Computing Linear Arithmetic Representation of Reachability Relation of One-counter Automata[⋆]

Xie Li[1,3], Taolue Chen[2], Zhilin Wu[1], and Mingji Xia[1]

[1] State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences, China
[2] Department of Computer Science, University of Surrey, UK
[3] University of Chinese Academy of Sciences, China

Abstract. One-counter automata (OCA) are a well-studied automata model that extends finite-state automata with one counter. The reachability problem of OCA was shown to be NP-complete when the integers in the OCA are encoded in binary. In this paper, we study the problem of computing the reachability relation of OCA. We show that, for each OCA, an existential Presburger arithmetic (EPA) formula of polynomial size can be computed in polynomial time to represent its reachability relation. This yields a polynomial-time reduction from the reachability problem of OCA to the satisfiability problem of EPA, enabling its solution via off-the-shelf SMT solvers. We implement the algorithm and provide the first tool OCAReach for the reachability problem of OCA. The experimental results demonstrate the efficacy of our approach.

1 Introduction

Counter automata have been extensively studied in computer science and have found numerous applications, notably in formal verification. Some examples include verification of programs with lists [7] and recursive or multi-threaded programs [22], XML query validation [8], parameterized hardware verification [29], and decision procedures for separation logics with data [30], to name a few. Historically, counter automata were introduced by Minsky as a formal model of computation. It is well-known that two counters are already sufficient for counter automata to simulate Turing machines, rendering almost all decision problems about them undecidable. In particular, this includes the reachability problem, arguably the most fundamental problem in verification.

To tame the undecidability, numerous restrictions on counter automata have been proposed, which were the subject of thorough investigation in the past 40

[⋆] This work is supported by Guangdong Science and Technology Department grant (No. 2018B010107004), Overseas Grant (KFKT2018A16) from the State Key Laboratory of Novel Software Technology, Nanjing University, China, the NSFC grants (No. 61872340), Natural Science Foundation of Guangdong Province, China (No. 2019A1515011689), the Open Project of Shanghai Key Laboratory of Trustworthy Computing (No. 07dz22304201601), and the INRIA-CAS joint research project VIP.

years. These restrictions include, for instance, the types of allowable tests on the counters (e.g., in Petri nets zero tests are disallowed), the set of paths under consideration (e.g., reversal boundedness [23]), the underlying structure of the automaton (e.g., flatness [27]). Probably the simplest restriction is to allow only one counter, giving rise to one-counter automata (OCA). We are primarily interested in the reachability problem of OCA. From a certain perspective, this is simple since OCA can be considered a special case of pushdown automata where the stack alphabet is a singleton. Indeed, Lafourcade et al. [26] showed that reachability in OCA in NL-compete, namely, it is no harder than the reachability in directed graphs. However, this result must be stated with caveat that it assumes that the updates in OCA are encoded in *unary*. On the contrary, we note that these updates involve integers which are most naturally encoded in *binary*. When this encoding is adopted, the NL-completeness does not hold any more, and it has been shown [20] that the reachability problem becomes NP-complete. Technically, for an OCA $\mathcal{A}$ the reachability problem is to decide, when given two configurations (q, n) and (q', n'), whether there exists a run of $\mathcal{A}$ from the configuration (q, n) to (q', n'). Note that in OCA all the counter values along the path must be nonnegative, which is the main source of the complication.

The formulation of reachability as a decision problem may not be sufficient for verification purposes from a practical perspective. Instead, one needs a characterization of the *reachability relation*, viz. the relation $R_{\mathcal{A},q,q'}$ comprising the pairs (n, n') of natural numbers such that there exists a run of $\mathcal{A}$ from (q, n) to (q', n'). Such a characterization turns out be possible in the existential fragment of Presburger arithmetic (EPA). That is to say, one can construct an EPA formula $\psi(x, y)$ such that $(n, n') \in R_{\mathcal{A},q,q'}$ if and only if $\psi(n, n')$ holds. Such a construction is important for at least two reasons: (1) one can feed the generated formula to, e.g., an off-the-shelf SMT solver to facilitate the reachability checking, especially when it is required as part of the decision procedure as in [30]; (2) it entails the NP membership of the reachability problem, since it is well-known that the satisfiability of EPA is NP-complete. Indeed, Haase [18] has shown the existence of such a formula. He gave an algorithm to generate an EPA formula ψ from the OCA and a pair of states. However, the algorithm therein runs in *nondeterministic* polynomial time. Whilst this may be sufficient for the purpose (2), it is not amenable to the purpose (1), because one needs to "guess" an EPA formula, rendering the algorithm implementation-unfriendly and inefficient.

In this paper, we provide a *deterministic* polynomial-time algorithm to construct an EPA formula to characterize the reachability relation in OCA, which enables us to utilize the off-the-shelf SMT solvers (e.g., Z3) to decide the reachability problem of OCA. The main idea is to utilize the existential quantifiers and arithmetic operations available in EPA to encode the nondeterministic guessing of the reachability certificates in [18]. For example, to account for the existence of a simple path, we introduce existentially quantified integer variables to index the edges along the path and specify that the indices of the edges are mutually distinct, and for any two edges sharing a common vertex, their indices must be consecutive. Moreover, we show that even more involved graph-theoretical concepts (e.g., edge decompositions and positive cycle templates [18]), can still be encoded by polynomial-sized EPA formulas. The new encoding yields a more direct, conceptually simpler approach to obtain an EPA formula for the reachability relation of OCA. As a proof-of-concept, we implement the algorithm in a tool OCAReach, which, to the best of our knowledge, is the first tool that is able to decide the reachability problem of OCA. We test OCAReach on both handcrafted and random generated benchmarks. The experimental results demonstrate the potential of OCAReach to be used in solving practical verification problems related to OCA.

Related Work. There is a large body of theoretical work on OCA and its variants, a survey of which is out of the scope of the current paper. Related to verification, Demri and Gascon investigated the problem of model checking an extension of LTL against OCA [11]. Moreover, model checking CTL and its fragments against OCA was also studied [17, 15, 16]. The similarity and bisimilarity problem of OCA and its variants have also been considered in [1, 24, 25, 5, 4], to name a few.

There have also been some verification tools for counter systems. For instance, the FAST tool [2] targets flattable counter systems, whose behavior can be captured by flat path schemes, i.e., concatenations of paths and simple cycles such that no two cycles share a vertex. If a counter system is flattable, then its reachability relation can be easily captured by an EPA formula. While zero-test free OCA are known to be flattable, the resulting path schemes are of exponential length [3]. Hence, EPA formulas of polynomial size appear to be difficult to be generated to capture the reachability relation via flattening. We instead utilize the polynomial-size reachability certificate [18], which is more involved than the flat path schemes, to construct a polynomial-size EPA formula.

An automata model closely related to counter automata is timed automata (TA), which equip finite-state automata with real-valued clocks rather than integer-valued counters. The relationship between reachability problems of TA and bounded counter automata (where counters take values from an arbitrary but fixed finite interval over the natural numbers) was established [21]. The reachability problem of TA is known to be PSPACE-complete, even when there are only two clocks [13]. The reachability relation of TA has also been studied. Comon and Jurski [9] first showed that the reachability relation of TA is effectively definable by a linear arithmetic formula over the integers and reals. This problem was revisited afterwards [12, 10], and very recently, Fränzel et al. provided a considerably simplified proof for this fact [14].

Structure of the paper. Preliminaries are given in Section 2. The algorithm to generate the EPA formula for a given OCA is presented in Section 3. The experimental results are given in Section 4. We conclude the paper in Section 5.

2 Preliminaries

Throughout the paper, $\mathbb{Z}$ and $\mathbb{N}$ denote the set of integers and natural numbers respectively. For a positive natural number n, $[n] := \{1, \cdots, n\}$. We also fix a set of operations $\mathsf{Op} = \{\mathsf{add}(c), \mathsf{zero} \mid c \in \mathbb{Z}\}$.

2.1 One-counter Automata

Definition 1 (OCA). *A* one-counter automaton *is a tuple $\mathcal{A} = (Q, F, \Delta)$ where Q is a finite set of* control locations*; $F \subseteq Q$ is the set of final location, $\Delta \subseteq Q \times \mathsf{Op} \times Q$ is the (finite) transition relation.*

The transitions $(q, \mathsf{zero}, q') \in \Delta$ are referred to as zero transitions. We write $N_{\mathcal{A}}$ for the maximum absolute value of the integer constants occurring in the transitions of $\mathcal{A}$. The set of all *configurations* of $\mathcal{A}$ is denoted by $C(\mathcal{A}) = Q \times \mathbb{N}$. The transition system generated by $\mathcal{A}$ is $(S, \xrightarrow{\mathcal{A}})$ where $S = C(\mathcal{A})$ and $(q, n) \xrightarrow{\mathcal{A}} (q', n')$ iff there is $(q, op, q') \in \Delta$ satisfying (1) in case $op = \mathsf{add}(c)$, $n' = n + c$; and (2) in case $op = \mathsf{zero}$, $n' = n = 0$. We use $\overset{\mathcal{A}}{\Rightarrow}$ to denote the reflexive and transitive closure of $\xrightarrow{\mathcal{A}}$.

