Reachability in Parametric Counter Automata*

Christoph Haase**, Stephan Kreutzer, Joël Ouaknine, and James Worrell

Oxford University Computing Laboratory, UK

Abstract. Counter automata are a fundamental class of infinite-state systems. They consist of a finite set of control locations, transitions between them and a finite set of counters over the positive integers. At each transition, a counter can be tested for zero or incremented by some integer value.
In this paper, we extend counter automata to parametric counter automata by allowing the counters to be updated by integer-valued parameters. The reachability problem asks whether there is an instantiation of the parameters such that there is a run between two given configurations of such an automaton. Our main result is that this problem is NP-complete for the class of parametric counter automata with only one counter, and that this is the only sub-class for which this problem is in general decidable. The NP-completeness result is shown by a reduction to quantifier free Presburger arithmetic with divisibility.

1 Introduction

Counter automata are a fundamental computational model, known to be equivalent to Turing machines [20], and there has been considerable interest in subclasses of counter machines for which reachability is decidable, such as Petri nets, one-counter automata and flat counter automata [5, 19]. As originally conceived by Minsky, counters are updated either by incrementation or decrementation instructions. However, for many applications of counter machines, including modeling computer programs, it is natural to consider more general types of updates, such as adding integer constants to a counter [2, 5, 15] or adding integer parameters [3, 14]. Parametric automata are used in various synthesis problems, and to model open programs, whose behavior depends on values input from the environment [1]. In [23] parameters are also used to model resources (e.g., time, memory, dollars) consumed by transitions. The reachability problem for parametric counter automata asks whether there exist values of the parameters such that a given configuration is reachable from another given configuration.

In this paper we show NP-completeness of the reachability problem for one-counter automata in which counters can be updated by adding integer constants,

* This paper is an extended version of the paper "Reachability in succinct and parametric one-counter automata" by the same authors which appeared in the proceedings of the 20th International Conference on Concurrency Theory.

** Corresponding author. Email: christoph.haase@comlab.ox.ac.uk

where the latter are encoded in binary. We also show decidability of reachability for parametric one-counter automata by reduction to existential Presburger arithmetic with divisibility [16]. We defer consideration of the complexity of the latter problem to the full version of this paper.

Related work. The verification literature contains a large body of work on decidability and complexity for various problems on restricted classes of counter automata. The work that is closest to our own is that of Demri and Gascon on model checking extensions of LTL over one-counter automata [8]. They consider automata with one integer-valued counter, with updates encoded in unary, and with sign tests on the counter. They show that reachability in this model is NL-complete. Determining the complexity of reachability when updates are encoded in binary is posed as an open problem by Demri in [7], Page 61, Problem 13. Since this last problem assumes an integer-valued counter with sign tests, it is more general than the one considered in our Theorem 2, and it remains open.

Also, there is an interesting link between reachability in one-counter automata and the compressed word problem for compressed regular expressions. A compressed word is given by a straight-line program and a compressed regular expression is a regular expression that allows for compressed words as primitives instead of only single alphabet symbols. The compressed word problem is to decide whether some compressed word is contained in the language defined by a compressed regular expression. It is shown by Plandowski and Rytter in [21] that this problem is NP-complete over a unary alphabet. The case of a non-unary alphabet is left open. It is not difficult to see that the compressed word problem over a unary alphabet can be rephrased as a reachability problem for one-counter automata in which all counter updates are positive integers. Thus, our main result generalises the result from [21], but does however not contribute to the open case of non-unary alphabets.

Another work closely related to our own is that of Ibarra, Jiang, Tran and Wang [14], which shows decidability of reachability for a subset of the class of deterministic parametric one-counter automata with sign tests. The decidability of reachability over the whole class of such automata is stated as an open problem in [14]. Note that although we do not allow negative counter values and sign tests, we allow nondeterminism. Thus our Theorem 2 is incomparable with this open problem.

2 Preliminaries

In this section, we establish the foundations for the remainder of the paper. We start with introducing the most general model of counter automata and their reachability problem. We recall some decidability and undecidability results known from the literature and continue with introducing parametric counter automata. It turns out that even in restricted subclasses the reachability problem for parametric counter automata with more than one counter is in general undecidable. This motivates the consideration of parametric counter automata with only one counter, for which we show that the reachability problem is NP-hard.

The main result of this paper, the NP upper bound, will be shown in the next section.

In the following, let $\mathbf{N}$ denote the set of natural numbers including 0 and $\mathbf{Z}$ be the set of integers. Given a set S, we denote by $\#S$ the cardinality of S.

Definition 1. *Let $k \in \mathbf{N}$ and let $Op := \{add(z) : z \in \mathbf{Z}\} \cup \{zero\}$ be a set of* counter operations*. A k-*counter automaton *$\mathcal{A} = \langle Q, q_{in}, F, \Delta, \lambda\rangle$ is a five tuple consisting of a finite set Q of* control locations*, an* initial location *$q_{in} \in Q$, a set $F \subseteq Q$ of* final locations*, a* transition relation *$\Delta \subseteq Q \times Q$ and a* transition labeling function *$\lambda : \Delta \to Op^k$.* Counter automata *are the family of all k-counter automata for $k \geq 1$.*

In this definition, we assume all numbers to be encoded in their standard binary encoding. A k-counter automaton is called *zero-test free* if $\lambda(\delta) \in \{\lambda(\delta)\} \cap \{add(z) : z \in \mathbf{Z}\}^k$ for every $\delta \in \Delta$. The *size* of a counter automaton is the cardinality of its set of control locations plus the number of symbols used to write down the integer increments. A *configuration* C of a k-counter automaton $\mathcal{A}$ is a tuple $(q, \mathbf{N}^k)$ consisting of a control location q, which we refer to as the *location component of* C, and an assignment of a value to each of the k counters. Denote by $\mathcal{C}_\mathcal{A}$ the set of all *configurations* of $\mathcal{A}$. The k-counter automaton $\mathcal{A}$ induces a transition system $(\mathcal{C}_\mathcal{A}, \to_\mathcal{A})$, where $\to_\mathcal{A} \subseteq \mathcal{C}_\mathcal{A} \times \mathcal{C}_\mathcal{A}$ such that $((q, c_1, \ldots, c_k), (q', c'_1, \ldots, c'_k)) \in \to_\mathcal{A}$ if and only if

- $(q, q') \in \Delta$,
- $\lambda(q, q') = (op_1, \ldots, op_k)$,
- $c_i = c'_i = 0$ if $op_i = zero$,
- $c_i = c'_i - z \geq 0$ if $op_i = add(z), 1 \leq i \leq k$.

We call $(q_{in}, 0, \ldots, 0)$ the *initial configuration*. Given configurations $C_1, C_2 \in \mathcal{C}_\mathcal{A}$, we subsequently use $\to_\mathcal{A}$ in infix notation, i.e., write $C_1 \to_\mathcal{A} C_2$ instead of $(C_1, C_2) \in \to_\mathcal{A}$, and denote by $\to^*_\mathcal{A}$ the transitive closure of $\to_\mathcal{A}$. A *run r of $\mathcal{A}$ in $(\mathcal{C}_\mathcal{A}, \to_\mathcal{A})$* is a possibly countably infinite sequence of configurations $r : C_1 \to_\mathcal{A} C_2 \to_\mathcal{A} \cdots$. By $|r|$ we denote the length of r, i.e. the number of transitions traversed by r or ∞ if r is infinite, and given $0 < i \leq |r| + 1$, we denote by $r(i)$ the configuration C_i. Given a finite run $r : C_1 \to_\mathcal{A} \cdots \to_\mathcal{A} C_n$ and a possibly infinite run $r' : C'_1 \to_\mathcal{A} C'_2 \to_\mathcal{A} \cdots$ such that $C_n = C'_1$, the *composition* $r \cdot r'$ of r and r' is the run $r \cdot r' : C_1 \to_\mathcal{A} \cdots \to_\mathcal{A} C_n \to_\mathcal{A} C'_2 \to_\mathcal{A} \cdots$. We say that the configuration C_2 is *reachable* from the configuration C_1 if and only if $C_1 \to^*_\mathcal{A} C_2$.

Definition 2. *Let $\mathcal{A} = \langle Q, q_{in}, F, \Delta, \lambda\rangle$ be a counter automaton and C_1, C_2 be configurations of $\mathcal{A}$. The* reachability problem *is to decide $C_1 \to^*_\mathcal{A} C_2$. The* emptiness problem *is to decide whether $(q_{in}, 0, \ldots, 0) \not\to^*_\mathcal{A} (q, 0, \ldots, 0)$ for all $q \in F$.*

Clearly, reachability is reducible in logarithmic space to non-emptiness and *vice versa*. The decidability of the emptiness problem was first considered by Minksy who showed that it is in general undecidable.

Proposition 1 ([20]). *The emptiness problem for k-counter automata is undecidable for $k \geq 2$.*

Minsky's result left two directions for the identification of decidable fragments of counter automata. First, it is easily seen that the reachability problem for one-counter automata (OCA) is decidable, since it can be reduced to reachability in a pushdown system with a unary stack alphabet, which is decidable, see e.g. [4]. Second, zero-test free counter automata have been considered in the literature, where they are commonly named *vector addition systems with states*.

Proposition 2 ([12]). *The reachability problem for vector addition systems with states is decidable.*

In this paper, we investigate the decidability and complexity of reachability for *parametric* counter automata, which generalise counter automata.

Definition 3. *A* parametric k-counter automaton (k-PCA) *is a six-tuple* $\mathcal{A} = \langle Q, q_{in}, F, P, \Delta, \lambda \rangle$. *The extra feature compared to counter automata is that Op additionally allows for adding or subtracting a parametric value from a finite set of* parameters P *to the counter, i.e.,* $Op = \{add(+p), add(-p) : p \in P\} \cup \{add(z) : z \in \mathbf{Z}\} \cup \{zero\}$. Parametric counter automata (PCA) *are the family of all parametric k-counter automata.*

A concrete instance of a PCA is obtained by an *instantiation* $I : P \to \mathbf{N}$ which assigns a natural number to each parameter. Given a PCA $\mathcal{A}$, the *instantiation $\mathcal{A}(I)$ of $\mathcal{A}$ with respect to I* is obtained from the counter automaton $\mathcal{A}$ by replacing the label of every edge labeled with $add(\pm p)$ with $add(\pm I(p))$. Let $C_1, C_2 \in \mathcal{C}_{\mathcal{A}}$, the *reachability problem for a* PCA $\mathcal{A}$ is to decide whether there exists an instantiation I such that $C_1 \to^*_{\mathcal{A}(I)} C_2$ in the transition system $(\mathcal{C}_{\mathcal{A}(I)}, \to_{\mathcal{A}(I)})$ induced by $\mathcal{A}(I)$. We write $C_1 \to^*_{\mathcal{A}} C_2$ if this is the case. Of course, the decidability of this problem is already limited by Proposition 1. Hence, there are two directions which are worth being investigated. On the one hand, we can restrict ourselves to the case of deciding reachability in the presence of only one counter. On the other hand, we can ban zero tests on the counters and thus decide reachability for *parametric k-vector addition system with states*. It is however not difficult to prove that the latter problem is in general undecidable.

Theorem 1. *The reachability problem for parametric $k-$vector addition system with states is undecidable for $k \geq 4$.*

Proof. We reduce from the emptiness problem for two-counter automata. A two-counter automaton $\mathcal{A}$ is going to be simulated by a parametric four-vector addition system with states $\mathcal{A}'$ with one parameter. In [18], Lipton showed ExpSpace-hardness of reachability for vector addition system with states. One key ingredient he uses in his hardness proof is to impose a parity condition on pairs of counters which allows him to simulate zero tests on counters. We subsequently adopt this technique in order to mimic $\mathcal{A}$ by $\mathcal{A}'$.

