IOS Press

Flat Parametric Counter Automata

Marius Bozga

Radu Iosif

Yassine Lakhnech

Verimag, CNRS, Université de Grenoble
2 Avenue de Vignate, 38610, Gières, France
{bozga,iosif,lakhnech}@imag.fr

Abstract. In this paper we study the reachability problem for parametric flat counter automata, in relation with the satisfiability problem of three fragments of integer arithmetic. The equivalence between non-parametric flat counter automata and Presburger arithmetic has been established previously by Comon and Jurski. We simplify their proof by introducing finite state automata defined over alphabets of a special kind of graphs (zigzags). This framework allows one to express also the reachability problem for parametric automata with one control loop as the satisfiability of a *1-parametric linear Diophantine systems*. The latter problem is shown to be decidable, using a number-theoretic argument. In general, the reachability problem for parametric flat counter automata with more than one loops is shown to be undecidable, by reduction from Hilbert's Tenth Problem. Finally, we study the relation between flat counter automata, integer arithmetic, and another important class of computational devices, namely the 2-way reversal bounded counter machines.

Keywords: Counter machines, Reachability problems, Diophantine systems

1. Introduction

Flat counter automata [5], [7], [3] have been extensively studied, as an important class of infinite-state systems, for which the reachability problem is decidable. The results obtained so far have been used in a number of successful system verification tools, like FAST [2], LASH [22] or TREX [1].

Comon and Jurski show in [5] that the relation between input and output counter values, for a flat counter automaton can be expressed in Presburger arithmetic, provided that the automata have transition relations that are conjunctions of relations of the form $x - y \leq c$, where x and y denote either the current

Address for correspondence: Verimag, 2 Avenue de Vignate, 38610, Gières, France

or the future (primed) values of the counters, and c is an integer constant. To our knowledge, their result concerns the most general class of flat counter automata, considered so far.

The contributions of the present paper are many fold. First, we give an alternative, easier, proof of the result of [5], using finite state automata defined over alphabets of graphs (zigzags). Second, we apply this framework to a more general class of flat counter automata, in which, besides integer constants, parameters are allowed to occur within transition relations. In general, the reachability of a designated control state in this case is equivalent to the existence of solutions of a Diophantine system. Since the latter problem is undecidable [16], this entails the undecidability of the reachability problem for parametric flat automata. However, when we restrict the control structure of parametric automata to one loop, the reachability problem can be expressed as existence of solutions of a particular class of Diophantine systems, called *1-parametric Diophantine systems*.

A *1-parametric Diophantine system* is a linear system with unknowns $x_1, \ldots, x_k$, whose coefficients are polynomials of any degree with one variable m. The satisfiability problem asks whether there exists a constant $c \in \mathbb{N}$, such that the linear system obtained by substituting m with c has a positive solution ? In this paper we show that this problem is decidable.

Last, we study the relation between flat counter automata, integer arithmetic, and another important class of computational devices, the 2-way reversal bounded counter machines [12]. We establish a three-level hierarchy relating flat counter automata and 2-way reversal bounded counter machines by the intermediate of three arithmetic theories : Presburger arithmetic, existential theory of addition and divisibility, and Diophantine systems.

Related Work

Work on the decidability of reachability problems for counter automata starts with the negative result of Minsky [18] regarding two counter machines. The two most studied restrictions of this model are the *reversal bounded* 2-way counter machines [12] and the *flat counter automata* [5], [7], [3]. The class of flat counter automata that is closest to the one considered in this paper is the one studied by Comon and Jurski [5], where the transition relations are conjunctions of inequalities of the form $x - y \leq c$, with $c \in \mathbb{Z}$. Their result is that the set of reachable configurations for such automata is definable in Presburger arithmetic. Our result considers parametric transition relations of the form $x - y \leq f(\mathbf{z})$, and defines the set of reachable configurations as solutions of a linear Diophantine system with one parameter. Decision procedures for this class of systems have been independently found by O. Ibarra and Z. Dang in [13], using a result from the theory of reversal-bounded counter automata, and by Y. Matyiasevich [17]. The latter result uses a similar number theoretic argument, but the proof is based on a more involved case analysis.

2. Preliminary Definitions

We denote by $\mathbb{N}$ the set of natural numbers, and by $\mathbb{Z}$ the set of integer numbers. Let $\mathbf{x} = \{x_1, \ldots, x_k\}$, $k > 1$ be a finite set of variables (counters) ranging over $\mathbb{Z}$, and $\mathbf{x}' = \{x'_i \mid 1 \leq i \leq k\}$. In what follows we will sometimes abusively use the name of a variable to denote its value also. For an arithmetic formula $\varphi(x_1, \ldots, x_k)$ with free variables from $\mathbf{x}$, and a tuple of integer constants $u_1, \ldots, u_k \in \mathbb{Z}$, we denote by $\varphi(u_1, \ldots, u_k)$ the closed formula in which all occurrences of x_i have been replaced by u_i, $1 \leq i \leq k$. For a closed formula φ, we denote by $\models \varphi$ the fact that it is valid, i.e. logically equivalent to true. The notation

$\mathbb{Z}[\mathbf{x}]$ stands for the set of all polynomials with unknowns from $\mathbf{x}$ and integer coefficients. Also $\mathrm{lin}\mathbb{Z}[\mathbf{x}]$ denotes the set of linear terms with unknowns $\mathbf{x}$ and integer coefficients.

For two formulae $\varphi(\mathbf{x},\mathbf{x}')$ and $\psi(\mathbf{x},\mathbf{x}')$ let $\varphi \circ \psi$ denote the relational composition $\exists \mathbf{y} \ . \ \varphi(\mathbf{x},\mathbf{y}) \wedge \psi(\mathbf{y},\mathbf{x}')$. The n-th composition φ^n, $n > 0$ is defined recursively : $\varphi^1 = \varphi$ and $\varphi^{n+1} = \varphi^n \circ \varphi$. The transitive closure φ^* is the infinite disjunction $\bigvee_{n>0} \varphi^n$. Intuitivelly, this represents the relation between $\mathbf{x}$ and $\mathbf{x}'$ after any number of iterations of φ.

A *transition table* over an alphabet Σ is a pair $T = \langle Q, \Delta \rangle$, where Q is a finite set of states and $\Delta \subseteq Q \times \Sigma \times Q$ is a transition relation. We denote $(q,\sigma,q') \in \Delta$ by $q \xrightarrow{\sigma} q'$, whenever T is clear from the context. We denote by $\overleftarrow{T} = \langle Q, \Delta' \rangle$ the *reversed table*, where $(q,\sigma,q') \in \Delta'$ if and only if $(q',\sigma,q) \in \Delta$. A *finite automaton* is a tuple $A = \langle T, Q_0, F \rangle$, where $T = \langle Q, \Delta \rangle$ is a transition table, $Q_0 \subseteq Q$ is a set of initial states, and $F \subseteq Q$ is a set of final states. A *run* of A is a sequence of states and transitions $\pi : q_0 \xrightarrow{\sigma_1} q_1 \xrightarrow{\sigma_2} q_2 \ldots \xrightarrow{\sigma_n} q_n$. We denote by $|\pi|$ the length of the run. The run is said to be *accepting* if and only if $q_n \in F$. The language of A is defined as $\mathcal{L}(A) = \{\sigma_1\sigma_2\ldots\sigma_n \mid q_0 \xrightarrow{\sigma_1} q_1 \xrightarrow{\sigma_2} q_2 \ldots \xrightarrow{\sigma_n} q_n$ is an accepting run of $A\}$. The *reversed automaton* is defined as $\overleftarrow{A} = \langle \overleftarrow{T}, F, Q_0 \rangle$. It is easy to see that $\mathcal{L}(\overleftarrow{A}) = \{\sigma_n\sigma_{n-1}\ldots\sigma_1 \mid \sigma_1\ldots\sigma_{n-1}\sigma_n \in \mathcal{L}(A)\}$.

2.1. Counter Machines

Let us fix a set of counters $\mathbf{x} = \{x_1, x_2, \ldots, x_k\}$ with an ordering of its elements. For any $x \in \mathbf{x}$, let $\zeta(x) = 1$ if $y \neq 0$ and $\zeta(y) = 0$ if $y = 0$. $\zeta(\mathbf{x})$ denotes the tuple $\langle \zeta(x_1), \ldots, \zeta(x_k) \rangle$, for the chosen ordering of the elements in $\mathbf{x}$. Let Σ be a finite alphabet, i.e. a set of symbols. By Σ^* we denote the set of all finite sequences of symbols from Σ. For a finite sequence σ, we denote by $|\sigma|$ its length, and by σ_i, $1 \leq i \leq |\sigma|$ the element at position i in σ.

A *2-way counter machine* over Σ is a tuple $M = \langle \mathbf{x}, Q, \delta, q_0, F \rangle$, where:

- $\mathbf{x}$ is the set of *working counters*,
- Q is the set of *control states*,
- $\delta : Q \times (\Sigma \cup \{\#, \&\}) \times \{0,1\}^k \ \rightarrow \ \mathcal{P}(Q \times \{-1,0,1\} \times \{-1,0,1\}^k)$ is the *transition mapping*,
- $q_0 \in Q$ the *initial state*,
- $F \subseteq Q$ the set of final states.

The input of the machine is a finite tape τ containing a word of the form $\#w\&$, where $\#, \& \notin \Sigma$ are the left and right end of tape markers, and $w \in \Sigma^*$ is a finite word over Σ. M is said to be *deterministic* if and only if $\|\delta(q,\sigma,\mathbf{b})\| \leq 1$, for all $q \in Q, \sigma \in \Sigma$, and $\mathbf{b} \in \{0,1\}^k$.

A configuration $\langle q, i, \mathbf{x} \rangle$ is an element of the set $Q \times \mathbb{N} \times \mathbb{Z}^k$, where q is the current control state, $1 \leq i \leq |w| + 2$ the current position of the input (read only) head, and $\mathbf{x}$ the current values of the counters[1]. A *configuration* $\langle q', j, \mathbf{x}' \rangle$ is said to be the successor of another configuration $\langle q, i, \mathbf{x} \rangle$, denoted by $\langle q, i, \mathbf{x} \rangle \rightarrow$

[1] In the classical literature [12], the counters are assumed to take only positive values, however this is not a restriction, since a machine with integer counters can be simulated by a machine working only on positive counters, by encoding the k-tuple of signs in the control state.

$\langle q', j, \mathbf{x}' \rangle$ if and only if there exists $\langle q', d, \mathbf{a} \rangle \in \delta(q, \tau_i, \zeta(\mathbf{x}))$ such that $j = i + d$ and $\mathbf{x}' = \mathbf{x} + \mathbf{a}$. Intuitivelly, the $\zeta(\mathbf{x})$ component of the transition relation δ denotes the *guard*, used to test whether a counter is zero or not, d is the change in the current position of the input head (-1 moves the head left, 0 leaves the head on the same position, and 1 moves the head right) and the $\mathbf{a}$ component is the *action*, that changes the values of the counters.

As usual, we require a special behavior in the presence of the end of tape markers # and &, preventing the input head to fall off the tape. A *run* of the machine is a sequence of configurations $c_1, c_2, \ldots, c_n$ such that $c_0 = \langle q_0, 1, \mathbf{0} \rangle$ and $c_i \to c_{i+1}$ for $1 \leq i < n$. Notice that we require the machine to start with the input head positioned on the left-end marker, and all counters set to zero.

The run is said to be *accepting* if the control state of the last configuration is an element of F. An input string w, i.e. $\tau = \#w\&$, is *accepted* if the machine has at least one accepting run on it. The set of all accepted strings is the *language* of M, denoted by $\mathcal{L}(M)$. In the following, we work also with the set $\mathcal{V}(M) = \{\mathbf{u} \in \mathbb{Z}^k \mid \langle q_0, 1, \mathbf{0} \rangle, \ldots, \langle q_f, j, \mathbf{u} \rangle$, is an accepting run of M, for some $q_f \in \mathcal{F}$ and $1 \leq j \leq |w| + 2\}$ of counter values produced by the accepting runs of M. Notice that $\mathcal{L}(M) = \emptyset$ if and only if $\mathcal{V}(M) = \emptyset$.

Let C be a class of counter machines. The following, referred to as F-problems, are the problems of deciding for any two machines $M_1, M_2 \in C$, whether:

- *emptiness*: $\mathcal{L}(M_1) = \emptyset$,
- *infiniteness*: $\mathcal{L}(M_1)$ is infinite,
- *disjointness*: $\mathcal{L}(M_1) \cap \mathcal{L}(M_2) = \emptyset$,
- *inclusion*: $\mathcal{L}(M_1) \subseteq \mathcal{L}(M_2)$,
- *universality*: $\mathcal{L}(M_1) = \Sigma^*$,
- *equivalence*: $\mathcal{L}(M_1) = \mathcal{L}(M_2)$.

CM(k, r, l) denotes the class of 2-way counter machines with k counters, and the restriction that, in every accepting run, each counter alternates from increasing to decreasing mode at most r times, and the input head changes direction at most l times. These machines are called (r, l)*-reversal bounded*. Further, let CM(k, r, l, n) be the class of (r, l)-reversal bounded machines restricted to work only on input tapes of the form $\#u_1^{i_1} \ldots u_n^{i_n}\&$, for some $u_j \in \Sigma$ and $i_j \in \mathbb{N}$, $1 \leq j \leq n$. Obviously, CM$(k, r, l, n) \subseteq CM(k, r, l)$. By DCM$(k, r, l, n)$ we denote the subclass of deterministic machines from CM(k, r, l, n). It is well-known that the emptiness, infiniteness and disjointness problems for the class CM(k, r, l, n) are decidable, and moreover, the universe, containment and equivalence problems for the class of DCM(k, r, l, n) are also decidable [12].

In the following, we may lift the restriction on either the number of counter reversals, direction changes, or on the number of input letters, by writing ∞ instead of r, l, n, respectively. We denote $(D)CM(*, r, l, n) = \bigcup_{k=0}^{\infty}(D)CM(k, r, l, n)$, $(D)CM(k, *, l, n) = \bigcup_{r=0}^{\infty}(D)CM(k, r, l, n)$ $(D)CM(k, r, *, n) = \bigcup_{l=0}^{\infty}(D)CM(k, r, l, n)$ and $(D)CM(k, r, l, *) = \bigcup_{n=0}^{\infty}(D)CM(k, r, l, n)$. Moreover, we have $(D)CM(*, *, l, n) = \bigcup_{r=0}^{\infty}(D)CM(*, r, l, n)$, and so on.

Remark To better understand the difference between e.g., $CM(1,*,1,\infty)$ and $CM(1,\infty,1,\infty)$, let us consider $M = \langle\{x\},\{q\},\delta,q,\{q\}\rangle$ a machine over $\Sigma = \{a,b\}$, where $\delta(q,a,1) = \{\langle q,1,1\rangle\}$ and $\delta(q,b,1)$ = $\{\langle q,1,-1\rangle\}$. Obviously $M \in CM(1,\infty,1,\infty)$. However the number of counter reversals in a run of M equals the number of alternations between a and b on the input tape, therefore $M \notin CM(1,r,1,\infty)$, for any $r \in \mathbb{N}$. □

2.2. Counter Automata

In this paper we investigate a subclass of counter machines, defined as follows. Let $\mathbf{x} = \{x_1,x_2,\ldots,x_k\}$ be a set of *working counters*, and $\mathbf{z} = \{z_1,z_2,\ldots,z_l\}$ be a set of *parameters*, intuitively a set of counters whose values never change. We formally require that $\mathbf{x} \cap \mathbf{z} = \emptyset$.