The *reachability* problem asks, given an OCA $\mathcal{A}$ and two configurations $C, C' \in C(\mathcal{A})$, does $C \overset{\mathcal{A}}{\Rightarrow} C'$ hold? In applications of OCA, it is usually more convenient to compute the reachability relation $R_{\mathcal{A},q,q'}$ for two given control locations q, q', defined as $R_{\mathcal{A},q,q'} = \{(n, n') \in \mathbb{N}^2 \mid (q, n) \overset{\mathcal{A}}{\Rightarrow} (q', n')\}$. The main purpose of the paper is to give a new representation of this relation in terms of Presburger arithmetic.

2.2 Presburger arithmetic

Presburger arithmetic (PA) is the first-order theory of integer numbers in the structure $(\mathbb{Z}, <, +, 0, 1)$. This is a decidable first-order theory, in contrast to the Peano arithmetic where multiplication is included. Let X be a set of first-order variables. PA Formulae are defined by

$$\varphi ::= \boldsymbol{a}^T \boldsymbol{x} \bowtie b \mid \varphi \wedge \varphi \mid \neg \varphi \mid \exists x. \varphi$$

where $\boldsymbol{a}$ is a vector over $\mathbb{Z}$, $b \in \mathbb{Z}$, and $\bowtie \in \{\geq, >, <, \leq\}$.

In this paper, we are primarily interested in the existential fragment of PA (EPA, aka. quantifier-free PA), which comprises the PA formulae where each existential quantifier is under the scope of an even number of negations. All EPA formulae can be easily rewritten into the prenex normal form $\varphi = \exists \boldsymbol{x}.\ \psi(\boldsymbol{x}, \boldsymbol{y})$, where no quantifiers are allowed in ψ. It is well-known that checking the satisfiability of EPA formulae is NP-complete [6, 19].

For a PA formula φ with free variables $x_1, \cdots, x_k$, we use $\varphi(x_1, \cdots, x_k)$ to highlight the free variables of φ. Moreover, we use $\varphi[n_1/x_1, \cdots, n_k/x_k]$ to denote φ under the assignment η with $\eta(x_j) = n_j$ for each $j \in [k]$.

2.3 Weighted graphs

Definition 2 (Weighted graph). *A weighted graph is a tuple $G = (V, E)$ where V is a finite set of vertices, $E \subseteq V \times \mathbb{Z} \times V$ is a finite set of directed edges with weights.*

Let $G = (V, E)$ be a weighted graph. For an edge $e = (v, z, v') \in E$, $s(e)$ and $t(e)$ denote the source (i.e., v) and the target (i.e., v') of e respectively, and $w(e)$ denotes the weight z. For $v \in V$, we use $E_{in}(v)$ (resp. $E_{out}(v)$) to denote the set of incoming (resp. outgoing) edges of v, namely, the set of edges e such that $t(e) = v$ (resp. $s(e) = v$). A *path* in G is a sequence of edges $e_1 \cdots e_n$ for $n \geq 1$ such that $t(e_i) = s(e_{i+1})$ for each $i \in [n-1]$, where $s(e_1)$ and $t(e_n)$ are called the source and target vertex of π respectively and n is called the *length* of π. A path $\pi = e_1 \cdots e_n$ is a simple path if each vertex occurs at most once along π. Moreover, we use ε to denote the empty path, i.e., a vacuous path containing no edges. If both the source and the target vertex of a path π are v, we say π is a v-cycle. π is a simple cycle if v is the only vertex which occurs twice along a v-cycle π. A weighted graph G is a *loop* if it is strongly connected and there is exactly one simple v-cycle for any vertex v. For a path π in G, we define

- $\mathsf{weight}(G, \pi)$: the sum over all weights of the edges along π,
- $\mathsf{drop}(G, \pi)$: the *minimum* accumulated weight of all prefixes of a path π.

If G is clear from the context, we simply write $\mathsf{weight}(\pi)$ and $\mathsf{drop}(\pi)$.

Example 1. Let $\pi = v_1 \xrightarrow{2} v_2 \xrightarrow{-3} v_3 \xrightarrow{2} v_4$. Then $\mathsf{weight}(\pi) = 2 - 3 + 2 = 1$ and $\mathsf{drop}(\pi) = \min(2, 2-3, 2-3+2) = -1$.

A cycle π is said to be a *positive* (resp. *negative*, resp. *zero*) cycle if $\mathsf{weight}(\pi) > 0$ (resp. $\mathsf{weight}(\pi) < 0$, resp. $\mathsf{weight}(\pi) = 0$).

For $v, v' \in V$, the reachability relation $R_{G,v,v'}$ comprises all the pairs $(n, n') \in \mathbb{N}^2$ such that there exists a path $\pi = v = v_1 \xrightarrow{z_1} v_2 \cdots v_k \xrightarrow{z_k} v_{k+1} = v'$ such that (1) $\mathsf{weight}(\pi) = n' - n$ and (2) for all $i \in [k]$, $n + \sum\limits_{j \in [i]} z_j \geq 0$. As a convention, we assume that $(n, n) \in R_{G,v,v}$ for all $v \in V$ and $n \in \mathbb{N}$. For convenience, we use $(v, n) \overset{G}{\Rightarrow} (v', n')$ to denote $(n, n') \in R_{G,v,v'}$.

For a weighted graph $G = (V, E)$, we use $G^{op} = (V, E^{op})$ to denote the weighted graph with $E^{op} = \{e^{op} \mid e \in E\}$, where $e^{op} = (v', -z, v)$ for $e = (v, z, v')$. For a path $\pi = e_1 \cdots e_n$ in G, π^{op} denotes the path $e_n^{op} \cdots e_1^{op}$ in G^{op}.

3 The EPA formula generation algorithm

Fix an OCA $\mathcal{A} = (Q, q_0, F, \varDelta)$ in this section. Let $G_{\mathcal{A}} = (Q, E)$ be the corresponding weighted graph. Recall that $E = \{(q, z, q') \mid (q, \mathsf{add}(z), q') \in \varDelta\}$. We shall show that, for any $q, q' \in Q$, an EPA formula $\varphi_{\mathcal{A},q,q'}$ can be computed *in polynomial time* to define the reachability relation. The crux of the algorithm is to show that the reachability relation from q to q' in the weighted graph $G_{\mathcal{A}}$ (without zero transitions) can be characterized by an EPA formula $\psi_{G_{\mathcal{A}},q,q'}$ of polynomial size. In the sequel, we first assume the existence of the EPA formula $\psi_{G_{\mathcal{A}},q,q'}$ and show how to formalize the reachability relation in EPA. We then show how the formula $\varphi_{G_{\mathcal{A}},q,q'}$ can be constructed.

3.1 Formalizing the reachability relation of $\mathcal{A}$ in EPA

Let $\mathsf{zt}_{\mathcal{A}}$ denote the set of zero transitions of $\mathcal{A}$. We define the *zero-transition graph* $G_{zt}[\mathcal{A}] = (\mathsf{zt}_{\mathcal{A}}, E_{zt})$ where E_{zt} comprises the pairs $((q_1, \mathsf{zero}, q_2), (q'_1, \mathsf{zero}, q'_2))$ satisfying $\psi_{G_{\mathcal{A}}, q_2, q'_1}(0, 0)$, i.e., $(q'_1, 0)$ is reachable from $(q_2, 0)$ in $G_{\mathcal{A}}$. Intuitively, $G_{zt}[\mathcal{A}]$ satisfies that for $(q_1, \mathsf{zero}, q_2) \in \mathsf{zt}_{\mathcal{A}}$ and $(q'_1, \mathsf{zero}, q'_2) \in \mathsf{zt}_{\mathcal{A}}$, $(q'_1, \mathsf{zero}, q'_2)$ is reachable from $(q_1, \mathsf{zero}, q_2)$ in $G_{zt}[\mathcal{A}]$ iff the configuration $(q'_1, 0)$ is reachable from $(q_2, 0)$ in $G_{\mathcal{A}}$. Note that our algorithm does not explicitly construct the graph $G_{zt}[\mathcal{A}]$; this is for the sake of presentation.

Lemma 1. *Let (q, n) and (q', n') be two configurations of $\mathcal{A}$. Then $(q, n) \overset{\mathcal{A}}{\Rightarrow} (q', n')$ iff one of the following conditions holds: either $(q, n) \overset{G_{\mathcal{A}}}{\Longrightarrow} (q', n')$; or there is a zero-transition $(p, \mathsf{zero}, p') \in \mathsf{zt}_{\mathcal{A}}$ such that $(q, n) \overset{G_{\mathcal{A}}}{\Longrightarrow} (p, 0)$ and $(p', 0) \overset{G_{\mathcal{A}}}{\Longrightarrow} (q', n')$; or there are zero-transitions $(p_1, \mathsf{zero}, p_2), (p'_1, \mathsf{zero}, p'_2) \in \mathsf{zt}_{\mathcal{A}}$ such that $(q, n) \overset{G_{\mathcal{A}}}{\Longrightarrow} (p_1, 0)$, $(p'_1, \mathsf{zero}, p'_2)$ is reachable from $(p_1, \mathsf{zero}, p_2)$ in $G_{zt}[\mathcal{A}]$, and $(p'_2, 0) \overset{G_{\mathcal{A}}}{\Longrightarrow} (t', n')$.*

