On the Automata Size for Presburger Arithmetic

Felix Klaedtke

Institut für Informatik, Albert-Ludwigs-Universität
klaedtke@informatik.uni-freiburg.de

Abstract. We prove that the number of states of the minimal automaton for a Presburger arithmetic formula is triple exponentially bounded in the length of the formula and we show that this triple exponential bound is tight. Furthermore, we improve the automata constructions for linear (in)equations found in the literature and show that our constructions are optimal.

1 Introduction

Presburger arithmetic is the first-order theory with addition and the ordering relation over the integers. Its decidability was first proved independently by M. Presburger [14, 15] and T. Skolem [22, 23] using the method of *quantifier elimination*. And its complexity has been investigated intensively since. For example, Fischer and Rabin [9] gave a lower bound on any decision procedure for Presburger arithmetic, namely double exponential in nondeterministic time, and in [13], Oppen showed that Cooper's quantifier elimination decision procedure [7] has a triple exponential worst-case complexity in deterministic time.

Another approach for deciding Presburger arithmetic or fragments of it that has recently become popular is to use automata; an idea that was already noted by Büchi in [6]. The idea of the automata-based approach is simple: Integers are represented as words, e. g., using the 2's complement representation and the automaton for a Presburger arithmetic formula accepts precisely the words that represent the integers making the formula true. The automaton can be recursively constructed from the formula, where automata operations handle the logical connectives and quantifiers. Specific algorithms for constructing automata for linear (in)equations have been developed in [1, 3, 5, 10, 26].

A crude complexity analysis on automata-based decision procedures leads to a non-elementary worst-case complexity. Namely, for every quantifier alternation there is an exponential blow-up in the worst-case. However, experimental comparisons [1, 10, 17] illustrate that automata-based decision procedures for Presburger arithmetic have competitive performance in comparison to other methods based on quantifier elimination techniques, like e. g., the Omega test [16]. The paper [5] claims that the minimal deterministic automaton for an arbitrary Presburger arithmetic formula has at most triple exponential states in the length of the formula. Unfortunately, the argument used in [5] to substantiate this claim is incorrect as explained in [26]. The main objective of this paper is to give a correct proof of this claim. Wolper and Boigelot gave in [26] an argument why there must be an upper bound on the size of the minimal automaton for a Presburger arithmetic formula. But Wolper and Boigelot's argument shows only the *existence* of an upper bound. Note that this is in contrast to automata-based decision procedures for the monadic second-order logic with "+1", or the first-order logic with "$<$" over finite words, where the minimal automaton for a formula can become non-elementary larger [19, 24] than the formula length.

In this paper, we give a detailed analysis of the automata for Presburger arithmetic formulas by comparing the constructed automata with the quantifier-free formulas constructed by the quantifier elimination method in [18] that is an improvement of Cooper's quantifier elimination method. Our main result is that there is a constant $c \geq 0$ such that the minimal deterministic automaton for an arbitrary Presburger arithmetic formula of length n has at most $2^{2^{2^{cn}}}$ states. Furthermore, we show that the triple exponential upper bound on the automata size for Presburger arithmetic formulas is tight.

Our analysis also provides (1) improvements of the automata constructions in [10, 26] for linear (in)equations, (2) optimality results of these constructions, and (3) a better understanding of the complexity of Presburger arithmetic, in particular for automata-based decision procedures. This improved understanding is also helpful for optimizing decision procedures using quantifier elimination.

The paper is organized as follows. In §2 we recall preliminaries. In §3 we describe the automata constructions for Presburger arithmetic formulas. In §4 we prove the upper bound on the automata size for arbitrary formulas, and in §5 we give a worst-case example. Finally, in §6 we draw conclusions.

2 Preliminaries

In this section, we recall common notations and definitions that we will use.