Let $\mathcal{A} = \langle Q, q_{in}, F, \Delta, \lambda \rangle$ be a two-counter automaton. We define a parametric four-vector addition system with states $\mathcal{A}' = \langle Q', q'_{in}, F, \{p\}, \Delta', \lambda' \rangle$, where

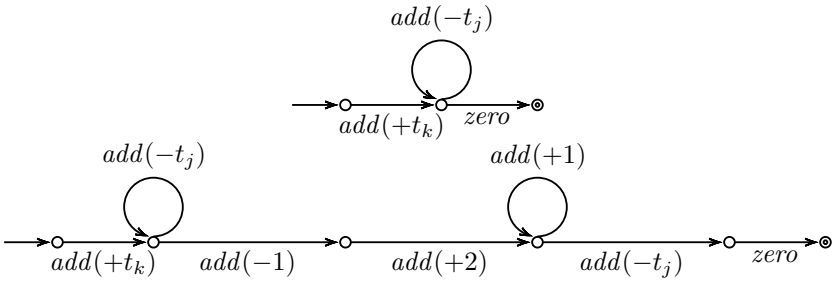


Fig. 1. Gadgets for testing $t_j|t_k$ (top) and $\neg(t_j|t_k)$ (bottom) assuming $t_j, t_k \geq 0$ in an instantiation.

- $Q' = Q \dot\cup \{q_e : e \in \Delta\} \dot\cup \{q'_{in}\}$
- $\Delta' = \{(q, q_e), (q_e, q') : e = (q, q') \in \Delta\} \cup \{(q'_{in}, q_{in})\}$
- $\lambda'(q'_{in}, q_{in}) = (add(0), add(+p), add(0), add(+p))$
- $\lambda'(q, q_e) = \begin{cases} (add(z_1), add(-z_1), add(z_2), add(-z_2)) & \text{if } \lambda(e) = (add(z_1), add(z_2)), z_1, z_2 \in \mathbf{Z} \\ (add(0), add(-p), add(z_2), add(-z_2)) & \text{if } \lambda(e) = (zero, add(z_2)), z_2 \in \mathbf{Z} \\ (add(z_1), add(-z_1), add(0), add(-p)) & \text{if } \lambda(e) = (add(z_1), zero), z_1 \in \mathbf{Z} \\ (add(0), add(-p), add(0), add(-p)) & \text{if } \lambda(e) = (zero, zero) \end{cases}$
- $\lambda'(q_e, q') = \begin{cases} (add(0), add(0), add(0), add(0)) & \text{if } \lambda(e) = (add(z_1), add(z_2)), z_1, z_2 \in \mathbf{Z} \\ (add(0), add(+p), add(0), add(0)) & \text{if } \lambda(e) = (zero, add(z_2)), z_2 \in \mathbf{Z} \\ (add(0), add(0), add(0), add(+p)) & \text{if } \lambda(e) = (add(z_1), zero), z_1 \in \mathbf{Z} \\ (add(0), add(+p), add(0), add(+p)) & \text{if } \lambda(e) = (zero, zero). \end{cases}$

The counter value c_i of i-th counter of $\mathcal{A}$ is represented by the counter value c'_{2i-1} of $\mathcal{A}'$ for $i \in \{1, 2\}$. The crucial point is that for any instantiation I, as soon as we have reached $(q_{in}, 0, p, 0, p)$ in $(\mathcal{C}_{\mathcal{A}'(I)}, \rightarrow_{\mathcal{A}'(I)})$ starting from the initial configuration, we have that $c'_{2i-1} + c'_{2i} = p$ in every configuration whose location component is from Q on every path in $(\mathcal{C}_{\mathcal{A}'(I)}, \rightarrow_{\mathcal{A}'(I)})$. That way, when imitating $\mathcal{A}$ by $\mathcal{A}'$, we can ensure that the counter value c_{2i-1} is zero if and only if we can subtract p from c'_{2i}, and this is exactly what is done when passing through the intermediate states $q_e \notin Q$. Thus, if $\mathcal{A}$ is non-empty then both counters c_1 and c_2 do not exceed some maximum value on a path witnessing non-emptiness and this value yields an instantiation I such that $(q'_{in}, 0, 0, 0, 0) \rightarrow^*_{\mathcal{A}'(I)} (q, 0, 0, 0, 0)$ for some $q \in F$ in $(\mathcal{C}_{\mathcal{A}'(I)}, \rightarrow_{\mathcal{A}'(I)})$. The other direction follows analogously.

In the light of this undecidability result, in the remainder of this paper we concentrate on *parametric one-counter automata (*POCA*)*. The main result of our paper is the following theorem.

Theorem 2. *The reachability problem for parametric one-counter automata is* NP*-complete.*

We are going to show the NP upper bound of the reachability problem in the next section. The easier part is to show NP-hardness. We reduce from the satisfiability

problem in quantifier free Presburger arithmetic with divisibility (QFPAD), i.e., the existential theory $\langle \mathbf{N}, +, -, |, 0, 1 \rangle$. Given a vector of free variables $\boldsymbol{X} = (x_1, \ldots, x_n)$, a formula $\varphi(\boldsymbol{X})$ of QFPAD is a Boolean combination of atoms $t_j | t_k$ and each t_i is a linear polynomial in the variables $\boldsymbol{X}$. The size of a QFPAD formula is the number of symbols used to write it down assuming numbers are encoded in binary. Given a linear polynomial t in $\boldsymbol{X}$ and natural numbers $\boldsymbol{Z} = (z_1, \ldots, z_n) \in \mathbf{N}^n$, denote by $t[\boldsymbol{X}/\boldsymbol{Z}]$ the $z \in \mathbf{Z}$ obtained from evaluating t by replacing each x_i with $z_i, 1 \leq i \leq n$. We define $(t_i | t_j)[\boldsymbol{X}/\boldsymbol{Z}] = \mathit{true}$ if there is an integer $k \in \mathbf{Z}$ such that $kt_i[\boldsymbol{X}/\boldsymbol{Z}] = t_j[\boldsymbol{X}/\boldsymbol{Z}]$ and $(t_i | t_j)[\boldsymbol{X}/\boldsymbol{Z}] = \mathit{false}$ otherwise. A QFPAD formula $\varphi(\boldsymbol{X})$ is satisfiable if there is $\boldsymbol{Z} = (z_1, \ldots, z_n) \in \mathbf{N}^n$ such that the Boolean formula obtained from substituting each $t_i | t_j$ with $(t_i | t_j)[\boldsymbol{X}/\boldsymbol{Z}]$ evaluates to *true*.

It was shown by Lipshitz in [16] that the satisfiability problem for QFPAD is decidable and later that it is NP-complete for a fixed size formula [17]. In our reduction, we construct for a given formula $\varphi(\boldsymbol{X})$ a POCA $\mathcal{A}_\varphi$ with parameters $x_1, \ldots, x_n$ such that φ is satisfiable if and only if the reachability problem for $\mathcal{A}_\varphi$ is solvable for two designated configurations. We follow a similar pattern to [1], in which the same problem is reduced to reachability in two-clock parametric timed automata. As a preparation, let us first consider literals of the form $t_j | t_k$ respectively $\neg(t_j | t_k)$. By exploiting the fact that division is just repeated subtraction, Figure 1 sketches two gadgets for testing divisibility respectively non-divisibility assuming that both terms t_j and t_k are positive in an instantiation. In the figure, control locations are depicted as circles and transitions with their labels as arrows between the control locations. The circle with an incoming edge is the initial location and the double circle is a final location. For brevity, we represent the sequence of locations and transitions that compute t_j respectively t_k by just one arrow. Clearly, whenever we can find an instantiation such that the final location is reachable from the initial location in Figure 1 then t_j divides respectively does not divide t_k. By changing the signs of the t_i in Figure 1, similar gadgets can be constructed for the cases where one or both of t_j and t_k are negative. Now assume $\varphi(\boldsymbol{X})$ to be a QFPAD formula in negation normal. For each literal of φ, the POCA $\mathcal{A}_\varphi$ consists of a control location that non-deterministically branches into four gadgets like those in Figure 1, one for each possible guess of the signs of t_j and t_k. Conjunction in φ can then be simulated in $\mathcal{A}_\varphi$ by sequential composition of those gadgets, and disjunction by non-deterministic branching. Finally, we can designate locations q and q' such that $(q', 0)$ is reachable from $(q, 0)$ if and only if φ is satisfiable. Obviously, the size of $\mathcal{A}_\varphi$ is potentially exponential in the size of φ, since summands of the form $ax, a \in \mathbf{N}$ in terms of φ need to be represented by $a + 1$ control locations, while a is encoded in binary in φ. However, in the formula used in [17] to show that QFPAD satisfiability is NP-hard, every variable has a constant multiplier. Thus, the following proposition holds.

Proposition 3. *The reachability problem for a parametric one-counter automaton of fixed size is* NP*-hard.*

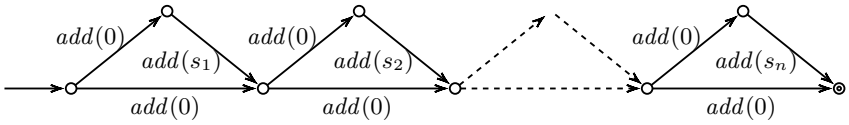


Fig. 2. One-counter automaton for the reduction from SubsetSum.

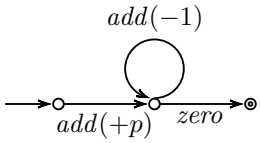
For brevity, if $\varphi(x_1, \ldots, x_k)$ is a QFPAD formula, we will not always explicitly declare all of the variables $x_1, \ldots, x_n$ to be free, but only those that are of particular interest for us. For example, if we want to emphasize that x_1 is free in φ, we just write $\varphi(x_1)$ instead of $\varphi(x_1, \ldots, x_k)$. Also, we allow for a partly evaluation of free variables, e.g., write $\varphi[x_1/z_1]$ for the QFPAD formula obtained from replacing every occurrence of x_1 with z_1 in φ.

Remark 1. It is worth mentioning that the reachability problem remains NP-hard for OCA in the presence of no parameters. There is a simple reduction from the NP-complete SubsetSum-problem [11]. SubsetSum is to decide for a given finite set $S = \{s_1, \ldots, s_n\} \subset \mathbf{N}$ and a target $t \in \mathbf{N}$ whether there exists a subset $S' \subseteq S$ such that $\sum_{s \in S'} s = t$. Figure 2 shows the OCA used for the reduction. Denote by q its initial and by q' its final location. On a run from q to q' the automaton can non-deterministically choose to whether or not add each s_i to the counter, $1 \leq i \leq n$. Hence a set S' with the above properties exists if and only if (q', t) is reachable from $(q, 0)$. This hardness result heavily depends on the binary encoding of the numbers in the automaton. In fact, it follows for example from [9] that the problem is NL-complete if numbers are encoded in unary.

3 Reachability in Parametric One-Counter Automata

In this section, we are going to show that the reachability problem for POCA is in NP and thus NP-complete. Before we start with the technical details, let us prepare ourselves with a high-level introduction to the proof.

Traditionally when deciding reachability for finite-state automata, an algorithm searches for a run of the automaton that connects the two control locations in question. The decidability of this problem follows from the fact that the length of such a run is finite and bounded by the number of control locations of the automaton. This idea can be adopted to OCA. It is for example shown in [9] that if there is a witnessing run for a reachability problem of an OCA, then there is a witnessing run whose length is exponentially bounded in the size of the OCA. However, this approach has no direct correspondence in the setting of POCA, as illustrated by the following example:



For any instantiation of the parameter p with a natural number n, a run witnessing non-emptiness of the instantiated automaton has length $n+2$ and is thus not bounded by the size of the automaton.