The characterization of $\overset{\mathcal{A}}{\Rightarrow}$ in Lemma 1 can be specified by an EPA formula $\varphi_{\mathcal{A}, q, q'}(x, y)$ defined as follows: Let $\mathsf{zt}_{\mathcal{A}} = \{\tau_1, \cdots, \tau_k\}$, where for each $i \in [k]$, $\tau_i = (p_{2i-1}, \mathsf{zero}, p_{2i})$. Then

$$\begin{aligned}\varphi_{\mathcal{A},q,q'}(x, y) \equiv \psi_{G_{\mathcal{A}},q,q'}(x, y) \vee \bigvee_{(p,\mathsf{zero},p')\in\mathsf{zt}_{\mathcal{A}}} (\psi_{G_{\mathcal{A}},q,p}(x, 0) \wedge \psi_{G_{\mathcal{A}},p',q}(0, y)) \vee \\ \bigvee_{i,j\in[k],i\neq j} \psi_{G_{\mathcal{A}},q,p_{2i-1}}(x, 0) \wedge \xi_{G_{zt}[\mathcal{A}]}(\tau_i, \tau_j) \wedge \psi_{G_{\mathcal{A}},p_{2j},q'}(0, y),\end{aligned}$$

where $\xi_{G_{zt}[\mathcal{A}]}(\tau_i, \tau_j)$ specifies that τ_j is reachable from τ_i in $G_{zt}[\mathcal{A}]$,

$$\begin{aligned}\xi_{G_{zt}[\mathcal{A}]}(\tau_i, \tau_j) \equiv \exists z_1. \cdots \exists z_k.\ z_i = 1 \wedge z_j > 1 \wedge \bigwedge_{\ell\in[k]} z_\ell \geq 0 \wedge \\ \bigwedge_{\ell',\ell''\in[k],\ell'\neq\ell''} ((z_{\ell'} > 0 \wedge z_{\ell''} > 0) \rightarrow z_{\ell'} \neq z_{\ell''}) \wedge \\ \bigwedge_{\ell\in[k]} \left(z_\ell > 1 \rightarrow \bigvee_{\ell'\in[k],\ell'\neq\ell} (z_{\ell'} > 0 \wedge z_{\ell'} + 1 = z_\ell \wedge \psi_{G_{\mathcal{A}},p_{2\ell'},p_{2\ell-1}}(0, 0)) \right).\end{aligned}$$

Intuitively, the variables $z_1, \cdots, z_k$ in $\xi_{G_{zt}[\mathcal{A}]}(\tau_i, \tau_j)$ represent the positions of some simple path from τ_i to τ_j in $G_{zt}[\mathcal{A}]$, where τ_i is in the first position (i.e. $z_i = 1$), τ_j is in the last position (i.e., z_j is maximal), and the vertices not in the path are assigned null (i.e. $z_\ell = 0$). Moreover, for each vertex in the path, except the one in the first position, there is a vertex in the position preceding it as well as an edge between them.

3.2 Characterizing the reachability relation of $G_{\mathcal{A}}$ in EPA

We first recall the core concepts of the decision procedure in [20, 18]. We then show how to construct the EPA formula $\psi_{G_{\mathcal{A}},q,q'}$ for $q, q' \in Q$. The main idea of the decision procedure is to characterize $\psi_{G_{\mathcal{A}},q,q'}$ by path flows satisfying some extra constraints.

Example 2 (Running example). We will use the OCA $\mathcal{A}$ in Figure 1 as a running example, where q_0 and q_{11} are the initial and final control locations respectively.

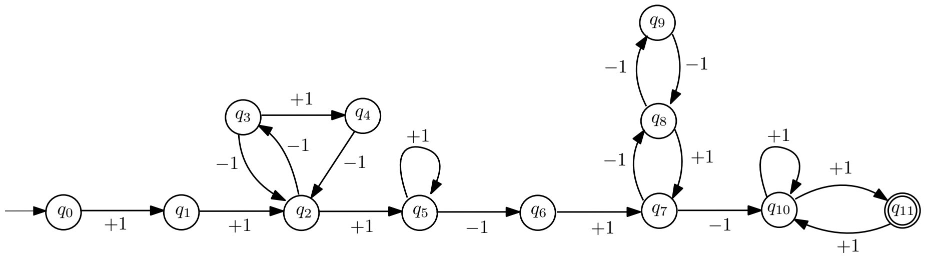

Fig. 1. $G_{\mathcal{A}}$ in the running example

Proposition 1. *Let (q, n) and (q', n') be two configurations of $\mathcal{A}$. Then (q', n') is reachable from (q, n) in $G_{\mathcal{A}}$ iff (q', n') is reachable from (q, n) through a path that contains no zero cycles.*

By Proposition 1, we will suppress zero cycles when constructing $\varphi_{G_{\mathcal{A}},q,q'}$.

Definition 3 (Flow and path flow). *Let $q, q' \in Q$. A flow from q to q' in $G_{\mathcal{A}}$ is a function $f : E \to \mathbb{N}$ such that*

- *if $q = q'$, then for all $p \in Q$, $\sum\limits_{e \in E_{in}(p)} f(e) = \sum\limits_{e \in E_{out}(p)} f(e)$,*
- *otherwise, for all $p \in Q \setminus \{q, q'\}$,*

$$\sum_{e \in E_{in}(p)} f(e) = \sum_{e \in E_{out}(p)} f(e), \text{ and}$$

$$1 + \sum_{e \in E_{in}(q)} f(e) = \sum_{e \in E_{out}(q)} f(e), \quad \sum_{e \in E_{out}(q')} f(e) = 1 + \sum_{e \in E_{in}(q')} f(e).$$

For a flow f, we use $\mathsf{weight}(f)$ to denote $\sum\limits_{e \in E} f(e)\mathsf{weight}(e)$. A path flow *from q to q' is a flow f corresponding to some path π from q to q', namely, $f = f_\pi$, where for each $e \in E$, $f_\pi(e)$ is the number of occurrences of e in π. In particular, for an edge e, f_e is a path flow such that $f_e(e) = 1$ and $f_e(e') = 0$ for each $e' \neq e$. Moreover, by convention, we assume that $f_\perp$ such that $f_\perp(e) = 0$ for every $e \in E$ is a path flow from q to q for each $q \in Q$.*

Example 3. Let f_1 be a flow from q_0 to q_5 in $G_{\mathcal{A}}$ presented in Figure 1, where $f_1((q_0, +1, q_1)) = f_1((q_1, +1, q_2)) = 1$, $f_1((q_2, -1, q_3)) = 2$, $f_1((q_3, -1, q_2)) = 1$, $f_1((q_3, +1, q_4)) = f_1((q_4, -1, q_2)) = f_1((q_2, +1, q_5)) = 1$, and $f_1(e') = 0$ for all the other edges e'. Then $f_1 = f_\pi$ where $\pi = q_0 \xrightarrow{+1} q_1 \xrightarrow{+1} q_2 \xrightarrow{-1} q_3 \xrightarrow{-1} q_2 \xrightarrow{-1} q_3 \xrightarrow{+1} q_4 \xrightarrow{-1} q_2 \xrightarrow{+1} q_5$. Therefore, f_1 is a path flow from from q_0 to q_5.

Definition 4 (Support). *Given a path flow $f : E \to \mathbb{N}$, the support of f is the weighted graph $G_f = (V_f, E_f)$ with $E_f = \{e \in E \mid f(e) > 0\}$. A subgraph $F \subseteq G_{\mathcal{A}}$ is called a q-q'* support *if there is a path flow f from q to q' such that $F = G_f$. In particular, the empty graph is a q-q support for each $q \in Q$.*

Example 4. Let F be the subgraph of $G_{\mathcal{A}}$ in Figure 1 comprising the edges $(q_0, +1, q_1)$, $(q_1, +1, q_2)$, $(q_2, -1, q_3)$, $(q_3, -1, q_2)$, $(q_3, +1, q_4)$, $(q_4, -1, q_2)$, and $(q_2, +1, q_5)$. Then F is a q_0-q_5 support since $F = G_{f_1}$ where f_1 is the path flow from q_0 to q_5 in Example 3.

It is well-known that path flows can be captured by an EPA formula, which specifies the conditions on the incoming and outgoing flows of all vertices and the constraints that the support of the path flow is connected.

Proposition 2 ([28]). *An EPA formula $\varphi^{(\mathrm{PF})}_{G_{\mathcal{A}},q,q'}((x_e)_{e \in E})$ can be constructed, in linear time, to capture the path flows from q to q'. Namely, for each flow f from q to q', f is a path flow iff $\varphi^{(\mathrm{PF})}_{G_{\mathcal{A}},q,q'}[(f(e)/x_e)_{e \in E}]$ holds.*

Note that not all path flows correspond to runs of $G_{\mathcal{A}}$ since the constraints of path flows do not address the non-negativeness requirements of the counter values. In the sequel, we recall the results [18] where extra constraints (called reachability criteria) were imposed.

For a path flow f, suppose $f = f_\pi$ for some path π. We can split f into multiple path flows by dividing π into segments according to the last occurrence of each edge in π (note that an edge may occur multiple times in π). This is formalized as the concept of edge decomposition as follows.

