \end{pmatrix}) \quad & \xrightarrow{lift?} (q_2, \begin{pmatrix} 0 \\ 0 \end{pmatrix}) \xrightarrow{quarter?} (q_2, \begin{pmatrix} 1 \\ 0 \end{pmatrix}) \\
& \xrightarrow{quarter?} (q_2, \begin{pmatrix} 2 \\ 0 \end{pmatrix}) \xrightarrow{dial?} (q_3, \begin{pmatrix} 2 \\ 0 \end{pmatrix}) \\
& \xrightarrow{connected?} (q_4, \begin{pmatrix} 2 \\ 0 \end{pmatrix}) \xrightarrow{signal?} (q_4, \begin{pmatrix} 2 \\ 1 \end{pmatrix})
\end{aligned}
$$

For instance, the last transition step is acceptable because

$$\begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix} \models y \leq x \wedge y' = y + 1$$

The *binary reachability relation* $\xrightarrow[q,q']{*}$ (resp. $\xrightarrow[q,q']{n}$) between two states of a counter automaton $\mathcal{A}$ is the relation on D^N (resp. on $\mathbb{N} \times D^N \times D^N$) defined by $\vec{v} \xrightarrow[q,q']{*} \vec{v'}$ iff there is an integer m such that $\vec{v} \xrightarrow[q,q']{m} \vec{v'}$ and $\vec{v} \xrightarrow[q,q']{m} \vec{v'}$ iff there are configurations $\gamma_0 = (q, \vec{v}), \gamma_1, \dots, \gamma_m = (q', \vec{v'})$ such that, for every $i < m$, $\mathcal{A}$ may move from γ_i to γ_{i+1}.

Example 3 In the last example, $\begin{pmatrix} 0 \\ 0 \end{pmatrix} \xrightarrow[q_1,q_4]{6} \begin{pmatrix} 2 \\ 1 \end{pmatrix}$.

Now we introduce the particular control we are interested in.

Definition 2 *An automaton is* flat *if, for each state q, there is at most one state sequence $q = q_0, q_1, \dots, q_n$ which does not contain any state twice and such that, for every $i < n$, there is a transition $(q_i, a_i, c_i, q_{i+1}) \in T$ and $(q_n, a_n, c_n, q_0) \in T$.*

In other words, there is at most one loop on each state. For instance the automaton of figure 2 is not flat as there are several loops e.g. on state q_2.

4 Loops, fixed points and n-graphs

The relation between counter values induced by a single loop of a counter automaton can be defined in the constraint system $\mathcal{C}$ enriched with a fixed point operator. On the other hand, constraints in $\mathcal{C}$ can be represented as weighted graphs whose vertices are the variables. In this section we make precise the relationships between these three points of view: loops in an automaton, fixed points of constraints and parametrized graphs.

4.1 The relation defined by a loop of a counter automaton

Consider a single state, single transition counter automaton. $\xrightarrow[q,q]{n}$ defines a ternary relation on $\mathbb{N} \times D^N \times D^N$: $(n, \vec{V}, \vec{V'}) \in R$ if and only if $\vec{V} \xrightarrow[q,q]{n} \vec{V'}$.

4.2 Fixed points

Definition 3 *We will say that a tuple* $m, \vec{V}, \vec{V'}$ *satisfies* $\mathsf{FIX}_n(c)$ *where* $c(\vec{x}, \vec{x'}) \in \mathcal{C}(\vec{x})$, $m \in \mathbb{N}$, $\vec{V}, \vec{V'} \in D^N$ *when*

- *either* $m \geq 1$ *and* $\vec{V}, \vec{V'} \models \exists \vec{x_1}, \ldots \overrightarrow{x_{m-1}}.c(\vec{x}, \vec{x_1}) \wedge \ldots \wedge c(\overrightarrow{x_{m-1}}, \vec{x'})$
- *or* $m = 0$ *and* $\vec{V} = \vec{V'}$

In particular, $1, \vec{V}, \vec{V'} \models \mathsf{FIX}_n(c)$ iff $\vec{V}, \vec{V'} \models c$.

Example 4 Let c be the constraint $x' = x \wedge y \leq x \wedge y' = y + 1$

$$\begin{array}{rcl}
0, \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}, \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} & \models & \mathsf{FIX}_n(c) \\
2, \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix} & \models & \mathsf{FIX}_n(c) \\
3, \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix} & \not\models & \mathsf{FIX}_n(c) \\
3, \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 4 \end{pmatrix} & \not\models & \mathsf{FIX}_n(c)
\end{array}$$

4.3 p-graphs

Weighted graphs are commonly used for the representation of constraints $x \leq y + c$: vertices of the graph are the variables and each such constraint is associated with an edge of weight c from x to y. If we want to iterate k times a graph (which will correspond to k iterations of the loop), where k is an integer, we get the notion of k, V-graph, which, roughly, is a weighted graph with $(k+1) \times |V|$ vertices where each level i $(0 \leq i \leq k)$ is related with the levels $i-1, i, i+1$ only. This notion is made precise in definition 4.

Next, we want the same notion, keeping however k as a parameter (free variable). This requires an additional abstraction which is formally defined as a p-graph in definition 6.

Definition 4 *If* k *is a non negative integer and* V *is a finite set, a* k, V*-graph* G *is a tuple* (S, E, W, op) *where*

- $S = V \times [0..k]$ *is the set of* vertices. $(v, i) \in S$ *is written* v_i *and* V_i *is a shorthand for* (V, i). V_0 *is the set of* input vertices *and* V_k *is the set of* output vertices.

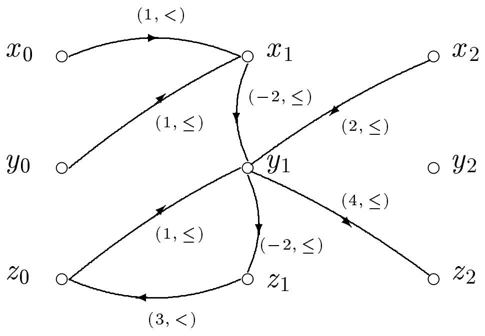


Figure 3: A $2,V$-graph

- $E \subseteq \bigcup_{i=1}^{k}((V_i \times V_i) \cup (V_i \times V_{i+1}) \cup (V_{i+1} \times V_i) \cup (V_{i+1} \times V_{i+1})) \subseteq S \times S$ *is the set of* edges, W *is a* weight function *from* E *to* D *and* op *is a* strictness function *from* E *to* $\{0,1\}$.

  We write shortly $v_i \xrightarrow[G,0]{w} v'_j$ *(resp.* $v_i \xrightarrow[G,1]{w} v'_j$*) if* $(v_i, v'_j) \in E$ *and* $W(v_i, v'_j) = w$ *and* $op(v_i, v'_j) = 0$ *(resp.* $op(v_i, v'_j) = 1$*).*

By abuse of notations, we will sometimes assume that *op* is a mapping from E into $\{<, \leq\}$: when convenient 0 will be identified with $<$ (strict inequality) and 1 with$\leq$ (large inequality), in this context.

Example 5 We displayed in figure 3 an example of a $2,V$-graph where $V = \{x, y, z\}$. The particularities of such graph are: there is no edge between e.g. x_0 and x_2 (only edges between neighbors), each edge is labeled with both a weight and a strictness sign (here $<$ for a strict edge, $\leq$ for a non-strict edge). Input vertices are x_0, y_0, z_0 and output vertices are x_2, y_2, z_2.

Definition 5 *Given a* (k,V)*-graph* G *and an* (m,V)*-graph* G'*, the* concatenation of G and G', *is the* $(k+m,V)$*-graph* $G \cdot G'$ *whose edges, weight function and strictness function are defined by:*

if $\min(i,j) < k$, $v_i \xrightarrow[G \cdot G', s]{w} v'_j$ *iff* $v_i \xrightarrow[G,s]{w} v'_j$

if $\max(i,j) \geq k+1$, $v_i \xrightarrow[G \cdot G', s]{w} v'_j$ *iff* $v_{i-k} \xrightarrow[G',s]{w} v'_{j-k}$

if $i = j = k$, $v_i \xrightarrow[G \cdot G', s]{w} v'_j$ *iff one of the following holds:*

- $(v_k, v'_k) \in E$, $(v_0, v'_0) \notin E'$ *and* $v_k \xrightarrow[G,s]{w} v'_k$

- $(v_0, v'_0) \in E'$, $(v_k, v'_k) \notin E$ *and* $v_0 \xrightarrow[G',s]{w} v'_0$

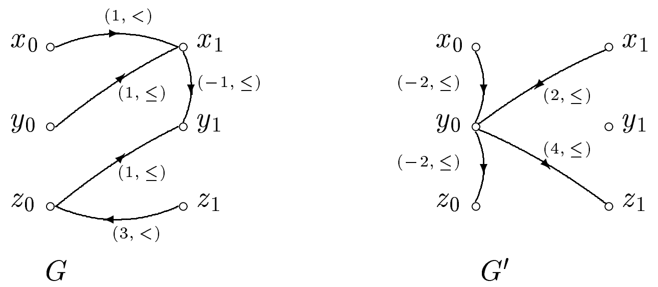


Figure 4: Two $1, V$-graphs

- $v_k \xrightarrow[G,s_1]{w_1} v'_k$ *and* $v_0 \xrightarrow[G',s_2]{w_2} v'_0$ *and* $w = \min(w_1, w_2)$ *and* $s = \min(s_1, s_2)$.

Example 6 We show on figure 4 two $1, V$-graphs G and G' whose composition is the graph given on figure 3.

Now, paths in such graph correspond to the transitive closure of inequalities, hence play an important role: a *path* γ in a k, V-graph is a sequence of consecutive edges $e_1, \dots, e_n$. The *weight* of γ is defined by $W(\gamma) = \sum_{i=1}^{n} W(e_i)$. The strictness of γ is defined by $op(\gamma) = \prod_{i=1}^{n} op(e_i)$.

With each path γ from a vertex v_i to a vertex v'_j is associated a constraint c_γ whose free variables are v_i, v'_j and which is defined by:

- if $op(\gamma) = 0$ then c_γ is $v_i < v'_j + W(\gamma)$
- if $op(\gamma) = 1$, then c_γ is $v_i \leq v'_j + W(\gamma)$.

γ has a *negative weight* if either $W(\gamma) < 0$ or else $W(\gamma) = 0$ and $op(\gamma) = 0$ (i.e. if c_γ is unsatisfiable).

The *relation* defined by a k, V-graph G is

- empty if G contains a cycle of negative weight,
- otherwise the set of pairs $(\overrightarrow{U_{in}}, \overrightarrow{U_{out}}) \in D^N \times D^N$ such that, for every path γ whose source and destination are either input or output vertices, $(\overrightarrow{U_{in}}, \overrightarrow{U_{out}}) \models c_\gamma$.

Example 7 Consider the graph of figure 3. It defines a binary relation on $\mathbb{Q}^3$. If $(u, v, w), (u', v', w')$ is in the relation, we must have e.g. $u < w' + 3$ because there is a path from x_0 to z_2 whose weight is 3 and which is strict. Similarly, we must have $u' < w + 3$ because the path going from x_2 to z_0 through y_1 and z_1 is strict and has a weight 3. We also have e.g. $u < w$ ...