In order to deal with this problem, we first need to put a different view on OCA by treating them as weighted directed graphs. A run in $(\mathcal{C}_\mathcal{A}, \rightarrow_\mathcal{A})$ of an OCA $\mathcal{A}$ then corresponds to a path in its corresponding weighted directed graph $G_\mathcal{A}$. The advantage is that a path π in $G_\mathcal{A}$ can be described in terms of a *path flow*. Such a path flow is a function that assigns to each edge the number of times the edge is traversed by π. An important property is that a path flow has a *finite* domain. Conversely, a path flow also induces a set of paths in $G_\mathcal{A}$ that however do not necessarily correspond to runs in $(\mathcal{C}_\mathcal{A}, \rightarrow_\mathcal{A})$. We overcome this problem by showing that if two configurations are reachable then there is a run of $\mathcal{A}$ in $(\mathcal{C}_\mathcal{A}, \rightarrow_\mathcal{A})$ that has a corresponding path in $G_\mathcal{A}$ which can be described by at most three path flows of a special type. Afterwards, we dedicate ourselves to POCA. We show that the question of deciding whether there exists an instantiation I of a POCA $\mathcal{A}$ such that a certain path flow in $G_{\mathcal{A}(I)}$ exists can be decided by solving a system of quadratic Diophantine equations. Solving such systems is in general undecidable, but the systems we obtain lie in a sub-class which can be shown to be decidable via a reduction to the satisfiability problem of a corresponding formula in QFPAD. As stated in the previous section, satisfiability in QFPAD is NP-complete and together with some technicalities it follows that reachability for one-counter automata is NP-complete.

We close this section by showing how the developed techniques can be used to show that deciding emptiness of POCA with Büchi acceptance condition is co-NP-complete.

3.1 Weighted Graphs and Path Flows

A *weighted directed graph* is a tuple $G = (V, E, w)$, where V is a finite set of *vertices*, $E \subseteq V \times V$ is a finite set of *directed edges* and $w : E \rightarrow \mathbf{Z}$ assigns a *weight* to each edge. In the following, we call weighted directed graphs just weighted graphs or graphs. The size $|G|$ of a graph G is the cardinality of its vertices plus the number of symbols it takes to write down the weights of its edges. The graph G^{op}, defined as $G^{op} := (V, E^{op}, w^{op})$ with $E^{op} = \{(v,u) : (u,v) \in E\}$ and $w^{op}(v,u) = -w(u,v)$, is called the *skew transpose* of G. Given $F \subseteq E$, we define G *restricted to* F as $G/F := (V \cap \{v \in V : (v,w) \in F \text{ or } (w,v) \in F\}, F, w)$. A finite sequence of vertices $\pi = v_0 v_1 \dots v_n$ such that $v_0 = s$, $v_n = t$ and $(v_i, v_{i+1}) \in E, 0 \leq i < n$ is called an *s-t path* π of length n, or just a path. We often write $\pi : s \rightarrow^* t$ to indicate that π is an s-t-path. By $|\pi|$ we denote the length of π. The set of *edges traversed by* π is $edges(\pi) := \{(v_i, v_{i+1}) : 0 \leq i < n\}$. A graph is called *connected* if there is an s-t path for all $s, t \in V$. Whenever

$s = t$, an s-t path is called an *s-cycle*, or just a cycle. Given paths $\pi : s \to^* t$ and $\pi' : t \to^* u$, $\pi \cdot \pi'$ denotes the s-u path obtained from composing π and π', similar to the composition of runs. For a given cycle $\ell : v \to^* v$, we define the cycles $\ell^0 = v$ and $\ell^{i+1} = \ell^i \cdot \ell$ for $i > 0$. The *weight* of a path $\pi : v_0 \dots v_n$ is the sum over the weights of all edges visited in π, i.e.,

$$\mathit{weight}(\pi) = \sum_{0 \le i < n} w(v_i, v_{i+1}). \tag{1}$$

If ℓ is a cycle such that $\mathit{weight}(\ell) > 0$ then we call ℓ a *positive cycle*. Likewise, we call ℓ a *negative cycle* if $\mathit{weight}(\ell) < 0$. A graph G contains a positive respectively negative cycle if there is a path π in G with positive respectively negative weight. A v-cycle ℓ is *chord-free* if $\ell = vv_1 \dots v_n v$ and $v_i \neq v_j$ for $1 \le i \neq j \le n$. Given $G = (V, E, w)$, if $V = \{s, t, v_1, \dots, v_n\}$ and $E = \{(s, v_1), (v_1, v_2), \dots, (v_n, t)\}$ we call G an *s-t path graph*. Similarly, if $V = \{v, v_1, \dots, v_n\}$ and $E = \{(v, v_1), (v_1, v_2), \dots, (v_n, v)\}$, G is called an *v-cycle graph*.

It is obvious that paths may in general have finite but potentially unbounded length. In order to give compact descriptions of paths that capture their essential properties we introduce the concept of path flows.

Definition 4. *Let $G = (V, E, w)$, $f : E \to \mathbf{N}$ be a function and $F := \{e \in E : f(e) > 0\}$ be the* support of f. *We call f an s-t* path flow *if it satisfies the following Eulerian path conditions:*

(i) (a) If $s = t$ then

$$\sum_{(v,u) \in E} f(v, u) = \sum_{(u,v) \in E} f(u, v) \text{ for all } u, v \in V. \tag{2}$$

(b) If $s \neq t$ then

$$\sum_{(v,u) \in E} f(v, u) = \sum_{(u,v) \in E} f(u, v) \text{ for all } u, v \in V \setminus \{s, t\}, \tag{3}$$

$$\sum_{(s,v) \in E} f(s, v) = \sum_{(v,s) \in E} f(v, s) - 1 \text{ for all } v \in V, \tag{4}$$

$$\sum_{(t,v) \in E} f(t, v) = \sum_{(v,t) \in E} f(v, t) + 1 \text{ for all } v \in V. \tag{5}$$

(ii) The sub-graph $G/(F \cup \{(t, s)\})$ is connected.

A path π determines a path flow f_π, where for each edge $e = (v', v') \in E$, $f_\pi(e)$ is defined to be the number of times vv' occurs in π. Conversely, the conditions from Definition 4 ensure that any path flow f induces at least one possible path. A flow f with support F contains a positive respectively negative cycle if G/F contains a positive respectively negative cycle. Subsequently, we call the

set $in(f) = \{v : \text{ there is } u \in V \text{ with } f(u,v) > 0\}$ the set of nodes with *incoming flow*. Just as paths can be sequentially composed, path flows can be composed by summation: given an s-t path flow f and a t-u path flow f, we define an s-u path flow $f + g$ by $(f + g)(e) = f(e) + g(e)$ for each edge $e \in E$. An s-t path flow f induces a path flow f^{op} in G^{op}, where $f^{op}(v,u) = f(u,v)$. The *weight* of a path flow f is defined to be

$$weight(f) = \sum_{e \in E} f(e) \cdot w(e). \tag{6}$$

Definition 5. *Let $G = (V, E, w)$ be a graph and $s, t \in V$. A sequence of path flows $f', f_1, \ldots, f_n$ with supports $F', F_1, \ldots, F_n$ is an s-t* cycle decomposition *if G/F' is an s-t path graph, G/F_i is a v_i-cycle graph for some $v_i \in V, 1 \leq i \leq n$ and the graph $G/(F \cup \{(t,s)\})$ is connected for $F = F' \cup \bigcup_{1 \leq i \leq n} F$.*

Proposition 4. *Let $G = (V, E, w)$ be a graph and let $z \in \mathbf{Z}$. There is an s-t path flow f with $weight(f) = z$ if and only if there is an s-t cycle decomposition $f', f_1, \ldots, f_n$ with $weight(f') + \sum_{1 \leq i \leq n} weight(f_i) = z$, $1 \leq n < \#E$.*

Proof. ($\Rightarrow$) Let F be the support of f. If $F = \emptyset$ or G/F is an s-t path graph then we are done.

Otherwise, let $\mathcal{L} = \{\pi : \pi \text{ is a chord-free cycle in } G/F\}$. Choose $e_1 = (v, v')$ such that $f(e_1) = min\{n \in \mathbf{N} : f(w,w') = n, ww' \ldots w \in \mathcal{L}\}$ and let $\ell = vv' \ldots v \in \mathcal{L}$ be a v-cycle in G/F. Define the v-v path flow f_1 such that for any $e \in E$,

$$f_1(e) := \begin{cases} f(e_1) \text{ if } e \in edges(\ell) \\ 0 \qquad \text{otherwise.} \end{cases}$$

and the s-t path flow f' such that $f = f' + f_1$. Obviously, we have $weight(f') + weight(f_1) = z$. We can then repeatedly apply this procedure to f', but at most $\#E - 1$ times until the support F' of f' is empty or G/F' is an s-t path, and eventually obtain the required path flows $f', f_1, \ldots f_n$ for some $1 \leq n < E$.

($\Leftarrow$) The fact that $G/(\bigcup_{1 \leq i \leq n} F_i \cup F' \cup \{(t,s)\})$ is connected guarantees that $f = f' + f_1 + \ldots + f_n$ is an s-t path flow and $weight(f) = z$.

Later in this paper, we will be more interested in the supports of s-t cycle decompositions of a graph $G = (V, E, w)$ and denote by $CDS(G, s, t)$ the set of all of them. Formally, $(F', F_1, \ldots, F_n) \in CDS(G, s, t)$ if $F', F_1, \ldots, F_n$ are supports of some s-t cycle decomposition as in Definition 5, $1 \leq n < \#E$.

3.2 Reachability Criteria

Let $\mathcal{A} = \langle Q, q_{in}, F, \Delta, \lambda \rangle$ be an OCA. In this section, we exclusively consider zero-test free OCA. Thus we can view $\mathcal{A}$ as a weighted graph $G_{\mathcal{A}} := (V, E, w)$ with $V := Q$, $E := \Delta$ and $w(q,q') := z$ if $\lambda(q,q') = add(z), q, q' \in Q, z \in \mathbf{Z}$. Subsequently, we identify the *graph corresponding to* $\mathcal{A}$ with $G_{\mathcal{A}}$. Just as we can relate $\mathcal{A}$ and $G_{\mathcal{A}}$, we can relate finite runs in $(\mathcal{C}_{\mathcal{A}}, \rightarrow_{\mathcal{A}})$ with paths in $G_{\mathcal{A}}$. Given

a finite run $r : (q_1, c_1) \to (q_2, c_2) \to \ldots \to (q_n, c_n)$ in $(\mathcal{C}_\mathcal{A}, \to_\mathcal{A})$, its *corresponding* q-q' *path* π *in* $G_\mathcal{A}$ is $\pi := q_1 q_2 \ldots q_n$.

Let (q, c) and (q', c') be configurations of $\mathcal{A}$, a run $r : (q, c) \to^* (q', c')$ of $\mathcal{A}$ implies the existence of a corresponding q-q' path π in $G_\mathcal{A}$ which in turn determines a path flow f_π. We regard f_π as a *reachability certificate.* It is obvious that a q-q' path flow f does not necessarily imply the existence of a path π that corresponds to a run $r : (q, c) \to^* (q', c')$ in $(\mathcal{C}_\mathcal{A}, \to_\mathcal{A})$ for some $c, c' \in \mathbf{N}$. Informally speaking, one needs to make sure that there is a path π with $f_\pi = f$ that does not cause the counter to go below zero in the run it corresponds to. Hence, in the remainder of this section, we seek for necessary and sufficient *reachability criteria* that allow a path flow f to serve as a reachability certificate, i.e. f can prove the existence of a path corresponding to a run.

Let $\pi = v_0 v_1 \ldots v_n$ be a path in a graph G, we define the *drop of* π as $drop(\pi) := min\{weight(v_0 \ldots v_i) : 0 \leq i \leq n\}$. The following proposition, which can easily be shown by induction on the length of π, states sufficient and necessary conditions that allow for relating a path to a run.

Proposition 5. *Let π be a path in $G_\mathcal{A}$. Then there is a run $r : (q, c) \to^*_\mathcal{A} (q', c')$ in $(\mathcal{C}_\mathcal{A}, \to_\mathcal{A})$ that π corresponds to if and only if $drop(\pi) \geq -c$ and $weight(\pi) = c - c'$.*

Our first application of this fact is the following proposition, which will later allow us to give a bounded description of positive and negative cycles.

Proposition 6. *Let $G = (V, E, w)$ be a weighted graph, $v \in V$ and $c \in \mathbf{N}$.*

(i) There exists a positive v-cycle ℓ such that $drop(\ell) \geq -c$ if and only if there exists a positive v-cycle ℓ' that can be written as $\ell' = \pi_1 \cdot \pi_2^k \cdot \pi_3$ such that $|\pi_1|, |\pi_2|, |\pi_3| \leq \#V$, π_2 is a positive cycle and $drop(\pi_1 \cdot \pi_2) \geq -c$ for some $k \in \mathbf{N}$.