A relation $R(\mathbf{x},\mathbf{x}',\mathbf{z})$ is said to be a *difference bound constraint* if and only if it can be equivalently written as a finite conjunction of atomic propositions of either one of the forms:

- $x_i - x_j \leq \alpha_{ij}(\mathbf{z})$, for some $1 \leq i,j \leq k$,
- $x'_i - x_j \leq \beta_{ij}(\mathbf{z})$, for some $1 \leq i,j \leq k$,
- $x_i - x'_j \leq \gamma_{ij}(\mathbf{z})$, for some $1 \leq i,j \leq k$,
- $x'_i - x'_j \leq \delta_{ij}(\mathbf{z})$, for some $1 \leq i,j \leq k$

where $\alpha_{ij},\beta_{ij},\gamma_{ij},\delta_{ij} \in \mathrm{lin}\mathbb{Z}[\mathbf{z}]$ are linear combinations of parameters from $\mathbf{z}$. The classical definition of difference bound constraints (see e.g. [5]) considers that $\alpha_{ij},\beta_{ij},\gamma_{ij}$, and δ_{ij} are (integer) constants, rather than linear combinations of parameters. One can understand the above definition as a generalization stating the *existence* of bounds on difference terms, rather than the values of these bounds.

For instance, $x - y' = 5z + 7$ is a difference bound constraint, as it is equivalent to the conjunction $x - y' \leq 5 \wedge y' - x \leq -5z - 7$. It is well-known that the class of difference bound constraints is closed under composition of relations. In other words, the existential quantifiers can be eliminated[2], the result being written as another difference bound constraint.

A *counter automaton* (CA) is a counter machine $A = \langle \mathbf{x} \cup \mathbf{z}, Q, \delta, q_0, F\rangle \in \mathrm{CM}(k,\infty,0,1)$ over a singleton alphabet Σ (i.e. $\|\Sigma\| = 1$), in which the transition relation is described by a set of rules of the form $q \xrightarrow{R(\mathbf{x},\mathbf{x}',\mathbf{z})} q'$, where:

- q and q' are control states, and
- R is a difference bound constraint between the values of counters when control is at q, and the corresponding values when control is at q'.

We assume moreover, that the input head is advanced to the right, with each transition. Formally, a configuration of a counter automaton is denoted by a pair $\langle q, \mathbf{xz}\rangle \in Q \times \mathbb{Z}^{k+l}$ (we willingly forget the position of the input head, as it is implicitly given by the position of the configuration in the run). A configuration $\langle q', \mathbf{x}'\mathbf{z}'\rangle$ is the successor of another configuration $\langle q, \mathbf{xz}\rangle$ if and only if there exists a transition rule $q \xrightarrow{R} q'$ such that $\models R(\mathbf{x},\mathbf{x}',\mathbf{z})$.

[2] By e.g. the Fourrier-Motzkin procedure.

For example, consider the counter automaton $A = \langle \{x\}, \{q\}, \{q \xrightarrow{-1 \leq x - x' \leq 1} q\}, q, \{q\} \rangle$ consisting of one control state and a self-loop. A possible run of this counter automaton is $\langle q, 0 \rangle \to \langle q, -1 \rangle \to \langle q, 1 \rangle \to \langle q, -1 \rangle \ldots$. Notice that, even very simple counter automata are not necessarily reversal bounded.

A control state q is said to be the *direct successor* of a control state p if and only if there exists a transition rule $q \xrightarrow{R} q'$. A *control path* is a sequence of states $q_1, q_2, \ldots, q_n$ such that, for all $0 \leq i < n$, q_{i+1} is a direct successor of q_i. The path is said to be non-trivial if $n > 0$. An *elementary cycle* is a non-trivial control path starting and ending with the same state, in which no other state, besides the initial state, occurs more than once.

A counter automaton is said to be *flat* (FCA) if and only if each control state belongs to at most one elementary cycle. A control state with two or more direct successors is said to be a *branching* state. A branching state with exactly two direct successors is said to be a *2-branching* state. A FCA is said to be *linear* (LFCA) if and only if the only branching states are 2-branching, and every elementary cycle contains at most one such state. Notice that every FCA can be effectively turned into a finite union of LFCA, such that the only branching state that is not 2-branching, is the initial state. The interested reader can also refer to [15] for an alternative, yet equivalent definition of flatness.

Since difference bound constraints are closed under relational composition, we can assume without losing generality that, each control path $q_1 \xrightarrow{R_1} q_2 \ldots q_{n-1} \xrightarrow{R_{n-1}} q_n$, with no branching and no transitions incoming along the way, is equivalent to a transition $q_1 \xrightarrow{R_1 \circ \ldots \circ R_{n-1}} q_n$. Applying this transformation to the whole counter automaton, we obtain a *normalized* counter automaton.

We denote by $\mathrm{FCA}(l,n)$ the class of flat counter automata with at most l parameters that occur in the transition relations, and with at most n cycles on each linear component. The class $\mathrm{FCA1A}(l,n)$ is the subclass of $\mathrm{FCA}(l,n)$ in which, the transition relation of each loop, in the normal form, is a formula of the form:

$$x'_i \leq x_j + \alpha \wedge x_k \leq x'_i + \beta \wedge \bigwedge_{j \neq i} x'_j = x_j \wedge \phi(\mathbf{x}) \wedge \psi(\mathbf{x}')$$

where ϕ, ψ are difference bound constraints. In other words, exactly one counter (x_i) is modified at the time. As in the previous, we denote $\mathrm{FCA(1)A}(*,n) = \bigcup_{l=0}^{\infty} \mathrm{FCA(1)A}(l,n)$, $\mathrm{FCA(1)A}(l,*) = \bigcup_{n=0}^{\infty} \mathrm{FCA(1)A}(l,*)$, and $\mathrm{FCA(1)A}(*,*) = \bigcup_{n=0}^{\infty} \mathrm{FCA(1)A}(*,n)$.

2.3. Integer Arithmetic

The undecidability of first-order arithmetic of integers $\langle \mathbb{Z}, +, \cdot, 0, 1 \rangle$ occurs as a consequence of Gödel's Incompleteness Theorem [9]. The basic result has been discovered by A. Church [4], and the essential undecidability (undecidability of its every consistent extension) by B. Rosser [21]. To complete the picture, the existential fragment of the full arithmetic, i.e. *Hilbert's Tenth Problem* [11] was proved undecidable by Y. Matiyasevich [16]. On the positive side, the decidability of the arithmetic of natural numbers with *addition* $\langle \mathbb{Z}, \geq, +, 0, 1 \rangle$ has been shown by M. Presburger [20].

Let us first introduce the theories of Presburger arithmetic [20] and 1-parametric linear Diophantine systems. Presburger arithmetic $\langle \mathbb{Z}, \geq, +, 0, 1 \rangle$ is the theory of first-order logic of addition. The interpretation of logical variables is the set of integers $\mathbb{Z}$, and the meaning of the function symbols $0, 1, +$ is the natural one. Due to the fact that $\langle \mathbb{Z}, \geq, +, 0, 1 \rangle$ admits quantifier elimination [20], every formula in this theory can be shown to be equivalent to a positive boolean combination of relations of one of the following forms :

$$\sum_{i=1}^{n} a_i x_i + b \quad \geq \quad 0 \tag{1}$$

$$\sum_{i=1}^{n} a_i x_i + b \quad = \quad 0 \mod d \tag{2}$$

A *Diophantine equation* is a formula of the form $P(\mathbf{x}) = 0$, where $P \in \mathbb{Z}[\mathbf{x}]$ is a polynomial of the form $P(\mathbf{x}) = \sum_{i=1}^{m} a_i t_i(\mathbf{x}) + a_0$, where t_i are multiplicative terms of the form $\Pi_{l=1}^{k} x_l^{i_l}$, with $i_1, \ldots, i_l \in \mathbb{N}$. An equation is said to be *linear with parameter* x_j, for $1 \leq j \leq k$, if for every multiplicative term of the form above, we have $\sum_{l \in \{1 \ldots k\}}^{l \neq j} i_l \leq 1$. In other words, the only variable that can occur at a power greater than one is x_j, and moreover, all multiplicative terms contain at most one variable, other than x_j. Note that any Diophantine linear equation with parameter m can be equivalently written as:

$$\sum_{i=1}^{n} p_i(m) x_i + p_0(m) = 0 \tag{3}$$

where $p_i \in \mathbb{Z}[m]$, $0 \leq i \leq n$ are polynomials of arbitrary degree in m. In the following, we denote by $\mathfrak{D}[1]$ the set of positive boolean combinations of linear Diophantine equations with one parameter. This definition can be easily generalized to n parameters, denoted by $\mathfrak{D}[n]$ in the following. We denote $\mathfrak{D}[*] = \bigcup_{n=0}^{\infty} \mathfrak{D}[n]$.

3. Counter Machines and Integer Arithmetic

This section is dedicated to a survey of the relations between various classes of counter machines and fragments of integer arithmetic. Intuitively, a class of counter machines is related to a fragment of arithmetic if and only if, for each machine in the class, the set of final configurations is definable in the corresponding arithmetic theory. Formally, a class of counter machines C is related to an arithmetic theory T, denoted as $C \longrightarrow T$ if and only if, for every machine $M \in C$ with k counters, there exists an open formula $\varphi(\mathbf{x})$, $\mathbf{x} = \{x_1, \ldots, x_k\}$ in the language of T, such that $\mathcal{V}(M) = \{\mathbf{u} \in \mathbb{Z}^k \mid \models \varphi(\mathbf{u})\}$.

Dually, a fragment of integer arithmetic is related to a class of counter machines if and only if, each set definable in the arithmetic theory can be produced by a counter machine in the corresponding class. Formally, an arithmetic theory T is related to a class of counter machines C if and only if, for every open formula $\varphi(\mathbf{x})$, $\mathbf{x} = \{x_1, \ldots, x_k\}$ in the language of T there exists a counter machine $M \in C$ with k counters such that $\mathcal{V}(M) = \{\mathbf{u} \in \mathbb{Z}^k \mid \models \varphi(\mathbf{u})\}$. If both $C \longrightarrow T$ and $T \longrightarrow C$, we say that C and T are inter-related, denoted as $C \longleftrightarrow T$.

With this notation, Figure 1 shows the relations between various subclasses of flat counter automata, integer arithmetic and 2-way counter machines. Let us start by proving the relations from the right-hand side of Figure 1. For $\langle \mathbb{Z}, \geq, +, 0, 1 \rangle \longleftrightarrow \mathrm{CM}(*, *, *)$, the right to left direction was proved in [12]. The other direction is proved by the following Lemma.

Lemma 3.1. $\langle \mathbb{Z}, \geq, +, 0, 1 \rangle \longrightarrow \mathrm{CM}(*, *, *)$.

$$\begin{tikzcd}
\mathit{FCA}(0,*) \arrow[r, leftrightarrow] \arrow[d, phantom, sloped, "\subseteq"] & \langle \mathbb{Z}, \geq, +, 0, 1 \rangle \arrow[r, leftrightarrow] \arrow[d, phantom, sloped, "\subseteq"] & \mathit{CM}(*,*,*) \arrow[d, phantom, sloped, "\subseteq"] \\
\mathit{FCA1A}(*,*) \arrow[r, leftrightarrow] \arrow[d, phantom, sloped, "\subseteq"] & \langle \mathbb{Z}, \geq, +, |, 0, 1 \rangle^{\exists} \arrow[r, leftrightarrow] \arrow[d, phantom, sloped, "\subseteq"] & \mathit{DCM}(1,*,\infty,*) \arrow[d, phantom, sloped, "\subseteq"] \\
\mathit{FCA}(*,*) \arrow[r, leftrightarrow] & \mathfrak{D}[*] \arrow[dr] & \mathit{DCM}(*,*,\infty,*) \arrow[l] \arrow[d, phantom, sloped, "\subseteq"] \\
 & & \mathit{CM}(*,*,\infty,*)
\end{tikzcd}$$

Figure 1. A Hierarchy of Flat Counter Automata, Arithmetic and Counter Machines

Proof:
Let $\varphi(\mathbf{x})$, $\mathbf{x} = \{x_1, \ldots, x_n\}$ be a formula of $\langle \mathbb{Z}, \geq, +, 0, 1 \rangle$, i.e. a positive boolean combination of relations of either one of the forms (1) or (2). Let $\Sigma = \{u_1, \ldots, u_n, v_1, \ldots, v_n\}$. We shall build a machine $M \in$ CM$(*,*,*)$ that recognizes the following language:

$$L_{\varphi} = \{s_1^{x_1} \ldots s_n^{x_n} \mid s_i \in \{u_i, v_i\} \text{ and } \models \varphi(\sigma(s_1)x_1, \ldots, \sigma(s_n)x_n)\}$$

where $\sigma(s) = 1$ if $s \in \{u_1, \ldots, u_n\}$ and $\sigma(s) = -1$ if $s \in \{v_1, \ldots, v_n\}$. Since CM$(*,*,*)$ is closed under the operations of union and intersection [12], it is sufficient to exhibit machines that recognize the languages corresponding to (1) and (2). Let $M \in$ CM$(1,n,1)$ be a machine such that :

- The expected input is an element of $(u_1^*|v_1^*) \ldots (u_n^*|v_n^*)$. Any input symbol $s \in \{u_i, v_i\}$ must be followed by either itself, u_{i+1} or v_{i+1}, else M rejects.
- While reading a symbol u_i (v_i) M adds (subtracts) the coefficient a_i from the current value of the counter. At the end of the input, the b value is added to the counter.
- The control of M keeps track of the sign of the counter. There are two disjoint sets of control states Q_+ and Q_-. M moves from Q_+ to Q_- whenever the counter was 0 and a decrement action is performed, and from Q_- back to Q_+ whenever the counter becomes 0 again.
- For (1) M accepts if, at the end of the tape, the control is in a state from Q_+. For (2), if at the end of the tape, M is in a state from Q_+ (Q_-), the value d is repeatedly subtracted (added) from (to) the counter. If, at any time between two consecutive subtractions (additions) the value of the counter is 0, M accepts, otherwise M rejects as soon as the control goes from Q_+ to Q_- (Q_- to Q_+).

Notice that M will halt on any input. It is straightforward to check that the language of the composition of such machines is non empty if and only if the original formula is satisfiable. □

The relations $\langle \mathbb{Z}, \geq, +, |, 0, 1 \rangle^{\exists} \longleftrightarrow$ DCM$(1,*,\infty,*)$ are described in [10]. In the proofs of these relations, it is essential that the counter machines are deterministic. The relation $\mathfrak{D}[n] \longrightarrow$ CM$(*,*,\infty,*)$ is described in [13], for the case $n = 1$, the generalization of their proof for $n > 1$ being straightforward. The relation DCM$(*,*,\infty,*) \longrightarrow \mathfrak{D}[*]$ is given by Lemma 3.2.