The following lemma essentially states that our definitions are adequate with the expected semantics: concatenation of paths correspond to the conjunction of constraints.

Lemma 1 *Let G be a k,V-graph and G' be a m,V-graph. If G (resp. G') defines a relation R_G (resp. $R_{G'}$) which is definable by a formula ϕ_G (resp. $\phi_{G'}$) in $\mathcal{T}_D$, then $R_{G\cdot G'}$ is definable in $\mathcal{T}_D$ by the formula:*

$$\exists \vec{x_1}.\phi_G(\vec{x},\vec{x_1}) \wedge \phi_{G'}(\vec{x_1},\vec{x'})$$

Proof:
We have to prove two inclusions. Assume first that $(\overrightarrow{U_{in}},\overrightarrow{U_{out}}) \models \exists \vec{x_1}.\phi_G(\vec{x},\vec{x_1}) \wedge \phi_{G'}(\vec{x_1},\vec{x'})$. Then there is a vector $\vec{U}$ such that $(\overrightarrow{U_{in}},\vec{U},\overrightarrow{U_{out}}) \models \phi_G(\vec{x},\vec{x_1}) \wedge \phi_{G'}(\vec{x_1},\vec{x'})$. By hypothesis, $(\overrightarrow{U_{in}},\vec{U}) \in R_G$ and $(\vec{U},\overrightarrow{U_{out}}) \in R_{G'}$. Let γ be any path in $G \cdot G'$ whose source and destination are input or output vertices. We decompose γ as $\gamma_1 \cdot \delta_1 \cdots \gamma_n \cdot \delta_n$ where each γ_i is a path of G and each δ_i is a path of G' (γ_1 and δ_n may be empty). By hypothesis, for every γ_i, $\overrightarrow{U_{in}},\vec{U} \models c_{\gamma_i}$ and, for every δ_i, $\vec{U},\overrightarrow{U_{out}} \models c_{\delta_i}$. Assume for instance (the other case is similar) that there is no strict edge in γ: c_γ is a constraint $v_i \leq v'_j + W(\gamma)$. Then the usual quantifier elimination on

$$\exists \vec{v}.c_{\gamma_1}(\vec{x},\vec{v}) \wedge c_{\delta_1}(\vec{v},\vec{y}) \wedge \ldots \wedge c_{\gamma_n}(\vec{x},\vec{v}) \wedge c_{\delta_n}(\vec{v},\vec{y})$$

gives c_γ because $W(\gamma) = (\sum_{i=1}^n W(\gamma_i)) + (\sum_{i=1}^n W(\delta_i))$. This means that $\overrightarrow{U_{in}},\overrightarrow{U_{out}} \models c_\gamma$. Since this is true for every path γ in $G \cdot G'$, $(\overrightarrow{U_{in}},\overrightarrow{U_{out}}) \in R_{G\cdot G'}$. (Note that, in particular, $R_{G\cdot G'}$ is not empty).

Conversely, assume $(\overrightarrow{U_{in}},\overrightarrow{U_{out}}) \in R_{G\cdot G'}$. Let us order the components of V using superscripts: $V = \{v^1,\ldots,v^N\}$. Let Γ_i (resp. Δ_i) be the set of all paths γ in $G\cdot G'$ whose source (resp. destination) is v^i_k and destination (resp. source) is in $V_0 \cup V_{m+k}$. We proceed now as for the quantifier elimination in the theory of successor or in the additive theory of $\mathbb{Q}$. In case $D = \mathbb{N}$ or $\mathbb{Z}$, we assume that all strict inequalities have been replaced with large ones by adding or subtracting 1 from the constant. Every c_γ, $\gamma \in \Gamma_i$ has one of the forms $v^i_k \leq v^j_0 + \alpha_{i,j}$, $v^i_k < v^j_0 + \alpha_{i,j}$, $v^i_k \leq v^j_{k+m} + \beta_{i,j}$, $v^i_k < v^j_{k+m} + \beta_{i,j}$. Let $\lambda_i = \min(\min_{\gamma\in\Gamma_i}(U^j_{in} + \alpha_{i,j}), \min_{\gamma\in\Gamma_i}(U^j_{out} + \beta_{i,j}))$. Similarly, let μ_i be the maximal value of lower bounds on v^i_k: $\mu_i = \max_{\gamma\in\Delta_i}(U^j_{in} + \delta_{i,j}, U^j_{out} + \eta_{i,j})$. Since, by hypothesis, $(\overrightarrow{U_{in}},\overrightarrow{U_{out}}) \in R_{G\cdot G'}$, $(\overrightarrow{U_{in}},\overrightarrow{U_{out}}) \models c_\gamma$ for every $\gamma = \gamma_1 \cdot \gamma_2$ with $\gamma_1 \in \Delta_i, \gamma_2 \in \Gamma_i$. This means that $\lambda_i \geq \mu_i$ for every i ($\lambda_i > \mu_i$ if there is a strict edge on γ). Then, we choose $U^i = \lambda_i$ if $\lambda_i = \mu_i$ and U^i in the interval $[\lambda_i,\mu_i] \cap D$ otherwise, in such a way that, for every i, for every $\gamma \in \Gamma_i \cup \Delta_i$, we get $U^i,\overrightarrow{U_{in}},\overrightarrow{U_{out}} \models c_\gamma$.

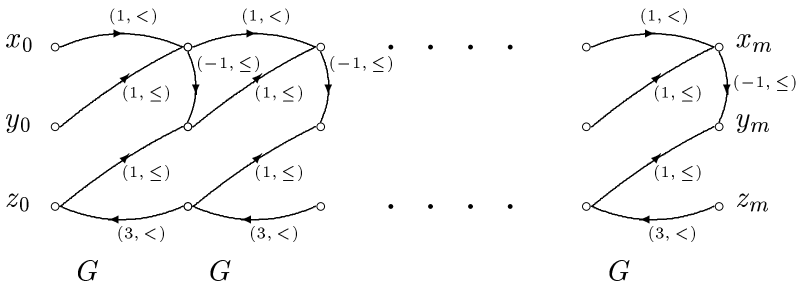


Figure 5: The iteration of G

Summing up, we constructed a vector $\vec{U} \in D^N$ such that $\overrightarrow{U_{in}}, \vec{U} \models \phi_G(\vec{x}, \vec{x_1})$ and $\vec{U}, \overrightarrow{U_{out}} \models \phi_{G'}(\vec{x_1}, \vec{x'})$. $\square$

Given a k, V-graph G and an integer m, G^m is the $(m \times k, V)$-graph $\underbrace{G \cdot \ldots \cdot G}_{m}$. The next definition abstracts the notion of a k, V-graph, keeping k as a parameter: a parametrized graph is a $1, V$-graph whose interpretation is the set of its iterations. This gives a set of graphs containing G as a repeated pattern.

Definition 6 *A* parametrized *V-graph (p-graph for short) is a $(1, V)$-graph G. It defines a relation $R_G \subseteq \mathbb{N} \times D^N \times D^N$ as follows: $(m, \vec{U}, \vec{U'}) \in R_G$ if and only if $(\vec{U}, \vec{U'}) \in R_{G^m}$.*

Example 8 Consider the graph G of figure 4. Its iteration G^m is depicted on figure 5 and the relation it defined corresponds to paths in such a (parametrized) iteration.

For instance, there are several paths from x_0 to x_m. One of the shortest (smallest weight) goes through $x_0, x_1, y_1, x_2, y_2...$ and has a weight 1. Hence $(m, \vec{U}, \vec{V}) \in R_G$ (G being considered here as a p-graph) only if, when $m \geq 1$, $U^1 < V^1 + 1$. Similarly, we must have $V^3 < 3 \times m + U^3$, following the path $z_m, z_{m-1}, \ldots, z_0$.

4.4 An equivalence result

The following lemma states the equivalence between the three representations.

Lemma 2 *A relation $R \subseteq \mathbb{N} \times D^N \times D^N$ is definable by a single state single transition counter automaton if and only if there is a constraint $c \in \mathcal{C}(\vec{x})$*

such that R is defined by $\mathsf{FIX}_n(c(\vec{x}, \vec{x'}))$ if and only if R is defined by a parametrized $\vec{x}$-graph.

It is simply a reformulation of the definitions.

5 The main theorem

This section is entirely devoted to the proof that relations which are defined by parametrized V-graphs are definable in $\mathcal{T}_D$.

First, we give some simple reductions, restricting further the class of graphs that have to be considered. Next, in section 5.2 we introduce the *folding* of a p-graph, which consists basically in connecting the inputs and the outputs. We prove in section 5.3 that the weights of paths in such a folded graph is definable in $\mathcal{T}_D$.

Next, we show in section 5.4 that if, for a sufficiently large n, there is no cycle of negative weight in G^n, then there is no cycle of negative weight in any G^p with $p \geq n$. Going back to the automaton formulation, this shows that a loop which can be iterated at least n times can be iterated arbitrarily many times. This also shows that we have only to consider graphs G such that, for every n, G^n does not contain a negative cycle.

Next, we consider in section 5.5 a well-founded quasi-ordering on paths $\trianglelefteq$ and we show that any path which is contained in a "central part" of G^n and which is minimal w.r.t. $\trianglelefteq$ is regular (in a sense which will be made precise). The weights of such regular paths are linearly dependent on n, hence expressible in $\mathcal{T}_D$. This is the central part of the proof and requires some technical results, constructing regular paths from non-regular ones.

Finally, we prove in section 5.6 that minimal paths go "back and forth" (zigzags). It follows that there are constants M, K (depending on G only and not on n) such that every minimal path can be decomposed into at most M paths in G^K and at most M paths in the "central part". The latter can be assumed regular, following the previous section.

We put everything together in the section 5.7, stating and proving the main result and its corollaries.

Finally, in section 5.8, we prove a side result: the decidability of the iterability of a loop (or a constraint). We consider there different notions of iterability such as "arbitrarily many times" or "infinitely many times".

5.1 A simple reduction

Lemma 3 *Let G be a p-graph. It is possible to compute (in linear time) a p-graph G' which does not contain edges between output vertices and such that, if the relation $R_{G'}$ defined by G' is also defined in $\mathcal{T}_D$ by a formula $\phi_{G'}(n, \vec{x}, \vec{x'})$, then R_G is defined in $\mathcal{T}_D$ by*

$$(\vec{x} = \vec{x'} \wedge n = 0) \vee (n \geq 1 \wedge \exists \vec{x_1}.\psi_0(\vec{x}, \vec{x_1}) \wedge \phi_{G'}(n-1, \vec{x_1}, \vec{x'}) \wedge \psi(\vec{x'}))$$

where $\psi_0, \psi_1 \in \mathcal{C}$.