(ii) There exists a negative v-cycle ℓ such that $drop(\ell) \geq -c$ if and only if there exists a negative v-cycle ℓ' that can be written as $\ell' = \pi_1 \cdot \pi_2^k \cdot \pi_3$ such that $|\pi_1|, |\pi_2|, |\pi_3| \leq \#V$, π_2 is a negative cycle and $drop(\pi_2 \cdot \pi_3) \geq -c$ for some $k \in \mathbf{N}$.

Proof. (i) ($\Rightarrow$) Let ℓ be a positive v-cycle with $|\ell| > \#V$. Without loss of generality, ℓ can be written as $\pi_1 \cdot \pi_2 \cdot \pi_3$ such that $\pi_1 : v \to^* w$, $\pi_2 : w \to^* w$ and $\pi_3 : w \to^* v$ for some $w \in V$ such that $|\pi_1|, |\pi_2| \leq \#V$, $drop(\pi_1 \cdot \pi_2) \geq -c$, $weight(\pi_2) > 0$ and any vertex occurs at most once in π_1 and π_2. Let π_3' be obtained from π_3 by deleting all cycles, hence $|\pi_3'| < \#V$. Let $k \in \mathbf{N}$ be chosen such that $weight(\pi_1) + k\,weight(\pi_2) \geq -drop(\pi_3)'$ and $k\,weight(\pi_2) > -(weight(\pi_1) + weight(\pi_3'))$. Set $\ell' = \pi_1 \cdot \pi_2^k \cdot \pi_3'$. We have

$$\begin{aligned} weight(\ell') &= weight(\pi_1) + weight(\pi_2^k) + weight(\pi_3) \\ &> weight(\pi_1) - (weight(\pi_1) + weight(\pi_3)) + weight(\pi_3) \\ &= 0. \end{aligned}$$

Also, we have

$$\begin{aligned} drop(\ell') &= min\{drop(\pi_1 \cdot \pi_2^k), weight(\pi_1 \cdot \pi_2^k) + drop(\pi_3')\} \\ &\geq min\{-c, weight(\pi_1) + kweight(\pi_2) + drop(\pi_3)\} \\ &= -c. \end{aligned}$$

($\Leftarrow$) This direction is trivially true.

(ii) The statement follows by applying the result from (i) to to the cycle ℓ^{op} in G^{op}.

The proposition shows that if we are looking for a positive or negative cycle in a graph, we can prove its existence by a finite number of vertices. We define the set of *v-cycle candidates* $CC(G, v)$ to be the set of all vectors (π_1, π_2, π_3) such that $0 \leq |\pi_1|, |\pi_2|, |\pi_3| \leq \#V$, $\pi_1 : v \rightarrow^* v', \pi_2 : v' \rightarrow^* v', \pi_3 : v' \rightarrow v$. Obviously, $CC(G, v)$ is finite and each of its elements and be guessed in time polynomial in $|G|$.

We will now seek for criteria of path flows that allow a path flow to prove the existence of a run. One key concept are vertex decompositions of path flows. Such a decomposition of some path flow f is a sequence of paths flows $f_0, \ldots, f_{n-1}$ that sum up to f and traverse the vertex v_i the last time in path flow $f_{i-1}, 1 \leq i < n$.

Definition 6. *Let f be an s-t path flow and $in(f) = \{v_1, \ldots, v_n\}$. A* vertex decomposition *of f is a sequence of paths flows $f_0, \ldots, f_{n-1}$ such that*

- *f_0 is an s-v_1 path flow,*
- *f_i is a v_i-v_{i+1} path flow,*
- *$f = f_0 + f_1 + \ldots + f_{n-1}$*
- *if $i \leq j$ then $v_i \notin in(f_j), 1 \leq i < n$.*

In the next section, we are going to be more interested in the supports of a vertex decomposition rather than the actual flows. The set of *vertex decomposition supports* $VDS(G, s, t)$ is the set of all vectors $(F_0, v_0, v_1, \ldots, F_{n-1}, v_{n-1}, v_n)$ such that there is an *s-t* path flow f that has a vertex decomposition $f_0, \ldots, f_{n-1}$ with each f_i being an v_i-v_{i+1} path flow having support $F_i, 0 \leq i < n$. It is easily checked that the set $VD(G, s, t)$ is finite and each of its elements can be guessed in time polynomial in $|G|$.

Definition 7. *Let G be a graph, f an s-t path flow with support F and $c, c' \in \mathbf{N}$. Then (G, f, c, c') fulfills the*

(i) type-1 reachability criteria *if*
  - *G/F does not contain positive cycles*
  - *$weight(f) = c' - c$*
  - *f has a vertex decomposition $f_0 + \ldots + f_n$ such that $\sum_{0 \leq i \leq j} weight(f_i) \geq -c, 0 \leq j \leq n$;*

(ii) type-2 reachability criteria *if*
  - *G/F does not contain negative cycles*
  - *$weight(f) = c' - c$*

– f^{op} *has a vertex decomposition* $f_0+\ldots+f_n$ *such that* $\sum_{0\leq i\leq j} weight(f_i) \geq -c', 0 \leq j \leq n$;

(iii) type-3 reachability criteria *if*
– $weight(f) = c' - c$
– *there is a positive* s*-cycle* ℓ *in* G *with* $drop(\ell) \geq c$
– *there is a negative* t*-cycle* ℓ' *in* G *with* $drop(\ell') \geq c'$

In the remainder of this section, we are now going to show that the type-1, type-2 and type-3 reachability criteria provide necessary and sufficient conditions for proving the existence of runs in an automaton.

Proposition 7. *Let* (q, c) *and* (q', c') *be configurations of an* OCA $\mathcal{A} = \langle Q, q_{in}, F, \Delta, \lambda\rangle$, $G_{\mathcal{A}}$ *the graph corresponding to* $\mathcal{A}$ *and* f *a* q-q' *path flow. We have that*

(i) if $(G_{\mathcal{A}}, f, c, c')$ *fulfills the type-1 reachability criteria,*
(ii) if $(G_{\mathcal{A}}, f, c, c')$ *fulfills the type-2 reachability criteria, or*
(iii) if $(G_{\mathcal{A}}, f, c, c')$ *fulfills the type-3 reachability criteria*

then $f = f_\pi$ *for some path* π *corresponding to a run* $r : (q, c) \rightarrow^*_{\mathcal{A}} (q', c')$ *in* $(\mathcal{C}_{\mathcal{A}}, \rightarrow_{\mathcal{A}})$.

Proof. (i) We use the terminology of Definition 7(i). Choose some path π_j with $f_{\pi_j} = f_j$ for $1 \leq j \leq n$ and set $\pi = \pi_0 \cdot \pi_1 \cdot \ldots \cdot \pi_n$. By assumption, $G_{\mathcal{A}}/F$ does not contain a positive cycle and consequently there is no positive cycle in π. Hence for two prefixes π_1, π_2 of π with $|\pi_1| \leq |\pi_2|$ that both end in the same vertex, we have $weight(\pi_1) \geq weight(\pi_2)$. It follows that we can obtain the drop of π by just considering the segments of π in which each vertex is visited the last time. We deduce that

$$\begin{aligned} drop(\pi) &= min\,\{weight(\pi_0 \ldots \pi_j) : 0 \leq j \leq n\} \\ &= min\left\{\sum_{0\leq i\leq j} weight(\pi_i) : 0 \leq j \leq n\right\} \\ &= min\left\{\sum_{0\leq i\leq j} weight(f_i) : 0 \leq j \leq n\right\} \\ &\geq -c. \end{aligned}$$

By applying Proposition 5, we deduce that a desired run $r : (q, c) \rightarrow^*_{\mathcal{A}} (q', c')$ in $(\mathcal{C}_{\mathcal{A}}, \rightarrow_{\mathcal{A}})$ exists.

(ii) This part is just the dual of part (i).

(iii) We use the terminology of Definition 7(iii). Let π be some path induced by f. Informally speaking, our strategy is to use the cycles ℓ and ℓ' in order to appropriately "pump up" and "pump down" π. Let $\omega = weight(\ell)$ and $\omega' = weight(\ell')$. Choose a such that $a\omega\omega' \geq drop(\pi)$ and define $\pi' = \ell^{a\omega'} \cdot \pi \cdot (\ell')^{a\omega}$. We have $drop(\pi') \geq c$ and $weight(\pi') = weight(\ell) + weight(\pi) + weight(\ell') = weight(\pi)$. Hence by Proposition 5, π' has corresponding run $r : (q, c) \rightarrow^*_{\mathcal{A}} (q', c')$ in $(\mathcal{C}_{\mathcal{A}}, \rightarrow_{\mathcal{A}})$.

Before we state the main result of this section, we prove a proposition that helps us to structure runs.

Proposition 8. *Let $r : (q, c) \to^*_{\mathcal{A}} (q', c')$ be a run in $(\mathcal{C}_{\mathcal{A}}, \to_{\mathcal{A}})$ with the corresponding path π in $G_{\mathcal{A}} = (V, E, w)$.*

(i) If π does not contain any positive cycle then either f_π does not contain any positive cycles, or there is a path $\mu = \mu_1 \cdot \mu_2 \cdot \mu_3$ in $G_{\mathcal{A}}$ corresponding to a run $r' : (q, c) \to^*_{\mathcal{A}} (q', c')$ in $(\mathcal{C}_{\mathcal{A}}, \to_{\mathcal{A}})$ such that $|\mu_1| < |\pi|$ and μ_2 is a positive cycle.

(ii) If π does not contain any negative cycle then either f_π does not contain any negative cycles, or there is a path $\mu = \mu_1 \cdot \mu_2 \cdot \mu_3$ in $G_{\mathcal{A}}$ corresponding to a run $r' : (q, c) \to^*_{\mathcal{A}} (q', c')$ in $(\mathcal{C}_{\mathcal{A}}, \to_{\mathcal{A}})$ such that $|\mu_3| < |\pi|$ and μ_2 is a negative cycle.

Proof. (i) Suppose that f_π contains a positive cycle ℓ. Let $v \in V$ be the first vertex of ℓ that occurs in π and let $d \in \mathbf{N}$ be such that the configuration (v, d) is first reached by r. We claim that there is a positive cycle at v in $G_{\mathcal{A}}$ that corresponds to a run $(v, d) \to^*_{\mathcal{A}} (v, d')$ in $(\mathcal{C}_{\mathcal{A}}, \to_{\mathcal{A}})$ for some $d' > d$.
In case ℓ does not correspond to a such a run starting from (v, d) we argue as follows. Factor ℓ as $\ell = \mu_1 \cdot \mu_2$ with $\mu_1 : v \to^* w$, $\mu_2 : w \to^* v$ such that w is the node with the maximum decrement in ℓ, i.e., $\mathit{weight}(\mu_1) = \mathit{drop}(\ell)$ and whence $\mathit{weight}(\mu_1) < -d$. Since v is the first vertex of ℓ visited by π, w is visited by π sometime after the first visit of v. So there is a v-w path μ_3 in $G_{\mathcal{A}}$ such that $\mathit{weight}(\mu_3) \geq \mathit{drop}(\mu_3) \geq d > \mathit{weight}(\mu_1)$. Consider now the cycle $\ell' = \mu_3 \cdot \mu_2$. It follows that ℓ' is a positive cycle, since

$$\begin{aligned} \mathit{weight}(\ell') &= \mathit{weight}(\mu_3) + \mathit{weight}(\mu_2) \\ &\geq \mathit{weight}(\mu_1) + \mathit{weight}(\mu_2) \\ &= \mathit{weight}(\ell). \end{aligned}$$

Moreover, we have

$$\begin{aligned} \mathit{drop}(\ell') &\geq \mathit{drop}(\mu_3) + \mathit{drop}(\mu_2) \\ &\geq -d + 0 \\ &= -d. \end{aligned}$$

Hence, Proposition 5 implies that ℓ' corresponds to a run from $(v, d) \to^*_{\mathcal{A}} (v, d')$ in $(\mathcal{C}_{\mathcal{A}}, \to_{\mathcal{A}})$.
Next we observe that the first occurrence of v in π actually lies on a negative cycle in π. This is because π must visit v in π again, otherwise ℓ would not exist in f_π. By assumption all cycles in π are negative. Thus we can decompose r as $r_1 \cdot r_2 \cdot r_3$ with

$$r_1 : (q, c) \to (v, d);\ r_2 : (v, d) \to (v, d'');\ r_3 :\to (q', c')$$

such that there is a positive cycle ℓ' that corresponds to a run from (v, d), and with the cycle π_2 corresponding to r_2 being negative. Let π_3 be the path corresponding to r_3.
In order to define the required path $\mu = \mu_1 \cdot \mu_2 \cdot \mu_3$, we reuse an idea from the proof of Proposition 7(iii). Let $\omega_1 = \mathit{weight}(\ell')$ and $\omega_2 = \mathit{weight}(\pi_2)$. Then define $\mu_1 = \pi_1$, $\mu_2 = (\ell')^{\omega_1}$ and $\mu_3 = (\pi_2)^{\omega_2+1} \cdot \pi_3$. Clearly, $|\mu_1| < |\pi|$ and μ_2 is a positive cycle, as required. Since the positive cycle μ_2 is canceled out by the negative cycle $(\pi_2)^{\omega_2}$ and by applying Proposition 5 we have that μ corresponds to a run from (q, c) to (q', c') in $(\mathcal{C}_\mathcal{A}, \rightarrow_\mathcal{A})$.