2.1 Automata

An (deterministic) **automaton** $\mathcal{A}$ is a tuple $(Q, \Sigma, q_{\mathrm{I}}, \delta, F)$, where Q is a finite set of states, Σ is a non-empty alphabet, $q_{\mathrm{I}} \in Q$ is the initial states, $\delta : Q \times \Sigma \to Q$ is the transition function, and $F \subseteq Q$ is the set of accepting states. A state $q \in Q$ is a **sink state** if $\delta(q, b) = q$, for all $b \in \Sigma$. The transition function δ is extended to words as follows. We define $\delta(q, \lambda) := q$, and $\delta(q, bw) := \delta(\delta(q, b), w)$, for $q \in Q$, $b \in \Sigma$, and $w \in \Sigma^*$. The **language** of $\mathcal{A}$ is the set $L(\mathcal{A}) := \{w \in \Sigma^* \mid \delta(q_{\mathrm{I}}, w) \in F\}$. A state $q \in Q$ is **reachable** from state $p \in Q$ if there is a word $w \in \Sigma^*$ with $\delta(p, w) = q$. The **size** of $\mathcal{A}$ is the cardinality of Q. The automaton $\mathcal{A}$ is **minimal** if there is not an automaton $\mathcal{B}$ with smaller size than $\mathcal{A}$ and $L(\mathcal{B}) = L(\mathcal{A})$. Note that the minimal automaton is uniquely determined up to isomorphism.

A **pre-automaton** $\mathcal{A}$ is an automaton without specifying the set of accepting states. The **product** of the pre-automata $\mathcal{A} = (Q, \Sigma, q_{\mathrm{I}}, \delta)$ and $\mathcal{A}' = (Q', \Sigma, q'_{\mathrm{I}}, \delta')$ is the pre-automaton $\mathcal{A} \times \mathcal{A}' := (Q \times Q', \Sigma, (q_{\mathrm{I}}, q'_{\mathrm{I}}), \eta)$ where the transition function is defined as $\eta((q, q'), b) := (\delta(q, b), \delta'(q', b))$, for $q \in Q$, $q' \in Q'$, and $b \in \Sigma$. Intuitively, $\mathcal{A}$ and $\mathcal{A}'$ run in parallel in $\mathcal{A} \times \mathcal{A}'$.

2.2 Presburger Arithmetic

Presburger arithmetic is the first-order logic over the structure $\mathfrak{Z} := (\mathbb{Z}, <, +)$.[1] As logical constructs, we allow the quantifiers $\exists$ and $\forall$, and the connectives $\neg$, $\vee$, $\wedge$, $\rightarrow$, and $\leftrightarrow$. Note that without changing the expressive power we can restrict ourselves to the logical constructs $\exists$, $\neg$, and $\vee$. But this restriction comes with the price of a blow-up in the length of the formulas. In particular, eliminating the connective $\leftrightarrow$ can produce formulas that are exponentially longer. We come back to this point in §4.3, where we apply quantifier elimination techniques.

We extend the logical language by (i) constants for the naturals 0 and 1, (ii) the unary function symbol "$-$" for the opposite of an integer, and (iii) for every $d > 0$, we extend the language by a predicate "$d|$", where $d|u$ is saying for $u \in \mathbb{Z}$ that u is a multiple of d, i.e., there is a $k \in \mathbb{Z}$ such that $kd = u$. It is straightforward to define these constants, predicates, and the unary function "$-$" in Presburger arithmetic. For instance, the formula $\exists x(x + \cdots + x = t)$ defines $d|t$, where x is repeated d times in the term $x + \cdots + x$, and x is a fresh variable, i.e., x does not appear in the term t.

The motivation for the extended logical language, where (i), (ii), and (iii) are handled as primitives, is that it admits quantifier elimination [7, 14, 22], i.e., for a formula $\exists x \varphi(x, y_1, \dots, y_r)$, where φ is quantifier-free, we can construct a logically equivalent quantifier-free formula $\psi(y_1, \dots, y_r)$. Recall that the formulas $\chi(x_1, \dots, x_r)$ and $\chi'(x_1, \dots, x_r)$ are **logically equivalent** if $\mathfrak{Z} \models \chi[a_1, \dots, a_r] \Leftrightarrow \mathfrak{Z} \models \chi'[a_1, \dots, a_r]$, for every $a_1, \dots, a_r \in \mathbb{Z}$. There are several quantifier elimination methods for Presburger arithmetic over the extended logical language. In §4 we use the quantifier elimination method in [18] that is an improvement of Cooper's quantifier elimination method [7].