Therefore, we may assume from now on without loss of generality that there is no edge between output variables.
Proof:
G', G_0 and G_1 are constructed as follows: G_0 is the graph G in which the edges between output vertices are removed. G' is defined as follows: for every $v, v' \in V$, $v_0 \xrightarrow[G',s]{w} v_1'$ iff $v_0 \xrightarrow[G,s]{w} v_1'$ and $v_1 \xrightarrow[G',s]{w} v_0'$ iff $v_1 \xrightarrow[G,s]{w} v_0'$ and $v_0 \xrightarrow[G',s]{w} v_0'$ iff

- $(v_0, v_0') \in E$ and $(v_1, v_1') \notin E$ and $v_0 \xrightarrow[G,s]{w} v_0'$
- or $(v_1, v_1') \in E$ and $(v_0, v_0') \notin E$ and $v_1 \xrightarrow[G,s]{w} v_1'$
- or $v_0 \xrightarrow[G,s_1]{w_1} v_0'$ and $v_1 \xrightarrow[G,s_2]{w_2} v_1'$ and $w = \min(w_1, w_2)$, $s = \min(s_1, s_2)$.

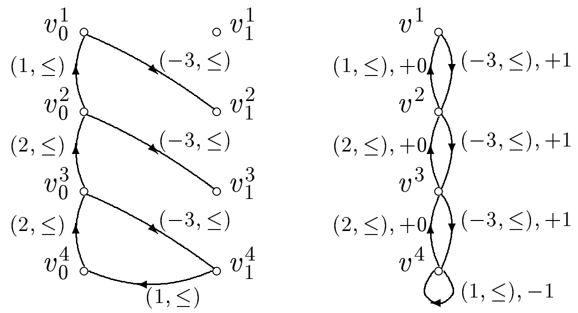
Moreover, there is no edge between v_1, v_1' in G'.

G_1 is the 0-graph such that $v_0 \xrightarrow[G_1,s]{w} v_0'$ if and only if $v_1 \xrightarrow[G,s]{w} v_1'$.

Then, for every $n \in \mathbb{N}$, $G^{n+1} = G_0 \cdot G'^n \cdot G_1$ and the result follows from lemma 1. □

5.2 The folded graph

In this section we consider the "repeated pattern" in G^n: it is obtained by connecting the inputs and the outputs of G. This gives a finite graph (not depending on n). We formally define here the connections between parametrized graphs and their folded version. In particular, we define a projection and an expansion function and show to which extend they are inverse of each other.



Graph G_1 Graph $K(G_1)$

Figure 6: An example of a graph and its folded version

Definition 7 *A* coloured weighted multigraph *is a tuple* (S, E, C, p, W, op) *where* S *is a finite set of vertices,* C *is a finite set of colors,* $E \subseteq S \times C \times S$ *is the set of edges,* p *is the mapping from* E *to* C *associating each edge with its colour,* W *is a mapping from* E *to* D *(the weight function) and* op *is a mapping from* E *to* $\{0, 1\}$ *(the strictness function).*

An edge from v *to* v' *is written* $v \xrightarrow[G,s,c]{w} v'$ *if* $(v, c, v') \in E$ *and* $W(v, c, v') = w, op(v, c, v') = s$.

Definition 8 *Given a parametrized graph* $G = (S, E, W, op)$ *, the* folded graph $K(G)$ *is the coloured weighted multigraph* $(S, E', \{-1, 0, 1\}, p, W, op)$ *such that*

$$\begin{array}{lll}
v_0 \xrightarrow[G,s]{w} v_0' & \textit{iff} & v \xrightarrow[K(G),s,0]{w} v' \\
v_0 \xrightarrow[G,s]{w} v_1' & \textit{iff} & v \xrightarrow[K(G),s,1]{w} v' \\
v_1 \xrightarrow[G,s]{w} v_0' & \textit{iff} & v \xrightarrow[K(G),s,-1]{w} v'
\end{array}$$

In the context of a folded graph p can be read as "progression": its value is $+1$ if the edge goes from V_0 to V_1, 0 if it stays in V_0 and -1 if it goes from V_1 to V_0.

Example 9 We show on figure 6 an example of a parametrized graph G_1 (without relations on its output vertices) and its folded version. We labeled first the edges with the pair (w, s) (the weight and strictness) and then with the colour.

p is extended to paths in $K(G)$: $p(\epsilon) = 0$ and $p(\gamma \cdot e) = p(\gamma) + p(e)$.

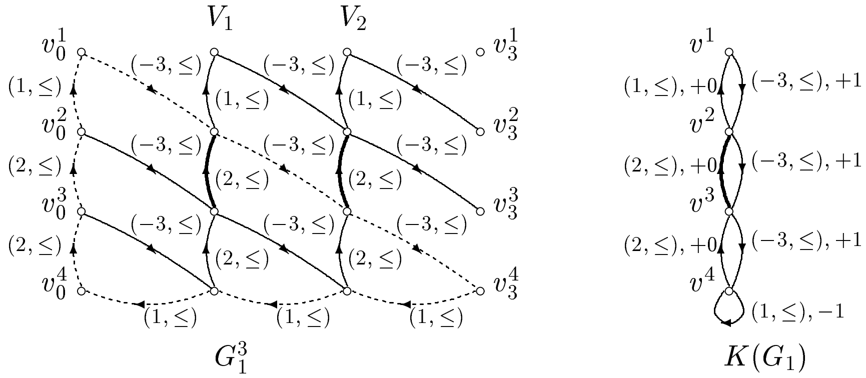


Figure 7: An expansion of an edge in $K(G_1)$

Each path γ in G^n is associated with its *projection* $\pi(\gamma)$ in $K(G)$: for each edge $e = v_i \xrightarrow[G^n,s]{w} v'_j$, $\pi(e) = v \xrightarrow[K(G),s,j-i]{w} v'$ and $\pi(e_1 \cdots e_k) = \pi(e_1) \cdots \pi(e_k)$.

Conversely, for each m, there is a (partial) expansion function associating with each integer n and each path γ in $K(G)$ a path $Expand_m(\gamma, n)$ in G^m:

- for an empty path γ with source (and destination) v, $Expand_m(\gamma, n)$ is the empty path with source and destination v_n, if $m \geq n \geq 0$ and is undefined otherwise.

- if $\gamma = e \cdot \gamma_1$ and $e = (v, i, v')$, then

$$Expand_m(\gamma, n) \stackrel{\text{def}}{=} (v_n, v'_{n+i}) \cdot Expand_m(\gamma_1, n + i),$$

  provided

  - $Expand_m(\gamma_1, n + i)$ is defined and
  - $m \geq n, m \geq n + i, n \geq 0, n + i \geq 0$ and
  - either $m > n$ or $i < 0$.

Example 10 We show on figure 7 the graph $K(G_1)$ and the iteration G_1^3. A distinguished edge of $K(G_1)$ is displayed with a thick line and two of its expansions in G_1^3 are also shown in bold face.

The following two lemmas establish that expansion and projection are inverse of each other. They assume, according to lemma 3, that G (and therefore G^m) does not contain any relation between output vertices.

Lemma 4 *Let G be a parametrized graph, then for every path γ in G^m, with source edge v_i, $Expand_m(\pi(\gamma), i)$ is defined and $Expand_m(\pi(\gamma), i) = \gamma$.*

Proof:
We proceed by induction on the length of γ: if γ is the empty path with source (and destination) v_i, then $\pi(\gamma)$ is the empty path with source (and destination v) and $Expand_i(\pi(\gamma), i)$ is again γ.

Now, assume $\gamma = e{\cdot}\gamma_1$. By definition, $\pi(\gamma) = \pi(e){\cdot}\pi(\gamma_1)$ and $Expand_m(\pi(\gamma), i) = Expand_m(\pi(e), i) \cdot Expand_m(\pi(\gamma_1), j)$, if v'_j is the target of e. By induction hypothesis, $Expand_m(\pi(\gamma_1), j) = \gamma_1$. Moreover $\pi(e) = v \xrightarrow[K(G),s,j-i]{w} v'$ and $0 \leq j \leq m$. It follows, by definition, that $Expand_m(\pi(e), i) = e$. □

Lemma 5 *For every path γ in $K(G)$, there are integers n, m (we may choose $m = 2 \times |\gamma|$ and $n = |\gamma|$) such that $Expand_m(\gamma, n)$ is defined and $\pi(Expand_m(\gamma, n)) = \gamma$.*

Proof:
Again, it is a direct consequence of the definitions. Since $p(\gamma) \in [-|\gamma|, ..., |\gamma|]$, choosing $n = |\gamma|$ and $m = 2 \times |\gamma|$ ensures that, starting at some vertex v_n, after $|\gamma|$ steps, we still are within the limits of G^m. □

5.3 Weights of paths in the folded graph

In this section, we show that it is possible to effectively express in $\mathcal{T}_D$ the set of weights of paths in the folded graph.

The idea here is to associate with each path the number of occurrences of each edge in the path. On one hand we show that, given the number of occurrences of each edge, the weight of the path is expressible in $\mathcal{T}_D$ (this is easy). On the other hand we give a formula which characterizes vectors of edges occurrences which correspond to actual paths in the graph.

Let $G = (V, E)$ be any finite directed multigraph. Let $E = \{e_1, \ldots, e_{n_E}\}$. A vector of edge occurrences is a tuple $M \in \mathbb{N}^{n_E}$. For each path γ of G, $M(\gamma)$ is the vector of edge occurrences in γ.

Consider the difficult part: assume given M, we are looking for a formula $\phi(k_1, \ldots, k_{n_E})$ which is satisfied by M if and only if there is a path γ such that $M = M(\gamma)$.

Let v, v' be two vertices of G. If $v \neq v'$, let $Path_{v,v'}$ be the formula with free variables $M = (k_1, \ldots, k_{n_E})$ defined by:

$$
\begin{aligned}
Path_{v,v'}(M) \quad &\stackrel{\text{def}}{=} \quad \bigwedge_{i=1}^{n_E} \mathsf{Int}(k_i) \\
& \qquad \bigwedge_{v'' \in V \setminus \{v,v'\}} In(v'', M) = Out(v'', M) \\
& \qquad \wedge\, Out(v, M) = 1 + In(v, M) \\
& \qquad \wedge\, In(v', M) = 1 + Out(v', M) \\
& \qquad \wedge\, \text{Connected}_v(M)
\end{aligned}
$$

where

$$
\begin{aligned}
In(v, M) &\stackrel{\text{def}}{=} \sum_{i, \exists v_i . e_i = (v, v_i)} k_i \\
Out(v, M) &\stackrel{\text{def}}{=} \sum_{i, \exists v_i . e_i = (v_i, v)} k_i
\end{aligned}
$$

$$
\text{Connected}_v(M) \stackrel{\text{def}}{=} \bigwedge_{v' \in V} \bigwedge_{j, \exists v_j . e_j = (v_j, v')} (k_j > 0 \Rightarrow \bigvee_{\gamma \in SP(v, v')} \bigwedge_{e_i \in \gamma} k_i > 0)
$$

and $SP(v, v')$ is the finite set of cycle-free paths in G from v to v'.

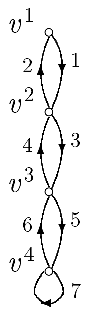
In English, $In(v, M)$ is the number of edges in M whose target is v (incoming edges), $Out(v, M)$ is the number of edges in M whose source is v (outgoing edges) and $\text{Connected}_v(M)$ expresses that, for each vertex v'' occurring in some edge of M, there is a possible path from v to v'' using edges of M. Then the formula expresses in particular that, except for the source and target of the path, the number of incoming and outgoing edges are equal for each vertex.