(ii) This part is just the dual of part (i).

Proposition 9. *There is a run $(q, c) \rightarrow^*_\mathcal{A} (q', c')$ in $(\mathcal{C}_\mathcal{A}, \rightarrow_\mathcal{A})$ if and only if there exists a run $r : (q, c) \rightarrow^*_\mathcal{A} (q', c')$ in $(\mathcal{C}_\mathcal{A}, \rightarrow_\mathcal{A})$ with a corresponding path π in $G_\mathcal{A}$ that can be written as $\pi = \pi_1 \cdot \pi_2 \cdot \pi_3$ such that there are $c_1, c_2 \in \mathbf{N}$ such that*

- *if $|\pi_1| > 0$ then $(G_\mathcal{A}, f_{\pi_1}, c, c_1)$ fulfills the type-1 reachability, criteria*
- *if $|\pi_2| > 0$ then $(G_\mathcal{A}, f_{\pi_2}, c_1, c_2)$ fulfills the type-3 reachability criteria; and*
- *if $|\pi_3| > 0$ then $(G_\mathcal{A}, f_{\pi_3}, c_2, c')$ fulfills the type-2 reachability criteria.*

Proof. ($\Rightarrow$) If there is a run $r : (q, c) \rightarrow^*_\mathcal{A} (q', c')$ with a corresponding path π_1 such that f_{π_1} does not contain any positive cycles then f_{π_1} induces a unique vertex decomposition and hence (G, f, c, c') fulfills the type-1 reachability criteria. Hence $\pi = \pi_1$ the required path.

Otherwise, let $\mu = q_1 \dots q_n$ be a path in $G_\mathcal{A}$ corresponding to some run $r : (q, c) \rightarrow^*_\mathcal{A} (q', c')$. By repeatedly applying Proposition 8(i) to μ, we can obtain $\pi_1 \cdot \mu'_2 \cdot \mu'_3$ from μ such that $\pi_1 : q_1 \rightarrow^* q_i$, f_{π_1} does not contain any positive cycles and $\mu' = \mu'_2 \cdot \mu'_3$ is a q_i-q_n path with μ'_2 being a positive cycle. If $f_{\mu'}$ does not contain any negative cycles, by setting $\pi_3 = \mu'$ and $c_1 = c + \mathit{weight}(\pi_1)$, we have $(G_\mathcal{A}, f_{\pi_1}, c, c_1)$ and $(G_\mathcal{A}, f_{\pi_3}, c_1, c')$ are type-1 respectively type-2 reachability certificates. It follows that $\pi = \pi_1 \cdot \pi_3$ is the required path.

Otherwise, by repeatedly applying Proposition 8(ii) to μ', we can obtain $\mu''_1 \cdot \mu''_2 \cdot \pi_3$ from μ' such that $\pi_3 : q_j \rightarrow^* q_n$, f_{π_3} does not contain any negative cycles and $\pi_2 = \mu''_1 \cdot \mu''_2$ is a q_i-q_j path with μ''_2 being a negative cycle. Let $c_1 = c + \mathit{weight}(\pi_1)$ and $c_2 = c_1 + \mathit{weight}(\pi_2)$. It follows from Proposition 5 and 8 that μ'_2 and μ''_2 witness the existence of a positive respectively negative cycle with $\mathit{drop}(\mu'_2) \geq c_1$ and $\mathit{drop}(\mu''_2) \geq c_2$. Thus $(G_\mathcal{A}, f_{\pi_2}, c_1, c_2)$ fulfills the type-3 reachability criteria. Similarly as above, $(G_\mathcal{A}, f_{\pi_1}, c, c_1)$ and $(G_\mathcal{A}, f_{\pi_3}, c_2, c)$ fulfill the type-1 respectively type-2 reachability certificates and hence $\pi = \pi_1 \cdot \pi_2 \cdot \pi_3$ is the required path.

($\Leftarrow$) This direction follows by appropriately combining the statements from Proposition 7(i)–(iii).

3.3 Reachability Formulas

Based on the observations on the correspondence between paths in a graph and runs in an OCA, we subsequently show NP-membership of the reachability

problem for POCA by showing that this problem can be decided by checking for satisfiability of a polynomial size QFPAD formula.

To this end, we need to introduce the concept of *parametric weighted directed graphs (PWDG)*. Similar to a graph, a PWDG is a tuple $G = (V, E, P, w)$ with the only difference that the weight function can additionally map into the set of parameters P, i.e. $w : E \to \mathbf{Z} \cup \{+p, -p : p \in P\}$. All definitions involving graphs, such as path flows, are adopted in a straight forward way to the setting of PWDG. In particular, a zero-test free POCA $\mathcal{A}$ induces a corresponding PWDG $G_{\mathcal{A}}$, which is defined in the obvious way. Given an instantiation I, $G(I)$ denotes the graph obtained from instantiating the parameters. Throughout this section, we assume a fixed set of parameters $P = \{p_1, \ldots, p_k\}$. Instantiations are going to be represented in QFPAD formulas by variables $\boldsymbol{Z} = (z_1, \ldots, z_k)$. Given a QFPAD formula $\varphi(\boldsymbol{Z})$, we slightly abuse notation and denote by $\varphi[\boldsymbol{Z}/I]$ the formula $\varphi[z_1/I(p_1), \ldots, z_k/I(p_k)]$. Likewise, any assignment of n_i to z_i induces a corresponding instantiation.

Given a zero-test free POCA $\mathcal{A} = \langle Q, q_{in}, F, P, \Delta, \lambda \rangle$, control locations $q, q' \in Q$ and natural numbers $d, d' \in \mathbf{N}$, we subsequently provide a set of formulas $RF(G_{\mathcal{A}}, q, q')$ such that there is some $\varphi(\boldsymbol{Z}, c, c') \in RF(G_{\mathcal{A}}, q, q')$ such that $\varphi[c/d, c'/d']$ is satisfiable if and only if $(q, d) \to^*_{\mathcal{A}} (q', d')$. Each $\varphi(\boldsymbol{Z}, c, c')$ can be guessed in time polynomial in $|\mathcal{A}|$, which together with the fact that satisfiability in QFPAD is NP-complete concludes that reachability for zero-test free POCA is in NP.

It is not difficult to see that this actually implies that the general reachability problem for POCA is in NP. Suppose there is an instantiation I such that there is a run $r : (q, d) \to^*_{\mathcal{A}(I)} (q', d')$. We may assume with no loss of generality that each transition labeled with *zero* is traversed at most once in r. Formally, r can be written as $r : (q_1, d_1) \to^*_{\mathcal{A}(I)} (q'_1, d'_1) \to_{\mathcal{A}(I)} (q_2, d_2) \to^*_{\mathcal{A}(I)} (q'_2, d'_2) \to^*_{\mathcal{A}(I)} \ldots \to^*_{\mathcal{A}(I)} (q_n, d_n) \to^*_{\mathcal{A}(I)} (q'_n, d'_n)$, where $q_1 = q$, $q'_n = q'$, $d_1 = d$, $d'_n = d'$, $d'_i = 0$ and $d_j = 0$ for $1 \leq i < n$, $1 < j \leq n$, $1 \leq n \leq \#\Delta$. Hence, there are zero-test free POCA $\mathcal{A}_i$ such that $(q_i, d_i) \to^*_{\mathcal{A}_i(I)} (q'_i, d'_i)$ and by assumption formulas $\varphi_i(\boldsymbol{Z}, c_i, c'_i) \in RF(G_{\mathcal{A}_i}, q_i, q'_i)$ such that the conjunction $\bigwedge_{1 \leq i \leq n} \varphi_i[c_i/d_i, c'_i/d'_i]$ is satisfiable, assuming with no loss of generality that the φ_i only share variables from $\boldsymbol{Z}$ in common. Conversely, we can guess the order in which transitions labeled with *zero* are traversed in a run witnessing reachability, their induced zero-test free POCA $\mathcal{A}_i$, the formulas $\varphi_i(\boldsymbol{Z}, c_i, c'_i) \in RF(G_{\mathcal{A}_i}, q_i, q'_i)$ and check for satisfiability of $\bigwedge_{1 \leq i \leq n} \varphi_i[c_i/d_i, c'_i/d'_i]$ in order to decide a general reachability problem.

Let us introduce some additional abbreviations to QFPAD that we will be using in the following. First, we can easily extend QFPAD with the standard Boolean abbreviations implication ($\to$) and equivalence ($\leftrightarrow$), where $\varphi \to \psi$ abbreviates $\neg\varphi \vee \psi$ and $\varphi \leftrightarrow \psi$ stands for $\varphi \to \psi \wedge \psi \to \varphi$. Second, let A, B be linear polynomials in some vectors of variables $\boldsymbol{X}$. Let y be a fresh variable, we introduce equivalence and inequalities between polynomials. The following identities can be easily verified:

$$
\begin{aligned}
A = B &\iff A|B \wedge B|A \wedge A+1|B+1 \wedge B+1|A+1 \\
A < B &\iff A + y + 1 = B \\
A > B &\iff -A < -B \\
A \leq B &\iff A < B \vee A = B \\
A \geq B &\iff A > B \vee A = B.
\end{aligned}
$$

Notice that when applying negation to formulas of the form $A \sim B, \sim \in \{<, \leq, \geq, >\}$ we rewrite this formula with the complement of $\sim$, e.g., $\neg(A < B)$ is rewritten with $A \geq B$.

Restricted systems of quadratic Diophantine equations will be our key tool for checking the existence of instantiations such that a path flow with a certain weight exists. In particular, in the setting of PWDG Equation (6) allows for describing the weight of a path flow in terms of a quadratic equation.

Definition 8. *Let $\boldsymbol{X} = (x_1, \ldots, x_m)$ and $\boldsymbol{Y} = (y_1, \ldots, y_n)$ be vectors of disjoint integer variables, and A_i, B_i be linear polynomials in $\boldsymbol{X}$, $1 \leq i \leq n$. A* restricted system S of quadratic Diophantine equations *is of the form*

$$
\begin{aligned}
y_1 A_1 + B_1 &= 0 \\
y_2 A_2 + B_2 &= 0 \\
&\vdots \\
y_n A_n + B_n &= 0.
\end{aligned}
$$

The system S has a solution if and only if there are $z'_i, z_j \in \mathbf{N}, 1 \leq i \leq m, 1 \leq j \leq n$, such that $z'_j A_j[x_1/z_1, \ldots, x_m/z_m] + B_j[x_1/z_1, \ldots, x_m/z_m] = 0, 1 \leq j \leq n$.