For convenience, we permit the use of several other symbols in writing terms and formulas. For example, 2 stands for $1 + 1$ and, in general, $k > 0$ stands for $1 + \cdots + 1$ (repeated k times), and $-(1 + \cdots + 1)$ for $k < 0$. We identify the terms k and $-k$ with the integers that they represent. Analogously, we write, e.g., $2 \cdot x$ for $x + x$ and, in general, $k \cdot x$ stands for $x + \cdots + x$ (repeated k times) if $k > 0$, and $-(x + \cdots + x)$ (repeated k times) if $k < 0$. In the remainder of the text, we assume that a term t is always of the form $k_1 \cdot x_1 + \cdots + k_r \cdot x_r + c$ where $c \in \mathbb{Z}$, $k_1, \dots, k_r \in \mathbb{Z} \setminus \{0\}$, and the variables $x_1, \dots, x_r$ are pairwise distinct. The k_is are the **coefficients** and c is the **constant**. If $c = 0$ we omit it. We define

$$||t||_- := \sum_{1 \le j \le r \text{ and } k_j < 0} |k_j| \qquad \text{and} \qquad ||t||_+ := \sum_{1 \le j \le r \text{ and } k_j > 0} k_j \,.$$

[1] In the literature, "Presburger arithmetic" sometimes means the first-order theory with addition over the *natural numbers*. This difference is immaterial. On the one hand, the restriction to the natural numbers in $\mathfrak{Z}$ is obvious. On the other hand, we can encode the integers in the natural numbers, where an even number $2n$ represents the positive integer n, and an odd number $2n+1$ represents the negative integer $-n$. Addition and the ordering relation on the integers are definable for this encoding.

Moreover, let $\|t\| := \|t\|_- + \|t\|_+$. The **greatest common divisor** of t, denoted by $\gcd(t)$, is the greatest common divisor of the coefficients of t, i. e. $\gcd(t) := \gcd(|k_1|, \dots, |k_r|)$. The term t is **homogeneous** if $r \geq 1$ and $c = 0$. $k \cdot t$ denotes the term where the constant and the coefficients of the term t are multipled by $k \in \mathbb{Z}$.

In the atomic formulas, we also allow the relations $\leq, >, \geq$, and $\neq$. Let $\lesseqgtr \in \{=, \neq, <, \leq, >, \geq\}$. If we do not say explicitly that an atomic formula α has a certain form, then we assume that α is in **normal form**, i. e., α is in one of the forms

$$1 < 0\,, \qquad 0 < 1\,, \qquad d \big| t + c\,, \qquad \text{or} \qquad t \lesseqgtr c\,,$$

where $c \in \mathbb{Z}$ and t is a homogeneous term. We omit the constant c in $d \big| t + c$ if $c = 0$.

The **maximal absolute integer** of a formula φ is the maximal absolute coefficient or constant occurring in a term of φ, or the d in a subformula of φ of the form $d \big| t + c$. We use the same definition of the length of a formula as in [9, 13, 18] since we refer to results in there. The **length** of a formula is the number of letters used in writing the formula, where integers are measured in unary. Note that the length of a formula depends significantly on this encoding since we actually need only logarithmically many digits to write down an integer k. For instance, $x = 11 \cdot y$ contains $2 + 4$ letters, but the "expanded version" has $2+11+10$ letters since $11 \cdot y$ abbreviates the term $y+y+y+y+y+y+y+y+y+y+y$. However, it is possible to express $k \cdot x$ by a formula of length $\mathrm{O}(\lg |k|)$. The idea is illustrated by $y = 11 \cdot x$: the formula is logically equivalent to $\exists z(y = z + z + x \wedge \exists y(z = y + y + x \wedge y = x + x))$. It is important here that variable reuse occur so that only a fixed number of variables are needed for any k, see [9]. For the sake of uniformity, we define the length of the formula $d \big| t$ has the length of the term t plus $d + 1$. Again, there is a logically equivalent formula of length $\mathrm{O}(\lg d)$ plus the length of t. For the results in this paper it does not matter if we measure the length of an integer k as $\mathrm{O}(\lg |k|)$ or as $\mathrm{O}(|k|)$.

3 Automata Constructions

In this section, we provide insights in the automata constructed for Presburger arithmetic formulas.