When $v = v'$, $Path_{v,v}$ is defined in a similar way:

$$
Path_{v,v}(M) \stackrel{\text{def}}{=} \bigwedge_{i=1}^{n_E} \mathsf{Int}(k_i) \bigwedge_{v'' \in V} In(v'', M) = Out(v'', M) \wedge \text{Connected}_v(M)
$$

Example 11 Consider the graph $K(G_1)$ displayed on figure 7. There are seven edges which we number $e_1, \ldots, e_7$ (see figure 8), in the lexicographic order (source, destination). Let $M = (m_1, \ldots m_7)$. It is a vector of occurrences of edges in a path (say, from v^2 to v^3) of $K(G_1)$ if and only if the following relations are satisfied:

$$
\begin{array}{llcll}
 & m_2 & = & m_1 & v^1 \text{ incoming edges} = v^1 \text{ outgoing edges} \\
\wedge & m_2 + m_3 & = & 1 + m_1 + m_4 & v^2 \text{ outgoing edges} = 1 + v^2 \text{ incoming edges} \\
\wedge & m_3 + m_6 & = & 1 + m_4 + m_5 & v^3 \text{ incoming edges} = 1 + v^3 \text{ outgoing edges} \\
\wedge & m_5 + m_7 & = & m_6 + m_7 & v^4 \text{ incoming edges} = v^4 \text{ outgoing edges} \\
\wedge & \multicolumn{4}{l}{\text{Connected}_{v_2}(M)}
\end{array}
$$



$K(G_1)$

Figure 8: Numbering of the edges in $K(G_1)$

And $\text{Connected}_{v_2}(M)$ is defined by:

$$\text{Connected}_{v_2}(M) \stackrel{\text{def}}{=} \left\{ \begin{array}{rrcl} & m_2 > 0 & \Rightarrow & m_2 > 0 \\ \wedge & (m_3 > 0 \vee m_6 > 0) & \Rightarrow & m_3 > 0 \\ \wedge & (m_5 > 0 \vee m_7 > 0) & \Rightarrow & (m_3 > 0 \wedge m_5 > 0) \end{array} \right.$$

each line corresponding to a choice of a vertex $v' \in V$. Of course, the formula can be simplified.

Then, the weight of a path represented by M is given by the formula

$$w = (-3 \times m_1) + m_2 + (-3 \times m_3) + (2 \times m_4) + (-3 \times m_5) + (2 \times m_6) + m_7$$

Lemma 6 *$M \models Path_{v,v'}$ if and only if there is a path γ in G from v to v' such that $M = M(\gamma)$.*

Proof:

If $M = M(\gamma)$, then $M \models Path_{v,v'}$, by induction on the length of γ: if γ has a zero length, then $M(\gamma) = (0, \ldots, 0)$ and $M(\gamma) \models \text{Connected}_v(M)$ for all v. Moreover, $In(v, M) = Out(v, M) = 0$ for every v. Assume now that $\gamma = \gamma_1 \cdot e_i$ and that γ_1 is a path from v to v_1 and e_i is an edge from v_1 to v'. $M(\gamma_1) = M(\gamma)$ except for the ith component for which $k_i^1 = k_i - 1$. Assume $v' \notin \{v, v_1\}$ (the proof is similar in the other cases). Then $In(v', M(\gamma)) = In(v', M(\gamma_1)) + 1$ and $Out(v_1, M(\gamma)) = Out(v_1, M(\gamma_1)) + 1$ and for every vertex $v'' \neq v'$, $In(v'', M(\gamma_1)) = In(v'', M(\gamma))$ and for every vertex $v'' \neq v_1$, $Out(v'', M(\gamma_1)) = Out(v'', M(\gamma))$.

Now, using the induction hypothesis, we can conclude for the part of the formula involving In and Out. Concerning the $\text{Connected}_v(M)$ formula, the only case we have to consider is the case where $k_i = 1$: $k_i^1 = 0$. Then, there is a $\gamma_0 \in SP(v, v_1)$ such that, for every $e_j \in \gamma_0$, $k_j^1 > 0$, by induction hypothesis. Then $\gamma_0 \cdot e_i \in SP(v, v')$.

If $M \models Path_{v,v'}$ then $M = M(\gamma)$ for some path γ from v to v' , by induction on the size (sum of components) of M. If M has size 0, then we must have $v = v'$ and the empty path ϵ satisfies $M = M(\epsilon)$.

Now, if M has a size at least one, either $v = v'$ and, since $M \models \text{Connected}_v(M)$, there is at least one edge e_i whose destination is v' and such that $k_i > 0$, or $v \neq v'$ and, since $M \models In(v', M) = 1 + Out(v', M)$, there is at least one edge e_i whose destination is v' and such that $k_i > 0$. In all cases, since $M \models \text{Connected}_v$, there is at least one path $\gamma_0 \in SP(v, v')$ such that, for the last edge e_i of γ_0, $M \models k_i > 0$.

We define an edge e_j which is intended to be removed from the occurrence vector M, yielding an M' to which the induction hypothesis can be applied. e_j is an edge whose target is v' and such that $e_i \neq e_j$, if possible. More formally, either $In(v', M) = 1$ and $j = i$ or else $In(v', M) > 1$. In the latter case, let e_j be an edge whose target is v' and such that either $j \neq i$ and $M \models k_j > 0$ or else $M \models k_j > 1$.

We let M' be identical to M, except for the jth component where $m'_j = m_j - 1$. We let v_1 be the source of e_j. We claim that $M' \models Path_{v,v_1}$. For the In and Out formulas, it is a routine work to check the satisfaction, in a way similar to the first part of this proof. Let us check now that $M' \models \text{Connected}_v$. Let $v'' \in V$ and $e_q = (v_q, v'')$ be such that $M' \models k_q > 0$ (and hence $M \models k_q > 0$).

Let $\gamma \in SP(v, v'')$ be such that $M \models \bigwedge_{e_l \in \gamma} k_l > 0$. Such a path does exist since $M \models \text{Connected}_v$ and $M \models k_q > 0$. There are basically three cases:

- If γ does not involve the edge e_j, then $M' \models \bigwedge_{e_l \in \gamma} k_l > 0$. Otherwise γ can be decomposed into $\gamma_1 \cdot e_j \cdot \gamma_2$ (see figure 9).
- If $e_j \neq e_i$, then let γ' be the path $\gamma_0 \cdot \gamma_2$ in which possible cycles have been removed: $\gamma' \in SP(v, v'')$ and $M' \models \bigwedge_{e_l \in \gamma'} k_l > 0$ since e_j does not occur in γ'.
- If $e_i = e_j$, then, by construction, $M \models k_j > 1$, hence $M' \models k_j > 0$, which implies $M' \models \bigwedge_{e_l \in \gamma} k_l > 0$.

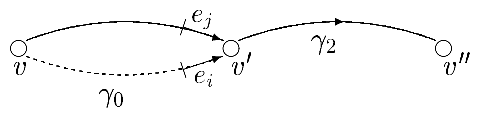


Figure 9: Construction of a path in the proof of lemma 6

In all cases, we have shown that $M' \models \bigvee_{\gamma \in SP(v,v'')} \bigwedge_{e_m \in \gamma} k_m > 0$. Hence $M' \models \text{Connected}_v$.

Applying the induction hypothesis to M', there is a path γ' such that $M' = M(\gamma')$. Then $M = M(\gamma' \cdot e_j)$.

$\square$

We can derive the following corollary:

Lemma 7 *Let G be a parameterized graph and v, v' be two vertices. There is a formula $WPath_{v,v'}(w)$ which is satisfied by the weights of the paths in $K(G)$ whose source is v and destination is v'.*

Proof:
It is sufficient to apply the previous lemma and to notice that the weight of a path γ is the sum of the weights of its edges:

$$WPath_{v,v'}(w) \stackrel{\text{def}}{=} \exists k_1, \dots, k_{n_E}.w = \sum_{i=1}^{n_E} k_i \times W(e_i) \wedge Path_{v,v'}(k_1, \dots, k_{n_E})$$

(Let us recall that n_E, the number of edges in $K(G)$, only depends on G). $\square$

5.4 n-Boundedness

In this section we prove that, for each parametrized graph G, there is a formula $Bound_G(n)$ of $\mathcal{T}_D$ with one free variable n, which is satisfied by the integers $m \in \mathbb{N}$ such that G^m contains a cycle of negative weight. This implies in particular that we can decide whether the relation defined by a parametrized graph G is a finite union of relations R_{G^i}.

Lemma 8 *Let G be a parametrized graph with N vertices and n_E edges. Let NP, PR and $Borne_G$ be the formulas in $\mathcal{T}_D$:*

$$NP(k_1, \dots, k_{n_E}) \stackrel{def}{=} ((\bigvee_{i, op(e_i)=0} k_i \neq 0) \wedge \sum_{i=1}^{n_E} k_i \times W(e_i) \leq 0) \vee \sum_{i=1}^{n_E} k_i \times W(e_i) < 0$$

$$PR(k_1, \ldots, k_{n_E}) \stackrel{def}{=} \sum_{i=1}^{n_E} k_i \times p(e_i) = 0.$$

$$Borne_G(n) \stackrel{def}{=} \exists k_1, \ldots, k_{n_E}. \bigwedge_{i=1}^{n_E} \mathsf{Int}(k_i)$$
$$\wedge \bigvee_{v \in S} (Path_{v,v}(k_1, \ldots k_{n_E}) \wedge n = 2 \times \sum_{i=1}^{n_E} k_i \wedge NP(k_1, \ldots, k_{n_E}) \wedge PR(k_1, \ldots, k_{n_E}))$$

Then

$$\begin{cases} m \models Borne_G(n) \Rightarrow R_G = \bigcup_{i=1}^{m} R_{G^i} \\ R_G = \bigcup_{i=1}^{m} R_{G^i} \Rightarrow \exists m_1 . m_1 \models Borne_G(n) \end{cases}$$

Proof:
$R_G = \bigcup_{i=1}^{m} R_{G^i}$ if and only if $R_{G^{m+1}}$ contains a cycle of negative weight.

Assume γ is a cycle of negative weight in G^{m+1}. Then $\pi(\gamma)$ is a cycle of negative weight in $K(G^{m+1})$ such that $p(\gamma) = 0$. Let $k_1^\gamma, \ldots, k_{n_E}^\gamma$ be the number of occurrences of $e_1, \ldots e_{n_E}$ in $\pi(\gamma)$. By lemma 6, $k_1^\gamma, \ldots, k_{n_E}^\gamma \models Path_{v,v}$ for some vertex v. They also satisfy NP and PR because respectively, $\pi(\gamma)$ has a negative weight and $p(\gamma) = 0$.