Definition 5 (Edge decomposition). *Given a q-q' support F, an* edge decomposition of F *is a sequence of tuples $\{(F_i, v_i, v'_i, e_i)\}_{i \in [m]}$, where $F_i \subseteq F$, $v_1 = q$, $v'_{m+1} = q'$ such that*

1. *for each $i \in [m]$, F_i is a v_i-v'_i support, $e_i = (v'_i, z_i, v_{i+1})$ for some $z_i \in \mathbb{Z}$,*
2. *all e_i's are mutually distinct,*
3. *for each $1 \leq i < j \leq m$, $e_i \notin F_j$,*
4. *$F = \bigcup\limits_{i \in [m]} F_i$.*

Note that if $v_i = v'_i$, then F_i may be the empty graph $\emptyset$.
Furthermore, given a path flow f, an edge decomposition of f *is a sequence of tuples $\{(f_i, v_i, v'_i, e_i)\}_{i \in [m]}$, where f_i is a path flow from v_i to v'_i, $f = \sum\limits_{i \in [m]} (f_i + f_{e_i})$, and $\{(G_{f_i}, v_i, v'_i, e_i)\}_{i \in [m]}$ is an edge decomposition of G_f.*

Example 5. Let f_1 be the path flow from q_0 to q_5 and $\pi = q_0 \xrightarrow{+1} q_1 \xrightarrow{+1} q_2 \xrightarrow{-1} q_3 \xrightarrow{-1} q_2 \xrightarrow{-1} q_3 \xrightarrow{+1} q_4 \xrightarrow{-1} q_2 \xrightarrow{+1} q_5$ in Example 3 such that $f_1 = f_\pi$. The edges in G_{f_1} can be ordered according to their last occurrences in π as follows: $(q_0, +1, q_1)$, $(q_1, +1, q_2)$, $(q_3, -1, q_2)$, $(q_2, -1, q_3)$, $(q_3, +1, q_4)$, $(q_4, -1, q_2)$, $(q_2, +1, q_5)$. Note that $(q_3, -1, q_2)$ is ordered before $(q_2, -1, q_3)$ since $(q_2, -1, q_3)$ occurs twice in π and the second occurrence of $(q_2, -1, q_3)$ is after the unique occurrence of $(q_3, -1, q_2)$. Then from this ordering, we can obtain an edge decomposition $\{(f'_i, v_i, v'_i, e_i)\}_{i \in [7]}$ of f, where

- $(f'_1, v_1, v'_1, e_1) = (f_\perp, q_0, q_0, (q_0, +1, q_1))$,
- $(f'_2, v_2, v'_2, e_2) = (f_\perp, q_1, q_1, (q_1, +1, q_2))$,
- $(f'_3, v_3, v'_3, e_3) = (f_{(q_2,-1,q_3)}, q_2, q_3, (q_3, -1, q_2))$,
- $(f'_4, v_4, v'_4, e_4) = (f_\perp, q_2, q_2, (q_2, -1, q_3))$,
- $(f'_5, v_5, v'_5, e_5) = (f_\perp, q_3, q_3, (q_3, +1, q_4))$,
- $(f'_6, v_6, v'_6, e_6) = (f_\perp, q_4, q_4, (q_4, -1, q_2))$, and
- $(f'_7, v_7, v'_7, e_7) = (f_\perp, q_2, q_2, (q_2, +1, q_5))$.

The reachability criteria to guarantee the non-negativeness of counter values in path flows are classified into three types, with the first two types formalized in the following two definitions.

Definition 6 (Type-1 reachability criteria). *Let $n, n' \in \mathbb{N}$. Then a path flow f from q to q' is said to satisfy the type-1 reachability criteria for (n, n') if the following constraints hold,*

- *G_f does not contain positive cycles,*
- *$\mathsf{weight}(f) = n' - n$,*
- *f has an edge decomposition $\{(f_i, v_i, v'_i, e_i)\}_{i\in[m]}$ such that $n + \sum\limits_{i\in[j]} (\mathsf{weight}(f_i) + \mathsf{weight}(e_i)) \geq 0$ for all $j \in [m]$.*

Note that the condition for $\{(f_i, v_i, v'_i, e_i)\}_{i\in[m]}$ in Definition 6 can be equivalently phrased as $n' - \sum\limits_{j<i\leq m} (\mathsf{weight}(f_i) + \mathsf{weight}(e_i)) \geq 0$ for all $j \in [m]$, which intuitively explains the dual of the type-1 reachability criteria, i.e. type-2 reachability criteria in Definition 7 .

Example 6. Let $(n, n') = (1, 1)$. Then the path flow f_1 from q_0 to q_5 in the Example 3 satisfies the type-1 reachability criteria for (n, n'): G_{f_1} does not contain positive cycles, $\mathsf{weight}(f_1) = 0 = 1 - 1$, f_1 has an edge decomposition $\{(f'_i, v_i, v'_i, e_i)\}_{i\in[7]}$ as shown in Example 5, moreover,

- $1 + \mathsf{weight}(f'_1) + \mathsf{weight}(e_1) = 1 + 0 + \mathsf{weight}((q_0, +1, q_1)) = 2 \geq 0$,
- $1 + \sum\limits_{j\in[2]} (\mathsf{weight}(f'_j) + \mathsf{weight}(e_j)) = 2 + 0 + \mathsf{weight}((q_1, +1, q_2)) = 3 \geq 0$,
- $1 + \sum\limits_{j\in[3]} (\mathsf{weight}(f'_j) + \mathsf{weight}(e_j)) = 3 + \mathsf{weight}(f_{(q_2,-1,q_3)}) + \mathsf{weight}((q_3, -1, q_2)) = 3 - 1 - 1 = 1 \geq 0$,
- $1 + \sum\limits_{j\in[4]} \mathsf{weight}(f'_j) + \mathsf{weight}(e_j) = 1 + 0 + \mathsf{weight}((q_2, -1, q_3)) = 0 \geq 0$,
- $1 + \sum\limits_{j\in[5]} \mathsf{weight}(f'_j) + \mathsf{weight}(e_j) = 0 + 0 + \mathsf{weight}((q_3, +1, q_4)) = 1 \geq 0$,
- $1 + \sum\limits_{j\in[6]} \mathsf{weight}(f'_j) + \mathsf{weight}(e_j) = 1 + 0 + \mathsf{weight}((q_4, -1, q_2)) = 0 \geq 0$,
- $1 + \sum\limits_{j\in[7]} \mathsf{weight}(f'_j) + \mathsf{weight}(e_j) = 0 + 0 + \mathsf{weight}((q_2, +1, q_5)) = 1 \geq 0$.

The type-2 reachability criteria are dual to the type-1 reachability criteria.

Definition 7 (Type-2 reachability criteria). *Let $n, n' \in \mathbb{N}$. Then a path flow f from q to q' is said to satisfy the type-2 reachability criteria for (n, n') if f^{op} satisfies the type-1 reachability criteria for (n', n) in G^{op}, where $f^{op}((v', -z, v)) = f((v, z, v'))$ for each $(v, z, v') \in E$.*

Example 7. Let $(n, n') = (1, 3)$ and f_3 be the path flow from q_7 to q_{11} such that $f_3((q_7, -1, q_{10})) = 1$, $f_3((q_{10}, +1, q_{10})) = 2$, and $f_3((q_{10}, +1, q_{11})) = 1$. Then f_3 satisfies the type-2 reachability criteria for $(1, 3)$ since f_3^{op} satisfies the type-1 reachability criteria for $(3, 1)$ in G^{op}.

- $G_{f_3^{op}}$ is the graph comprising the edges $(q_{11}, -1, q_{10})$, $(q_{10}, -1, q_{10})$, and $(q_{10}, +1, q_7)$. It contains no positive cycles.
- $\mathsf{weight}(f_3^{op}) = (-1) \times 1 + (-1) \times 2 + (+1) \times 1 = -2 = 1 - 3$.
- f_3^{op} has an edge decomposition $\{(f'_i, v_i, v'_i, e_i)\}_{i \in [3]}$ where $(f'_1, v_1, v'_1, e_1) = (f_\perp, q_{11}, q_{11}, (q_{11}, -1, q_{10}))$, $(f'_2, v_2, v'_2, e_2) = (f_{(q_{10}, -1, q_{10})}, q_{10}, q_{10}, (q_{10}, -1, q_{10}))$, $(f'_3, v_3, v'_3, e_3) = (f_\perp, q_{10}, q_{10}, (q_{10}, +1, q_7))$, moreover,
  - $3 + \mathsf{weight}(f'_1) + \mathsf{weight}(e_1) = 2 \geq 0$,
  - $3 + \sum\limits_{j \in [2]} (\mathsf{weight}(f'_j) + \mathsf{weight}(e_j)) = 2 - 1 - 1 = 0 \geq 0$,
  - $3 + \sum\limits_{j \in [3]} (\mathsf{weight}(f'_j) + \mathsf{weight}(e_j)) = 0 + 0 + 1 = 1 \geq 0$.

It remains to present the type-3 reachability criteria.

Definition 8 (Cycle template). *Let $G = (V', E')$ be a subgraph of $G_{\mathcal{A}}$, $v \in V'$ and $n \in \mathbb{N}$. A* positive v-cycle template *w.r.t. n in G is a cycle $\pi = \pi_1 \cdot \pi_2 \cdot \pi_3$ such that there is a vertex $v' \in V'$ satisfying that*

- *π_2 is a positive simple v'-cycle,*
- *if $v = v'$, then $\pi_1 = \pi_3 = \varepsilon$, otherwise, π_1 (resp. π_3) is a simple path from v to v' (resp. from v' to v),*
- $\mathsf{drop}(\pi_1 \cdot \pi_2) \geq -n$.