3.1 Representing Sets of Integers with Automata

In order to use automata to recognize tuples of numbers, we give a mapping from words to tuples of numbers. We use an encoding scheme that is based on the 2's complement representation of integers, where the most significant bit is the first digit, i. e., the word $b_n b_{n-1} \dots b_0 \in \{0,1\}^*$ encodes the integer $-2^n b_n + b_{n-1} 2^{n-1} + \dots + b_0 2^0$. A word $\overline{b}_n \overline{b}_{n-1} \dots \overline{b}_0 \in (\{0,1\}^r)^*$ represents the tuple $(x_1, \dots, x_r) \in \mathbb{Z}^r$ of integers, where the ith "track" of the word encodes

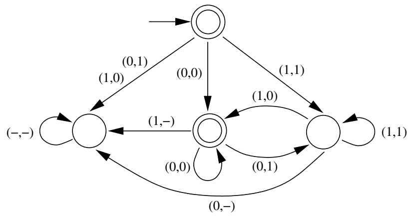

Fig. 1. Automaton representing the set $\{(x, y) \in \mathbb{Z}^2 \,|\, y = 2x\}$.

the integer x_i, i.e., $x_i = -2^n b_{n,i} + 2^{n-1} b_{n-1,i} + \cdots + b_{0,i}$ with $1 \leq i \leq r$ and $\overline{b}_j = (b_{j,1}, \ldots, b_{j,r})$ for $0 \leq j \leq n$. We also say that $\overline{b}_n \overline{b}_{n-1} \ldots \overline{b}_0$ represents $(x_1, \ldots, x_r)$. This encoding has been also used in [3, 10, 26].

A natural choice of representing sets of tuples of integers as languages is the following [26]. The set $U \subseteq \mathbb{Z}^r$ is **represented** by the language $L \subseteq (\{0,1\}^r)^*$ if for every $\overline{x} \in \mathbb{Z}^r$ it holds that $\overline{x} \in U$ iff *all* words that represent $\overline{x}$ are in L. The reason for the requirement that *all* encodings of the elements in U have to be in the set L is that the representation of $\mathbb{Z}^r \setminus U$ is just the complement of L, i.e. $(\{0,1\}^r)^* \setminus L$. An automaton $\mathcal{A}$ **represents** the set $U \subseteq \mathbb{Z}^r$ if $L(\mathcal{A})$ represents U. Note that under this definition not every language over $\{0,1\}^r$ encodes a set of tuples of integers, and not every automaton with alphabet $\{0,1\}^r$ represents a subset of $\mathbb{Z}^r$. The language $L \subseteq (\{0,1\}^r)^*$ encodes a subset of $\mathbb{Z}^r$ iff L satisfies the two properties $\lambda \in L \Leftrightarrow \overline{0} \in L$ and $\overline{b}u \in L \Leftrightarrow \overline{b}\overline{b}u \in L$, for all $u \in (\{0,1\}^r)^*$ and $\overline{b} \in \{0,1\}^r$.

Example 1. The set of pairs $(x, y) \in \mathbb{Z}^2$ where y equals $2x$ is represented by the automaton depicted in Figure 1.

Note that other encodings are possible. For instance, it is possible to interpret the first letter of a word as the least significant bit [1, 5]. This corresponds to reversing the language. On the one hand, the automata size can be exponentially shorter. For example, the family of formulas for $n \geq 1$,

$$\begin{aligned}\varphi_n(y) := y \geq 0 \wedge \exists x_0 \ldots \exists x_{n-1} \big(& \textstyle\bigwedge_{0 \leq i < n} (x_i = 0 \vee x_i = 2^i) \wedge \\ & \textstyle\bigwedge_{0 \leq i < n} (2^{i+1} | y - (x_0 + \cdots + x_i)) \wedge \\ & \neg 2^{n+1} | y - (x_0 + \cdots + x_{n-1}) \big)\end{aligned}$$

encode the language of words over $\{0,1\}$ where the $(n+1)$st first letter is 1 if we use the encoding where the least significant bit is the first letter. It is straightforward to define an automaton for this language that has $\mathrm{O}(n)$ states. It is well-known that the minimal automaton for the reverse language has $2^{\mathrm{O}(n)}$ states. On the other hand, some experiments show that the minimal automaton is often larger using the encoding where the first letter is the least significant bit. It is an open problem how the two encodings are related and which encoding is superior in practice. We want to point out that the results in this paper rely on the 2's complement encoding where the first letter is the most significant bit.