Now, assume that $m \models Borne_G(n)$. Then, by lemma 6, there is a cycle γ in $K(G)$ whose number of edge occurrences $k_1^\gamma, \ldots, k_{n_E}^\gamma$ satisfies $NP(k_1, \ldots, k_{n_E})$ and $PR(k_1, \ldots, k_{n_E})$, which implies that γ has a negative weight and $p(\gamma) = 0$. By lemma 5, $\gamma' = Expand_{m+1}(\gamma, n_1)$ is a path in G^{m+1} for a well-chosen n_1, since $m = 2 \times |\gamma|$. Moreover, by definition of *Expand*, γ' is a cycle since $p(\gamma) = 0$ and it has a negative weight since $W(\gamma) = W(\pi(\gamma))$. It follows that R_{G^i} is empty for $i \geq m + 1$. $\square$

Example 12 Let G_1 be the parametrized graph of figure 7. Its iteration G_1^3 contains a path of negative weight, as depicted using dashed edges on figure 7.

Let us see how this is found using the previous lemma. On the example, we have:

$$NP(M) \stackrel{\text{def}}{=} -3 \times m_1 + m_2 - 3 \times m_3 + 2 \times m_4 - 3 \times m_5 + 2 \times m_6 + m_7 < 0$$

since there is no strict edge in G_1 and

$$PR(M) \stackrel{\text{def}}{=} m_1 + m_3 + m_5 - m_7 = 0$$

Moreover, after simplification, $Path_{v,v}$ is given by:

$$\text{Connected}_v(M) \wedge m_1 = m_2 \wedge m_3 = m_4 \wedge m_5 = m_6$$

Hence, after simplification, we get for $Path_{v,v}(M) \wedge NP(M) \wedge PR(M)$:

$$m_1 > 0 \wedge m_1 = m_2 \wedge m_3 = m_4 \wedge m_5 = m_6 \wedge m_7 = m_1 + m_3 + m_5 \wedge \text{Connected}_v(M)$$

Moreover, since the constraint implies both $m_1 > 0$ and $m_7 > 0$, for every v, it can be simplified to:

$$m_1 > 0 \wedge m_3 > 0 \wedge m_5 > 0 \wedge m_1 = m_2 \wedge m_3 = m_4 \wedge m_5 = m_6 \wedge m_7 = m_1 + m_3 + m_5$$

which is satisfiable; for instance $m_1 = m_2 = m_3 = m_4 = m_5 = m_6 = 1$ and $m_7 = 3$ satisfies the formula. This means, according to our lemma, that there is a bound on the number of iterations, given by $n = 2 \times \sum m_i = 18$. This bound is not optimal, as we can see: on the example, the actual bound is 3.

As a corollary of lemma 8 and lemma 2 we may derive:

Theorem 1
- *It is decidable whether or not $\mathsf{FIX}_n(c)$ is satisfiable for finitely many values of n only.*
- *It is decidable whether or not there is a (uniform) bound on the number of iterations of a loop in a counter automaton.*

Thanks to lemma 1, we may derive from lemma 8:

Lemma 9 *if $Bounded_G(n)$ is satisfiable, then R_G is definable in $T_{\mathcal{D}}$.*

Hence, from now on, we may restrict our attention to graphs G such that $Bounded_G$ is unsatisfiable, i.e. to the case where, for every i, G^i does not contain any path of negative weight.

5.5 Regular paths

First, we define a quasi-ordering on paths in G^m:

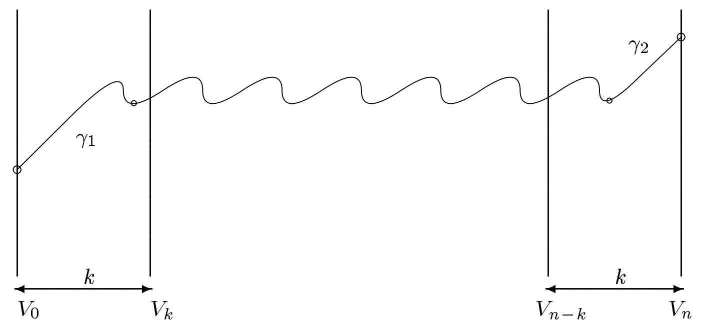


Figure 10: A k-regular path

Definition 9 *$\gamma \trianglelefteq \gamma'$ if and only if γ and γ' have the same source and the same destination and, moreover $(W(\gamma), op(\gamma), |\gamma|) \leq (W(\gamma'), op(\gamma'), |\gamma'|)$, the pairs being compared lexicographically.*

Note that, for paths whose source and destination are input or output vertices of G^m, then $\gamma \trianglelefteq \gamma'$ if and only if $c_\gamma \models c_{\gamma'}$. Hence the relation R_{G^m} may be actually defined considering minimal paths w.r.t. $\trianglelefteq$ only.

The idea now is to prove that, for the "central part" of the graph, minimal paths are regular. Let us make this precise.

Definition 10 *Let k be a positive integer. A path γ in G^m is* k-regular *if it can be decomposed into $\gamma = \gamma_1 \cdot \sigma_1 \cdots \sigma_p \cdot \gamma_2$ where $|\gamma_1|, |\gamma_2| < k$ and there exists an elementary cycle σ in $K(G)$ such that, for $i = 1, ..., p$, $\pi(\sigma_i) = \sigma$.*

Example 13 An illustration of a k-regular path is given on figure 10. For a more concrete example, see figure 11, where we display an iteration of a graph G_2 together with its folded version $K(G_2)$. The dashed path corresponds to the iteration of the dashed cycle in $K(G_2)$. The bold path corresponds to the iteration of the bold cycle in $K(G_2)$. Both paths are regular. However, a path which would sometimes use dashed edges and sometimes bolded edges would not be regular.

Now, we explain what we mean by "central part":

Definition 11 *A path in G^m is* k-central *if all its vertices belong to $\bigcup_{i=k}^{m-k} V_i$. A path in G^m is* k-external *if either the destination of its first edge is in $\bigcup_{i=m-k+1}^{m} V_i$ or the source of its last edge is in $\bigcup_{i=1}^{k-1} V_i$.*

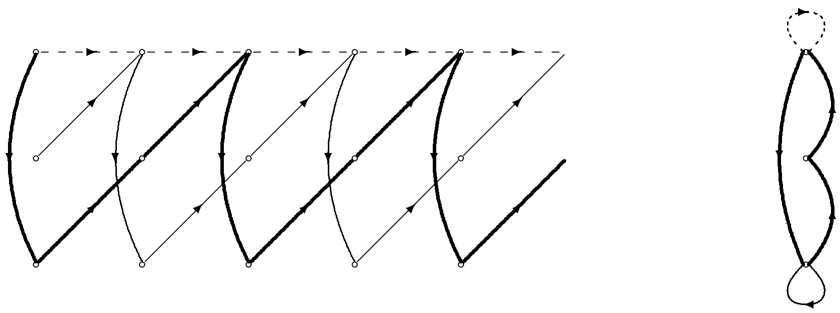


Figure 11: Regular paths in G_2

Finally, we let the constants b_1 and b_2 be defined as:

$$b_1 \stackrel{\text{def}}{=} p_G^2$$

$$b_2 \stackrel{\text{def}}{=} (NC_G \times P_G^2 + 1) \times N + 1$$

where $p_G = \max\{p(\sigma) \mid \sigma \in Cycles(G)\}$, $Cycles(G)$ is the finite set of elementary cycles in $K(G)$ and $NC_G = |Cycles(G)|$.

Lemma 10 *For every b_1-central path γ, there is a path $\gamma' \trianglelefteq \gamma$ such that $\gamma' = \delta_1 \cdot \gamma'' \cdot \delta_2$ where γ'' is b_1-central and b_2-regular and δ_1, δ_2 are either empty or b_1-external.*

Proof:
The idea of the proof is to find in γ several subpaths whose projections are the same cycle and then to gather them together, re-arranging the ordering of the subpaths.

We prove, by induction on the pair $(\gamma, |\gamma| - |\sigma| \times p)$, lexicographically ordered with $\trianglelefteq$ on the first component that, for all γ, for all $\sigma \in Cycles(G)$, for all p,

> If $\gamma = \gamma_1 \cdot \sigma_1 \cdots \sigma_p \cdot \gamma_2$ is b_1-central and for every i $\pi(\sigma_i) = \sigma$, then there is a path $\gamma' = \delta_1 \cdot \gamma'' \cdot \delta_2$ such that $\gamma' \trianglelefteq \gamma$ and γ'' is b_1-central and b_2-regular and δ_1, δ_2 are either b_1-external or empty.

If $|\gamma| - |\sigma|^p$ is null, then it suffices to choose $\gamma' = \gamma$ and δ_1, δ_2 empty. If $|\gamma| < b_2$, then it suffices to choose again $\gamma' = \gamma'' = \gamma$.

Let $\gamma = \gamma_1 \cdot \sigma_1 \cdots \sigma_p \cdot \gamma_2$ with $\pi(\sigma_i) = \sigma \in Cycles(G)$

If $p < P_G$ Since $|\gamma| \geq b_2$, γ contains at least $NC_G \times P_G^2 + 2$ occurrences of the same vertex. This defines $NC_G \times P_G^2 + 1$ segments of γ, each of which containing a subpath whose projection is a cycle. Now, at least $P_G^2 + 1$ of them should be identical:

$$\gamma = \gamma_1 \cdot \sigma_1^1 \cdot \gamma_2 \ldots \cdot \sigma_\alpha^1 \cdot \gamma_{\alpha+1}$$

where $\alpha \geq P_G^2 + 1$ and $\pi(\sigma_i^1) = \sigma^1 \in Cycles(G)$. Let β be the integral part of $\frac{P_G^2}{|\sigma^1|}$. Now, since γ is b_1-central and $p(\sigma^1) \leq P_G$, the path

$$\gamma_0 = \gamma_1 \cdot \sigma_1^1 \cdot \sigma_2^1 \cdots \sigma_\beta^1 \cdot \gamma_2 \cdots \gamma_{\beta+1} \cdot \sigma_{\beta+1}^1 \cdots \cdot \gamma_{\alpha+1}$$

and the path

$$\gamma_0' = \gamma_1 \cdot \gamma_2 \cdots \gamma_\beta \cdot \sigma_1^1 \cdot \sigma_2^1 \cdots \sigma_\beta^1 \cdot \gamma_{\beta+1}\sigma_{\beta+1}^1 \cdots \cdot \gamma_{\alpha+1}$$

are paths in G^m which are equivalent (w.r.t. $\trianglelefteq$) to γ. We write them $\mu \cdot \delta$ (resp. $\delta \cdot \mu$) with μ b_1-central and containing $\sigma_1^1 \cdots \sigma_\beta^1$ and δ is b_1-external. Then,

$$|\sigma^1|^\beta > P_G^2 - |\sigma^1| \geq P_G \times (P_G - 1) \geq p \times |\sigma| = |\sigma|^p.$$

Hence, we may apply the induction hypothesis to μ and we get the expected result, concatenating δ on the right or on the left of γ'

If $p \geq P_G$ If $|\gamma_1|, |\gamma_2| < b_2$, then the result is proved. Assume for instance that $|\gamma_2| \geq b_2$ (the other case is similar). γ_2 is b_1-central, hence, as we did above in the case $p < P_G$, there is a path γ_1' which is equivalent to γ_2 and which can be decomposed into $\gamma_2' \cdot \theta_1 \ldots \cdot \theta_q \cdot \gamma_3$ with $\pi(\theta_i) = \theta \in Cycles(G)$ and $q \geq P_G$. If this path is not b_1-central, then γ is equivalent to some path

$$\gamma_1 \cdot \sigma_1 \ldots \cdot \sigma_p \cdot \mu \cdot \delta$$

where δ is b_1 external and non-empty. Then we apply the induction hypothesis to $\gamma_1 \cdot \sigma_1 \ldots \cdot \sigma_p \cdot \mu$ and we get the result.