It is important to mention that, in particular, we have $\mathfrak{D}[1] \longleftrightarrow$ CM$(k,r,\infty,1) =$ DCM$(k,r,\infty,1)$. The proof of the latter equivalence being given in [14]. Since language emptiness for CM$(k,r,\infty,1)$ is decidable [14], this entails the decidability of the satisfiability problem for the class $\mathfrak{D}[1]$, which is the main result of [13]. This problem is tackled also in Section 5, using a number-theoretic argument instead.

Lemma 3.2. $\mathrm{DCM}(*,*,\infty,*) \longrightarrow \mathfrak{D}[*]$.

Proof:
We need to introduce a definition from [10], generalized to the case of $k \geq 1$. We say that a counter machine $M \in \mathrm{DCM}(k,r,\infty,n)$ is *canonical* if and only if the following hold:

1. M halts on all computations.
2. M changes the direction of the input head only at the tape markers (#, &).
3. M reverses its counters only at the left tape marker (#).

One can show that, for every $M \in \mathrm{DCM}(k,r,\infty,n)$ it is possible to effectively build a finite set of canonical machines $M_I \in \mathrm{DCM}(k,r,\infty,n)$, $1 \leq I \leq N$ for some $N \in \mathbb{N}$ depending only on M, such that $\mathcal{L}(M) = \emptyset$ if and only if $\mathcal{L}(M_I) = \emptyset$, for all $1 \leq I \leq N$. The first two conditions can be satisfied by modifying M to perform only full sweeps of the input, as described in [10]. Next, we simulate M using machines that work on inputs of the form:

$$\#(u_1,\langle l_1,0\rangle)^*(u_1,\langle l_1,1\rangle)(u_1,\langle l_1,2\rangle)^*(u_1,\langle l_2,0\rangle)^*(u_1,\langle l_2,1\rangle)(u_1,\langle l_2,2\rangle)^* \ldots$$
$$(u_n,\langle l_{k-1},0\rangle)^*(u_n,\langle l_{k-1},1\rangle)(u_n,\langle l_{k-1},2\rangle)^*(u_n,\langle l_k,0\rangle)^*(u_n,\langle l_k,1\rangle)(u_n,\langle l_k,2\rangle)^*\&$$

where $\{l_1,\ldots,l_k\}$ is a permutation of $\{1,\ldots,k\}$. We will have a different machine M_I for each such possible indexing. In order to simulate M, M_I treats symbols $(u_i,\langle l_j,l\rangle)$, $1 \leq i \leq n$, $1 \leq l \leq 3$ just as M treats u_i. If M_I attempts to reverse any counter other than x_{l_j} on a segment of the form $(u_i,\langle l_j,0\rangle)^*$ $(u_i,\langle l_j,1\rangle)$ $(u_i,\langle l_j,2\rangle)^*$, the input is rejected. Otherwise, if M_I attempts to reverse x_{l_j} on a symbol that is not $(u_i,\langle l_j,1\rangle)$, the input is rejected. In the last case, M_I remembers $(u_i,\langle l_j,1\rangle)$ in its finite control, moves the head until it encounters the left end marker # (possibly by reversing the direction once at the right end marker &), reverses x_{l_j} and continues until it restores the position of the input head.

Now for any canonical machine M_I, we will show the existence of a system $S_I \in \mathfrak{D}[*]$ with parameters $\mathbf{z} = \{z_1,\overline{z_1},\ldots,z_m,\overline{z_m}\}$, for some m, depending on M_I. In the following, we ignore the subscript I. Without loss of generality, assume that each counter reverses mode from non-decreasing to non-increasing at most once, and that M accepts when all counters have the value zero. Let the input tape be of the form $\#u_1^{y_1}\ldots u_n^{y_n}\&$. Each time M sweeps across $u_i^{y_i}$, $1 \leq i \leq n$, the j-th counter increases by $f_{ij} = a_{ij}y_i + b_{ij}$, $1 \leq j \leq k$, if M is in non-decreasing mode with respect to x_j and decreases by $g_{ij} = c_{ij}y_i + d_{ij}$, otherwise. The machine accepts if, for each counter there exists two values z_j and $\overline{z_j}$ such that $\sum_{i=1}^{n} z_j f_{ij} + \overline{z_j} g_{ij} = 0$. Intuitively, z_j ($\overline{z_j}$) represents the number of times M sweeps across the entire input being in the non-decreasing (non-increasing) mode, with respect to x_j. Moreover, it is required that $z_i + \overline{z_i} = z_j + \overline{z_j}$, for all $1 \leq i < j \leq k$. Altogether, these conditions form a system of Diophantine equations of the form (3). □

Next, we prove the reductions between several classes of FCA and the theories $\langle \mathbb{Z}, \geq, +, 0, 1\rangle$, $\langle \mathbb{Z}, \geq, +, |, 0, 1\rangle^{\exists}$ and $\mathfrak{D}[*]$. We distinguish the FCA according to the number of parameters, and the form of the transition relations. The first class, denoted as $\mathrm{FCA}(0,*)$, consists of flat counter automata with no parameters. The direction $\mathrm{FCA}(0,*) \longrightarrow \langle \mathbb{Z}, \geq, +, 0, 1 \rangle$ is a direct consequence of the fact that, for any counter automaton $M \in \mathrm{FCA}(0,*)$, the set $\mathcal{V}(M)$ is definable in Presburger arithmetic, as will be shown in Section 4. The other direction is handled by the following lemma :

Lemma 3.3. $\langle \mathbb{Z}, \geq, +, 0, 1 \rangle \longrightarrow \mathrm{FCA}(0,*)$.

Proof:
If $\models \varphi(\mathbf{x}) \leftrightarrow \perp$, the machine M corresponding to φ has a final state that is unreachable, i.e. $\mathcal{V}(M) = \emptyset$. Otherwise, φ can be effectively written in the equivalent semilinear form, as a finite disjunction of formulae of the form $\mathbf{x} = \mathbf{a_0} + \sum_{i=1}^{n} \mathbf{a_i}\lambda_i$, where $\mathbf{a_i} \in \mathbb{Z}^{\|\mathbf{x}\|}$, $0 \leq i \leq n$ and $\lambda_1, \ldots, \lambda_n$ are fresh variables. The automaton corresponding to φ is composed of disjoint linear flat counter automata, one for each disjunct in the semilinear form of φ, and defined as follows. All LFCA have in common only the initial state. Then, for each term $\mathbf{a_0}$, we have a transition from the initial state to the first state of the LFCA, which is also a 2-branching state, and the transition relation is $\mathbf{x}' = \mathbf{x} + \mathbf{a_0}$. Finally, for any term of the form $\mathbf{a_i}\lambda_i$, the LFCA has a self-loop with a transition relation $\mathbf{x}' = \mathbf{x} + \mathbf{a_i}$. All loops are linked by transitions of the form $\mathbf{x} = \mathbf{x}'$. □

The reductions $\mathrm{FCA1A}(*,*) \longleftrightarrow \langle \mathbb{Z}, \geq, +, |, 0, 1 \rangle^{\exists}$ are handled by the following Lemma:

Lemma 3.4. $\langle \mathbb{Z}, \geq, +, |, 0, 1 \rangle^{\exists} \longleftrightarrow \mathrm{FCA1A}(*,*)$.

Proof:
"$\longrightarrow$" We show how to translate a relation of the form $f(\mathbf{x}) \mid g(\mathbf{x})$, where f and g are linear functions on $\mathbf{x}$. First, the values of the functions are computed using two separate counters, i.e. $y = f(\mathbf{x})$ and $z = g(\mathbf{x})$. Notice that this can be done using a flat control structure, with only one assignment per loop. Then the automaton guesses the value of y using a parameter p, by testing $y = p$. In order to ensure that $p|z$, the automaton uses another loop with the transition relation whose only assignment is $z' = z - p$, and an exit transition with condition $z = 0$, that leads to the accepting state.

"$\longleftarrow$" The input-output relation of a counter automaton $A \in \mathrm{FCA1A}(*,*)$ can be shown to be of the form $\exists n \,.\, x_i' \sim x_j + (p+c)n \,\wedge\, \bigwedge_{j \neq i} x_j' = x_j \wedge \phi(\mathbf{x}) \wedge \psi(\mathbf{x}')$ independently, for each loop of M. Each such relation can be expressed in the language of $\langle \mathbb{Z}, \geq, +, |, 0, 1 \rangle^{\exists}$, e.g. $\exists n \,.\, x_i' \leq x_j + (p+c)n \iff \exists y \,.\, y \geq 0 \wedge p + c | x_i' + y - x_j$. □

The reduction $\mathrm{FCA}(*,*) \longrightarrow \mathfrak{D}[*]$ is a direct consequence of the fact that, for any $M \in \mathrm{FCA}(*,1)$ the set $\mathcal{V}(M)$ is defined by a formula of $\mathfrak{D}[1]$, as it is shown in Section 4. The inverse direction is proved in the following lemma :

Lemma 3.5. $\mathfrak{D}[*] \longrightarrow \mathrm{FCA}(*,*)$.

Proof:
By the definition of $\mathfrak{D}[n]$, φ is a finite disjunction of parametric linear Diophantine systems, consisting of equations of the form (3). Notice that each such equation can be reduced to a system in which all polynomials p_i, q_i are of degree one, by introducing new variables. For instance, if $x_{1,2}$ are variables and

$z_{1,2}$ are parameters, the equation $x_1 = z_1^2 z_2 \cdot x_2$ is equivalent to the system:

$$\begin{cases} x_1 &= z_1 \cdot y_1 \\ y_1 &= z_1 \cdot y_2 \\ y_3 &= z_2 \cdot x_2 \end{cases}$$

where y_1, y_2, y_3 are fresh variables. We can assume w.l.o.g. that $\varphi(\mathbf{x})$ is in this form. In order to build an automaton $M \in \mathrm{FCA}(p,m)$ such that $\mathcal{V}(M) = \{\mathbf{v} \in \mathbb{Z}^{\|\mathbf{x}\|} \mid\ \models \varphi(\mathbf{v})\}$, it is sufficient to encode the relations of the form (1) $x_1 = z \cdot x_2$ and (2) $x_1 = x_2 + x_3$. Each such relation will be encoded by a loop, using counters $x_{1,2,3}$ and an extra counter y. For (1) we initialize $x_1' = 0 \wedge y' = x_2$. The transition relation of the loop is $x_1' = x_1' + z \wedge y' = y' - 1 \wedge y > 0$, and the exit condition is $y = 0$. For (2) we initialize $x_1' = x_2 \wedge y' = x_3$. The transition relation of the loop is $x_1' = x_1 + 1 \wedge y' = y - 1 \wedge y > 0$ and the exit condition is $y = 0$. Notice that the number of loops needed for M is greater or equal than the number of parameters in φ. □

4. Deciding the Reachability Problem for Flat Counter Automata

One of the goals of this paper is to investigate flat counter automata with respect to their emptiness problem. This problem naturally translates into a *reachability problem*: is there any final control state that is reachable? An important role is played by the set of values $\mathcal{V}(A)$, since the 1-letter language of a CA is empty if and only if its set of values is. As we show in the following, the set $\mathcal{V}(A)$ can be defined in various subfragments of the arithmetic of integer numbers. Hence, the reachability problem is decidable for a class of counter automata (machines) whenever $\mathcal{V}(A)$ can be defined in a decidable fragment of integer arithmetic.

Let $\mathbf{x} = \{x_1, x_2, \ldots, x_k\}$ be a set of working counters, and $\mathbf{z} = \{z_1, z_2, \ldots, z_l\}$ be a set of parameters. Given a counter automaton $A = \langle \mathbf{x} \cup \mathbf{z}, Q, \delta, q_0, F \rangle$ and a control state $q \in Q$, the *reachability problem*[3] asks whether there exists tuples of values $\mathbf{u} \in \mathbb{Z}^k$ and $\mathbf{v} \in \mathbb{Z}^l$, and a run of A from an initial configuration $\langle q_0, \mathbf{0v} \rangle$ to $\langle q, \mathbf{uv} \rangle$. One way to prove decidability of the aforementioned reachability problem is to define the relation between the input and output values of the counters of A using a decidable arithmetic theory. In other words, we aim at building an arithmetic formula $\nu_A^q(\mathbf{x}, \mathbf{x}', \mathbf{z})$ depending on A and q, such that, for every $\mathbf{u}, \mathbf{u}' \in \mathbb{Z}^k$, $\mathbf{v} \in \mathbb{Z}^l$, there is a run in A from $\langle q_0, \mathbf{uv} \rangle$ to $\langle q, \mathbf{u}'\mathbf{v} \rangle$ if and only if $\models \nu_A^q(\mathbf{u}, \mathbf{u}', \mathbf{v})$. The reachability problem for A and q reduces then to checking the satisfiability of the (open) formula $\nu_A^q(\mathbf{0}, \mathbf{x}, \mathbf{z})$.

In order to define ν_A^q, we first observe that each $A \in \mathrm{FCA}(p,n)$ can be defined as a union of disjoint linear flat counter automata, each being composed of a sequence of cycles, connected by non-trivial control paths (cf. Figure 2). Without loss of generality, we will assume that A is in normal form, i.e. each control path with no incoming edges and no branching has been reduced to one transition, by composing the transition relations along the way. It follows that $\nu_A^q(\mathbf{x}, \mathbf{x}', \mathbf{z})$ is of the following form (cf. Figure 2):

$$\exists \mathbf{y_{1\ldots n}} \exists \mathbf{y'_{1\ldots n}} \bigvee_{i=1}^{r} \eta_{i1}(\mathbf{x}, \mathbf{y_1}, \mathbf{z}) \wedge \bigwedge_{j=1}^{m_i} \left[\xi_{ij}^*(\mathbf{y_j}, \mathbf{y'_j}, \mathbf{z}) \wedge \eta_{ij}(\mathbf{y'_j}, \mathbf{y_{j+1}}, \mathbf{z}) \right] \wedge \mathbf{x}' = \mathbf{y_{m_i}}$$

[3] A more general way to formulate the reachability problem is: given a set C of configurations, is there a run of A from an initial configuration $\langle q_0, \mathbf{0v} \rangle$ to a configuration in C, for some valuation of the parameters $\mathbf{v} \in \mathbb{Z}^l$?

where $n = \max\{m_i \mid 1 \leq i \leq r\}$ is the maximum number of loops in a linear component, η_{ij} are the difference bound constraints corresponding to the transitions between cycles, and ξ^*_{ij} represent the transitive closures of the cycle relations, for $1 \leq i \leq r$ and $1 \leq j \leq m_i$.