3.2 Linear Equations and Inequations

In this subsection, we give automata constructions for (in)equations of the form $t \lessgtr c$, where $t(x_1, \dots, x_r)$ is a homogeneous term, $\lessgtr \in \{=, \neq, <, \leq, >, \geq\}$, and $c \in \mathbb{Z}$. Similar constructions for this kind of formulas can be found in [1,5,10,26]. We build on the constructions in [10,26] and optimize them. Additionally, we prove that our construction are optimal in the sense that the constructed automaton is minimal.

We recall the underlying idea of the constructions in [10,26]. First, note the following observation for a word $w \in (\{0,1\}^r)^*$ encoding the tuple of integers $\overline{v} \in \mathbb{Z}^r$. Let $w' := w\overline{b}$ for some $\overline{b} \in \{0,1\}^r$. If w is not the empty word then the word w' represents the tuple of integers $2\overline{v} + \overline{b}$. For the special case where w is the empty word, w' represents the tuple of integers $-\overline{b}$, since we are using 2's complement representation. Given this, it is relatively straightforward to obtain an automaton $\mathcal{A}$ with an *infinite* set of states for $t \lessgtr c$. The set of state is $\{q_\mathrm{I}\} \cup \mathbb{Z}$ where q_I is the initial state. Note that we identify integers with states. The idea is to keep track of the value of t as successive bits are read. Thus, except for the special initial state, a state in $\mathbb{Z}$ represents the current value of t. Lemma 2 justifies this intuition for a state. The transition function δ of $\mathcal{A}$ is defined as follows for a letter $\overline{b} \in \{0,1\}^r$. For $q \in \mathbb{Z}$, we define

$$\delta(q, \overline{b}) := 2q + t(\overline{b}) \,,$$

and for the initial state, $\delta(q_\mathrm{I}, \overline{b}) := -t(\overline{b})$. The set of accepting states is

$$F := \{q \in \mathbb{Z} \mid q \lessgtr c\} \cup \begin{cases} \{q_\mathrm{I}\} & \text{if } 0 \lessgtr c, \\ \emptyset & \text{otherwise.} \end{cases}$$

Lemma 2. *For* $\overline{b}_0, \dots, \overline{b}_n \in \{0,1\}^r$,

(a) $\delta(q, \overline{b}_n \overline{b}_{n-1} \dots \overline{b}_0) = q2^{n+1} + t\big(\sum_{0 \leq i \leq n} 2^i \overline{b}_i\big)$*, where* $q \in \mathbb{Z}$*, and*
(b) $\delta(q_\mathrm{I}, \overline{b}_n \overline{b}_{n-1} \dots \overline{b}_0) = t(-2^n \overline{b}_n + \sum_{0 \leq i < n} 2^i \overline{b}_i)$.

Proof. (a) is easily proven by induction over n, and (b) follows directly from the definition $\delta(q_\mathrm{I}, \overline{b}_n) = -t(\overline{b}_n)$ and (a). ⊣

The constructions in [10,26] are based on the observation that states $q, q' \in \mathbb{Z}$ of $\mathcal{A}$ can be merged if, intuitively speaking, q and q' are both small or both are large. Here, the meaning "small" and "large" depend on the coefficients of t and on the constant c. More precisely, we say that $q \in \mathbb{Z}$ is **small** if $q < \min\{c, -\|t\|_+\}$, and it is **large** if $q > \max\{c, \|t\|_-\}$. The only difference between the construction in [26] and [10] are the bounds that determine the meaning of "small" and "large". Note that from a small value we can only obtain smaller values, and from a large value we can only obtain larger values, i.e., for all $\overline{b} \in \{0,1\}^r$, if $q > \|t\|_-$ then $2q + t(\overline{b}) > q$, and if $q < -\|t\|_+$ then $2q + t(\overline{b}) < q$. We obtain an automaton $\mathcal{B}$ with the set of states

$$Q := \big\{q_\mathrm{I}, q_{\mathrm{small}}, \min\{c, -\|t\|_+\}, \dots, \max\{c, \|t\|_-\}, q_{\mathrm{large}}\big\} \,.$$

The correctness of $\mathcal{B}$ follows from the following lemma which is an adoption of Lemma 2 to $\mathcal{B}$.