Assume now that γ_1' is b_1-central. There are several cases to investigate, depending on the relative signs of $p(\sigma)$ and $p(\theta)$ on one hand, and how their weights are ordered.

If $p(\sigma) \times p(\theta) \geq 0$ **and** $(|p(\theta)| \times W(\sigma), op(\sigma), |p(\theta)| \times |\sigma|) \leq_{lex} (|p(\sigma)| \times W(\theta), op(\theta), |p(\sigma)| \times |\theta|)$, then we are going to move some θ_i to the left, replacing them with σ_j's. More precisely, we claim that there is a path

$$\gamma' = \gamma_1 \cdot \sigma_1 \ldots \cdot \sigma_p \cdot \sigma_{p+1} \ldots \cdot \sigma_{p+|p(\theta)|} \cdot \gamma_2'' \cdot \theta_{|p(\sigma)|+1} \cdot \ldots \theta_q \cdot \gamma_3$$

in G^m which is smaller or equivalent to γ. The paths σ_i for $i > p$ are defined by $\sigma_i = Expand_m(\sigma, i_0 + |p(\sigma)| \times (i-p))$ if the source of γ_2' is some vertex v_{i_0} . $\gamma_2'' = Expand_m(\pi(\gamma_2'), p(\sigma) \times p(\theta) + i_0)$. γ' is a path in G^m indeed: since $\pi(\sigma_i)$ is a cycle which is independent of i, the target vertex of every piece of γ' coincides with the source vertex of the next piece. Moreover, since γ is b_1-central and since we have increased (resp. decreased) the maximal index (resp. minimal) of a vertex by at most $p(\sigma) \times p(\theta) \leq P_G^2$, the path γ' stays within the limits of G^m. Finally, the weight of γ' is smaller or equal to the weight of γ, thanks to the inequality $|p(\theta)| \times W(\sigma) \leq |p(\sigma)| \times W(\theta)$ and, in case of equality, either γ' is stricter than γ or γ' is shorter than γ: in all cases $\gamma' \trianglelefteq \gamma$. If they are equivalent, then the second component of the measure: $|\gamma'| - |\sigma_1 \ldots \sigma_{p+|p(\theta)|}|$ is strictly smaller than the same measure for γ: we may apply the induction hypothesis to γ'.

If $p(\sigma) \times p(\theta) \geq 0$ **and** $(|p(\theta)| \times W(\sigma), op(\sigma), |p(\theta)| \times |\sigma|) >_{lex} (|p(\sigma)| \times W(\theta), op(\theta), |p(\sigma)| \times |\theta|)$ the case is symmetric of the previous one: instead of moving θ_i's to the left, we move the σ_i's to the right, yielding again a smaller γ'.

If $p(\sigma) \times p(\theta) < 0$ **and** $(|p(\theta)| \times W(\sigma) + |p(\sigma)| \times W(\theta), op(\sigma) \times op(\theta)) <_{lex} (0, 1)$, then we add σ's and θ's in both parts: let

$$\gamma' = \gamma_1 \cdot \sigma_1 \cdots \sigma_p \cdot \sigma_{p+1} \cdots \sigma_{p+|p(\theta)|} \cdot \gamma_2'' \cdot \theta_1' \cdots \theta_q' \cdot \theta_{q+1}' \cdots \theta_{q+|p(\sigma)|}' \cdot \gamma_3$$

γ' is a path in G^m for the same reasons as before. For instance, the maximal index (resp. the minimal index) of a vertex has been increased (resp. decreased) by at most $|p(\sigma) \times p(\theta)|$ because $p(\sigma)$ and $p(\theta)$ have opposite signs. And, since γ is b_1-central, indices of vertices in γ' stay within $[0..m]$. We have also to check that we can glue the different paths; for instance the source of γ_3 is the target of the path before because:

$$\begin{aligned} p(\sigma^{p+|p(\theta)|}) + p(\theta^{q+|p(\sigma)|}) &= p \times p(\sigma) + p(\sigma) \times |p(\theta)| + p(\theta) \times q + p(\theta) \times |p(\sigma)| \\ &= q \times p(\theta) + p \times p(\sigma) \quad \text{since } p(\sigma) \times p(\theta) \leq 0 \\ &= p(\sigma^p) + p(\theta^q) \end{aligned}$$

Now, thanks to the hypothesis on the weights and strictness, $\gamma' \trianglelefteq \gamma$ and they are not equivalent; we may again conclude using the induction hypothesis.

$p(\sigma) \times p(\theta) < 0$ **and** $(|p(\theta)| \times W(\sigma) + |p(\sigma)| \times W(\theta), op(\sigma) \times op(\theta)) \geq_{lex} (0,1)$, then we remove σ_i's and θ_j's, yielding a new path

$$\gamma' = \gamma_1 \cdot \sigma_1 \cdots \sigma_{p-|p(\theta)|} \cdot \gamma_2'' \cdot \theta_1' \cdots \theta_{q-|p(\sigma)|}' \cdot \gamma_3$$

As in the previous case, we indeed get a path in G^m. Its weight is smaller or equal to the weight of γ. In case of equality, by hypothesis, $op(\sigma) = op(\theta) = 1$, hence $op(\gamma') = op(\gamma)$. Then γ' is shorter than γ and we may apply the induction hypothesis.

$\square$

5.6 Zigzags

Lemma 10 shows that the central parts of the paths can be assumed to be regular. If we consider now an arbitrary path, either it is looping, or it contains at most n central parts (n is the number of vertices of G). Hence we may decompose every path into a fixed number of regular paths:

Lemma 11 *For every path γ in G^m, there is a path $\gamma' \trianglelefteq \gamma$ such that γ' can be decomposed as:*

$$\gamma' = \delta_0 \cdot \gamma_1 \cdot \delta_1 \cdots \gamma_k \cdot \delta_k$$

where k is smaller than the number of vertices of G, every γ_i is b_1-central and b_2-regular and every δ_j is either empty or its vertices belong to $\bigcup_{i=1}^{b_1} V_i \cup \bigcup_{i=m-b_1}^{m} V_i$.

Proof:
We may assume w.l.o.g. that γ is loop-free (if it is not, replace it with a path in which the loops have been removed; such a new path is smaller w.r.t. $\trianglelefteq$).

Every path γ can be decomposed (in only one way) into $\delta_0 \cdot \gamma_1 \cdot \delta_1 \cdots \gamma_k \cdot \delta_k$ with γ_i b_1-central and each δ_i has its vertices in $\bigcup_{i=1}^{b_1} V_i \cup \bigcup_{i=m-b_1}^{m} V_i$. Let $C(\gamma)$ be the sum of the lengths of the paths γ_i in such a decomposition. and $k(\gamma)$ be the number of components in such a decomposition.

We prove the lemma by induction on the triple $(\gamma, -k(\gamma), C(\gamma))$ lexicographically ordered (by $\trianglelefteq$ on the first component). This gives a well-founded ordering since $k(\gamma)$ is smaller than the number of vertices of G (otherwise there is a loop in γ).

If $C(\gamma) = 0$, then there is no central part in γ, hence the lemma (and the base case). If there is at least one central part, then, thanks to lemma 10, for each γ_i, there is a path $\gamma'_i \trianglelefteq \gamma_i$ such that:

$$\gamma'_i = \eta_i \cdot \gamma''_i \cdot \theta_i$$

and η_i, θ_i are either empty or b_1-external and γ''_i is b_1-central and b_2-regular. Replacing γ_i with $\eta_i \cdot \gamma''_i \cdot \theta_i$ in γ, we get a path $\gamma' \trianglelefteq \gamma$:

$$\gamma' = \delta_0 \cdot \eta_1 \cdot \gamma''_1 \cdot \theta_1 \cdot \delta_1 \cdot \eta_2 \cdots \gamma''_k \cdot \theta_k \cdot \delta_k$$

γ' contains at least as many central parts as γ and $\gamma' \trianglelefteq \gamma$, $C(\gamma') \leq C(\gamma)$. If one of the inequalities is strict, then we apply the induction hypothesis. If γ and γ' are equivalent and $C(\gamma) = C(\gamma')$. γ and γ' being equivalent, they do have the same length, hence

$$\sum_{i=1}^{k} |\gamma''_i| = \sum_{i=1}^{k} |\gamma_i| \geq \sum_{i=1}^{k} (|\gamma''_i| + \eta_i| + |\theta_i|)$$

This implies that η_i and θ_i are empty, hence the above decomposition of γ' satisfies the properties stated in the lemma. $\square$

5.7 The main result

Now, we can complete the main result:

Theorem 2 *For every parametrized graph G such that, for every m, G^m does not contain a cycle of negative weight, there are (computable) finite sets $F^{b,p}_{i,j} \subset D \times D$, where $b \in \{0,1\}^3$, $1 \leq i,j \leq |G|$, $0 \leq p \leq N_G$, N_G is the least common multiple of the length of cycles in $K(G)$, such that $(m, \vec{U}, \vec{U'}) \in R_G$ if and only if m is equal to p modulo N_G and $(\frac{m-p}{N_G}, \vec{U}, \vec{U'})$ satisfies the formula*

$$\phi_p(n, \vec{x}, \vec{x'}) \stackrel{def}{=} \bigwedge_{b \in \{0,1\}^3} \phi^b_p(n, \vec{x}, \vec{x'})$$

$$\phi^{000}_p \stackrel{def}{=} \bigwedge_{(\alpha,\beta) \in F^{000,p}_{i,j}} x_i \leq x_j + n\alpha + \beta$$

$$\phi_p^{001} \stackrel{def}{=} \bigwedge_{(\alpha,\beta)\in F_{i,j}^{001,p}} x_i \le x'_j + n\alpha + \beta$$

$$\phi_p^{010} \stackrel{def}{=} \bigwedge_{(\alpha,\beta)\in F_{i,j}^{010,p}} x'_i \le x_j + n\alpha + \beta$$

$$\phi_p^{011} \stackrel{def}{=} \bigwedge_{(\alpha,\beta)\in F_{i,j}^{011,p}} x'_i \le x'_j + n\alpha + \beta$$

And the other formulas are defined replacing the large inequality with a strict one and setting the first bit of b to 1.