A negative v-cycle template *w.r.t. n is a cycle $\pi = \pi_1 \cdot \pi_2 \cdot \pi_3$ such that $\pi^{op} = \pi_3^{op} \cdot \pi_2^{op} \cdot \pi_1^{op}$ is a positive v-cycle template w.r.t. n in G^{op}.*

Example 8. The cycle $\pi_1 \cdot \pi_2 \cdot \pi_3$, where $\pi_1 = \pi_3 = \varepsilon$ and $\pi_2 = (q_5, +1, q_5)$, is a positive q_5-cycle template in $G_{\mathcal{A}}$ w.r.t. 1 since $\mathsf{drop}(\pi_1 \cdot \pi_2) = 1 \geq -1$. Moreover, $\pi_4 \cdot \pi_5 \cdot \pi_6$, where $\pi_4 = (q_7, -1, q_8)$, $\pi_5 = (q_8, -1, q_9)(q_9, -1, q_8)$, and $\pi_6 = (q_8, +1, q_7)$, is a negative q_7-cycle template w.r.t. 1 in $G_{\mathcal{A}}$ since $\pi_6^{op} \cdot \pi_5^{op} \cdot \pi_4^{op}$ satisfies that $\mathsf{drop}(\pi_6^{op} \cdot \pi_5^{op}) = -1 \geq -1$, thus is a positive q_7-cycle template w.r.t. 1 in $G_{\mathcal{A}}^{op}$.

Definition 9 (Type-3 reachability criteria). *Let $n, n' \in \mathbb{N}$. Then a path flow f from q to q' is said to satisfy the type-3 reachability criteria for (n, n') if the following constraints hold.*

- *there is a positive q-cycle template w.r.t. n in $G_{\mathcal{A}}$,*
- $\mathsf{weight}(f) = n' - n$,

- *there is a negative q'-cycle template w.r.t. n' in $G_{\mathcal{A}}$.*

Example 9. Let f_2 be the path flow from q_5 to q_7 such that $f_2((q_5, -1, q_6)) = f_2((q_6, +1, q_7)) = 1$ and $f_2(e') = 0$ for all the other edges e'. Then f_2 satisfies the type-3 reachability criteria for $(1, 1)$: $\mathsf{weight}(f_2) = 0 = 1 - 1$, moreover, $\pi_1 \cdot \pi_2 \cdot \pi_3$ in Example 8 is a positive q_5-cycle template w.r.t. 1 in $G_{\mathcal{A}}$ and $\pi_4 \cdot \pi_5 \cdot \pi_6$ is a negative q_7-cycle template w.r.t. 1 in $G_{\mathcal{A}}$.

The following lemma captures reachability in $G_{\mathcal{A}}$.

Lemma 2 ([18]). *Let $q, q' \in Q$ and $n, n' \in \mathbb{N}$. Then (q', n') is reachable from (q, n) in $G_{\mathcal{A}}$ iff there is a path flow f from q to q' which can be split into three path flows f_1, f_2, f_3 such that*

- *$f = f_1 + f_2 + f_3$,*
- *there are $q_1, q_2 \in Q$ and $n'', n''' \in \mathbb{N}$ satisfying that*
  - *f_1 is a path flow from q to q_1 (note that f_1 may be the zero flow $f_\perp$, in this case, $q_1 = q$), moreover, if $f_1 \neq f_\perp$, then f_1 satisfies the type-1 reachability criteria for (n, n''),*
  - *f_2 is a path flow from q_1 to q_2 (note that f_2 may be the zero flow $f_\perp$, in this case, $q_2 = q_1$), moreover, if $f_2 \neq f_\perp$, then f_2 satisfies the type-3 reachability criteria for (n'', n'''),*
  - *f_3 is a path flow from q_2 to q' (note that f_3 may be the zero flow $f_\perp$, in this case, $q' = q_2$), moreover, if $f_3 \neq f_\perp$, then f_3 satisfies the type-2 reachability criteria for (n''', n').*

Example 10. Let $f = f_1 \cdot f_2 \cdot f_3$ be path flow from q_0 to q_{11}, where f_1 is the path flow from q_0 to q_5 in Example 6, f_2 is a path flow from q_5 to q_7 in Example 9, and f_3 is a path flow from q_7 to q_{11} in Example 7. Then from Example 6, Example 9, and Example 7, we know that f_1 satisfies the type-1 reachability criteria for $(1, 1)$, f_2 satisfies the type-3 reachability criteria for $(1, 1)$, and f_3 satisfies the type-2 reachability criteria for $(1, 3)$. Therefore, according to Lemma 2, $(q_{11}, 3)$ is reachable from $(q_0, 1)$ in $G_{\mathcal{A}}$.

In the sequel, we show how the constraints in Lemma 2 can be defined by EPA formulae. We use the variables $(x_e)_{e \in E}$ to represent the path flow f in Lemma 2. Moreover, we use the variables $(y_{e,1})_{e \in E}$, $(y_{e,2})_{e \in E}$, $(y_{e,3})_{e \in E}$ to represent the path flows f_1, f_2, and f_3.

Type-1 reachability criteria. Our goal is to formalize by an EPA formula $\psi^{(\mathrm{T1RC})}_{q,q_1}$ that the path flow f_1 from q to q_1 represented by $(y_{e,1})_{e \in E}$ satisfies the type-1 reachability criteria. Let the variables x, x_1 represent the counter values of q, q_1 respectively. From the definition of the type-1 reachability criteria, it is sufficient to show that the absence of positive cycles and the existence of an edge decomposition in G_{f_1} can be encoded in EPA. In the sequel, we illustrate how to encode by an EPA formula the existence of an edge decomposition. The EPA formula $\psi^{(\mathrm{APC})}((y_{e,1})_{e \in E})$ to encode the absence of positive cycles is omitted, due to the page limit.

For each edge e, we introduce integer variables idx_e and sum_e, and the integer variables $(y_{e,e'})_{e' \in E}$. Intuitively, each edge e is associated with an index idx_e indicating the position of the last occurrence of e along the edge decomposition, $(y_{e,e'})_{e' \in E}$ specifies the flow of e' associated with the edge e, i.e., the number of occurrences of e' along the path up to the last occurrence of e. We use sum_e to represent the sum of the weights of all the edges preceding the last occurrence of e in the edge decomposition. Besides, x, x_1 represent the counter value at q and q_1 respectively. Then the existence of an edge decomposition from q to q_1 is encoded by the EPA formula

$$\begin{aligned}
&\psi^{\mathrm{EDC}}_{q,q_1}((y_{e,1})_{e \in E}, (idx_e, sum_e)_{e \in E}, (y_{e,e'})_{e,e' \in E}) ::= \\
&\qquad \psi^{(\mathrm{IDX})}_{q,q_1}((y_{e,1})_{e \in E}, (idx_e)_{e \in E}) \wedge \psi^{(\mathrm{EDG})}_{q,q_1}((y_{e,1})_{e \in E}, (idx_e)_{e \in E}, (y_{e,e'})_{e,e' \in E}) \wedge \\
&\qquad\quad \psi^{(\mathrm{NN})}_{q,q_1}((y_{e,1})_{e \in E}, (idx_e, sum_e)_{e \in E}, (y_{e,e'})_{e,e' \in E}),
\end{aligned}$$

where $\psi^{(\mathrm{IDX})}_{q,q_1}((y_{e,1})_{e \in E}, (idx_e)_{e \in E})$ intuitively specifies that the variables idx_e with $y_{e,1} > 0$ are mutually distinct and represent an order of the edges corresponding to their last occurrences in a path flow from q to q_1. Formally, it specifies that $\{idx_e \mid y_{e,1} > 0\} = [i]$, where i is the number of edges e with $y_{e,1} > 0$. Moreover, $idx_e = i$ for some e with $t(e) = q_1$,

$$\begin{aligned}
\psi^{(\mathrm{IDX})}_{q,q_1} ::= &\bigwedge_{e \in E} (y_{e,1} > 0 \to idx_e > 0 \wedge y_{e,1} = 0 \to idx_e = 0) \wedge \\
&\bigvee_{e \in E} (y_{e,1} > 0 \wedge idx_e = 1) \wedge \bigwedge_{e,e' \in E, e \neq e'} ((y_{e,1} > 0 \wedge y_{e',1} > 0) \to idx_e \neq idx_{e'}) \wedge \\
&\bigwedge_{e \in E} \left((y_{e,1} > 0 \wedge idx_e > 1) \to \bigvee_{e' \in E} (y_{e',1} > 0 \wedge idx_{e'} + 1 = idx_e) \right) \wedge \\
&\bigvee_{e \in E, t(e) = q_1} \left(y_{e,1} > 0 \wedge \bigwedge_{e' \in E} idx_{e'} \leq idx_e \right),
\end{aligned}$$

and $\psi^{(\mathrm{EDG})}_{q,q_1}((y_{e,1})_{e \in E}, (idx_e)_{e \in E}, (y_{e,e'})_{e,e' \in E})$ specifies the constraints on the occurrences of edges in an edge decomposition,

$$\begin{aligned}
\psi^{(\mathrm{EDG})}_{q,q_1} := &\bigwedge_{e \in E} \left((y_{e,1} > 0 \wedge idx_e = 1) \to \psi^{(\mathrm{PF})}_{q,s(e)}((y_{e,e'})_{e' \in E}) \right) \wedge \\
&\bigwedge_{e,e' \in E} \left((y_{e',1} > 0 \wedge y_{e,1} > 0 \wedge idx_{e'} + 1 = idx_e) \to \psi^{(\mathrm{PF})}_{t(e'),s(e)}((y_{e,e''})_{e'' \in E}) \right) \wedge \\
&\bigwedge_{e,e' \in E} ((y_{e,1} > 0 \wedge y_{e',1} > 0 \wedge idx_e < idx_{e'}) \to y_{e',e} = 0) \wedge \\
&\bigwedge_{e \in E} \left(y_{e,1} > 0 \to \left(\textstyle\sum_{e' \in E} y_{e',e} \right) + 1 = y_{e,1} \right),
\end{aligned}$$

(Note that $y_{e',e} = 0$ specifies that e does not occur in the path flow for e'.)