Lemma 3. *Let* $\overline{b}_0, \dots, \overline{b}_n \in \{0,1\}^r$.

(a) For $q \in Q \cap \mathbb{Z}$, *it holds that*

$$\delta(q, \overline{b}_n \overline{b}_{n-1} \dots \overline{b}_0) = \begin{cases} q_{\text{small}} & \textit{if } s < \min\{c, -\|t\|_-\}, \\ q_{\text{large}} & \textit{if } s > \max\{c, \|t\|_+\}, \\ s & \textit{otherwise,} \end{cases}$$

where s *is* $q2^{n+1} + t\big(\sum_{0 \le i \le n} 2^i \overline{b}_i\big)$.

(b) It holds that

$$\delta(q_{\mathrm{I}}, \overline{b}_n \overline{b}_{n-1} \dots \overline{b}_0) = \begin{cases} q_{\text{small}} & \textit{if } s < \min\{c, -\|t\|_-\}, \\ q_{\text{large}} & \textit{if } s > \max\{c, \|t\|_+\}, \\ s & \textit{otherwise,} \end{cases}$$

where s *is* $t(-2^n \overline{b}_n + \sum_{0 \le i < n} 2^i \overline{b}_i)$.

From this construction [10, 26] we obtain the following upper bounds of the size of the minimal automaton.

Theorem 4. *The minimal automaton for an (in)equation* $t \lessgtr c$ *has at most* $3 + \max\{c, \|t\|_-\} + \max\{|c|, \|t\|_+\}$ *states.*

In §3.4 we need the following definition. The pre-automaton $\mathcal{A}^t_{[c,c']}$ for the homogeneous term t and $c < c'$ is the pre-automaton obtained from the automaton (with an infinite number of states) $\mathcal{A}$, where the states $q \le c$ are collapsed into a single state q_{small} and the states $q \ge c'$ are collapsed into a single state q_{large}. Note that the pre-automaton of the automaton $\mathcal{B}$ is the pre-automaton $\mathcal{A}^t_{[\min\{c, -\|t\|_+\}-1, \max\{c, \|t\|_-\}+1]}$.

For establishing the upper bound on the automata size for Presburger arithmetic formulas, we make use only of the upper bound in Theorem 4. In the following subsubsection, we optimize the automata construction [10, 26] described above in several ways.

Optimized Constructions

Before we optimize the automaton construction [10, 26] for (in)equations, we need the following definitions. Let $\mathcal{A}$ be an automaton $\mathcal{A} = (Q, \Sigma, q_{\mathrm{I}}, \delta, F)$ where all states are reachable from the initial state q_{I}. We define an equivalence relation on Q as follows. The states $p, q \in Q$ are equivalent, $p \sim_{\mathcal{A}} q$ for short, if for all words $w \in \Sigma^*$, $\delta(p, w) \in F$ iff $\delta(q, w) \in F$. We denote by $\widetilde{q}$ the equivalence class of $q \in Q$. The states that are equivalent can be merged. The minimal automaton of $\mathcal{A}$ is the automaton $\widetilde{\mathcal{A}} := (\{\widetilde{q} \mid q \in Q\}, \Sigma, \widetilde{q}_{\mathrm{I}}, \widetilde{\delta}, \{\widetilde{q} \mid q \in F\})$ where the transition function is defined as $\widetilde{\delta}(\widetilde{q}, b) := \widetilde{\delta(q, b)}$. If $p \not\sim_{\mathcal{A}} q$ then $\widetilde{p} \neq \widetilde{q}$. This means that two states of $\mathcal{A}$ that are not equivalent with respect to $\sim_{\mathcal{A}}$ determine different states of the minimal automaton.