Proof:
Let us consider for instance the case of non strict inequalities and relations between x_l and x_j (the 7 other cases are similar). We only have to consider the minimal paths (w.r.t. $\trianglelefteq$) in G^m whose source is x_l^0 and target is x_j^0. By lemma 11, such paths γ can be decomposed as

$$\delta_0 \cdot \gamma_1 \cdots \gamma_k \cdot \delta_k$$

where the paths δ_i range (up to translation) in a finite set of paths (the paths of G^{b_1}), which does not depend on m, $k < |G|$ and each γ_i can be written

$$\gamma_i^1 \cdot \sigma_i^1 \cdots \sigma_i^{p_i} \cdot \gamma_i^2$$

where $\pi(\sigma_i^j) = \sigma_i$ is independent of j and is an elementary cycle in $K(G)$. Moreover, $|\gamma_i^1|, |\gamma_i^2| \le b_2$, which only depends on G. the constraint c_γ is then

$$x_l \le x_j + W(\delta_0) + \sum_{i=1}^{k} (W(\gamma_i^1) + p_i \times W(\sigma_i) + W(\gamma_i^2) + W(\delta_i))$$

and, for every i, the centrality condition implies $m - 2 \times b_1 = |p(\gamma_i^1) + p(\gamma_i^2) + p_i \times p(\sigma_i)|$, which can be written (assuming e.g. that $p(\gamma_i) > 0$), $p_i = \frac{m-2b_1-p(\gamma_i^1)-p(\gamma_i^2)}{p(\sigma_i)}$. If we let m' be $\frac{m-p}{N_G}$ (which is an integer by hypothesis), we get

$$p_i = m' \frac{N_G}{p(\sigma_i)} + \frac{p - 2b_1 - p(\gamma_i^1) - p(\gamma_i^2)}{p(\sigma_i)}$$

each fraction being an integer. Then the inequality becomes:

$$x_l \leq x_j + \underbrace{\sum_{i=1}^{k} \frac{N_G}{p(\sigma_i)} W(\sigma_i)}_{\alpha} \times m' + \beta$$

where

$$\beta = W(\delta_0) + \sum_{i=1}^{k} (W(\gamma_i^1) + W(\gamma_i^2) + W(\delta_i) + \frac{p - 2b_1 - p(\gamma_i^1) - p(\gamma_i^2)}{p(\sigma_i)} \times W(\sigma_i))$$

α, β range in a finite set which does not depend on m', hence the result. $\square$

Corollary 1 *For any parametrized graph G, R_G is definable in $\mathcal{T}_D$.*

Proof:
This is a consequence of theorem 2 and lemma 8. $\square$

This can also be translated, according to lemma 2:

Corollary 2 *For every $c \in \mathcal{C}(\vec{x})$, $\mathsf{FIX}_n(c(\vec{x}, \vec{x'}))$ is definable in $\mathcal{T}_D$.*

Corollary 3 *Every single state, single transition counter automaton defines a relation which is also definable in $\mathcal{T}_D$.*

5.8 Iterability

We saw in theorem 1 that the existence of a uniform bound on the number of iterations of a loop is decidable. In other words, we decided whether $\mathsf{FIX}_n(c)$ is logically equivalent to some $n \leq m \wedge \mathsf{FIX}_n(c)$ for some given m.

This does not answer the following question: can a loop be *infinitely* iterated ? It might be the case that, for every given m, there is an initial value $\vec{x_0}$ such that the loop can be iterated more than m times, but there always a bound on the number of iterations, which depends on the initial value. For instance consider a constraint $x' \leq x - 1$ (interpreted in $\mathbb{N}$). For each starting value x_0, the loop can be iterated at most x_0 times. However, there is no uniform bound on the number of iterations.

To be more precise, we distinguish 3 notions of iterability:

Definition 12 *Let G be a p-graph and ϕ_G be the formula defining R_G (as in corollary 1). We say that*

1. *The iterations of G are* uniformly bounded *if*

$$\models \exists N \forall \vec{x}, \vec{x'}, \forall n. \phi_G(n, \vec{x}, \vec{x'}) \Rightarrow n \leq N$$

2. *The iterations of G are* bounded *if,*

$$\models \forall \vec{x}, \exists N, \forall \vec{x'}, \forall n. \phi(n, \vec{x}, \vec{x'}) \Rightarrow n \leq N$$

3. *G is* infinitely iterable *if there is an infinite sequence $\vec{U_i} \in D^n$ such that, for every i, $\vec{U_i}, \overrightarrow{U_{i+1}} \models \phi_G(1, \vec{x}, \vec{x'})$.*

Note that this last property cannot be expressed as a first-order formula in $\mathcal{T}_D$. Let us also illustrate the difference between the two last notions with an example

Example 14 Consider the following transition system: configurations consist of pairs of natural numbers and there is a move from (i, j) to (i', j') if and only if either $i' = i+1$ and $j' = j-1 \geq 0$ or $i = j = 0$ and $i' = 1, j' \geq 0$. It is easy to see that, from $(0, 0)$ it is possible to iterate n times the loop, for every n (there are paths of arbitrary length). However it is impossible to iterate the loop infinitely often as there is no infinite path in the transition graph.

As a consequence of theorem 2, we can derive the following lemma which states that for every initial value for which G can be iterated arbitrarily many times, there is a particular sequence of configurations, which yields an infinite iteration of G.

Lemma 12 *Let G be any parametrized graph. Either the iterations of G are bounded, or else, for every $\vec{V_0}$ such that $\vec{V_0} \models \forall n, \exists \vec{V'}. \phi_G(n, \vec{V_0}, \vec{V'})$, there is a vector $\vec{V_1}$ such that, for every $m \geq 1$, $\vec{V_0}, m \times N_G, \vec{V_0} + m \times \vec{V_1} \models \phi_G(n, \vec{x}, \vec{x'})$*

Proof:
Let $S = \{\vec{V} \mid \vec{V} \models \forall n. \exists \vec{V'_n}. \phi_G(n, \vec{V}, \vec{V'_n})\}$. S is the set of vectors from which it is possible to iterate G an unbounded number of times. We assume here that $S \neq \emptyset$. Let $\vec{V_0} \in S$. By theorem 2, for every pair of variables x, y there are constants $\alpha_x, \beta_x, \gamma_x, \delta_x, \epsilon_{x,y}, \eta_{x,y}$ (resp. $\alpha'_x, \beta'_x, \gamma'_x, \delta'_x, \epsilon'_{x,y}, \eta'_{x,y}$) such that, for large enough m,

$$m \times N_G, \vec{V_0}, \vec{V'} \models \phi_G(n, \vec{x}, \vec{x'})$$

if and only if

$$\vec{V'}, m \models \psi_{\vec{V_0}}(\vec{V}, n)$$

where

$$\psi_{\vec{V_0}} \stackrel{\text{def}}{=} \bigwedge_{x \in \vec{V}} \begin{cases} \quad x \leq \alpha_x \times n + \beta_x \\ \wedge x \geq \gamma_x \times n + \delta_x \\ \wedge x - y \leq \epsilon_{x,y} \times n + \eta_{x,y} \\ \wedge x < \alpha'_x \times n + \beta'_x \\ \wedge x > \gamma'_x \times n + \delta'_x \\ \wedge x - y < \epsilon'_{x,y} \times n + \eta'_{x,y} \end{cases}$$

For instance α_x, β_x can be defined as follows:

$$\alpha_x \stackrel{\text{def}}{=} \min\{\alpha \mid (\alpha, \beta) \in F_{i,j}^{000,0}, 1 \leq j \leq |G|\}$$

if i is the index corresponding to x and

$$\beta_x \stackrel{\text{def}}{=} \min\{\beta + x_j \mid (\alpha_x, \beta) \in F_{i,j}^{000,0}, 1 \leq j \leq |G|\}$$

We may also assume that such formulas are closed by transitivity, i.e. the constants are the minimal (resp. maximal) possible ones. Assume first that there is no strict inequalities (i.e. they are all consequences of the large inequalities).

Let $\vec{V_1}$ be the vector whose components are the γ_x. We are going to show that, if $\vec{V}, m \models \psi$, then $\vec{V} + \vec{V_1}, m + 1 \models \psi$. There are three sets of inequalities:

- Concerning the first set of inequalities, it is sufficient to prove that

  $$\vec{V}, m \models x + \gamma_x \leq \alpha_x \times (n+1) + \beta_x$$

  which is a consequence of $\alpha_x \geq \gamma_x$. The latter inequality holds because $\vec{V_0} \in S$, hence ψ is satisfiable for every n,

- The second set of inequalities is obviously satisfied

- Concerning the third set of inequalities, it is sufficient to prove that

  $$\gamma_x - \gamma_y \leq \epsilon_{x,y}$$

  which is a consequence of the closure by transitivity:

  $$\psi \models y \geq (-\epsilon_{x,y} + \gamma_x)n - \eta_{x,y} + \delta_x$$

  hence $\gamma_y \geq -\epsilon_{x,y} + \gamma_x$ by maximality of γ_y.

Now, if there are strict inequalities, for each group of inequalities and each variable(s) we keep in ψ either a strict inequality or a large one: depending on how e.g. α_x and α'_x compare, one relation is the consequence of the other. Then, everything works as above, except that $\vec{V_1}$ contains, for each component x, γ_x if $\gamma_x > \gamma'_x$ and γ'_x otherwise. □

Corollary 4 *The iterations of G are unbounded iff G is infinitely iterable.*

Proof:
Following lemma 12, if the iterations of G are unbounded, we can construct an infinite sequence of configurations, which shows that G is infinitely iterable. The converse is straightforward. □

Corollary 5 *It is decidable whether G (resp. a loop of a counter automaton) is infinitely iterable.*

Proof:
This is a consequence of corollary 4, corollary 1 and the (first-order) definition of boundedness. □

6 The reachability relation for flat counter automata

Once we have shown that the iteration of a loop of a counter automaton is definable in the additive theory of the underlying structure, it is easy to extend the results to flat automata.

Lemma 13 *Let $\mathcal{A}$ be any counter automaton. For every sequence of transitions $t_1, \ldots, t_k$ of $\mathcal{A}$ such that, for every i, $t_i = (q_{i-1}, a_i, c_i, q_i)$, there is a constraint $c(\vec{x_0}, \vec{x_k}) \in \mathcal{C}$ which is logically equivalent to*

$$\exists \vec{x_1}, \ldots, \exists \vec{x_{k-1}}. \bigwedge_{i=1}^{k} c_i(\vec{x_{i-1}}, \vec{x_i}).$$

Proof:
We can proceed by induction on k: if $k = 1$ the result is immediate since there is no quantification. If $k > 1$, by induction hypothesis, it is sufficient to show that, for any two constraints $c_1, c_2 \in \mathcal{C}$, $\exists \vec{y}. c_1(\vec{x}, \vec{y}) \wedge c_2(\vec{y}, \vec{z})$ is logically

equivalent to a constraint $c(\vec{x}, \vec{z}) \in \mathcal{C}$. This is in turn a consequence of the classical variable elimination procedure; we assume that every inequality is written in such a way that one of its members is in $\vec{y}$ (or else it does not contain any occurrence of a variable in $\vec{y}$). Then c is the conjunction, for every index i, of the formulas

$$\bigwedge_{u \leq y_i} \bigwedge_{y_i \leq v} u \leq v$$

as well as similar conjunctions for strict inequalities. Now, the resulting constraint is a conjunction of inequalities $u \leq v$ where u, v are of the form $x_j + \alpha$ or $z_j + \beta$. Such inequalities reduce to the required form by subtracting α and β. $\square$

Theorem 3 *If $\mathcal{A}$ is a flat counter automaton, then for every states q, q' of $\mathcal{A}$ the binary relation $\xrightarrow[q,q']{*}$ (resp. the ternary relation $\xrightarrow[q,q']{n}$) is effectively definable in the additive theory of D.*

Proof:
Consider the relation $\geq$ on states which is defined by $q \geq q'$ iff there is a transition sequence from q to q'. If $\approx$ is the associated equivalence relation, then, when $q \approx q'$ implies that there is at most one (loop-free) transition sequence from q to q'. We also let $>$ be $\geq \setminus \approx$.