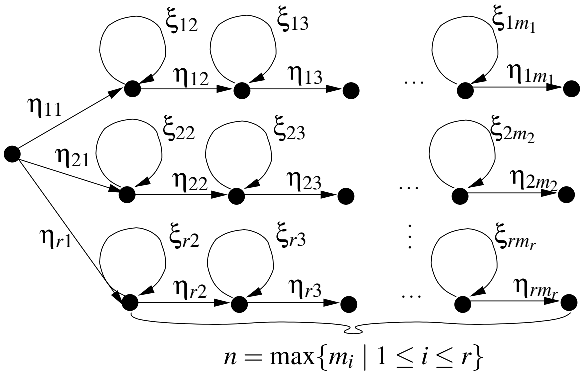


Figure 2. Flat Counter Automaton

Since η_{ij} are difference bound constraints, it follows that ν^q_A is a formula in the language of $\langle \mathbb{Z}, \geq, +, 0, 1\rangle$, if and only if the transitive closure formulae ξ^*_{ij} belong to the same language. Moreover, if all $m_i \leq 1$, ν^q_A is a formula of $\mathfrak{D}[1]$ if and only if all transitive closure formulae ξ^*_{ij} are. It is therefore sufficient to analyze the definability of ν^q_A when A has only one self-loop (i.e. one transition of the form $q \xrightarrow{\varphi(\mathbf{x},\mathbf{x}',\mathbf{z})} q$). In the following developments, we will silently assume that this is the case.

The main results of this section are summarized by the following Theorem:

Theorem 4.1. Let $\mathbf{x}$ be a set of working counters, $\mathbf{z}$ be a set of parameters, such that $\mathbf{x} \cap \mathbf{z} = \emptyset$, and $R(\mathbf{x}, \mathbf{x}', \mathbf{z})$ be a difference bound constraint. Then the transitive closure R^* of R is definable $\mathfrak{D}[1]$. Moreover, if $\mathbf{z} = \emptyset$ then R^* is definable in $\langle \mathbb{Z}, \geq, +, 0, 1\rangle$.

Notice that, the second part of Theorem 4.1 has been already proved in [5]. Here we give a different proof, based on the notion of weighted automata, that allows us to prove the first, more general, statement. Since $\langle \mathbb{Z}, \geq, +, 0, 1\rangle$ is a logic decidable for satisfiability, the reachability problem for flat counter automata with no parameters is decidable. In case where we allow parameters, decidability of the reachability problem follows from the result of Section 5 on decidability of the satisfiability problem for $\mathfrak{D}[1]$. This entails the decidability of the reachability problem for flat counter automata with at most one loop per linear component i.e., the class FCA$(*, 1)$.

However, for an arbitrary number of loops per linear component, one needs the full expressivity of Diophantine systems, in order to encode the reachability problem. In the light of Hilbert's Tenth Problem [11], the following lemma shows the undecidability of the reachability problem for parametric FCA with unrestricted number of loops.

Lemma 4.1. Given a Diophantine system $S(\mathbf{x})$, it is possible to build a parametric FCA $A = \langle \mathbf{y} \cup \mathbf{z}, Q, \delta, q_0, F\rangle$ with working counters $\mathbf{y}$ and parameters $\mathbf{z}$, such that $\mathbf{x} \subseteq \mathbf{z}$, and for some control state $q \in Q$, and for all $\mathbf{x} \in \mathbb{Z}$, we have $\models S(\mathbf{x})$ if and only if there exists a run of A $\langle q_0, \mathbf{0z}\rangle \to \ldots \to \langle q, \mathbf{yz}\rangle$

Proof:
The system $S(\mathbf{x})$ is a set of equations of the form $P(\mathbf{x}) = 0$, with $P \in \mathbb{Z}[\mathbf{x}]$. We transform each such equation in a system $T(\mathbf{y})$, by introducing new variables, as follows. Start with $\mathbf{y} = \mathbf{x}$ and $T = \emptyset$. Then iterate the following steps:

- choose one subterm t of the form $y_i \circ y_j$, $\circ \in \{+, \cdot\}$ and two variables $z, z' \notin \mathbf{y}$
- $S \leftarrow S[z/t]$, $T \leftarrow T \cup \{y_i \circ z = z',\ y_j = z\}$, and $\mathbf{y} \leftarrow \mathbf{y} \cup \{z, z'\}$

until a fixpoint is reached. It is easy to check that, for all $\mathbf{x} \in \mathbb{Z}$, $\models S(\mathbf{x})$ if and only if there exist integer values $\mathbf{y} \setminus \mathbf{x}$ such that $\models T(\mathbf{y})$. For a suitable indexing of the set $\mathbf{y} = \{y_1, y_2, \ldots\}$, T has only equations of the form (1) $y_i \cdot y_j = y_k$, (2) $y_i + y_j = y_k$ where $i < j < k$, and (3) $y_i = y_j$.

Now we build a parametric FCA $A = \langle \{x_1, x_2\}, \mathbf{y}, Q, \delta, q_0 \rangle$ with two working counters, and the set of parameters same as the set of the variables of $T(\mathbf{y})$. Note that we have $\mathbf{x} \subseteq \mathbf{y}$, from the construction of T. For every equation of the form (1) we have a control loop $q_1 \xrightarrow{\varphi_1} q_2$, $q_2 \xrightarrow{\varphi_2} q_2$ and $q_2 \xrightarrow{\varphi_3} q_3$, where:

- the entry relation is $\varphi_1 : x'_1 = y_i \ \wedge \ x'_2 = 0$
- the loop relation is $\varphi_2 : x'_1 = x_1 - 1 \ \wedge \ x'_2 = x_2 + y_j \ \wedge \ x_2 > 0$
- the exit relation is $\varphi_3 : x_1 = 0 \wedge x_2 = y_k$

For equations of type (2) we introduce a similar loop, the only exception being the loop relation, which is $\varphi_2 : x'_1 = x_1 - 1 \ \wedge \ x'_2 = x_2 + 1 \ \wedge \ x_2 > 0$. Equalities of type (3) are assigned a single transition with guard $y_i = y_j$. Notice that A is flat by construction. It is easy to check now that $T(\mathbf{y})$ has a solution if and only if A has a run ending in $\langle q, \mathbf{xy} \rangle$. □

4.1. Outline of the Proof

The rest of this section is concerned with the proof of Theorem 4.1. Before giving the actual proof, let us sketch the main lines of the construction. The first step is to represent the given difference bound relation $\varphi(\mathbf{x}, \mathbf{x}', \mathbf{z})$ as a constraint graph G_φ, whose nodes are the working counters from $\mathbf{x} \cup \mathbf{x}'$, and the edges represent the constraints between the counters. Namely, there is an edge labeled α between x and y if and only if the atomic proposition $x - y \leq \alpha$ occurs in φ[4]. Our goal is to characterize the relation between the initial values of $\mathbf{x}$ and the values after $n \geq 1$ iterations, by a formula $\psi(n, \mathbf{x}, \mathbf{x}')$, in which n occurs as a free variable. The transitive closure $\varphi^*(\mathbf{x}, \mathbf{x}')$ is then $\exists n \ . \ \psi(n, \mathbf{x}, \mathbf{x}')$.

For a given $n \geq 1$, the relation φ^n can be represented by the graph obtained by producing n copies of G_φ, and identifying the primed nodes of the i-th copy with the unprimed nodes of the $(i-1)$-th copy of G_φ, for $1 < i \leq n$ (see Figure 3). This graph, call it G^n_φ, represents all constraints between the intermediate values of the $\mathbf{x}$ counters, in n steps. An important property of this construction is that the strongest constraints between the initial value of a variable x and the final value (after n steps) of a value y are given by the minimal weight paths between the extremal nodes corresponding to x and y. Since the edges of G^n_φ are labeled with linear combinations of parameters $\mathbf{z}$, the choice of the minimal path depends on the initial choice of values for the parameters.

[4] This representation is also used in [5], with the difference that we allow parameters as labels in the constraint graph.

Let us explain the need for finding minimal weight paths in more detail. By the definition of relational composition, $\varphi^n(\mathbf{x},\mathbf{x}') = \exists \mathbf{y_1} \exists \mathbf{y_2} \ldots \exists \mathbf{y_{n-1}} \,.\, \varphi(\mathbf{x},\mathbf{y_1}) \wedge \varphi(\mathbf{y_2},\mathbf{y_3}) \wedge \ldots \wedge \varphi(\mathbf{y_{n-1}},\mathbf{x}')$, for any $n > 1$. Since φ is a difference bound constraint, so is φ^n, and its equivalent difference bound constraint form can be effectively computed by eliminating the existential quantifiers. In practice, this is done by first strengthening[5] φ^n, and then dropping all atomic propositions involving $\mathbf{y_1},\mathbf{y_2},\ldots,\mathbf{y_{n-1}}$. Notice that the effect of strengthening is equivalent to computing minimal weight paths between the extremal points of G^n_φ. Therefore we need means to define minimal weight paths in G^n_φ, as functions of n.

The main idea of the proof is that any path (zigzag) between two extremal nodes of G^n_φ can be seen as a word of length n, over the finite alphabet of subgraphs of G_φ (see, for instance, Figure 5 (a)). Noticeably, the set of paths between two given variables can be recognized by a (weighted) finite state automaton, that can be effectively constructed from G_φ. The problem of computing the minimal weights among all paths between two given nodes of G^n_φ is reduced to computing the minimal weight among all runs of length n in a given finite state automaton, with weights on transitions. We show that, if $\mathbf{z} = \emptyset$ the set of weights corresponding to runs of length n (between two given nodes of G^n_φ) can be defined in $\langle \mathbb{Z}, \geq, +, 0, 1 \rangle$, and in general ($\mathbf{z} \neq \emptyset$) in $\mathfrak{D}[1]$. As a result the formula $\psi(n,\mathbf{x},\mathbf{x}')$ can be defined in $\langle \mathbb{Z}, \geq, +, 0, 1 \rangle$ if $\mathbf{z} = \emptyset$, and in $\mathfrak{D}[1]$, otherwise. This entails our results concerning the definability of the transitive closure φ^* in $\langle \mathbb{Z}, \geq, +, 0, 1 \rangle$ and $\mathfrak{D}[1]$, respectively.

4.2. Constraint Graph Execution Model

In general, a difference bound constraint $\varphi(\mathbf{x},\mathbf{x}',\mathbf{z})$ can be represented as a directed weighted graph whose set of vertices is the set of variables $\mathbf{x} \cup \mathbf{x}'$, and there is an edge with weight α from x to y if and only if there is an explicit constraint $x - y \leq \alpha$ in φ, where $\alpha \in \mathrm{lin}\mathbb{Z}[\mathbf{z}]$. The n-th iteration of φ (denoted φ^n) is represented by a *constraint graph* G^n_φ, defined as the minimal graph whose set of vertices is $\bigcup_{i=0}^n \mathbf{x}^i$, where $\mathbf{x}^i = \{x^i_j \mid 1 \leq j \leq k\}$ and, for all $0 \leq i < n$, there is an edge labeled α from:

- x^i to y^i, if there is a constraint $x - y \leq \alpha$ in φ.
- x^{i+1} to y^{i+1}, if there is a constraint $x' - y' \leq \alpha$ in φ.
- x^i to y^{i+1}, if there is a constraint $x - y' \leq \alpha$ in φ.
- x^{i+1} to y^i, if there is a constraint $x' - y \leq \alpha$ in φ.

For example, Figure 3 shows the constraint graph for the relation $\varphi : x_1 - x'_2 \leq z_1 \wedge x'_2 - x_3 \leq z_2 \wedge x_3 - x'_1 \leq z_3 \wedge x_1 - x'_3 \leq z_4$. Intuitively, the nodes $\mathbf{x^i}$ in the execution graph represent the possible values of the counters after i steps of execution. Let $G^\infty_\varphi = \bigcup_{n>0} G^n_\varphi$.

We say that a path in G^∞_φ *stretches between n and m*, for some $n \leq m$, if the path contains at least one node from $\mathbf{x}^i$, for each $n \leq i \leq m$. If $\pi : x^i \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_m} y^j$, $0 \leq i,j \leq n$ is a path in G^n_φ, let $\omega(\pi)$ denote the sum of all labels along the path, i.e. $\omega(\pi) = \sum_{k=1}^m \alpha_k$. Notice that $\omega(\pi) \in \mathrm{lin}\mathbb{Z}[\mathbf{z}]$, for any constant $m \in \mathbb{N}$. Clearly, we have $x^i - y^j \leq \omega(\pi)$. We define $\min\{x^i \to y^j\} = \min\{\omega(\pi) \mid \pi : x^i \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_m} y^j\}$.

[5] The strengthening of a difference bound constraint φ consists in adding, between each two variables x and y, a constraint $x - y \leq \alpha$, where α is the minimal bound on $x - y$ that can be inferred from φ.

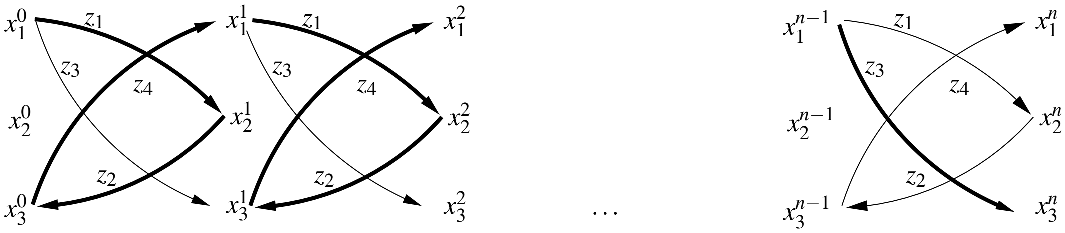


Figure 3. Iterated Constraint Graph for $x_1 - x_2' \leq z_1 \wedge x_2' - x_3 \leq z_2 \wedge x_3 - x_1' \leq z_3 \wedge x_1 - x_3' \leq z_4$

By convention, if there are no paths in G_φ^n, between x^i and y^j, we take $\min\{x^i \longrightarrow y^j\} = \infty$. On the other hand, if the set of paths between x^i and x^j does not have a minimal element, we take $\min\{x^i \longrightarrow y^j\} = -\infty$. Notice that this can only be the case if G_φ^n has a cycle whose sum is less than zero. Thus, the satisfiability of φ^n entails the absence of negative cycles from G_φ^n.

With this notation, we have $x^i - y^j \leq \min\{x^i \longrightarrow y^j\}$. The minimal weight among all paths between x^i and y^j gives the strongest difference constraint between the values of x and y at iteration steps i and j, respectively. As previously explained, this is a consequence of the fact that the class of difference bound constraints admits quantifier elimination, and that the quantifier-free form can be computed by first computing the minimal paths between each two nodes (strengthening) and then eliminating the quantified variables from the strengthened formula. Hence, for any $n \geq 1$, φ^n is equivalent to the following difference bound constraint:

$$\bigwedge_{x,y \in \mathbf{x}} x - y \leq \min\{x^0 \longrightarrow y^0\} \wedge x' - y' \leq \min\{x^n \longrightarrow y^n\} \wedge x - y' \leq \min\{x^0 \longrightarrow y^n\} \wedge x' - y \leq \min\{x^n \longrightarrow y^0\}$$

The next step is to define the functions (in n) $\min\{x^i \longrightarrow y^j\}$, $i, j \in \{0, n\}$ using the arithmetic of integers. These functions are definable in $\langle \mathbb{Z}, \geq, +, 0, 1 \rangle$, if φ has no parameters, and in $\mathfrak{D}[1]$, otherwise. The reduction method, based on weighted finite automata, is the same in both cases, and will be presented in the rest of this section.