Moreover, $\psi^{(\mathrm{NN})}_{q,q_1}$ specifies that the sum of x and the weights of the path flows and edges in the edge decomposition are non-negative,

$$\begin{aligned}
\psi^{(\mathrm{NN})}_{q,q_1} ::= &\bigwedge_{e \in E} \left((y_{e,1} > 0 \wedge idx_e = 1) \to sum_e = \mathsf{weight}(e) + \sum_{e' \in E} \mathsf{weight}(e') \cdot y_{e,e'} \right) \wedge \\
&\bigwedge_{e,e' \in E} \left(\begin{array}{l} (y_{e,1} > 0 \wedge y_{e',1} > 0 \wedge idx_e + 1 = idx_{e'}) \to \\ sum_e + \mathsf{weight}(e') + \sum\limits_{e'' \in E} \mathsf{weight}(e'') \cdot y_{e',e''} = sum_{e'} \end{array} \right) \wedge \\
&\bigwedge_{e \in E} (y_{e,1} > 0 \to x + sum_e \geq 0).
\end{aligned}$$

Then we encode the type-1 reachability criteria by the following EPA formula,

$$\psi^{(\mathrm{T1RC})}_{G_{\mathcal{A}},q,q_1}(x, x_1, (y_{e,1})_{e\in E}) ::= \psi^{(\mathrm{APC})}((y_{e,1})_{e\in E})) \wedge$$
$$\exists (idx_e, sum_e)_{e\in E}, (y_{e,e'})_{e,e'\in E}.\ \begin{pmatrix} \psi^{(\mathrm{EDC})}_{q,q_1}((y_{e,1})_{e\in E}, (idx_e, sum_e)_{e\in E}, (y_{e,e'})_{e,e'\in E}) \wedge \\ \psi^{(\mathrm{WGT})}_{q,q_1}((y_{e,1})_{e\in E}, (idx_e, sum_e)_{e\in E}) \end{pmatrix},$$

where $\psi^{(\mathrm{WGT})}_{q,q_1}$ specifies that the sum of x and the weights of all the path flows and edges in the edge decomposition is equal to x_1,

$$\psi^{(\mathrm{WGT})}_{q,q_1} ::= \bigvee_{e\in E, t(e)=q_1} \left(y_{e,1} > 0 \wedge \bigwedge_{e'\in E} idx_{e'} \leq idx_e \wedge x + sum_e = x_1 \right).$$

One can observe that the size of $\psi^{(\mathrm{T1RC})}_{G_{\mathcal{A}},q,q_1}$ is polynomial in the size of $\mathcal{A}$.

Type-2 *reachability criteria.* Suppose that $(y_{e,3})_{e\in E}$ represents a path flow f_3 from q_2 to q'. Then Lemma 2 says that f_3 satisfies the type-2 reachability criteria, that is, the flow f_3^{op} in $G^{op}_{\mathcal{A}}$ satisfies the type-1 reachability criteria, which is encoded by the EPA formula $\psi^{(\mathrm{T2RC})}_{G_{\mathcal{A}},q_2,q'}$ defined below. Let x_2, x' represents the counter values of q_2 and q' respectively. Then

$$\psi^{(\mathrm{T2RC})}_{G_{\mathcal{A}},q_2,q'}(x_2, x', (y_{e,3})_{e\in E}) ::= \exists (y^{op}_{e',3})_{e'\in E^{op}}.\ \varphi^{(\mathrm{T1RC})}_{G^{op}_{\mathcal{A}},q',q_2}(x', x_2, (y^{op}_{e',3})_{e'\in E^{op}}) \wedge$$
$$\bigwedge_{e=(p,c,p')\in E, e'=(p',-c,p)\in E^{op}} y^{op}_{e',3} = y_{e,3}.$$

Type-3 *reachability criteria.* Our goal is to construct an EPA formula $\psi^{(\mathrm{T3RC})}_{G_{\mathcal{A}},q_1,q_2}$ to characterize the type-3 reachability criteria for a path flow represented by $(y_{e,2})_{e\in E}$ from q_1 to q_2. Let x_1, x_2 represent the counter values of q_1, q_2 respectively. Recall that the type-3 reachability criteria specify that there exist a positive q_1-cycle template and a negative q_2-cycle template, as well as a path flow from q_1 to q_2. Since negative cycle templates are the dual of positive cycle templates and we know how to encode a path flow in EPA, it is sufficient to show that the existence of a positive q_1-cycle template can be specified by an EPA formula $\psi^{(\mathrm{PCT})}_{G_{\mathcal{A}},q_1}$. To this end, we introduce integer variables $idx_{e,1}, idx_{e,2}, idx_{e,3}$ for $e \in E$ to represent the three simple paths (or cycles) π_1, π_2, π_3 in a positive q_1-cycle template. Moreover, we introduce integer variables $sum_{p,1}, drop_{p,1}$ and $sum_{p,2}, drop_{p,2}$ for $p \in Q$ to describe the computation of the sum of edge weights and the drop in the prefixes of π_1 and π_2 respectively. Then

$$\psi^{(\mathrm{PCT})}_{G_{\mathcal{A}},q_1}(x_1, (idx_{e,i})_{e\in E, i=1,2,3}, (sum_{p,j}, drop_{p,j})_{p\in Q, j=1,2}) ::=$$
$$\bigvee_{p'\in Q} \begin{pmatrix} \psi^{(\mathrm{SP1})}_{q_1,p'}((idx_{e,1})_{e\in E}, (sum_{p,1}, drop_{p,1})_{p\in Q}) \wedge \\ \psi^{(\mathrm{SC})}_{p',p'}((idx_{e,2})_{e\in E}, (sum_{p,2}, drop_{p,2})_{p\in Q}) \wedge \\ \psi^{(\mathrm{SP2})}_{p',q_1}((idx_{e,3})_{e\in E}) \wedge \\ \psi^{(\mathrm{NN})}(x_1, sum_{p',1}, drop_{p',1}, drop_{p',2}) \end{pmatrix},$$

where $\psi^{(\mathrm{SP1})}_{q_1,p'}$, $\psi^{(\mathrm{SC})}_{p',p'}$, and $\psi^{(\mathrm{SP2})}_{p',q_1}$ specify the existence of three simple paths (or cycles) π_1, π_2, π_3 in a positive q_1-cycle template, as well as the computation of the the sum of edge weights and the drop in the prefixes of π_1 and π_2. Concretely,

$$\begin{array}{l}
\psi^{(\mathrm{SP1})}_{q_1,p'}((idx_{e,1})_{e\in E},(sum_{p,1},drop_{p,1})_{p\in Q}) ::= \\
\left(q_1 = p' \wedge \bigwedge\limits_{e\in E} idx_{e,1} = 0 \wedge sum_{p',1} = 0 \wedge drop_{p',1} = 0\right) \vee \\
\left(\begin{array}{l}
q_1 \neq p' \wedge \psi^{(\mathrm{SPIDX})}_{q_1,p'}((idx_{e,1})_{e\in E}) \wedge \\
\quad \bigwedge\limits_{e=(q_1,c,p)\in E} (idx_{e,1} = 1 \to (sum_{p,1} = c \wedge drop_{p,1} = \min(c,0))) \ \wedge \\
\quad \bigwedge\limits_{e=(p_1,c,p_2)\in E} \quad idx_{e,1} > 1 \to \left(\begin{array}{l} sum_{p_1,1} + c = sum_{p_2,1} \ \wedge \\ drop_{p_2,1} = \min(drop_{p_1,1}, sum_{p_2,1}) \end{array}\right)
\end{array}\right),
\end{array}$$

where $\psi^{(\mathrm{SPIDX})}_{q_1,p'}$ specifies how the integer variables $idx_{e,1}$ for $e \in E$ can be constrained to represent a simple path from q_1 to p',

$$\begin{array}{l}
\psi^{(\mathrm{SPIDX})}_{q_1,p'}((idx_{e,1})_{e\in E}) ::= \bigwedge\limits_{e\in E} idx_{e,1} \geq 0 \wedge \bigwedge\limits_{e=(p,z,p)\in E} idx_{e,1} = 0 \wedge \\
\bigvee\limits_{e\in E, s(e)=q_1} idx_{e,1} = 1 \ \wedge \bigvee\limits_{e\in E, t(e)=p'} \bigwedge\limits_{e'\in E} idx_{e',1} \leq idx_{e,1} \wedge \\
\quad \bigwedge\limits_{e,e'\in E, e\neq e'} ((idx_{e,1} > 0 \wedge idx_{e',1} > 0) \to idx_{e,1} \neq idx_{e',1}) \wedge \\
\bigwedge\limits_{e\in E} (idx_{e,1} > 1 \to \bigvee\limits_{e'\in E, t(e')=s(e)} idx_{e',1} + 1 = idx_{e,1}) \wedge \\
\quad \bigwedge\limits_{e,e'\in E, t(e)=s(e')} ((idx_{e,1} > 0 \wedge idx_{e',1} > 0) \to idx_{e,1} + 1 = idx_{e',1}).
\end{array}$$

The formula $\psi^{(\mathrm{SC})}_{p',p'}((idx_{e,2})_{e\in E},(sum_{p,2},drop_{p,2})_{p\in Q})$ and $\psi^{(\mathrm{SP2})}_{p',q_1}((idx_{e,3})_{e\in E})$ can be defined similarly.