Lemma 1. *Let S be a restricted system of quadratic Diophantine equations. There exists a QFPAD formula φ of size polynomial in the size of S such that φ is satisfiable if and only if S has a solution.*

Proof. We use the terminology from Definition 8. Since each variable y_i only occurs *once* in each row, the statement follows by setting

$$
\varphi(\boldsymbol{X}) := \bigwedge_{1 \leq i \leq n} \left(A_i | B_i \wedge (A_i > 0 \leftrightarrow B_i > 0)\right).
$$

Notice that since $0|0$ and $\neg(0|z)$ for $z \in \mathbf{Z} \setminus \{0\}$ are tautologies in QFPAD, the A_i are allowed to be equivalent to the constant polynomial 0.

Remark 2. It immediately follows from a result by Ibarra and Dang [13] that generalising Definition 8 to allow the same variable y_i to appear in two separate quadratic polynomials leads to an undecidable problem.

Let $G = (V, E, P, w)$ be a PWDG. For every $(F', F_1, \ldots, F_n) \in CDS(G, s, t)$, the set of *flow instantiations* $FI(G, s, t)$ contains the equiv-satisfiable QFPAD formula $\varphi(\boldsymbol{Z}, c, c')$ of the following system $S_{s,t}$ of restricted quadratic Diophantine equations

$$\begin{aligned} weight(\pi') - x' &= 0 \\ y_1 weight(\ell_1) - x_1 &= 0 \\ &\vdots \\ y_n weight(\ell_n) - x_n &= 0 \\ x' + x_1 + \ldots + x_n - c' + c &= 0, \end{aligned}$$

where π' is the only s-t path in G/F' and ℓ_i is a cycle in G/F_i, $1 \leq i \leq n$.

Proposition 10. *Let $G = (V, E, P, w)$ be a PWDG and $s, t \in V$. For all instantiations I and $d, d' \in \mathbf{N}$, we have the following:*

(a) For all $\varphi(\boldsymbol{Z}, c, c') \in FI(G, s, t)$, if $\varphi[\boldsymbol{Z}/I, c/d, c'/d']$ is satisfiable then there exists an s-t path flow f in $G(I)$ with $weight(f) = d' - d$.
(b) If there exists an s-t path flow f in $G(I)$ with $weight(f) = d' - d$ then there exists some $\varphi(\boldsymbol{Z}, c, c') \in FI(G, F, s, t)$ such that $\varphi[\boldsymbol{Z}/I, c/d, c'/d']$ is satisfiable.

Proof. (a) Let $\varphi[\boldsymbol{Z}/I, c/d, c'/d']$ be satisfiable and derived from $(F', F_1, \ldots, F_n) \in CDS(G, s, t)$ with the respective path π' and cycles $\ell_1, \ldots, \ell_n$. Since $S_{s,t}$ has a solution, there are $y_1, \ldots, y_i$ such that $y' weight(\pi') + \sum_{1 \leq j \leq i} y_i weight(\ell_i) = d' - d$. Hence for $e \in E$, define $f'(e) = 1$ if $e \in F'$ and $f'(e) = 0$ otherwise, and $f_j(e) = y_j$ if $e \in F_i$ and $f_i(e) = 0$ otherwise. It follows from Proposition 4 that $f = f' + \sum_{1 \leq j \leq i} f_i$ is the required flow with $weight(f) = d' - d$.
(b) Let f be a flow with support f and $weight(f) = d' - d$ in $G(I)$. By Proposition 4, there exists an s-t-cycle decomposition $f', f_1, \ldots, f_n$ with supports $F', F_1, \ldots, F_i, 1 \leq i \leq \#E$ such that $weight(f) = weight(f = f' + \sum_{1 \leq j \leq i} f_i)$, Consequently, there are $y_1, \ldots, y_i$ such that $weight(f) = weight(f') + \sum_{1 \leq j \leq i} y_j weight(f_j) = d' - d$. Hence, there is $\varphi(\boldsymbol{Z}, c, c') \in FI(G, s, t)$ such that $\varphi[\boldsymbol{Z}/I, c/d, c'/d']$ is satisfiable.

We now define QFPAD formulas that are satisfiable whenever a graph does not contain any positive respectively negative cycles.

Proposition 11. *Let $G = (V, E, P, w)$ be a PWDG. There exist singleton sets $PC(G)$ and $NC(G)$ such that for any instantiation I,*

(i) for $\varphi^+(\boldsymbol{Z}) \in PC(G)$, $\varphi^+[\boldsymbol{Z}/I]$ is satisfiable if and only if $G(I)$ does not contain any negative cycle; and
(ii) for $\varphi^-(\boldsymbol{Z}) \in NC(G)$, $\varphi^-[\boldsymbol{Z}/I]$ is satisfiable if and only if $G(I)$ does not contain any positive cycle.

Proof. (i) In order to show the statement we provide a polynomial-time algorithm that checks for the non-existence of negative cycles in a weighted graph. By loop-unraveling we can then translate this algorithm into the required QFPAD formula.

Consider Algorithm 1, a variant of the celebrated Bellman-Ford algorithm, which is a polynomial-time algorithm that computes the shortest path in weighted graphs with the ability to detect negative cycles, see e.g. [6]. Given a weighted graph G, for each $v \in V$ Algorithm 1 works on variables $d_v^0, \ldots, d_v^{|V|-1}$ which are all assumed to be initialised with 0. The intention of the d_v^i is to store the minimal weight of a path to v in G of length at most i. Since we are exclusively interested in finding negative cycles, we have $d_v^i \leq 0$ and $d_v^i \leq d_v^{i+1}$ when the algorithm has terminated for all $v \in V, 0 \leq i < |V| - 1$. At the beginning, the algorithm loops $|V| - 1$ times and updates in each round the d_v^i for each $v \in V$. If there is a node $v' \in V$ and an incoming edge (v', v) such that $d_{v'}^{i-1} + w(v', v) < d_v^{i-1}$ then d_v^i is set to $d_{v'}^{i-1} + w(v', v)$ for some $v' \in V$. Otherwise d_v^i is set to d_v^{i-1}. In the last round of the loop, the algorithm checks whether there is a node $v \in V$ such that $d_v^{|V|-1}$ can be decreased. If this is the case then there exists a cycle whose weight sums up below zero and the algorithm returns false. Otherwise G does not a negative cycle and the algorithm eventually exits the loop and returns true.

It is now not hard to see that the following formula translates Algorithm 1 into a polynomial size Presburger formula with additional free variables $d_v^i, 0 \leq i < |V|, v \in V$ and P:

$$\begin{aligned}
\varphi^- := & \bigwedge_{v \in V} d_v^0 = 0 \wedge \\
& \wedge \bigwedge_{1 \leq i < |V|} \bigwedge_{v \in V} \bigwedge_{(v',v) \in E} \left(\bigwedge_{(v'',v) \in E} d_{v''}^{i-1} + w(v'', v) \geq d_{v'}^{i-1} + w(v', v) \right) \rightarrow \\
& \qquad \rightarrow ((d_{v'}^{i-1} + w(v', v) < d_v^{i-1} \rightarrow d_v^i = d_{v'}^{i-1} + w(v', v)) \wedge \\
& \qquad\quad \wedge ((d_{v'}^{i-1} + w(v', v) \geq d_v^{i-1} \rightarrow d_v^i = d_v^{i-1}))) \wedge \\
& \wedge \bigwedge_{v \in V} \bigwedge_{(v',v) \in E} d_{v'}^{|V|-1} + w(v', v) \geq d_v^{|V|-1}.
\end{aligned}$$

It follows that $\varphi[p_1/z_1, \ldots, p_k/z_k]$ is satisfiable only for values $z_1, \ldots, z_k \in \mathbf{Z}$ such that $I(p_i) = z_i, 1 \leq i \leq k$ and $G(I)$ does not contain any positive cycles.

(ii) This is just the dual of part (i).

Given a $G = (V, E, P, w)$, we are now going to introduce sets of QFPAD formulas $RC_1(G, s, t), RC_2(G, s, t)$ and $RC_3(G, s, t)$, $s, t \in V$. The set RC_1 will be used to decide the existence of path flows that fulfill the type-1 reachability criteria. Thus RC_1 contains formulas $\varphi(\boldsymbol{Z}, c, c')$ such that for any instantiation I and $d, d' \in \mathbf{N}$, if $\varphi[\boldsymbol{Z}/I, c/d, c'/d']$ is satisfiable then there is an s-t

Algorithm 1 Variant of the Bellman-Ford algorithm for checking that the weighted graph $G = (V, E, w)$ does not contain a negative cycle.

```
Input: G = (V, E, w)
  for i = 1 to |V| do
    for all v ∈ V do
      d_v^i := d_v^{i-1}
      for all (v', v) ∈ E do
        if d_{v'}^{i-1} + w(v', v) < d_v^{i-1} && i < |V| then
          d_v^i := d_{v'}^{i-1} + w(v', v)
        end if
        if d_{v'}^{i-1} + w(v', v) < d_v^{i-1} && i = |V| then
          return false
        end if
      end for
    end for
  end for
  return true
```

path flow f such that $(G(I), f, d, d')$ fulfills the type-1 reachability criteria. Conversely, if $(G(I), f, d, d')$ fulfills the type-1 reachability criteria then there is some $\varphi(\boldsymbol{Z}, c, c') \in RC_1(G, s, t)$ that is satisfiable. Similar conditions hold for the formulas in the sets RC_2 and RC_3.

For each $(F_0, s_0, s_1, \ldots, F_{m-1}, s_{m-1}, s_m) \in VDS(G, s, t)$ with $F := \bigcup_{0 \leq i < m} F_i$, each $\varphi_i(\boldsymbol{Z}, c_i, c_i') \in PF(G/F_i, s_i, s_{i+1}), 0 \leq i < m$ and $\varphi^- \in NC(G/F)$, we have $\varphi(\boldsymbol{Z}, c, c') \in RC_1(G, s, t)$ with

$$\varphi := \bigwedge_{0 \leq i < m} \varphi_i \wedge \varphi^- \wedge \bigwedge_{0 \leq j < m} \sum_{0 \leq i \leq j} c_i' - c_i \geq -c \wedge \sum_{0 \leq i < m} c_i' - c_i = c' - c. \quad (7)$$

The set $RC_2(G, s, t)$ is defined analogously for capturing type-2 reachability criteria.