4.3. Even and Odd Automata

In the following, we will work with a more convenient (yet equivalent) form of the transition relation $\varphi(\mathbf{x}, \mathbf{x}', \mathbf{z})$. Namely, all constraints of the form $x - y \leq \alpha$ are replaced by $x - t' \leq \alpha \wedge t' - y \leq 0$, and all constraints of the form $x' - y' \leq \alpha$ are replaced by $x' - t \leq \alpha \wedge t - y' \leq 0$, by introducing fresh variables $t \notin \mathbf{x} \cup \mathbf{z}$. In other words, we can assume without loss of generality that the constraint graph corresponding to φ (G_φ) is *bipartite*, i.e. it does only contain edges from $\mathbf{x}$ and $\mathbf{x}'$ and viceversa.

As previously mentioned, the presence of any cycle of negative weight within G_φ^n indicates that φ^n is not satisfied. On the other hand, a path that has a cycle of positive weight is not minimal, as one can obtain a path of smaller weight by eliminating the cycle. So, in principle, we need one tool for recognizing cycles of negative weight, and another one for recognizing acyclic paths within G_φ^n. Both tools will be finite state automata with weighted transitions, defined on two different alphabets.

Intuitively, a path π between, say, x^0 and x^n, with $x, y \in \mathbf{x}$ is represented by a word w of length n, as follows: the w_i symbol represents *simultaneously* all edges of π that involve only nodes from $\mathbf{x}^i \cup \mathbf{x}^{i+1}$, $0 \leq i < m$. Since we assumed that G_φ is bipartite, it is easy to see that, for a path from x^0 to y^n, coded by a word w, the number of times the w_i symbol is traversed by the path is odd, whereas for a path from x^0 to y^0, or from x^n to y^n, this number is even. Hence the names of *even* and *odd automata*. As an example, Figure 5 (a) shows the word encoding of the highlighted path between x_1^0 and x_3^n, from Figure 3.

Given a difference bound constraint $\varphi(\mathbf{x}, \mathbf{x}', \mathbf{z})$, the *even alphabet* of φ, denoted as Σ_φ^e, is the set of all graphs satisfying the following conditions, for each $G \in \Sigma_\varphi^e$:

1. the set of nodes of G is $\mathbf{x} \cup \mathbf{x}'$,
2. for any $x, y \in \mathbf{x} \cup \mathbf{x}'$, there is an edge labeled α from x to y, only if $x - y \leq \alpha$ occurs in φ.
3. the in-degree and out-degree of each node are at most one.
4. the number of edges from $\mathbf{x}$ to $\mathbf{x}'$ equals the number of edges from $\mathbf{x}'$ to $\mathbf{x}$.

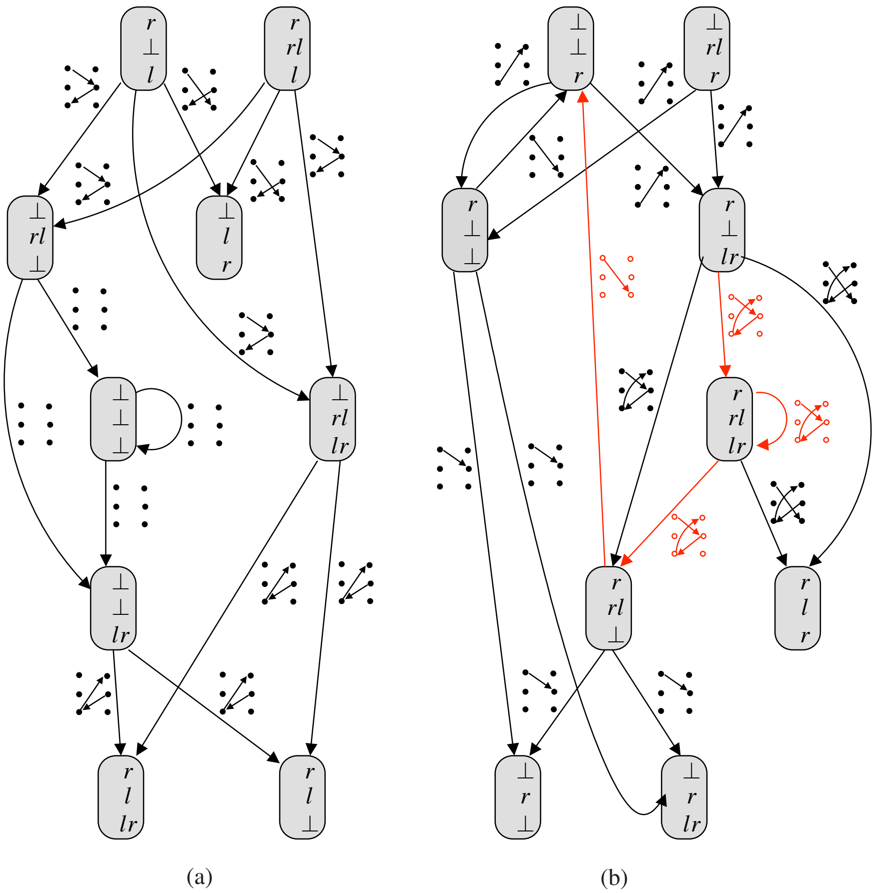
The *odd alphabet* of φ, denoted by Σ_φ^o, is defined in the same way, with the exception of the last condition:

4. the difference between the number of edges from $\mathbf{x}$ to $\mathbf{x}'$ and the number of edges from $\mathbf{x}'$ to $\mathbf{x}$ is either 1 or -1.

Let $\Sigma_\varphi = \Sigma_\varphi^e \cup \Sigma_\varphi^o$. Notice that, the number of edges in all symbols of Σ_φ^e is even, while the number of edges in all symbols of Σ_φ^o is odd. The label of G is the sum of the weights that occur on its edges. Clearly, the weight of a path through G_φ^n is the weight of the word it is represented by. We denote by $\omega(w)$ the weight of a word $w \in {\Sigma_\varphi}^*$. Notice that $\omega(w) \in \mathrm{lin}\mathbb{Z}[\mathbf{z}]$, for any given $w \in {\Sigma_\varphi}^*$, where $\mathbf{z}$ is the set of parameters of φ.

We are now ready for the definition of automata recongizing words that represent encodings of paths from G_φ^n. The even and odd automata share the same transition table, whereas the input alphabet is Σ_φ^e for the former, and Σ_φ^o for the latter. More precisely, we define the common transition table as $T_\varphi = \langle Q, \Delta \rangle$, where $Q = \{l, r, lr, rl, \perp\}^k$, and there is a transition $\langle q_1 \ldots q_k \rangle \xrightarrow{G} \langle q_1', \ldots, q_k' \rangle$ if and only if the following conditions hold, for all $1 \leq i \leq k$:

- $q_i = l$ iff G has one edge whose destination is x_i, and no other edge involving x_i.
- $q_i' = l$ iff G has one edge whose source is x_i', and no other edge involving x_i.
- $q_i = r$ iff G has one edge whose source is x_i, and no other edge involving x_i.
- $q_i' = r$ iff G has one edge whose destination is x_i', and no other edge involving x_i.
- $q_i = lr$ iff G has exactly two edges involving x_i, one having x_i as source, and another as destination.
- $q_i' = rl$ iff G has exactly two edges involving x_i', one having x_i' as source, and another as destination.
- $q_i' \in \{lr, \perp\}$ iff G has no edge involving x_i'.
- $q_i \in \{rl, \perp\}$ iff G has no edge involving x_i.



(a) (b)

Figure 4. The even (a) and odd (b) transition tables for $x_1 - x_2' \leq z_1 \wedge x_2' - x_3 \leq z_2 \wedge x_3 - x_1' \leq z_3 \wedge x_1 - x_3' \leq z_4$

As an example, the odd transition table for $\varphi = x_1 - x_2' \leq z_1 \wedge x_2' - x_3 \leq z_2 \wedge x_3 - x_1' \leq z_3 \wedge x_1 - x_3' \leq z_4$ is depicted in Figure 4 (b). If we consider the automaton obtained from this table, by setting the initial state to $\langle r, \bot, lr \rangle$ and the final state to $\langle \bot, \bot, r \rangle$, a run of this automaton is shown in Figure 5 (b). Intuitively, $q_{ij} = l$ means that the node x_j^i of G_φ^n is traversed from right to left by a path, and no other path comes across this node. Also, $q_{ij} = lr$ means that there is a path coming into x_j^i from $\mathbf{x}^{i+1}$ (left), and leaving also towards $\mathbf{x}^{i+1}$ (right), while no other path comes across this node.

Let $\pi : \mathbf{q}_0 \xrightarrow{G_1} \mathbf{q}_1 \xrightarrow{G_2} \ldots \mathbf{q}_{n-1} \xrightarrow{G_n} \mathbf{q}_n$ be a run pf A_φ. Each node in $G(\pi)$ is labeled by a symbol from the set $\{l, r, lr, rl, \bot\}$, and we write, e.g. $q_{ij} = l$, meaning that q_{ij} is labeled with l. We denote by $G(\pi)$ the graph $G_1 G_2 \ldots G_n$ labeling this run, and by $\omega(\pi)$ the weight of this graph i.e., the sum of the labels of all edges in $G(\pi)$. Notice that $\omega(\pi)$ is again a linear combination of parameters i.e., $\omega(G(\pi)) \in \mathrm{lin}\mathbb{Z}[\mathbf{z}]$.

The following Lemma is needed for technical reasons.

Lemma 4.2. Let $\pi : \mathbf{q}_0 \xrightarrow{G_1} \mathbf{q}_1 \xrightarrow{G_2} \ldots \mathbf{q}_{n-1} \xrightarrow{G_n} \mathbf{q}_n$ be a run of A_φ. Then each node q_{ij}, $0 \leq i \leq n$, $1 \leq j \leq k$, from $G(\pi)$, has at most one predecessor and at most one successor.

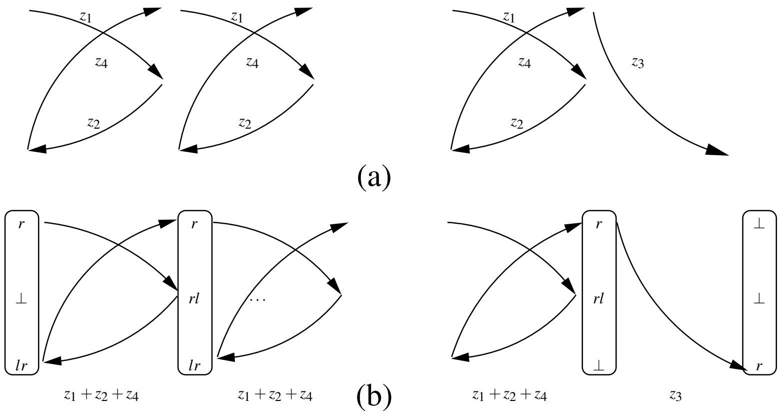


Figure 5. Sample run of the automaton in Fig. 4 (b)

Proof:
Suppose that $q_{i\,j}$ is a node in $G(\pi)$, and that $q_{i\,j}$ has two or more predecessors. We distinguish three cases: $i=0$, $1 \leq i < n$, and $i=n$. For the case $i=0$, $q_{1\,j}$ must have two or more incoming edges from $\mathbf{q}_2$, which is in contradiction with the third point in the definition of Σ^e_φ. The case $i=n$ is symmetrical.

In the case $1 \leq i < n$ we have $\mathbf{q}_{i-1} \xrightarrow{G_i} \mathbf{q}_i \xrightarrow{G_{i+1}} \mathbf{q}_{i+1}$. If $q_{i\,j}$ has two incoming edges, it can only have one edge from $\mathbf{q}_{i-1}$, and another from $\mathbf{q}_{i+1}$. Now if G_i has one edge incoming to $q_{i\,j}$, $1 \leq j \leq k$ it must be that $q_{i\,j} \in \{r, rl\}$. On the other hand, if G_{i+1} has one edge incoming to $q_{i\,j}$, it must be that $q_{i\,j} \in \{l, lr\}$, and we obtain a contradiction. The proof for $q_{i\,j}$ having at most one outgoing edge is symmetrical. □

The *even automaton* recognizes paths that start and end on the same side of G^n_φ i.e., either paths from x^0_i to x^0_j, or from x^n_i to x^n_j, for some $1 \leq i, j \leq n$, respectively. We call the first type of automata *forward* even automata, and the second one *backward* even automata. The distinction between the two is in the sets of initial and final states.

Formally, let $A^e_{ij} = \langle T_\varphi, Q_0, F \rangle$ be the forward even automaton, over the alphabet Σ^e_φ, where:

$$Q_0 = \begin{cases} \{q \mid q_i = r,\ q_j = l \text{ and } q_h \in \{lr, \perp\},\ 1 \leq h \leq k,\ h \notin \{i,j\}\} & \text{if } i \neq j \\ \{q \mid q_i = q_j = lr \text{ and } q_h \in \{lr, \perp\},\ 1 \leq h \leq k,\ h \neq i\} & \text{otherwise} \end{cases}$$

is the set of initial states, and $F = \{rl, \perp\}^k$ is the set of final states. In the case when $i=j$, we denote A^e_{ij} by A^e_i. The next Lemma relates the runs of A^e_{ij}, for $i \neq j$, to the paths between x^0_i and x^0_j, $1 \leq i, j \leq k$. The reader may refer to Figure 6 (a) for a depiction of the case covered by the following:

Lemma 4.3. For any $1 \leq i, j \leq k$, $i \neq j$, (1) A^e_{ij} has an accepting run of length at most n if and only if there exists a path in G^n_φ, from x^0_i to x^0_j, that stretches between 0 and some $m \leq n$. Moreover, (2) if G^n_φ does not have cycles of negative weight, the minimal weight among all paths from x^0_i to x^0_j, stretching from 0 to some $m \leq n$, equals the minimal weight among all accepting runs of A^e_{ij} of length at most n.

Proof:
(1) ”⇒” Assume that A^e_{ij} has an accepting run $\pi : \mathbf{q}_0 \xrightarrow{G_1} \mathbf{q}_1 \xrightarrow{G_2} \dots \mathbf{q}_{m-1} \xrightarrow{G_m} \mathbf{q}_m$, with $\mathbf{q}_0 \in Q_0$ and $\mathbf{q}_m \in F$,

$m \leq n$. We need to build a path in G^n_φ, from x^0_i to x^0_j, stretching between 0 and at most m. Note first that $G(\pi)$ is isomorphic with a subset of G^∞_φ. Our argument is the following: suppose that we have already built a path from q_{0_i} to q_{h_t}, for some $0 \leq h \leq m$, $1 \leq t \leq k$, then either $h = 0$ and $t = j$, case in which we are done, or else we can extend this path further. Since $G(\pi)$ is finite, we will eventually find either a path from q_{0_i} to q_{0_j}, or a cycle. But, in the last case, $G(\pi)$ must have a node with two incoming edges and at least one outgoing edge. For, otherwise q_{0_i} would belong to a cycle, having an incoming edge from $\mathbf{q}_1$. But this contradicts with the fact that $q_{0_i} = r$ and the only edge involving q_{0_i} is an outgoing edge. According to Lemma 4.2, no node in $G(\pi)$ can have two incoming edges, thus we have reached a contradiction.