Moreover, we define the formula

$$\begin{array}{l}
\psi^{(\mathrm{NN})}(x_1, sum_{p',1}, drop_{p',1}, drop_{p',2}) ::= \\
\quad x_1 + drop_{p',1} \geq 0 \wedge x_1 + sum_{p',1} + drop_{p',2} \geq 0.
\end{array}$$

Symmetrically, the existence of a negative q_2-cycle template can be specified by an EPA formula $\psi^{(\mathrm{PCT})}_{G^{op}_{\mathcal{A}},q_2}(x_2,(idx_{e',i})_{e'\in E^{op},i=4,5,6},(sum_{p,j},drop_{p,j})_{p\in Q,j=3,4})$, where the variables $idx_{e',4}, idx_{e',5}, idx_{e',6}$ and $sum_{p,3}, drop_{p,3}, sum_{p,4}, drop_{p,4}$ are similar to the variables $idx_{e,1}, idx_{e,2}, idx_{e,3}$ and $sum_{p,1}, drop_{p,1}, sum_{p,2}, drop_{p,2}$ respectively. It follows that

$$\begin{array}{l}
\psi^{(\mathrm{T3RC})}_{G_{\mathcal{A}},q_1,q_2}(x_1,x_2,(y_{e,2})_{e\in E}) ::= \varphi^{(\mathrm{PF})}_{q_1,q_2}((y_{e,2})_{e\in E}) \wedge x_1 + \sum\limits_{e\in E} \mathsf{weight}(e)\cdot y_{e,2} = x_2 \wedge \\
\quad \exists (idx_{e,i})_{e\in E,i\in[6]}(sum_{p,j},drop_{p,j})_{p\in Q,j\in[4]}. \\
\qquad \left(\begin{array}{l}
(\psi^{(\mathrm{PCT})}_{G_{\mathcal{A}},q_1}(x_1,(idx_{e,i})_{e\in E,i=1,2,3},(sum_{p,j},drop_{p,j})_{p\in Q,j=1,2}) \wedge \\
\psi^{(\mathrm{PCT})}_{G^{op}_{\mathcal{A}},q_2}(x_2,(idx_{e,i})_{e\in E,i=4,5,6},(sum_{p,j},drop_{p,j})_{p\in Q,j=3,4}))
\end{array}\right).
\end{array}$$

Finally, let x and y denote the initial and final counter values of state q and q' respectively. By combining formulae for the type-1, type-2 and type-3 reachability criteria, the EPA formula $\varphi^{(\mathrm{RC})}_{G_{\mathcal{A}},q,q'}$ is defined as

$$
\begin{array}{l}
\psi^{(\mathrm{RC})}_{G_{\mathcal{A}},q,q'}(x,y) ::= \exists x_1 x_2 \exists (y_{e,i})_{e\in E, i\in[3]}.\ x_1 \geq 0 \wedge x_2 \geq 0 \wedge \bigwedge_{e\in E, i\in[3]} y_{e,i} \geq 0 \wedge \\
\left(\begin{array}{l}
(q = q' \wedge x = x_1 \wedge x_1 = x_2 \wedge x_2 = y) \ \vee \\
\left(x = x_1 \wedge x_1 = x_2 \wedge \bigvee \psi^{(\mathrm{T2RC})}_{G_{\mathcal{A}},q,q'}(x_2, y, (y_{e,3})_{e\in E})\right) \ \vee \\
\left(x = x_1 \wedge \psi^{(\mathrm{T3RC})}_{G_{\mathcal{A}},q,q'}(x_1, x_2, (y_{e,2})_{e\in E}) \wedge x_2 = y\right) \ \vee \\
\bigvee\limits_{q_2\in Q} \left(x = x_1 \wedge \psi^{(\mathrm{T3RC})}_{G_{\mathcal{A}},q,q_2}(x_1, x_2, (y_{e,2})_{e\in E}) \wedge \psi^{(\mathrm{T2RC})}_{G_{\mathcal{A}},q_2,q'}(x_2, y, (y_{e,3})_{e\in E})\right) \ \vee \\
\left(\psi^{(\mathrm{T1RC})}_{G_{\mathcal{A}},q,q'}(x, x_1, (y_{e,1})_{e\in E}) \wedge x_1 = x_2 \wedge x_2 = y\right) \ \vee \\
\bigvee\limits_{q_1\in Q} \left(\psi^{(\mathrm{T1RC})}_{G_{\mathcal{A}},q,q_1}(x, x_1, (y_{e,1})_{e\in E}) \wedge x_1 = x_2 \wedge \psi^{(\mathrm{T2RC})}_{G_{\mathcal{A}},q_1,q'}(x_2, y, (y_{e,3})_{e\in E})\right) \ \vee \\
\bigvee\limits_{q_1\in Q} \left(\psi^{(\mathrm{T1RC})}_{G_{\mathcal{A}},q,q_1}(x, x_1, (y_{e,1})_{e\in E}) \wedge \psi^{(\mathrm{T3RC})}_{G_{\mathcal{A}},q_1,q'}(x_1, x_2, (y_{e,2})_{e\in E}) \wedge x_2 = y\right) \ \vee \\
\bigvee\limits_{q_1,q_2\in Q} \left(\begin{array}{l}
\psi^{(\mathrm{T1RC})}_{G_{\mathcal{A}},q,q_1}(x, x_1, (y_{e,1})_{e\in E}) \wedge \psi^{(\mathrm{T3RC})}_{G_{\mathcal{A}},q_1,q_2}(x_1, x_2, (y_{e,2})_{e\in E}) \wedge \\
\psi^{(\mathrm{T2RC})}_{G_{\mathcal{A}},q_2,q'}(x_2, y, (y_{e,3})_{e\in E})
\end{array}\right)
\end{array}\right).
\end{array}
$$

4 Experiments

We implement in Java the algorithm in the preceding Section and develop a tool OCAReach.[4] OCAReach computes, for a given OCA $\mathcal{A}$ and a pair of states q, q', an EPA formula $\varphi_{\mathcal{A},q,q'}(x,y)$ representing $R_{\mathcal{A},q,q'}$. Moreover, it integrates the SMT solver Z3 to eliminate the existential quantifiers in $\varphi_{\mathcal{A},q,q'}(x,y)$ as well as to solve the reachability problem from (q,n) to (q',n') for two additional $n, n' \in \mathbb{N}$, by evaluating $\varphi_{\mathcal{A},q,q'}(x,y)$ on n, n'. The performance of OCAReach are evaluated on two benchmark suites: MOCA, which is manually constructed, and ROCA, which is randomly generated.

MOCA We created 17 OCA benchmarks manually, of sizes ranging from (2 states, 1 transitions) to (10 states, 11 edges). The OCA instances in MOCA have relatively simple transition graphs so that for each instance $(\mathcal{A}, q, q')$ in MOCA, we are able to manually construct an EPA formula $\psi'_{\mathcal{A},q,q'}$ as the ground truth for the reachability relation, then use the SMT solver Z3 to test the equivalence of $\psi'_{\mathcal{A},q,q'}$ and $\psi_{\mathcal{A},q,q'}$ (the output of OCAReach), so that the correctness of OCAReach is validated.

ROCA This benchmark suite consists of randomly generated OCA instances by first determining the number of states n, then randomly generating the transitions, based on a sparsity parameter $\eta \in [0,1]$, with the intention that for each pair of states, there exist edges between them, with the probability η. Moreover, assuming that there exist edges between a given pair of states, then the probabilities of zero-transition, $+1$-transition, and -1-transition, are 1/8, 7/16, and 7/16 respectively. We first fix $\eta = 0.2$, and generate 50 instances for each $n \in \{5, 7, 10\}$. Then we fix $n = 4$ and generate 50 instances for each $\eta = 0.2, 0.4, 0.5$.

All the experiments were performed on a laptop with Intel Core i5-8450 processor and 8GB main memory.

[4] Available at https://github.com/SpencerL-Y/OCAReach.

Experimental results on MOCA. The results are given in Table 1, where time refers to the time to generate the EPA formula, and size refers to the size of the generated formula. We can see that the running time and the generated formula size are roughly proportional to the number of states and transitions. Moreover, for each MOCA instance, we use Z3 to validate the equivalence of the generated formula and the manually constructed ground truth formula.

state num.	2	2	2	2	3	3	4	4	4
transtion num.	1	2	2	5	2	3	3	3	6
zero-test num.	0	1	1	0	0	1	1	1	1
time (s)	0.066	0.062	0.078	0.076	0.066	0.072	0.061	0.079	0.093
size (kB)	0.302	0.404	0.697	0.302	0.133	0.929	0.348	0.325	2.592

state num.	5	6	6	6	7	8	10	10	
transtion num.	6	6	7	8	9	7	11	11	
zero-test num.	1	2	2	2	2	2	2	3	
time (s)	0.087	0.078	0.106	0.091	0.106	0.090	0.116	0.117	
size (kB)	2.057	2.469	7.457	3.078	6.427	4.807	8.443	7.515	

Table 1. Experimental results on MOCA

Experimental results on ROCA. The results are given in Table 2. We can see that when $\eta = 0.2$, if the the number of states n is increased from 5 to 10, then the average number of transitions, the average running time, and the average size of the generated formula grow quickly. Moreover, from the experimental results, we can also see that when the number of states $n = 4$, if the sparsity parameter η is increased from 0.2 to 0.5, then the average number of transitions, the average running time, and the average size of the generated formula also grow quickly. We remark that, in practice, the transition graphs of OCA are generally sparse so our approach is potentially scalable.