Last, we define the set of QFPAD formulas for capturing type-3 reachability criteria. Given $G = (V, E, P, w)$ and $s, t \in V$, for every $\varphi_f(\boldsymbol{Z}, c_f, c_f') \in PF(G, s, t)$, for every $(\pi_1, \pi_2, \pi_3) \in CC(G, s)$ and $(\pi_1', \pi_2', \pi_3') \in CC(G, t)$ such that $\pi_1 = v_0 v_1 \ldots v_{l_1}$, $\pi_2 = v_{l_1} \ldots v_l$, $\pi_1' = v_0' v_1' \ldots v_{l_1'}$ and $\pi_2 = v_{l_1'}' \ldots v_{l'}'$, there is $\varphi(\boldsymbol{Z}, c, c') \in RC_3(G, s, t)$ with

$$\varphi := \bigwedge_{0 \leq j \leq l} \sum_{0 \leq i < j} w(v_i v_{i+1}) \geq -c \wedge \sum_{l_1 \leq i < l} w(v_i v_{i+1}) > 0 \wedge \quad (8)$$

$$\wedge \bigwedge_{0 \leq j \leq l'} \sum_{0 \leq i < j} w(v_i' v_{i+1}') \geq -c \wedge \sum_{l_1' \leq i < l'} w(v_i' v_{i+1}') < 0 \wedge \quad (9)$$

$$\wedge \varphi_f \wedge c_f = c \wedge c_f' = c'. \quad (10)$$

Proposition 12. *Let $G = (V, E, P, w)$ be a PWDG. Let $s, t \in V, d, d' \in \mathbf{N}$ and I be an instantiation.*

(i) *(a)* *For all $\varphi(\boldsymbol{Z}, c, c') \in RC_1(G, s, t)$ such that $\varphi[\boldsymbol{Z}/I, c/d, c'/d']$ is satisfiable there is an s-t path flow f such that $(G(I), f, d, d')$ fulfills the type-1 reachability criteria.*

(b) *If there is an s-t path flow f such that $(G(I), f, d, d')$ fulfills the type-1 reachability criteria then there is some $\varphi(\boldsymbol{Z}, c, c') \in RC_1(G, s, t)$ such that $\varphi[\boldsymbol{Z}/I, c/d, c'/d']$ is satisfiable.*

(ii) *(a)* *For all $\varphi(\boldsymbol{Z}, c, c') \in RC_2(G, s, t)$ such that $\varphi[\boldsymbol{Z}/I, c/d, c'/d']$ is satisfiable there is an s-t path flow f such that $(G(I), f, d, d')$ fulfills the type-2 reachability criteria.*

(b) *If there is an s-t path flow f such that $(G(I), f, d, d')$ fulfills the type-2 reachability criteria then there is some $\varphi(\boldsymbol{Z}, c, c') \in RC_2(G, s, t)$ such that $\varphi[\boldsymbol{Z}/I, c/d, c'/d']$ is satisfiable*

(iii) *(a)* *For all $\varphi(\boldsymbol{Z}, c, c') \in RC_3(G, s, t)$ such that $\varphi[\boldsymbol{Z}/I, c/d, c'/d']$ is satisfiable there is an s-t path flow f such that $(G(I), f, d, d')$ fulfills the type-3 reachability criteria.*

(b) *If there there is some $\varphi(\boldsymbol{Z}, c, c') \in RC_3(G, s, t)$ such that $\varphi[\boldsymbol{Z}/I, c/d, c'/d']$ is satisfiable then there is some $\varphi(\boldsymbol{Z}, c, c') \in RC_3(G, s, t)$ such that $\varphi[\boldsymbol{Z}/I, c/d, c'/d']$ is satisfiable.*

Proof. (i) (a) Let some $\varphi[\boldsymbol{Z}/I, c/d, c'/d']$ be satisfiable. Then φ is derived from some vertex decomposition $(F_0, s_0, s_1, \ldots, F_{m-1}, s_{m-1}, s_m) \in VDS(G, s, t)$. Since each φ_i in (7) is satisfied, by Proposition 10(i) there exist s_i-s_{i+1} path flows f_i with weight $c'_i - c_i, 0 \leq i < m$. Moreover, since φ^- is satisfied, there is no positive cycle in $G(I)/F$. It is also ensured that the sum of the weights over all sums of path flows in the relevant order does not go below d and that that their overall weight is $d' - d$. Hence, the type-1 reachability criteria are fulfilled for $(G(I), f, d, d')$, where $f = \sum_{0 \leq i \leq m} f_i$.

(b) Let f be a path flow with support F such that $(G(I), f, d, d')$ fulfills the type-1 reachability criteria. Thus, let $(F_0, s_0, s_1, \ldots, F_{m-1}, s_{m-1}, s_m) \in VDS(G, s, t)$ be the supports of the corresponding vertex decomposition. Since there are s_i-s_{i+1} path flows f_i in G/F_i with $\mathit{weight}(f_i) = d'_i - d_i$, by Proposition 10(ii) there are $\varphi_i \in PF(G/F_i, s_i, s_{i+1})$ such that each $\varphi(G/F_i, s_{m-1}, s_m)$ is satisfiable, $0 \leq i < m$. Moreover, φ^- is also satisfiable, since $G(I)/F$ does not contain any positive cycle. The remaining type-1 reachability criteria enforce that the remaining conjuncts from Equation (7) are also satisfied, hence there exists some $\varphi[\boldsymbol{Z}/I, c/d, d']$ that is satisfiable.

(ii) This is just the dual of part (i).

(iii) (a) Let some $\varphi[\boldsymbol{Z}/I, c/d, d']$ be satisfiable. Then φ is derived from some $(\pi_1, \pi_2, \pi_3) \in CC(G, s)$ and $(\pi'_1, \pi'_2, \pi'_3) \in CC(G, t)$. The conjuncts from (8) together with Proposition 6(i) ensure that there is a positive cycle ℓ^+ at s in $G(I)$ with $\mathit{drop}(\ell^+) \geq d$. Likewise, the conjuncts from (9) together with Proposition 6(ii) ensure that there is a negative cycle ℓ^- at t in $G(I)$ with $\mathit{drop}(\ell^-) \geq c'$. Last, the conjunct from (10) together with Proposition 10(i) ensures that there is an s-t path flow

in $G(I)$ with weight $d' - d$. Hence $(G(I), f, d, d')$ fulfills the type-3 reachability criteria.

(b) Let f be a path flow such that $(G(I), f, d, d')$ be such that it fulfills the type-3 reachability criteria. Then there is a positive cycle at s and a negative cycle at t. We derive from Proposition 6 that there exists some corresponding $(\pi_1, \pi_2, \pi_3) \in CC(G, s)$ and $(\pi_1', \pi_2', \pi_3') \in CC(G, t)$ such that the conjuncts from (8) and (9) are fulfilled. Moreover, Proposition 10(ii) implies that there is some $\varphi_f(\boldsymbol{Z}, c_f, c_f') \in PF(G, s, t)$ such that $\varphi_f[\boldsymbol{Z}/I, c_f/d, c_f'/d']$ and all the remaining conjuncts from (10) are satisfied. Hence the required $\varphi(\boldsymbol{Z}, c, c') \in RC_3(G, s, t)$ exists.

We finally reached the point where we can define the set $RF(G, s, t)$ of *reachability formulas*. The previously defined sets RC_1, RC_2 and RC_3 allow us to check for the existence of path flows that fulfill the type-1, type-2 respectively type-3 reachability criteria. Proposition **??** allows us to decide a reachability problem by just considering at most three path flows that fulfill the type-i reachability criteria. Hence, the set $RF(G, s, t)$ consists of conjunction of formulas from RC_1, RC_2 and RC_3.

In the following, let us assume without loss of generality that all formulas from RC_1, RC_2 and RC_3 only share the variables $z_1, \ldots, z_k$. Define $RF(G, s, t)$ to be a finite set of QFPAD formulas $\varphi(\boldsymbol{Z}, c, c')$ such that

$$
\begin{aligned}
RF(G,s,t) := \{&c_1 = c \wedge c_1' = c' \wedge \varphi_1 : \varphi_1(\boldsymbol{Z}, c_1, c_1') \in RC_1(G, s, t)\} \\
&\cup \{c_1 = c \wedge c_1' = c_2 \wedge c_2' = c' \wedge \varphi_1 \wedge \varphi_2 : \\
&\quad \varphi_1(\boldsymbol{Z}, c_1, c_1') \in RC_1(G, s, s'), \varphi_2(\boldsymbol{Z}, c_2, c_2') \in RC_2(G, s', t), s' \in V\} \\
&\cup \{c_1 = c \wedge c_1' = c_2 \wedge c_2' = c_3 \wedge c_3' = c' \wedge \varphi_1 \wedge \varphi_2 \wedge \varphi_3 : \\
&\quad \varphi_1(\boldsymbol{Z}, c_1, c_1') \in RC_1(G, s, s'), \varphi_2(\boldsymbol{Z}, c_2, c_2') \in RC_3(G, s', t'), \\
&\quad\ \varphi_3(\boldsymbol{Z}, c_3, c_3') \in RC_2(G, t', t), s', t' \in V\}.
\end{aligned}
$$

Proposition 13. *Let $\mathcal{A} = \langle Q, q_{in}, F, P, \Delta, \lambda \rangle$ be a zero-test free* POCA *with the corresponding PWDG $G_{\mathcal{A}}$. Let $q, q' \in Q$, $d, d \in \mathbf{N}$ and I be an instantiation.*

(a) For all $\varphi(\boldsymbol{Z}, c, c') \in RF(G, q, q')$ such that $\varphi[\boldsymbol{Z}/I, c/d, c'/d']$ is satisfiable we have $(q, d) \rightarrow^*_{\mathcal{A}(I)} (q', d')$ in $(\mathcal{C}_{\mathcal{A}(I)}, \rightarrow_{\mathcal{A}(I)})$.

(b) If $(q, d) \rightarrow^*_{\mathcal{A}(I)} (q', d')$ in $(\mathcal{C}_{\mathcal{A}(I)}, \rightarrow_{\mathcal{A}(I)})$ then there is $\varphi(\boldsymbol{Z}, c, c') \in RF(G, q, q')$ such that $\varphi[\boldsymbol{Z}/I, c/d, c'/d']$ is satisfiable.

Proof. (a) We only consider the most general case $\varphi(\boldsymbol{Z}, c, c') = c_1 = c \wedge c_1' = c_2 \wedge c_2' = c_3 \wedge c_3' = c' \wedge \varphi_1 \wedge \varphi_2 \wedge \varphi_3$, the other cases follow in a similar fashion. Since φ is satisfiable, there are $d_1, d_2 \in \mathbf{N}$ and $q_1, q_2 \in Q$ such that $\varphi_1[\boldsymbol{Z}/I, c_1/d, c_1'/d_1]$, $\varphi_2[\boldsymbol{Z}/I, c_2/d_1, c_2'/d_2]$ and $\varphi_3[\boldsymbol{Z}/I, c_3/d_2, c_3'/d']$ are satisfiable for $\varphi_1(\boldsymbol{Z}, c_1, c_1') \in RC_1(G, q, q_1)$, $\varphi_2(\boldsymbol{Z}, c_2, c_2') \in RC_3(G, q_1, q_2)$ and $\varphi_3(\boldsymbol{Z}, c_3, c_3') \in RC_2(G, q_2, q')$. It follows from Proposition 12 that there is a q-q_1 path flow f_1 such that $(G(I), f_1, d, d_1)$ fulfills the type-1 reachability

criteria, and hence by Proposition 9(i) $(q,d) \rightarrow^*_{\mathcal{A}(I)} (q_1,d_1)$. By the same argumentation, we have $(q_1,d_1) \rightarrow^*_{\mathcal{A}(I)} (q_2,d_2)$ and $(q_2,d_2) \rightarrow^*_{\mathcal{A}(I)} (q',d')$, whence $(q,d) \rightarrow^*_{\mathcal{A}(I)} (q',d')$.

(b) By Proposition 9, in the most general case there exist a q-q_1 path flow f_1, a q_1-q_2 path flow f_2, a q_2-q' path flow f_3 and $d_1, d_2 \in \mathbf{N}$ such that $(G_\mathcal{A}, f_1, d, d_1)$, $(G_\mathcal{A}, f_2, d_1, d_2)$ and $(G_\mathcal{A}, f_3, d_2, d')$ fulfill the type-1, type-2 respectively type-3 reachability criteria. By Proposition 12, there are formulas $\varphi_1(\boldsymbol{Z}, c_1, c_1') \in RC_1(G_\mathcal{A}, q, q_1)$, $\varphi_2(\boldsymbol{Z}, c_2, c_2') \in RC_3(G_\mathcal{A}, q_1, q_2)$ and $\varphi_3(\boldsymbol{Z}, q_3, c_3') \in RC_2(G_\mathcal{A}, q_2, q')$ such that $\varphi_1[\boldsymbol{Z}/I, c_1/d, c_1'/d_1]$, $\varphi_1[\boldsymbol{Z}/I, c_1/d_1, c_1'/d_2]$ and $\varphi_1[\boldsymbol{Z}/I, c_1/d_2, c_1'/d']$ are satisfiable. Hence, the formula $\varphi(\boldsymbol{Z}, c, c') := c_1 = c \wedge c_1' = c_2 \wedge c_2' = c_3 \wedge c_3' = c' \wedge \varphi_1 \wedge \varphi_2 \wedge \varphi_3$, which obviously is contained in $RF(G, q, q')$, is satisfiable.