We proceed by induction on the number of equivalence classes modulo $\approx$. If all states of Q are equivalent, then there is exactly one loop (the graph structure of the automaton is a ring). More precisely, there are exactly k transitions $t_1, \ldots t_k$ in $\mathcal{A}$ such that, for every i, $t_i = (q_{i-1}, a_i, c_i, q_i)$ and $q_k = q_0$. Assume w.l.o.g that $q = q_0$ and $q' = q_i$. By lemma 13 there is a constraint c which is equivalent to $\exists \vec{x_1}, \ldots, \exists \vec{x_{k-1}}. \bigwedge_{i=1}^{k} c_i(\vec{x_{i-1}}, \vec{x_i})$ and a constraint c' which is logically equivalent to $\exists \vec{x_1}, \ldots, \exists \vec{x_{i-1}}. \bigwedge_{j=1}^{i} c_j(\vec{x_{j-1}}, \vec{x_j})$. $\vec{v} \xrightarrow[q,q']{m} \vec{v'}$ iff $m = k \times m_1 + i$ for some m_1 and $m_1, \vec{v}, \vec{v_1} \models \mathsf{FIX}_n(c)$, $\vec{v_1}, \vec{v'} \models c'$ for some $\vec{v_1}$. By corollary 2, $\mathsf{FIX}_n(c)$ is effectively expressible as a formula ϕ in the additive theory of D. Therefore, $\xrightarrow[q,q']{n}$ is expressible in the additive theory of D by

$$\exists \vec{v_1} \exists m_1 . \mathsf{Int}(m_1) \wedge n = k \times m_1 + i \wedge \phi(m_1, \vec{v}, \vec{v_1}) \wedge c'(\vec{v_1}, \vec{v'})$$

Now assume that there is more than one class modulo $\approx$. Consider a maximal class ϵ. As shown above, for every pair of states q_1, q_1' in this class, we can compute a formula in $\mathcal{T}_D$ expressing the relation $\xrightarrow[q_1,q_1']{n}$. By maximality, all states of $\mathcal{A}$ which are not in ϵ are strictly smaller than any state which is in ϵ. If we remove from the automaton all elements of ϵ and all transitions whose source is in ϵ, we get an automaton with strictly less classes. Then, by induction hypothesis, for every pair of states q_2, q_2' there is a formula of $\mathcal{T}_D$ expressing $\xrightarrow[q_2,q_2']{n}$.

If $q, q' \in \epsilon$ or if $q, q' \notin \epsilon$, then the result is proved. Moreover, by construction, we cannot have $q \notin \epsilon$ and $q' \in \epsilon$. hence only one case remains to be considered: $q \in \epsilon$ and $q' \notin \epsilon$. $\xrightarrow[q,q']{n}$ can then be expressed by guessing the (single !) transition point between a state of ϵ and a state not in ϵ:

$$\exists n_1, n_2.\vec{v_1}, \vec{v_2}. \bigvee_{\substack{q_1 \in \epsilon, q_2 \notin \epsilon \\ (q_1, a, c_0, q_2) \in T}} (\vec{v} \xrightarrow[q,q_1]{n_1} \vec{v_1} \wedge c_0(\vec{v_1}, \vec{v_2}) \wedge \vec{v_2} \xrightarrow[q_2,q']{n_2} \vec{v'} \wedge n = n_1 + n_2 + 1)$$

expresses $\xrightarrow[q,q']{n}$. $\square$

7 Applications and perspectives

We do not intend to develop here some applications, but only to sketch some possible directions.

7.1 Safety analysis

Safety questions often reduce to the reachability of a catastrophic state. There are classically two approaches for the reachability problem. The *forward analysis* which consist in starting from an initial (set of) configurations and compute the successor configurations until either a fixed point is reached or the catastrophic state belongs to the successors sets. For models with counters, this computation may not terminate and there are several speedup procedures (also called *widening*) [15, 16]. This is known as the $Post^*$ computation (or approximation): $Post^*(S)$ is the set of all configurations which can be reached when starting from a configuration in S.

The *backward analysis* consists in the converse: start from the catastrophic state(s) and compute the predecessor configurations until either a fixed point is reached or there is a non-empty intersection with the set of initial configurations. This is the $Pre^*(S)$ computation.

In both cases, we need a symbolic representation of sets of configurations, which is effectively closed by one-step successor (resp. predecessor) and by conjunction and disjunction and whose emptiness is decidable. The Presburger constraints (or, more generally, additive constraints) are a good candidate and are indeed used in several such situations. (See for instance [7] and [13, 21] for Presburger arithmetic and [2, 10] for real arithmetic).

In this area, our main result (theorem 2) has an important application: if we assume that the counter automaton is flat, then $Pre^*(S)$ as well as $Post^*(S)$ are computable: there are effectively computable constraints whose solutions are the configurations in $Pre^*(S)$ (resp. $Post^*(S)$). This explains and generalize for instance a similar result concerning BPP (a special class of Petri nets) [12, 11].

Moreover, S can be kept as a parameter. This means that we may "compile" the binary reachability relation (tailored to a given automaton); for each set of states S, we will only have to simplify the formula, roughly replacing the parameters with actual values.

We have seen that our computation is quite complex (our algorithm is exponential). However, it has to be done only once for the model: the result can be reused for several verifications. As we saw above, the reachability analysis will simply consist in "evaluating" our program on actual data.

7.2 Meta-transitions

The advantage of computing the binary reachability relation is the ability to faithfully abstract the corresponding automaton using a *meta transition* (see e.g. [4]). The idea is to replace a possibly complex part of an automaton (with a flat control !) with a single transition, labeled with the constraint relating the inputs and the outputs.

7.3 Timed automata

One weakness of our result is the flatness requirement. Of course, we know that multiplication can be implemented with two loops and hence reachability becomes undecidable. This means that this condition is necessary. However, how wide is the class of systems to which our result can be applied? We have shown since the first version of this paper that Alur and Dill's timed automata [1] fall into the class of flat systems [8]. More precisely, as far as only relations between clock values are concerned, it is possible, for every timed automaton, to compute an equivalent flat timed automaton. Then, using a simple transformation, we may use theorem 2. This shows that, besides BPP, there is a non trivial class of models to which our result can be applied.

Acknowledgements

We acknowledge L. Fribourg for many discussions on multiple counters automata. We acknowledge V. Bruyère and S. Pinchinat for their comments on an earlier version of this paper.

References

[1] R. Alur and D. Dill. Automata for modeling real-time systems. In *Proc. 17th Int. Coll. on Automata, Languages and Programming, Warwick, LNCS 443*, pages 322–335. Springer-Verlag, 1990.

[2] B. Bérard and L. Fribourg. Reachability analysis of (timed) petri nets using real arithmetic. In *Proc. CONCUR'99*, volume 1664 of *Lecture Notes in Computer Science*. Springer-Verlag, 1999.

[3] O. Bernholtz, M. Vardi, and P. Wolper. An automata-theoretic approach to branching time model checking. In *Proc. Computer Aided Verification*, 1994.

[4] B. Boigelot. *Symbolic Methods for Exploring Infinite State Spaces*. PhD thesis, Faculty of Applied Sciences of the University of Liège, 1998.

[5] B. Boigelot, S. Rassart, and P. Wolper. On the expressiveness of real and integer arithmetic automata. In *Proc. 25th Int. Colloqium on Automata Languages and Programming*, volume 1443 of *Lecture Notes in Computer Science*, pages 152–163, 1998.

[6] B. Boigelot and P. Wolper. Symbolic verification with periodic sets. In *Computer Aided Verification, Proc. 6th Int. Conerence*, LNCS, Stanford, June 1994. Springer-Verlag.

[7] T. Bultan, R. Gerber, , and W. Pugh. Symbolic model checking of infinite state systems using presburger arithmetic. In O. Grumberg, editor, *Proc. Computer Aided Verification*, volume 1254 of *LNCS*, Haifa, Israel, 1997. Springer-Verlag.

[8] H. Comon and Y. Jurski. Timed automata and the theory of real numbers. In *Proc. Conf. on Concurrency Theory (CONCUR)*, number 1664 in Lecture Notes in Computer Science, pages 242–257. Springer-Verlag, 1999.

[9] P. Cousot and N. Halbwachs. Automatic discovery of linear restraints among variables of a program. In *Proc. Int. Conf. on Princinples Of Programming Languages (POPL)*, 1978.

[10] G. Delzanno and A. Podelski. Model checking in clp. In *Proc. TACAS'99*, volume 1579 of *Lecture Notes in Computer Science*. Springer-Verlag, 1999.

[11] J. Esparza. Decidability of model checking for infinite-state concurrent systems. *Acta Informatica*, 34:85–107, 1997.

[12] L. Fribourg and H. Olsen. A decompositional approach for computing least fixed-point of datalog programs with z-counters. *J. Constraints*, 1997.

[13] L. Fribourg and H. Olsèn. Proving safety properties of infinite state systems by compilation into presburger arithmetic. In *Proc. CONCUR'97*, volume 1243 of *Lecture Notes in Computer Science*, pages 96–107. Springer-Verlag, 1997.

[14] L. Fribourg and J. Richardson. Symbolic verification with gap-order constraints. Research Report LIENS-96-3, Ecole Normale Supérieure, Paris, Feb. 1996.

[15] N. Halbwachs. Delay analysis in synchronous programs. In *Proc. Computer Aided Verification*, LNCS 697, pages 333–346. Springer-Verlag, 1993.

[16] N. Halbwachs, Y.-E. Proy, and P. Raymond. Verification of linear hybrid systems by means of convex approximations. In *Proc. Static Analysis Symposium*, Lecture Notes in Computer Science, pages 223–237. Springer-Verlag, 1994.

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

[18] V. Pratt. Two easy theories whose combination is hard. Research report, Massachusetts Institute of Technology, 1977.

[19] M. Presburger. über die Vollständingen einer gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. In *Comptes Rendus du premier Congrès des Mathématiciens des Pays slaves*, Warszawa, 1929.

[20] P. Revesz. A closed form for datalog queries with integer order. In *Proc 3rd International Conference on Database Theory*, pages 187–201, Paris, 1990.

[21] T. Shiple, J. Kukula, and R. Ranjan. A comparison of presburger engines for EFSM reachability. In *Proc. Computer Aided verification (CAV)*, volume 1427 of *Lecture Notes in Computer Science*, pages 280–292, 1998.

[22] D. Toman, J. Chomicki, and D. S. Rogers. Datalog with integer periodicity constraints. In *Int. Symp. on Logic Programming*, 1994.