(state num. n, sparsity param. η)	(5, 0.2)	(7, 0.2)	(10, 0.2)	(4, 0.2)	(4, 0.3)	(4, 0.5)
Avg. transition num.	4	10	19	3	5.34	8.4
Avg. time (s)	0.012	16.161	362	0.021	0.492	23.334
Avg. size (kB)	6.29	4,470	37,241	4.823	3.161	235.140

Table 2. Experimental results on ROCA

5 Conclusion

In this paper, we have shown that the reachability relation of OCA can be represented by an existential Presburger arithmetic formula which can be computed in polynomial time. This result generalizes the well-known result that an existential Presburger arithmetic formula can be computed in polynomial time to define the Parikh image of the regular language of finite automata. We developed a tool OCAReach and conducted experiments to evaluate the efficiency of our approach. To the best of our knowledge, OCAReach provides the first tool support for solving the reachability problem of OCA.

References

1. P. A. Abdulla and K. Cerans. Simulation is decidable for one-counter nets (extended abstract). In D. Sangiorgi and R. de Simone, editors, *CONCUR '98: Concurrency Theory, 9th International Conference, Nice, France, September 8-11, 1998, Proceedings*, volume 1466 of *Lecture Notes in Computer Science*, pages 253–268. Springer, 1998.
2. S. Bardin, J. Leroux, and G. Point. FAST extended release. In T. Ball and R. B. Jones, editors, *Computer Aided Verification, 18th International Conference, CAV 2006, Seattle, WA, USA, August 17-20, 2006, Proceedings*, volume 4144 of *Lecture Notes in Computer Science*, pages 63–66. Springer, 2006.
3. M. Blondin, A. Finkel, S. Göller, C. Haase, and P. McKenzie. Reachability in two-dimensional vector addition systems with states is pspace-complete. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 32–43. IEEE Computer Society, 2015.
4. S. Böhm, S. Göller, and P. Jancar. Bisimilarity of one-counter processes is pspace-complete. In P. Gastin and F. Laroussinie, editors, *CONCUR 2010 - Concurrency Theory, 21th International Conference, CONCUR 2010, Paris, France, August 31-September 3, 2010. Proceedings*, volume 6269 of *Lecture Notes in Computer Science*, pages 177–191. Springer, 2010.
5. S. Böhm, S. Göller, and P. Jancar. Equivalence of deterministic one-counter automata is nl-complete. In D. Boneh, T. Roughgarden, and J. Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 131–140. ACM, 2013.
6. I. Borosh and L. B. Treybig. Bounds on positive integral solutions of linear diophantine equations. *Proceedings of the American Mathematical Society*, 55(2):299–304, 1976.
7. A. Bouajjani, M. Bozga, P. Habermehl, R. Iosif, P. Moro, and T. Vojnar. Programs with lists are counter automata. In T. Ball and R. B. Jones, editors, *Computer Aided Verification, 18th International Conference, CAV 2006, Seattle, WA, USA, August 17-20, 2006, Proceedings*, volume 4144 of *Lecture Notes in Computer Science*, pages 517–531. Springer, 2006.
8. C. Chitic and D. Rosu. On validation of XML streams using finite state machines. In S. Amer-Yahia and L. Gravano, editors, *Proceedings of the Seventh International Workshop on the Web and Databases, WebDB 2004, June 17-18, 2004, Maison de la Chimie, Paris, France, Colocated with ACM SIGMOD/PODS 2004*, pages 85–90. ACM, 2004.
9. H. Comon and Y. Jurski. Timed automata and the theory of real numbers. In J. C. M. Baeten and S. Mauw, editors, *CONCUR '99: Concurrency Theory, 10th International Conference, Eindhoven, The Netherlands, August 24-27, 1999, Proceedings*, volume 1664 of *Lecture Notes in Computer Science*, pages 242–257. Springer, 1999.
10. Z. Dang. Pushdown timed automata: a binary reachability characterization and safety verification. *Theor. Comput. Sci.*, 302(1-3):93–121, 2003.
11. S. Demri and R. Gascon. The effects of bounding syntactic resources on presburger LTL. In *14th International Symposium on Temporal Representation and Reasoning (TIME 2007), 28-30 June 2007, Alicante, Spain*, pages 94–104. IEEE Computer Society, 2007.
12. C. Dima. Computing reachability relations in timed automata. In *17th IEEE Symposium on Logic in Computer Science (LICS 2002), 22-25 July 2002, Copenhagen, Denmark, Proceedings*, page 177. IEEE Computer Society, 2002.

13. J. Fearnley and M. Jurdzinski. Reachability in two-clock timed automata is pspace-complete. *Inf. Comput.*, 243:26–36, 2015.
14. M. Fränzle, K. Quaas, M. Shirmohammadi, and J. Worrell. Effective definability of the reachability relation in timed automata. *Inf. Process. Lett.*, 153, 2020.
15. S. Göller, C. Haase, J. Ouaknine, and J. Worrell. Model checking succinct and parametric one-counter automata. In S. Abramsky, C. Gavoille, C. Kirchner, F. M. auf der Heide, and P. G. Spirakis, editors, *Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part II*, volume 6199 of *Lecture Notes in Computer Science*, pages 575–586. Springer, 2010.
16. S. Göller and M. Lohrey. Branching-time model checking of one-counter processes and timed automata. *SIAM J. Comput.*, 42(3):884–923, 2013.
17. S. Göller, R. Mayr, and A. W. To. On the computational complexity of verifying one-counter processes. In *Proceedings of the 24th Annual IEEE Symposium on Logic in Computer Science, LICS 2009, 11-14 August 2009, Los Angeles, CA, USA*, pages 235–244. IEEE Computer Society, 2009.
18. C. Haase. *On the complexity of model checking counter automata*. PhD thesis, 2012.
19. C. Haase. Subclasses of presburger arithmetic and the weak exp hierarchy. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, CSL-LICS '14, New York, NY, USA, 2014. Association for Computing Machinery.
20. C. Haase, S. Kreutzer, J. Ouaknine, and J. Worrell. Reachability in succinct and parametric one-counter automata. In *Proceedings of the 20th International Conference on Concurrency Theory*, CONCUR 2009, pages 369–383. Springer-Verlag, 2009.
21. C. Haase, J. Ouaknine, and J. Worrell. Relating reachability problems in timed and counter automata. *Fundam. Inform.*, 143(3-4):317–338, 2016.
22. M. Hague and A. W. Lin. Model checking recursive programs with numeric data types. In *Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings*, pages 743–759, 2011.
23. O. H. Ibarra. Reversal-bounded multicounter machines and their decision problems. *J. ACM*, 25(1):116–133, 1978.
24. P. Jancar, A. Kucera, F. Moller, and Z. Sawa. DP lower bounds for equivalence-checking and model-checking of one-counter automata. *Inf. Comput.*, 188(1):1–19, 2004.
25. A. Kucera. Efficient verification algorithms for one-counter processes. In U. Montanari, J. D. P. Rolim, and E. Welzl, editors, *Automata, Languages and Programming, 27th International Colloquium, ICALP 2000, Geneva, Switzerland, July 9-15, 2000, Proceedings*, volume 1853 of *Lecture Notes in Computer Science*, pages 317–328. Springer, 2000.
26. P. Lafourcade, D. Lugiez, and R. Treinen. Intruder deduction for AC-like equational theories with homomorphisms. In J. Giesl, editor, *Term Rewriting and Applications, 16th International Conference, RTA 2005, Nara, Japan, April 19-21, 2005, Proceedings*, volume 3467 of *Lecture Notes in Computer Science*, pages 308–322. Springer, 2005.
27. J. Leroux and G. Sutre. Flat counter automata almost everywhere! In D. A. Peled and Y. Tsay, editors, *Automated Technology for Verification and Analysis, Third International Symposium, ATVA 2005, Taipei, Taiwan, October 4-7, 2005,*

Proceedings, volume 3707 of *Lecture Notes in Computer Science*, pages 489–503. Springer, 2005.

28. H. Seidl, T. Schwentick, A. Muscholl, and P. Habermehl. Counting in trees for free. In *ICALP*, pages 1136–1149, 2004.
29. A. Smrcka and T. Vojnar. Verifying parametrised hardware designs via counter automata. In K. Yorav, editor, *Hardware and Software: Verification and Testing, Third International Haifa Verification Conference, HVC 2007, Haifa, Israel, October 23-25, 2007, Proceedings*, volume 4899 of *Lecture Notes in Computer Science*, pages 51–68. Springer, 2007.
30. Z. Xu, T. Chen, and Z. Wu. Satisfiability of compositional separation logic with tree predicates and data constraints. In L. de Moura, editor, *Automated Deduction - CADE 26 - 26th International Conference on Automated Deduction, Gothenburg, Sweden, August 6-11, 2017, Proceedings*, volume 10395 of *Lecture Notes in Computer Science*, pages 509–527. Springer, 2017.