We are now left with proving that every node, other than q_{0_j}, that is reachable from q_{0_i}, has at least one successor. Obviously, this is the case for q_{0_i}, since $q_{0_i} = r$. For other nodes $q_{0_j} \neq \bot$, $j \neq i$, we have $q_{0_j} = lr$, and we are done. Any other node q_{h_t}, $1 \leq h \leq m$, $1 \leq t \leq k$ that is reachable from q_{0_i} has at least one incoming edge. Hence it must be that $q_{h_t} \neq \bot$. If $q_{h_t} \in \{lr, rl\}$, we are done. Otherwise, if $q_{h_t} = l$, and since $\mathbf{q}_{h-1} \xrightarrow{G_h} \mathbf{q}_h$, then G_h has an edge $x'_t \xrightarrow{c} x_u$, for some $1 \leq u \leq k$, which becomes an outgoing edge of q_{h_t}. For the case $q_{h_t} = r$, we must distinguish between $h < m$ and $h = m$. In case $h < m$, we have $\mathbf{q}_h \xrightarrow{G_{h+1}} \mathbf{q}_{h+1}$ and we do a similar reasoning, as in the case $q_{h_t} = l$. Else, if $h = m$, we cannot have $q_{m_t} = r$, because that would contradict with the fact that $\mathbf{q}_m \in F = \{rl, \bot\}^k$.

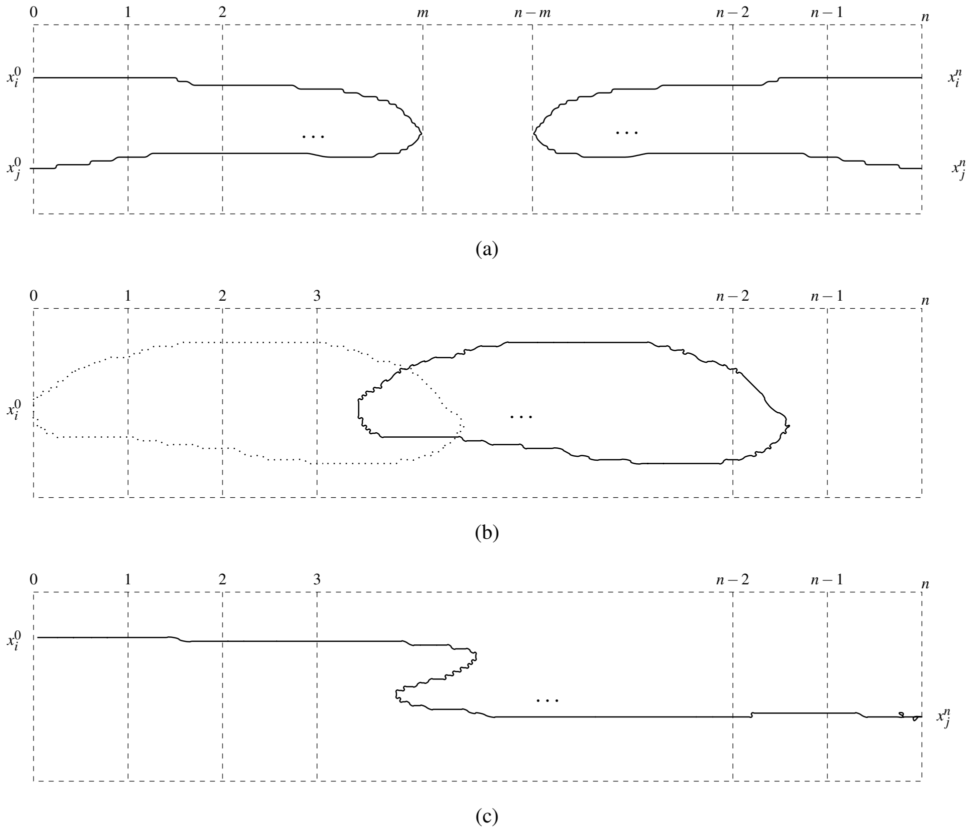
"$\Leftarrow$" We show the existence of the run by induction on m. Note that one only needs to consider states and transitions that occur within the path, and verify that the transitions are valid. This check is trivial.

(2) Let A^e_{ij} have an accepting run $\rho : \mathbf{q}_0 \xrightarrow{G_1} \mathbf{q}_1 \xrightarrow{G_2} \ldots \mathbf{q}_{m-1} \xrightarrow{G_m} \mathbf{q}_m$, with $\mathbf{q}_0 \in Q_0$ and $\mathbf{q}_m \in F$. By the first point, there exists path π from q_{0_i} to q_{0_j} in $G(\rho)$. Moreover, this path should be unique, since no node in the graph can have two different successors. We prove first that each node $q_{h_t} \neq \bot$, $1 \leq h \leq m$, $1 \leq t \leq k$, that is not on π, must belong to a cycle, which does not intersect with π. This is done along the same lines as the proof for existence of the path, at the previous point. First, each node $q_{h_t} \neq \bot$, $q_{h_t} \notin \pi$, must have a successor. Moreover, the successor must not be on π, or else one node from π would have two different predecessors. Hence, q_{h_t} belongs to an infinite path π', and $\pi \cap \pi' \neq \emptyset$. Now we need to show that π' is a cycle. But if this would not be the case, since π' is infinite, then at least one node on π' must have two different predecessors, which results in a contradiction with Lemma 4.2.

Since for every accepting run ρ of A^e_{ij} of length m, there exists a path π from q_{0_i} to q_{0_j}, stretching from 0 to at most m, and moreover, every edge not on π belongs to a cycle, we have that $\omega(\rho) = \omega(\pi) + \sum_{\gamma \text{ is a cycle in } G(\rho)} \omega(\gamma)$. Since there are no cycles γ such that $\omega(\gamma) < 0$, we have that $\omega(\rho) \geq \omega(\pi)$. Dually, for each path π from q_{0_i} to q_{0_j}, one can build an accepting run ρ such that $\omega(\pi) = \omega(\rho)$. Suppose now that, for instance we had:

$$\begin{gathered}\min\{\omega(\pi) \mid \pi \text{ is a path from } q_{0_i} \text{ to } q_{0_j} \text{ stretching from 0 to } m \leq n\} \\ \neq \min\{\omega(\rho) \mid \rho \text{ is an accepting run of } A^e_{ij} \text{ of length } m \leq n\}\end{gathered}$$

Suppose that the left hand side is strictly greater than the right hand side. Then there exists an accepting run of A^e_{ij} of length $m \leq n$ which has smaller weight than any path from q_{0_i} to q_{0_j} stretching from 0 to some $m' \leq m$. But this is in contradiction with the fact that for each run of length m, there exists a path stretching from 0 to $m' \leq n$ of smaller weight. Else, if the right hand side is strictly greater than the left hand side, then there exists a path from q_{0_i} to q_{0_j}, stretching from 0 to some $m \leq n$ of weight smaller



(a)

(b)

(c)

Figure 6. Runs of Even and Odd Automata

than that of any accepting run of A^e_{ij} of length $m' \leq m$. But this is in contradiction with the existence of a run of length exactly m, and of weight equal to the weight of the path. □

The backward even automaton is defined as $\overleftarrow{A^e_{ij}} = \langle \overleftarrow{T_\varphi}, F, Q_0 \rangle$, where Q_0 and F are the ones defined for the forward even automaton. The next Lemma relates the runs of $\overleftarrow{A^e_{ij}}$, for $i \neq j$, to the paths between x_i^n and x_j^n, $1 \leq i, j \leq k$. The reader may refer to Figure 6 (a) for a depiction of the case covered by the following:

Lemma 4.4. For any $1 \leq i, j \leq k$, $i \neq j$, (1) $\overleftarrow{A^e_{ij}}$ has an accepting run of length at most n if and only if there exists a path in G^n_φ, from x_i^n to x_j^n, that stretches between $n-m$ and n, for some $1 \leq m < n$. Moreover, (2) if G^n_φ does not have cycles of negative weight, the minimal weight among all paths from x_i^n to x_j^n, stretching from $n-m$ to n, equals the minimal weight among all accepting runs of $\overleftarrow{A^e_{ij}}$, of length at most n.

Proof:
Similar to the proof of Lemma 4.3.□

The next Lemma relates the cycles in G^∞_φ to the runs of A^e_i. In the proof, it is essential that G^∞_φ is composed of copies of the same graph G_φ. As a consequence, if there is a cycle in G_φ^∞, there is also a cycle starting and ending with x_0^i, for some $1 \leq i \leq k$. The reader may refer to Figure 6 (b) for a depiction of the case covered by the following:

Lemma 4.5. For any $1 \leq i \leq k$, A_i^e has an accepting run of negative weight if and only if there exists a cycle of negative weight in G_φ^∞.

Proof:
"⇒" Suppose that A_i^e has an accepting run $\rho : \mathbf{q}_0 \xrightarrow{G_1} \mathbf{q}_1 \xrightarrow{G_2} \ldots \mathbf{q}_{n-1} \xrightarrow{G_n} \mathbf{q}_n$, with $\mathbf{q}_0 \in Q_0$ and $\mathbf{q}_n \in F$, such that $\omega(\rho) < 0$. Then, by the first point of Lemma 4.3, there exists a cycle π in $G(\rho)$, such that q_{0i} is on that cycle. Also, by the proof of the second point of Lemma 4.3, we have that $\omega(\rho) = \omega(\pi) + \sum_{\gamma \text{ is a cycle in } G(\rho)} \omega(\gamma) < 0$. Then either $\omega(\pi) < 0$ or $G(\rho)$ has another cycle $\gamma \neq \pi$ whose weight is negative. Since $G(\rho)$ is isomorphic with a subset of G_φ^∞, we are done.

"⇐" Suppose that G_φ^∞ has a cycle of negative weight, stretching between n and m, for some $n < m$. Note that all subgraphs of G_φ^∞, that consist only of nodes $\mathbf{x}^i$ and $\mathbf{x}^{i+1}$ (together with the edges between them), are isomorphic. Hence there exists a cycle of negative weight, stretching between n and m if and only if there exists a cycle of negative weight stretching between 0 and $m-n$. Now one can build a run of A_i^e that has exactly the same weight as the latter cycle. □

We define now the *odd automata*, that recognize paths from one side of G_φ^n to another. The automata recognizing paths from x_0^i to x_n^j are called *forward* odd automata, whereas the ones recognizing paths from x_n^i to x_0^j are called *backward* odd automata. The reader may refer to Figure 6 (c) for a depiction of these cases.

Formally, $A_{ij}^o = \langle T_\varphi, Q_0, F \rangle$ be the forward odd automaton, over Σ_φ^o, where:

$$Q_0 = \{q \mid q_i = r \text{ and } q_h \in \{lr, \bot\},\ 1 \leq h \leq k,\ h \neq i\}$$
$$F = \{q \mid q_j = r \text{ and } q_h \in \{rl, \bot\},\ 1 \leq h \leq k,\ h \neq j\}$$

An example of an odd automaton is given in Figure 4. For $i = 1$ the initial states are $\langle r, \bot, lr \rangle$ and $\langle r, \bot, \bot \rangle$. For $j = 3$ the final state is $\langle \bot, \bot, r \rangle$. An accepting run of A_{13}^o is shown in Figure 5 (b). The next Lemma relates the runs of A_{ij}^o to the paths between x_i^0 and x_j^n, $1 \leq i, j \leq k$.

Lemma 4.6. For any $1 \leq i, j \leq k$, A_{ij}^o has an accepting run of length n if and only if there exists a path in G_φ^n, from x_i^0 to x_j^n. Moreover, if G_φ^n does not have cycles of negative weight, then the minimal weight among all paths from x_i^0 to x_j^n equals the minimal weight among all accepting runs of length n.

Proof:
This proof is done along the same lines as the proof of Lemma 4.3. "⇒ If ρ is an accepting run of A_{ij}^o of length n, then there exists a path π from q_{0_i} to q_{0_j} in $G(\rho)$, and moreover, since $G(\rho)$ does not have cycles of negative weight, $\omega(\rho) \geq \omega(\pi)$. "⇐" If G_φ^n has a path between x_i^0 and x_j^n, then there exists an accepting run ρ of A_{ij}^o, of length m, such that $G(\rho)$ has a path π from q_{0_i} to q_{n_j}, and no other edges,

except for the ones in π. Hence $\omega(\rho) = \omega(\pi)$. The equality of minimal weights follows in the same way as in the proof of Lemma 4.3. □

The definition of backward odd automata is symmetrical. Let $\overleftarrow{A^o_{ij}} = \langle \overleftarrow{T_\varphi}, F, Q_0 \rangle$ be the backward odd automaton over the alphabet Σ^o_φ, where F and Q_0 are the ones defined for the forward odd automaton. The next Lemma relates the runs of A^o_{ij} to the paths between x^n_i and x^0_j, $1 \leq i, j \leq k$.

Lemma 4.7. For any $1 \leq i, j \leq k$, $\overleftarrow{A^o_{ij}}$ has an accepting run of length n if and only if there exists a path in G^n_φ, from x^n_i to x^0_j. Moreover, if G^n_φ does not have cycles of negative weight, then the minimal weight among all paths from x^n_i to x^0_j equals the minimal weight among all accepting runs of length n.

4.4. Defining Optimal Paths in Weighted Finite Automata

Given a finite automaton with linear weights on transitions, we consider the problem of defining the set of accepting runs of a given length and of minimal weight. This solves the previous problem of defining the functions $\min\{x^i \rightarrow y^j\}$, which is needed for computing the transitive closure of a difference bound relation.

Let $T = \langle Q, \Delta \rangle$ be a transition table, and $A = \langle T, q_0, F \rangle$ be a finite automaton. A weight function $\omega : Q \times Q \rightarrow \mathrm{lin}\mathbb{Z}[\mathbf{z}]$ associates each transition $q \rightarrow r$ a linear expression $\omega(q,r) \in \mathrm{lin}\mathbb{Z}[\mathbf{z}]$. For a run π of A, $\omega(\pi)$ denotes the sum of all weights on the transitions. We aim at defining the set $\{(|\pi|, \omega(\pi)) \mid \pi \text{ is a run of } A\}$ using integer arithmetic.

Theorem 4.2. Let $\mathbf{z} = \{z_1, \ldots, z_l\}$ be a set of parameters. Given a finite automaton $A = \langle T, q_0, F \rangle$ where $T = \langle Q, \Delta \rangle$, and a weight function $\omega : Q \times Q \longrightarrow \mathrm{lin}\mathbb{Z}[\mathbf{z}]$ associating each transition a linear expression, one can effectively construct a formula $\psi_A(x, y, \mathbf{z}) \in \mathfrak{D}[1]$ such that, for any $n \in \mathbb{N}$, $w \in \mathbb{Z}$, $\mathbf{m} \in \mathbb{Z}^l$ we have $\models \psi_A(n, w, \mathbf{m})$ if and only if w is the minimal weight among all accepting runs of length n of A, under the valuation $\mathbf{m}$ of the parameters. Moreover, if $\mathbf{z} = \emptyset$, ψ_A is equivalent to a finite disjunction of linear inequality systems.