It is easily verified that any formula from $\varphi \in RF(G, s, t)$ can be guessed in polynomial time in $|G|$, since all the formulas from FI, PC, NC, RC_1, RC_2 and RC_3 can be guessed respectively constructed in time polynomial in $|G|$. Recalling that checking for satisfiability in QFPAD is in NP, we have thus shown that the reachability problem for POCA without zero tests is in NP. As discussed at the beginning of this section, this shows that the general POCA reachability problem is in NP. Moreover, in Proposition 3 we proved that the POCA reachability problem is NP-hard, hence our main theorem follows.

Theorem 2. *The reachability problem for parametric one-counter automata is* NP*-complete.*

Remark 3. In Section 2, we defined an instantiation to be a function that maps the set of parameters to the set of positive integers. It is now easily seen that this definition can be generalised to allow for instantiations mapping the set of parameters to the whole set of integers without changing the complexity of the reachability problem. Given a POCA $\mathcal{A}$, by guessing beforehand the signs of each parameter and replacing p with $-p$ respectively $-p$ with p at the transitions of $\mathcal{A}$ if the sign of p is guessed to be negative, this generalised reachability problem reduces to the original problem.

3.4 Emptiness with Büchi Acceptance Condition

In the literature, Büchi automata have been introduced for the specification, modeling and reasoning about non-terminating systems, see e.g. [22]. A Büchi automaton is defined in a similar way to a finite state automaton, but its definition of emptiness differs. A Büchi automaton $\mathcal{A}$ is empty if there is an infinite run r of $\mathcal{A}$ such that there is a state from the set of final states that occurs infinitely often in r. This condition is known as Büchi acceptance condition. The concept of Büchi acceptance condition can be introduced in a straight-forward way to the setting of (parametric) one-counter automata. We show in this section that checking emptiness for parametric one-counter automata with Büchi acceptance condition is CO-NP-complete. The decidability of this problem has independently been established by Demri and Sangnier in [10].

Let $\mathcal{A} = \langle Q, q_{in}, F, \Delta, \lambda \rangle$ be an OCA. Given a run r in $(\mathcal{C}_\mathcal{A}, \rightarrow_\mathcal{A})$, $r(i)$ is defined to be the i-th configuration of r, $1 \leq i \leq |r| + 1$. The set $inf(r) \subseteq Q$ of *control locations occurring infinitely often in* r is defined as $inf(r) := \{q \in Q : \#\{i \in \mathbf{N} : r(i) = (q, c), c \in \mathbf{N}\} \text{ is infinite}\}$.

Definition 9. *A* POCA $\mathcal{A} = \langle Q, q_{in}, F, P, \Delta, \lambda \rangle$ *is* empty with respect to Büchi acceptance condition *if* $inf(r) \cap F = \emptyset$ *for all instantiations* I *and all runs* r *in* $(\mathcal{C}_{\mathcal{A}(I)}, \rightarrow_{\mathcal{A}(I)})$ *with* $r(1) = (q_{in}, 0)$.

In the setting of Büchi automata, non-emptiness can be decided by finding a strongly connected component in the graph underlying the automaton that contains a final location and is reachable from the initial location. We can modify this approach for deciding non-emptiness for POCA with Büchi acceptance condition. Given a POCA $\mathcal{A}$, we aim for finding an instantiation I such that we can find a cycle reachable from the initial configuration between configurations (q, c) and (q, c') in $(\mathcal{C}_{\mathcal{A}(I)}, \rightarrow_{\mathcal{A}(I)})$ such that $q \in F$ and either there is no zero-test between (q, c) and (q, c') and the counter value increases, or there is a zero-test and the counter value remains the same. This idea is formalised in the following proposition.

Proposition 14. *A* POCA $\mathcal{A} = \langle Q, q_{in}, F, P, \Delta, \lambda \rangle$ *is not empty with respect to Büchi acceptance condition if and only if there are an instantiation* I, $q \in F$ *and* $c \in \mathbf{N}$ *such that*

(i) there is $c' \in \mathbf{N}$ *with* $c' \geq c$ *and there are runs* $r_1 : (q_{in}, 0) \rightarrow^* (q, c)$ *and* $r_2 : (q, c) \rightarrow^* (q, c')$ *in* $(\mathcal{C}_{\mathcal{A}(I)}, \rightarrow_{\mathcal{A}(I)})$ *and* r_2 *is zero-test free; or*

(ii) there is $q' \in Q$ *and there are runs* $r_1 : (q_{in}, 0) \rightarrow^* (q', 0)$, $r_2 : (q', 0) \rightarrow^* (q, c)$ *and* $r_3 : (q, c) \rightarrow^* (q', 0)$ *in* $(\mathcal{C}_{\mathcal{A}(I)}, \rightarrow_{\mathcal{A}(I)})$.

Proof. ($\Rightarrow$) Suppose I is an instantiation such that there is a run r in $(\mathcal{C}_{\mathcal{A}(I)}, \rightarrow_{\mathcal{A}(I)})$ with $inf(r) \cap F \neq \emptyset$. Let $q \in inf(r) \cap F$. If r can be partitioned into $r = r_1 \cdot r_2 \cdot r_3$ such that $r_2 : (q, c) \rightarrow^*_{\mathcal{A}(I)} r_2' \cdot (q, c')$, r_2 is zero-test free and $c' \geq c$ we are done. Otherwise there is some $q' \in inf(r)$ such that $(q', 0)$ occurs infinitely often in r, since some zero test is performed infinitely often in r. Hence there is some $c \in \mathbf{N}$ such that r can be decomposed as $r = r_1 \cdot r_2 \cdot r_3 \cdot r_4$ such that $r_1 : (q_{in}, 0) \rightarrow^*_{\mathcal{A}(I)} (q', 0)$, $r_2 : (q', 0) \rightarrow^*_{\mathcal{A}(I)} (q, c)$ and $r_3 : (q, c) \rightarrow^*_{\mathcal{A}(I)} (q', 0)$.

($\Leftarrow$) Case (i): Suppose there are an instantiation I, $q \in F$, $c, c' \in \mathbf{N}$ with $c' \geq c$ and runs $r_1 : (q_{in}, 0) \rightarrow^* (q, c)$ and $r_2 : (q, c) \rightarrow^* (q, c')$ in $(\mathcal{C}_{\mathcal{A}(I)}, \rightarrow_{\mathcal{A}(I)})$ with r_2 being zero-test free. Using the latter fact, we have $(q, c+d) \rightarrow^* (q, c'+d)$ in $(\mathcal{C}_{\mathcal{A}(I)}, \rightarrow_{\mathcal{A}(I)})$ for all $d \in \mathbf{N}$. It follows that $r = r_1 \cdot r_2 \cdot (r_2 + (c' - c)) \cdot (r_2 + 2(c' - c)) \cdot (r_2 + 3(c' - c)) \ldots$ is a run witnessing non-emptiness.

Case (ii): It obviously follows that $r = r_1 \cdot r_2 \cdot r_3 \cdot r_2 \cdot r_3 \cdot r_2 \ldots$ is a run in $(\mathcal{C}_{\mathcal{A}(I)}, \rightarrow_{\mathcal{A}(I)})$ witnessing non-emptiness.

From this proposition we can derive the main result of this section. We will omit the proof since it is just a straight forward combination of Proposition 14 together with the proof of Theorem 2.

Theorem 3. *The emptiness problem for parametric one-counter automata with respect to Büchi acceptance condition is* co-NP*-complete.*

4 Conclusion

In this paper, we have considered the reachability problem for parametric counter automata. By previous results on non-parametric counter automata, this problem is undecidable in general. We have shown that—in contrast to the non-parametric case—the reachability problem remains undecidable in general even if we ban zero tests. However, for the sub-class of parametric one-counter automata we have shown that the reachability problem is NP-complete by showing that it is inter-reducible to quantifier free Presburger arithmetic with divisibility. Based on our result on reachability, we have shown that deciding emptiness for parametric one-counter automata with Büchi acceptance condition is co-NP-complete.

An interesting aspect for future work could be the investigation of the effect of introducing additional operations on the counter such as non-equality tests on the decidability and complexity of the reachability problem. According to [10], this could give new decidability respectively undecidability results for freeze LTL model checking of one-counter automata. It is however unlikely that the techniques we used in this paper can be adopted to this setting in a straightforward way. One of our main technical tools, lifting up paths by pumping of positive and negative cycles, does not seem to have a correspondent when non-equality test with parameters are allowed.

Acknowledgments. We would like to thank Leonard Lipshitz for making reference [17] available to us.

References

1. R. Alur, T.A. Henzinger, and M. Y. Vardi. Parametric real-time reasoning. In *Proceedings of the 25th Symposium on Theory of Computing (STOC)*, pages 592–601. ACM, 1993.
2. A. Bouajjani, M. Bozga, P. Habermehl, R. Iosif, P. Moro, and T. Vojnar. Programs with lists are counter automata. In *CAV*, volume 4144 of *LNCS*. Springer, 2006.
3. M. Bozga, R. Iosif, and Y. Lakhnech. Flat parametric counter automata. In *Proc. ICALP'06*, volume 4052 of *LNCS*. Springer, 2006.
4. J. R. Büchi. Regular canonical systems. *Archive for Mathematical Logic*, 6(3-4):91, April 1964.
5. H. Comon and Y. Jurski. Multiple counters automata, safety analysis and presburger arithmetic. In *Proc. CAV'98*, volume 1427 of *LNCS*. Springer, 1998.
6. T. Cormen, C. Leiserson, and R. Rivest. *Introduction to algorithms*. MIT Press and McGraw-Hill, 1990.
7. S. Demri. *Logiques pour la spécification et vérification*. Mémoire d'habilitation, Université Paris 7, 2007.
8. S. Demri and R. Gascon. The effects of bounding syntactic resources on Presburger LTL. In *Proc. TIME'07*. IEEE Computer Society Press, 2007.
9. Stéphane Demri and Régis Gascon. The effects of bounding syntactic resources on Presburger LTL. *Journal of Logic and Computation*, 2009. To appear.
10. Stéphane Demri and Arnaud Sangnier. When model-checking freeze LTL over counter machines becomes decidable. In C.-H. Luke Ong, editor, *Proceedings of the*

13th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'10), volume 6014 of *Lecture Notes in Computer Science*, Paphos, Cyprus, March 2010. Springer. To appear.
11. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.
12. J. E. Hopcroft and J. Pansiot. On the reachability problem for 5-dimensional vector addition systems. Technical report, Ithaca, NY, USA, 1976.
13. O. H. Ibarra and Z. Dang. On two-way finite automata with monotonic counters and quadratic diophantine equations. *Theoretical Computer Science*, 312(2-3):359–378, 2004.
14. O. H. Ibarra, T. Jiang, N. Trân, and H. Wang. New decidability results concerning two-way counter machines and applications. In *ICALP*, volume 700 of *LNCS*. Springer, 1993.
15. J. Leroux and G. Sutre. Flat counter automata almost everywhere! In *Proc. ATVA'05*, volume 3707 of *LNCS*. Springer, 2005.
16. L. Lipshitz. The diophantine problem for addition and divisibility. *Transaction of the American Mathematical Society*, 235:271–283, 1976.
17. L. Lipshitz. Some remarks on the diophantine problem for addition and divisibility. In *Proceedings of the Model Theory Meeting*, volume 33, pages 41–52, 1981.
18. R. Lipton. The reachability problem requires exponential space. Technical report, New Haven, CT, USA, 1975.
19. E. W. Mayr. An algorithm for the general petri net reachability problem. In *Proc. STOC'81*, pages 238–246, New York, NY, USA, 1981. ACM.
20. M. Minsky. Recursive unsolvability of post's problem of "tag" and other topics in theory of turing machines. *Annals of Mathematics*, 74(3), 1961.
21. Wojciech Plandowski and Wojciech Rytter. Complexity of language recognition problems for compressed words. In *Jewels are Forever, Contributions on Theoretical Computer Science in Honor of Arto Salomaa*, pages 262–272, London, UK, 1999. Springer-Verlag.
22. Wolfgang Thomas. Automata on infinite objects. pages 133–191, 1990.
23. G. Xie, Z. Dang, and O. H. Ibarra. A solvable class of quadratic diophantine equations with applications to verification of infinite-state systems. In *Proc. ICALP 2003*, volume 2719 of *LNCS*. Springer, 2003.