Throughout this subsubsection, let $t(x_1, \dots, x_r) \lessgtr c$ be an (in)equation, and let $\mathcal{B} = (Q, \{0,1\}^r, q_\mathrm{I}, \delta, F)$ be the automaton for this (in)equation that is obtained by the above explained construction [10, 26]. Recall that Q is the set of states $\{q_\mathrm{I}, q_\mathrm{small}, \min\{c, -\|t\|_+\}, \dots, \max\{c, \|t\|_-\}, q_\mathrm{large}\}$. Let $S := \{-\|t\|_+ + 1, \dots, \|t\|_- - 1\}$.

Without loss of generality, we can restrict ourselves to the cases where $\lessgtr \in \{=, <, >\}$. The cases with $\lessgtr \in \{\neq, \leq, \geq\}$ reduce to automata complementation and the cases for $=$, $<$, and $>$ since $t \neq c$ is logically equivalent to $\neg t = c$, $t \leq c$ is logically equivalent to $\neg t > c$, and $t \geq v$ is logically equivalent to $\neg t < c$. Note that complementation of an automaton can be done be flipping accepting and non-accepting states. The complemented automaton is minimal iff the original automaton is minimal.

Optimization #1. An obvious optimization is to filter out the states in $Q \cap \mathbb{Z}$ that are not a multiple of $\gcd(t)$ since they are not reachable from the initial state q_I.

Lemma 5. *Let $q \in Q \cap \mathbb{Z}$. The state $q \in Q \cap \mathbb{Z}$ is reachable from the initial state q_I iff q is a multiple of* $\gcd(t)$.

Proof. ($\Rightarrow$). This direction is easy to prove by induction: for all $\overline{b} \in \{0,1\}^r$, it holds that (i) $-t(\overline{b})$ is a multiple of $\gcd(t)$, and (ii) if $p \in \mathbb{Z}$ is a multiple of $\gcd(t)$ then $2p + t(\overline{b})$ is a multiple of $\gcd(t)$.

($\Leftarrow$). Assume that q is a multiple of $\gcd(t)$. There are $v_1, \dots, v_r \in \mathbb{Z}$ with $t(v_1, \dots, v_r) = q$. Let $\overline{b}_n \dots \overline{b}_0 \in (\{0,1\}^r)^*$ be a 2's complement representation of $(v_1, \dots, v_r) \in \mathbb{Z}^r$. From Lemma 3(b) it follows that $\delta(q_\mathrm{I}, \overline{b}_n \dots \overline{b}_0) = t(v_1, \dots, v_r) = q$. ⊣

Alternatively, instead of filtering out the states $q \in \mathbb{Z}$ that are not a multiple of $\gcd(t)$ we can rewrite the (in)equation $t \lessgtr c$ to the logically equivalent atomic formula α and then construct the automaton for α, where α is defined as

$$\alpha := \begin{cases} t' \lessgtr \left\lceil \frac{c}{\gcd(t)} \right\rceil & \text{if } \lessgtr \text{ is } <, \\ t' \lessgtr \left\lfloor \frac{c}{\gcd(t)} \right\rfloor & \text{if } \lessgtr \text{ is } >, \\ t' \lessgtr \frac{c}{\gcd(t)} & \text{if } \lessgtr \text{ is } = \text{ and } c \text{ is a multiple of } \gcd(t), \\ 1 < 0 & \text{otherwise,} \end{cases}$$

where the coefficients in t' are the coefficients of t divided by $\gcd(t)$. In the following, we assume without loss of generality that $\gcd(t) = 1$. By this assumption all states in $\mathcal{B}$ are reachable from q_I.

Lemma 6. *Let $p \in S$.*

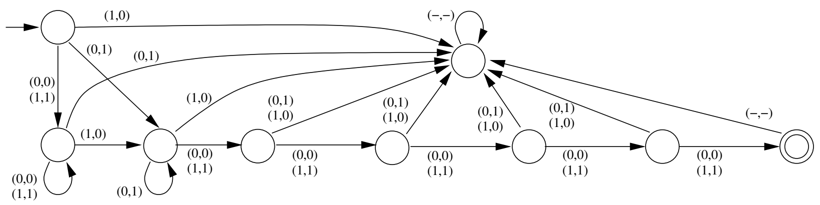

Fig. 2. Minimal automaton for the equation $x - y = 32$.