On one hand, this gives an alternative proof for the result of Comon and Jurski [5], namely that the transitive closure of a parameter-free difference bound relation is Presburger-definable. In practice, our proof gives also a direct method of expressing the n-th step iteration as a difference bound constraint, in which n occurs free. The advantage is that, in this way, we can express the reachability problem for a flat counter automaton as a finite disjunction of linear inequality systems, and apply state-of-the-art satisfiability solvers to it.

On the other hand, the reachability problem for single loop automata with parametric transition relations is definable in $\mathfrak{D}[1]$. As we show in Section 5, the problem concerning the existence of solutions for such systems is decidable, which entails the decidability of the reachability problem for the class of FCA$(l, 1)$.

Let us proceed now with the proof of Theorem 4.2. We associate with any transition $q \rightarrow r \in \Delta$ a variable x_{qr} and take $\mathbf{x}$ to be the set $\{x_{qr} \mid q \rightarrow r \in \Delta\}$. Intuitively, x_{qr} is the number of times the transition $q \rightarrow r$ occurs within a run. Hence we take as an implicit condition the fact that all such x_{qr} range over positive integers, i.e. $\bigwedge_{q \longrightarrow r \in \Delta} x_{qr} \geq 0$. The formula characterizing an accepting run of A of length l and weight w is :

$$\phi_A(l,w) \stackrel{\Delta}{=} \exists \mathbf{x} \, . \bigvee_{q_f \in F} \varphi_{q_f}(\mathbf{x}) \wedge \sum_{q \longrightarrow r \in \Delta} x_{qr} = l \wedge \sum_{q \longrightarrow r \in \Delta} x_{qr} \omega(q,r) = w \tag{4}$$

where $\varphi_{q_f}(\mathbf{x})$ expresses the necessary and sufficient conditions in order for $\mathbf{x}$ to define a path of A leading from q_0 to q_f. The definition of φ_{q_f} in Presburger arithmetic follows a method described in [5], which is based on the fact that the set of states Q of A is finite. For self-containment reasons, we give the definition of φ_{q_f} below.

$$\varphi_{q_f}(\mathbf{x}) \stackrel{\Delta}{=} \begin{cases} \varphi^1_{q_0}(\mathbf{x}) \wedge \varphi^2_{q_f}(\mathbf{x}) \wedge \varphi^3_{q_0 q_f}(\mathbf{x}) \wedge \psi(\mathbf{x}) & \text{if } q_f \neq q_0 \\ \varphi^4(\mathbf{x}) \wedge \psi(\mathbf{x}) & \text{otherwise} \end{cases}$$

where

$$\varphi^1_q(\mathbf{x}) \stackrel{\Delta}{=} 1 + \sum_{p \longrightarrow q \in \Delta} x_{pq} = \sum_{q \longrightarrow r \in \Delta} x_{qr} \tag{5}$$

$$\varphi^2_q(\mathbf{x}) \stackrel{\Delta}{=} \sum_{p \longrightarrow q \in \Delta} x_{pq} = \sum_{q \longrightarrow r \in \Delta} x_{qr} + 1 \tag{6}$$

$$\varphi^3_{qq'}(\mathbf{x}) \stackrel{\Delta}{=} \bigwedge_{s \in Q \setminus \{q,q'\}} \sum_{p \longrightarrow s \in \Delta} x_{ps} = \sum_{s \longrightarrow r \in \Delta} x_{sr} \tag{7}$$

$$\varphi^4(\mathbf{x}) \stackrel{\Delta}{=} \bigwedge_{s \in Q} \sum_{p \longrightarrow s \in \Delta} x_{ps} = \sum_{s \longrightarrow r \in \Delta} x_{sr} \tag{8}$$

and $\psi(\mathbf{x})$ is detailed next. Intuitively, the first three formulae above are the flow equations for the initial state (5), the final state of the run (6), and any state, other than the initial and the final, that may occur on the run (7). The case of (possibly empty) circular runs (i.e., $q_0 = q_f$) needs only one flow equation (8). Notice that the above flow equations can be also satisfied by two strongly connected components of A with no transition relating them. In order to define the set of paths in A, we need one extra condition. The final condition $\psi(\mathbf{x})$ is that a path must be connected.

Let $\mathcal{P}_A(q)$ be the set of all paths leading from q_0 to q, *with no repeated transitions.* Since Q is finite, this set is also finite. With these considerations, the connectivity condition can be given as follows, where $\alpha \cdot \beta$ denotes the concatenation of paths α and β :

$$\psi(\mathbf{x}) \stackrel{\Delta}{=} \bigwedge_{q \in Q} \bigvee_{q \longrightarrow r \in \Delta} x_{qr} > 0 \longrightarrow \bigvee_{\rho \cdot p \longrightarrow q \in \mathcal{P}_{\mathcal{A}}(q)} x_{pq} > 0$$

If $\mathbf{m} \in \mathbb{N}^k$ is an interpretation of $\mathbf{x}$, we refer as a $\mathbf{m}$-path to a path $s_0, s_1, \ldots, s_n$, where $s_i \in Q$, such that each transition $s_i \rightarrow s_{i+1}$ is taken *exactly* $m_{s_i s_{i+1}}$ times.

The following lemma proves that the formula defined in the previous correctly characterizes all accepting runs of A:

Lemma 4.8. For all $n \in \mathbb{N}$, $w \in \mathbb{Z}$, A has an accepting run of length n and weight w if and only if $\models \phi_A(n,w)$.

Proof:
We need to prove that, for all $n \in \mathbb{N}$ and $w \in \mathbb{Z}$, $\models \phi_A(n,w)$ if and only if A has a run ρ of length n and weight w, from q_0 to some $q_f \in F$.

"$\Rightarrow$" If ρ is an accepting run of length n and weight w, let $\mathbf{m}$ be the vector of occurrences of each transition within ρ and check $\models \varphi(\mathbf{m})$, which is trivial.

"$\Leftarrow$" Let $\mathbf{m}$ be the witness for $\varphi(\mathbf{x})$ i.e., $\models \varphi(\mathbf{m})$. We shall build an accepting $\mathbf{m}$-run of A, by induction on $M = \sum_{q,r \in Q} m_{qr}$. The base case is $M = 0$, which implies $n = 0$, hence the only run to be considered is the empty run. Let us prove that this is also an accepting run i.e. $q_0 \in F$. Assuming the contrary, we have $q_0 \neq q_f$ for all $q_f \in F$. But $\models \varphi_{q_f}(\mathbf{m})$, for some $q_f \in F$ must be the case, hence, also $\models \varphi^1_{q_0}(\mathbf{m})$, which is a contradiction, since we have assumed that $M = 0$.

For the inductive step $M > 1$, assume that for all $M' < M$, where $M' = \sum_{q \to r \in \Delta} m'_{qr}$, if $\models \varphi_{q_f}(\mathbf{m}')$ for some $q_f \in F$, then A has an $\mathbf{m}'$-run from q_0 to some $q_f \in F$. If $M > 0$ then there exists $q \to r \in \Delta$ such that $m_{qr} > 0$. Since $\mathbf{m} \models \psi$, there exists an $\mathbf{m}$-path from q_0 to q. Assume $r \notin F$, for else we were done. We show the existence of an $\mathbf{m}$-path from r to some $q_f \in F$. Since $r \notin F$, either $\mathbf{m} \models \varphi^3_{q_0 q_f}$, for some $q_f \in F$, or $\mathbf{m} \models \varphi^4$. In both cases, $\sum_{r' \in Q} m_{rr'} > 0$, hence there exists $r' \in Q$ such that $m_{rr'} > 0$. By repeating this argument, we discover an $\mathbf{m}$-path starting in r and ending either in a state p from the path, or in a final state $q_f \in F$. We consider the first case, the second leading immediately to the conclusion. In this case there exists a cyclic $\mathbf{m}$-path γ from p to itself. Let $\mathbf{m}'$ be the vector defined as $m'_{qq'} = m_{qq'} - 1$, if $q \to q'$ is on γ, and $m'_{qq'} = m_{qq'}$ otherwise. Obviously, $\sum_{q \to r \in \Delta} m'_{qr} < \sum_{q \to r \in \Delta} m_{qr}$, and the induction hypothesis applies, i.e. A has an accepting $\mathbf{m}'$-run. But in this case A has also an accepting $\mathbf{m}$-run, obtained by appending the γ cycle back to the $\mathbf{m}'$-run. □

Notice that, if A does not have parameters, ϕ_A is a formula in the language of $\langle \mathbb{Z}, \geq, +, 0, 1 \rangle$, hence we can already define the minimal weight w among all runs of length n by the following formula : $\phi_A(n,w) \ \wedge \ \forall z \ [z \leq w \to \neg \phi_A(n,z)]$. However, this is not the case when A has parameters, due to the multiplicative terms of the form $x_{qr}\omega(q,r)$ that occur within ϕ_A. Nevertheless, it is possible to build from ϕ_A, a formula of $\mathfrak{D}[1]$ defining optimal runs.

The main idea is to find the elementary cycles of optimal weight/length ratio $\frac{w}{n}$ within the weighted even/odd automata. In the parameter-free case (the weighted automata are labeled by integer constants), finding optimal cycles can be implemented using efficient algorithms [6], known to perform in almost linear average time. In the case of parameters, one has to consider a case split of size linear in the number of cycles. In the following, we present the technical details of the construction leading to a quantifier-free formula defining the minimal weight path function, for a given weighted automaton. In the parameter-free case, the function giving the value of w is a finite union of linear functions in n.

We apply the following transformation to each disjunct from the definition (4), i.e. for each $q_f \in F$. Let $\mathbf{x} = \{x_1, \ldots, x_m\}$ be a renaming of the existentially quantified variables, and if x_i, $1 \leq i \leq m$ is the renaming of x_{qr}, then let ω_i, denote the term $\omega(q,r)$. Since the subformula $\varphi_{q_f}(\mathbf{x}) \wedge \sum_{q \to r \in \Delta} x_{qr} = y$ is an open Presburger formula, it is either false or it defines a non-empty semilinear set, equivalent to a finite disjunction of formulae of the following form [8] :

$$\begin{pmatrix} x_1 \\ \vdots \\ x_m \\ y \end{pmatrix} = \begin{pmatrix} a_{01} \\ \vdots \\ a_{0m} \\ b_0 \end{pmatrix} + \begin{pmatrix} a_{11} \\ \vdots \\ a_{1m} \\ b_1 \end{pmatrix} \lambda_1 + \ldots + \begin{pmatrix} a_{k1} \\ \vdots \\ a_{km} \\ b_k \end{pmatrix} \lambda_k$$

for some new existentially quantified variables $\lambda_1, \ldots, \lambda_k$, with $a_{ij}, b_i \in \mathbb{Z}$, $1 \leq i \leq k$, $1 \leq j \leq m$. Since $z = \sum_{q \longrightarrow r \in \Delta} x_{qr} \omega(q,r)$, we obtain, for each disjunct of the original formula:

$$\begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} b_0 \\ \sum_{j=1}^m a_{0j}\omega_j \end{pmatrix} + \begin{pmatrix} b_1 \\ \sum_{j=1}^m a_{1j}\omega_j \end{pmatrix} \lambda_1 + \ldots + \begin{pmatrix} b_k \\ \sum_{j=1}^m a_{kj}\omega_j \end{pmatrix} \lambda_k$$

Since $y > 0$, it must be that $b_i > 0$, for all $0 \leq i \leq n$. Otherwise, if some $b_i < 0$ we can obtain a negative value for y by increasing λ_i sufficiently. On the other hand, if some $b_i = 0$, there would be an infinite number of weights z corresponding to the same path length y, resulting in a contradiction.

Let κ_i be the following formula :

$$\bigwedge_{p=1}^{k} \frac{\sum_{j=1}^m a_{ij}\omega_j}{b_i} \leq \frac{\sum_{j=1}^m a_{pj}\omega_j}{b_p}$$

with free variables from the set $\{z_1, \ldots, z_l\}$ of parameters of φ. Intuitively, κ_i is true if the i-th cycle in the weighted automaton is optimal. Note that, if φ is parameter-free, κ_i reduces to either true or false. Also, it is easy to see that $\models \bigvee_{i=1}^k \kappa_i$, i.e. the union of all κ_i covers the space $\mathbb{Z}^l$.

We perform a case split, in which the i-th case corresponds to a choice of parameter values that satisfy κ_i. Notice that $\models \bigvee_{1 \leq i \leq k} \kappa_i$. If $\models \kappa_i$, the minimal value z can take, in the above formula, for a given y, is encoded by the following:

$$\begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} b_0 \\ \sum_{j=1}^m a_{0j}\omega_j \end{pmatrix} + \sum_{1 \leq l \leq k}^{l \neq i} \begin{pmatrix} b_l \\ \sum_{j=1}^m a_{lj}\omega_j \end{pmatrix} r_l + \begin{pmatrix} b_i \\ \sum_{j=1}^m a_{ij}\omega_j \end{pmatrix} (\lambda_i + \sum_{1 \leq l \leq k}^{l \neq i} q_l b_l)$$

where q_l and r_l are the quotient and the remainder of λ_l divided by b_i, for all $l \in \{1, \ldots, i-1, i+1, \ldots, k\}$. This is because b_i can be subtracted from any λ_l, $l \neq i$, at most q_l times, by adding at the same time $q_l b_l$ to λ_i, and without modifying the value of y. Since $0 \leq r_l < b_i$, we perform another case split and replace r_l by constants in the formula above. Moreover, we introduce a fresh variable m, define $m = \lambda_i + \sum_{1 \leq l \leq k}^{l \neq i} q_l b_l$, and make the substitution accordingly, in the formula above.

Notice however that the above expresses the minimum for only one disjunct of the semilinear transform of $\varphi_{q_f}(\mathbf{x}) \wedge \sum_{q \longrightarrow r \in \Delta} x_{qr} = y$ considered for only one disjunct from the definition (4). We have to extend the construction to the case of more than one disjuncts. Let us consider the case of two disjuncts, call them D_1 and D_2, the generalization to more than two being rather straightforward. For each D_i, $i = 1, 2$ we have a formula of the above form, defining the minimal z :

$$D_i : \begin{pmatrix} y \\ z_i \end{pmatrix} = \begin{pmatrix} b_i \\ \beta_i \end{pmatrix} + \begin{pmatrix} c_i \\ \gamma_i \end{pmatrix} m, \ \ i = 1, 2$$

Our goal is to define the following set:

$$\begin{aligned} D &= \{\min(z_1,z_2) \mid \exists y\ (y,z_1) \in D_1 \ \wedge\ (y,z_2) \in D_2\} \\ &= \{z \mid \exists y\ (y,z) \in D_1 \ \wedge\ \forall z'\ (y,z') \notin D_2\} \cup \{\ z \mid \exists y\ (y,z) \in D_2 \ \wedge\ \forall z'\ (y,z') \notin D_1\} \\ &\quad \cup \{z \mid \exists y\ (y,z) \in D_1 \cap D_2 \ \wedge\ z = \min\{z' \mid (y,z') \in D_1 \cap D_2\}\} \end{aligned}$$

The first two sets can be defined directly by adjoining Presburger conditions on m. Let us take for example $\{z \mid (y,z) \in D_1 \ \wedge\ \forall z\ (y,z) \notin D_2\}$, which can be defined by :

$$\begin{aligned} \exists m\ .\ z = \beta_1 + \gamma_1 m \wedge \forall m'\ b_1 + c_1 m \neq b_2 + c_2 m' &\iff \\ \exists m\ .\ z = \beta_1 + \gamma_1 m \wedge\ \nexists m'\ .\ b_1 + c_1 m = b_2 + c_2 m' &\iff \\ \exists m\ .\ z = \beta_1 + \gamma_1 m \wedge c_2 \not| c_1 m + b_1 - b_2 &\iff \\ \exists m \exists q\ .\ \bigvee_{r=1}^{c_2-1} z = \beta_1 + \gamma_1 m \wedge c_1 m + b_1 - b_2 = c_2 q + r & \end{aligned}$$

where q is a fresh variable.

To represent the latter set from the definition of D, we have two symmetrical cases, as follows:

$$b_1 + c_1 m_1 = b_2 + c_2 m_2 \wedge \beta_1 + \gamma_1 m_1 \sim_i \beta_2 + \gamma_2 m_2 \wedge z = \beta_i + \gamma_i m_i$$

where $\sim_1$ is $\leq$, and $\sim_2$ is $\geq$. Now we can set $m = m_1$ (m_2), by writing m_2 (m_1) as a linear function of m_1 (m_2).

In conclusion, for the parameter-free case, the function relating the length n of a path in the weighted automaton to its weight w is a finite union of linear functions, while in the case with parameters, it is a finite union of 1-parametric Diophantine systems.

5. Solving Parametric Linear Diophantine Systems

In this section we give a proof for the decidability of the class of formulae $\mathfrak{D}[1]$. The problem considered here has been independently solved by O. Ibarra and Z. Dang in [13], using a property of reversal bounded counter machines. Another proof has been suggested to us by Y. Matiyasevich [17], using a more involved case analysis. Our proof is based on a result of L. Pottier [19], quoted by Theorem 5.1.

Let us fix a linear Diophantine system with parameter m, i.e. a system of the form $\{\sum_{j=1}^{n} p_{ij}(m)x_j + q_i(m) = 0\}_{i=1}^{r}$, with $p_{ij}, q_i \in \mathbb{Z}[m]$. We are interested in the existence of a solution $m, x_1, \ldots, x_n$ in natural numbers, although this is not a restriction.[6] We denote by $A(m)$ the matrix $[p_{ij}(m)]$.

Let us consider first that the system is homogeneous, i.e. $q_i(m)$ is the zero polynomial, for all $1 \leq i \leq n$. The general case will be dealt with in the following, by adding a new variable x_{n+1}, replacing each occurrence of $q_i(m)$ by $q_i(m)x_{n+1}$, and looking only after solutions in which $x_{n+1} = 1$. Let $P(m)$ be the greatest common divisor of all $p_{ij}(m)$ with respect to (symbolic) polynomial division, i.e. obtained by applying Euclid's algorithm in $\mathbb{Z}[m]$. Since $P(m)$ is a polynomial in one variable, its set of roots is finite and effectively computable. If $P(m_0) = 0$ for some $m_0 \in \mathbb{Z}$, then $\langle m_0, x_1, \ldots, x_n \rangle$ is a solution of the

[6] The satisfiability problem for integers can be reduced to 2^{n+1} instances of the same problem on natural numbers, by performing a case split on the signs of $m, x_1, \ldots, x_n$.

system $A(m)\mathbf{x} = 0$, for any choice of $x_1, \ldots, x_n \in \mathbb{Z}$. Thus, we assume in the following that $P(m) \neq 0$, for all $m \in \mathbb{N}$, in other words that, for no value of m, $p_{ij}(m)$ will all become zero at the same time.

Next, we are interested in the minimal solutions of the system. For a given $m \in \mathbb{N}$, a solution $(x_1, \ldots, x_n)$ is said to be *minimal* if it is a least solution with respect to the pointwise ordering on $\mathbb{N}^n$: $(u_1, \ldots, u_n) \preceq (v_1, \ldots, v_n) \iff u_i \leq v_i$, $1 \leq i \leq n$. The following Theorem has been proved in [19]:

Theorem 5.1. For a fixed $m_0 \in \mathbb{N}$, let $x_1, \ldots, x_n$ be any minimal solution of $A(m_0)\mathbf{x} = \mathbf{0}$. Then, for all $1 \leq i \leq n$, we have: $x_i \leq (n - r_0)\left(\frac{\sum_{i,j} a_{ij}(m_0)}{r_0}\right)^{r_0}$, where r_0 is the rank of $A(m_0)$.

Let $C > 0$ be the maximal absolute value of all coefficients of $a_{ij}(m)$, $1 \leq i \leq r$, $1 \leq j \leq n$, and $K \geq 0$ be the maximum degree of these polynomials. The following is a direct consequence of Theorem 5.1:

Corollary 5.1. For a fixed $m_0 \geq \max(C, n, r)$, let $x_1, \ldots, x_n$ be any minimal solution of $A(m_0)\mathbf{x} = \mathbf{0}$. Then, for all $1 \leq i \leq n$, we have $x_i \leq m_0^{(K+3)r+1}$.

Proof:
We have $a_{ij}(m_0) \leq Cm_0^K + Cm_0^{K-1} + \ldots + C = C\frac{m_0^{K+1}-1}{m_0-1} \leq m_0^{K+1}$, and therefore $\sum_{i,j} a_{ij} \leq nr \cdot m_0^{K+1} \leq m_0^{K+3}$. Since $0 < r_0 \leq r$, we have $(n - r_0)\left(\frac{\sum_{i,j} a_{ij}}{r_0}\right)^{r_0} \leq n\left(\frac{\sum_{i,j} a_{ij}}{r_0}\right)^{r} \leq n\left(\frac{m_0^{K+3}}{r_0}\right)^{r} \leq n(m_0^{K+3})^r \leq m_0^{(k+3)r+1}$. By Theorem 5.1, we obtain the result. □

Hence, one can enumerate all $0 \leq m < \max(C, n, r)$, and stop as soon as a solution of the linear Diophantine system $A(m)\mathbf{x} = \mathbf{0}$ has been found. Otherwise, for any $m \geq \max(C, n, r)$ the solution $x_1, \ldots, x_n$ can be represented in base m using at most $M = (K+3)r + 1$ digits. Let $(x_i)_m = \sum_{j=0}^{M} \chi_{ij} m^j$, with $0 \leq \chi_{ij} < m$ be the polynomial representing x_i in base m. The entire system $A(m)\mathbf{x} = \mathbf{0}$ can be now represented in base m, as it will be explained in the following.

First, we write the system as a set of equations of the form $P(m, x_1, \ldots, x_n) = Q(m, x_1, \ldots, x_n)$, with all coefficients of P and Q being positive. Since m is assumed to be greater than C, the maximal value of all coefficients c of the system, we have $(c)_m = c$. The operations of addition, multiplication by a constant $0 < c < m$, and multiplication by m, respectively, can be defined now using Presburger arithmetic. Let $(d)_m = \sum_{i=0}^{M} \delta_i m^i$, $(e)_m = \sum_{i=0}^{M} \varepsilon_i m^i$ and $(f)_m = \sum_{i=0}^{M} \phi_i m^i$, with $0 \leq \delta_i, \varepsilon_i, \phi_i < m$. We have:

$$\begin{aligned}
(f)_m = (d)_m + (e)_m \quad &\iff \quad \bigvee_{\mathbf{r} \in \{0\} \times \{0,1\}^{k-1} \times \{0\}} \bigwedge_{i=0}^{M} \delta_i + \varepsilon_i + r_i = \phi_i + m r_{i+1} \\
(e)_m = c(d)_m \quad &\iff \quad \bigvee_{\mathbf{r} \in \{0\} \times \{0,\ldots,c-1\}^{k-1} \times \{0\}} \bigwedge_{i=0}^{M} c\delta_i + r_i = \varepsilon_i + m r_{i+1} \\
(e)_m = m(d)_m \quad &\iff \quad \delta_M = \phi_0 = 0 \wedge \bigwedge_{i=0}^{M-1} \delta_i = \phi_{i+1}
\end{aligned}$$

The result of applying this transformation to the system $A(m)\mathbf{x} = \mathbf{0}$ is a formula $\Psi_A(m, \chi)$ in Presburger arithmetic, defining all minimal solutions of the original system $(x_i)_m = \sum_{j=0}^{M} \chi_{ij} m^j$, for $m \geq \max(C, n, r)$, with $\chi = \{\chi_{ij} \mid 1 \leq i \leq n,\ 1 \leq j \leq r\}$. The original system has a solution $(m, x_1, \ldots, x_n)$ if and only if, for some $m \in \mathbb{N}$, it has a minimal solution $(x_1^m, \ldots, x_n^m)$. Hence $\Psi_A(m, \chi)$ is satisfiable. Dually, if $\Psi_A(m, \chi)$ is satisfiable, we can construct a solution (not necessarily minimal) of $A(m)\mathbf{x} = \mathbf{0}$.

Example Consider the equation: $1 \cdot x_1 + m \cdot x_2 = m^2 + 2$. By Corollary 5.1 we should have considered the case $m \geq 6$, however the following is true even for $m \geq 3$: $(1)_m \cdot x_1 + (10)_m \cdot x_2 = (102)_m$. We look for solutions of the form $x_1 = (a_2a_1a_0)_m$ and $x_2 = (b_1b_0)_m$, and therefore, our equation is equivalent to: $(a_2a_1a_0)_m + (b_1b_00)_m = (102)_m$. This is reduced to the following two systems:

$$\left\{ \begin{array}{rcl} a_0 & = & 2 \\ a_1 + b_0 & = & 0 \\ a_2 + b_1 & = & 1 \end{array} \right. \qquad \left\{ \begin{array}{rcl} a_0 & = & 2 \\ a_1 + b_0 & = & m \\ a_2 + b_1 + 1 & = & 1 \end{array} \right.$$

from which we get the set of all solutions (x_1, x_2): $\{((102)_m, (0)_m), ((2)_m, (10)_m)\} \cup \{ ((c\ 2)_m, (m - c\ 0)_m) \mid 1 < c < m\}$ □

The non-homogeneous case is handled in the proof of the following:

Theorem 5.2. The satisfiability problem for linear parametric Diophantine systems $\mathfrak{D}[1]$ is decidable.

Proof:
Let $A(m)\mathbf{x} = B(m)$ (1) be the original (non-homogeneous) system, where $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$ and $\mathbf{x}' = \langle x_1, \ldots, x_n, x_{n+1} \rangle$, and $A'(m)\mathbf{x}' = 0$ (2) be the homogeneous system $A(m)\mathbf{x} = B(m)x_{n+1}$. Also let S and S' be the sets of non-trivial solutions of the systems (1) and (2). It is sufficient to show the following: $S \neq \emptyset$ iff S' has a minimal element with x_{n+1}. Since the latter is decidable (by adding the condition $x_{n+1} = 1$ to the Presburger systems derived using the m base representation), we obtain the result. □

Theorem 5.2, together with Theorem 4.2 entail the main novel result of this paper:

Corollary 5.2. The reachability problem for single loop parametric flat counter automata FCA$(p, 1)$ is decidable.

6. Conclusions

We have studied a generalization of the flat counter automata considered by Comon and Jurski in [5], obtained by adding parameters to the transition relations. We reduce the reachability problem for these automata to either Presburger arithmetic, in the non-parametric case, and to linear Diophantine systems with one parameter, for single-loop automata with multiple parameters. The existence of solutions for the latter class of systems is shown to be decidable. This entails the decidability of the reachability problem for counter automata with parameters and one control loop, while in general, this problem is undecidable for flat automata with more than one control loop.

References

[1] Annichini, A., Bouajjani, A., M.Sighireanu: TreX: A Tool for Reachability Analysis of Complex Systems, *Proc.CAV*, 2102, Springer, 2001.

[2] Bardin, S., Finkel, A., Leroux, J., Petrucci, L.: FAST: Fast Accelereation of Symbolic Transition systems, *Proc. TACAS*, 2725, Springer, 2004.

[3] Boigelot, B.: On Iterating Linear Transformations over Recognizable Sets of Integers, *TCS*, **309**(2), 2003, 413–468.

[4] Church, A.: An unsolvable problem of elementary number theory, *American Journal of Mathematics*, **58**, 1936, 345 – 363.

[5] Comon, H., Jurski, Y.: Multiple Counters Automata, Safety Analysis and Presburger Arithmetic, *Proc. CAV*, 1427, Springer, 1998.

[6] Dasdan, A., Irani, S., Gupta, R. K.: Efficient Algorithms for Optimum Cycle Mean and Optimum Cost to Time Ratio Problems, *Design Automation Conference*, 1999.

[7] Finkel, A., Leroux, J.: How to compose Presburger-accelerations: Applications to broadcast protocols, *Proc. FST&TCS*, 2556, Springer, 2002.

[8] Ginsburg, S., Spanier, E. H.: Semigroups, Presburger Formulas and Languages, *Pacific Journal of Mathematics*, **16**(2), 1966, 285–296.

[9] Gödel, K.: Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I, *Monatshefte für Mathematik und Physik*, **38**, 1931, 173 – 198.

[10] Gurari, E. M., Ibarra, O. H.: Two-way Counter Machines and Diophantine Equations, *Journal of the Association for Computing Machinery*, **29**(3), July 1982, 863–873.

[11] Hilbert, D.: Mathematische Probleme. Vortrag, gehalten auf dem internationalen Mathematiker-Kongress zu Paris, *Nachrichten von der Königliche Gesellschaft der Wissenschaften zu Göttingen*, 1900.

[12] Ibarra, O. H.: Reversal-Bounded Multicounter Machines and Their Decision Problems, *Journal of the Association for Computing Machinery*, **25**(1), January 1978, 116 – 133.

[13] Ibarra, O. H., Dang, Z.: On the Solvability of a Class of Diophantine Equations and Applications, *Theoretical Computer Science*, **352**, 2006, 342 – 346.

[14] Ibarra, O. H., Jiang, T., Tran, N., Wang, H.: New Decidability Results Concerning Two-way Counter Machines, *SIAM J. Comput.*, **24**(1), February 1995, 123–137.

[15] Leroux, J., Sutre, G.: On flatness for 2-dimensional vector addition systems with states, *Proc. CONCUR*, 3170, Springer, 2004.

[16] Matiyasevich, Y.: Enumerable Sets are Diophantine, *Journal of Sovietic Mathematics*, **11**, 1970, 354 – 358.

[17] Matiyasevich, Y.: Personal communication, 2005.

[18] Minsky, M.: *Computation: Finite and Infinite Machines*, Prentice-Hall, 1967.

[19] Pottier, L.: *Solutions minimales des systemes diophantiens lineaires: bornes et algorithmes*, Technical Report 1292, INRIA Sophia Antipolis, 1990.

[20] Presburger, M.: Über die Vollstandigkeit eines gewissen Systems der Arithmetik, *Comptes rendus du I Congrés des Pays Slaves*, Warsaw 1929.

[21] Rosser, B.: Extensions of some theorems of Gödel and Church, *The Journal of Symbolic Logic*, **1**, 1936, 87 – 91.

[22] Wolper, P., Boigelot, B.: Verifying Systems with Infinite but Regular State Spaces, *Proc. CAV*, 1427, Springer, 1998.