(a) Every state in $Q \setminus \{q_{\mathrm{I}}\}$ is reachable from p.
(b) For $q \in Q \setminus \{q_{\mathrm{I}}\}$, it holds that if $p \neq q$ then $p \not\sim_{\mathcal{B}} q$.

Proof. (a). See the appendix.

(b). First, assume that $q = q_{\mathrm{large}}$. Since q is a sink state and from p we can reach accepting and non-accepting states (see (a)), it follows that $p \not\sim_{\mathcal{B}} q$. The case $q = q_{\mathrm{small}}$ is proven analogously.

Now, assume that $q \in Q \cap \mathbb{Z}$. Without loss of generality assume that $p < q$. By (a) there is a word $u := \overline{b}_n \dots \overline{b}_0$ with $\delta(p, u) = c$. It holds that

$$c = \delta(p, u) = p2^{n+1} + t(\sum_{0 \leq i \leq n} 2^i \overline{b}_i) < q2^{n+1} + t(\sum_{0 \leq i \leq n} 2^i \overline{b}_i)\,.$$

This means that $\delta(q, u)$ is either q_{large}, or $\delta(p, u) < \delta(q, u)$. In the case where $\lessgtr$ is $=$, we have that $\delta(p, u) \in F$ and $\delta(q, u) \notin F$. If $\lessgtr$ is $>$, we have that $\delta(p, u) \notin F$ and $\delta(q, u) \in F$. In the case where $\lessgtr$ is $<$, we can conclude similarly that $p \not\sim_{\mathcal{B}} q$ by looking at a word u' with $\delta(q, u') = c$. ⊣

From the lemma it follows that we cannot merge any state in $Q \setminus \{q_{\mathrm{I}}\}$ with a state in S. The automata construction for $t \lessgtr c$ can only be optimized for states in $Q \setminus S$.

Corollary 7. *The minimal automaton for $t \lessgtr c$ has at least $\|t\| - 1$ states.*

Optimization #2. The following example illustrates that the automaton construction in [10, 26] for an equation $t = c$ produce too many states if c is significantly larger than $\|t\|_-$ or c is significantly smaller than $-\|t\|_+$.

Example 8. Consider the equation $x - y = 32$. The construction in [10, 26] yields an automaton with the set of states $\{q_{\mathrm{I}}, q_{\mathrm{small}}, -1, 0, \dots, 32, q_{\mathrm{large}}\}$. The only accepting state is 32, and state 32 is only reachable from the states $q_{\mathrm{I}}, 1, 2, 4, 8, 16, 32$. All the other states can be merged to a single non-accepting sink state. Figure 2 shows the minimal automaton for $x - y = 32$.

Motivated by the example, an optimized construction for an equation is to identify the states from which the accepting states are not reachable. In the following, we exclude the special cases $x = 0$, $-x = 0$, $x - y = 0$, and $-x + y = 0$. It is straightforward to give minimal automata for these four equations.

We construct an automaton $\mathcal{B}'$ for the equation $t = c$ as follows. We first determine the set R of states in $Q \cap \mathbb{Z}$ from which state c is reachable, and then we construct the transition function. Let R' be the set $\{\min\{c, -\|t\|_+ + 1\}, \dots, \max\{c, \|t\|_- - 1\}\}$. The set R is defined as $\bigcup_{i\geq 0} R_i$ with $R_0 := \{c\}$ and

$$R_{i+1} := R_i \cup \{p \in R' \mid \text{there are } q \in R_i \text{ and } \overline{b} \in \{0,1\}^r \text{ with } 2p + t(\overline{b}) = q\},$$

for $i \geq 0$. It is straightforward to define an algorithm that computes R. The automaton $\mathcal{B}'$ is defined as $\mathcal{B}' := (\{q_{\mathrm{I}}, q_{\mathrm{sink}}\} \cup R, \{0,1\}^r, q_{\mathrm{I}}, \delta', F')$ with

$$F' := \begin{cases} \{q_{\mathrm{I}}, 0\} & \text{if } c = 0, \\ \{c\} & \text{if } c \neq 0, \end{cases}$$

and for $p \in \{q_{\mathrm{I}}, q_{\mathrm{sink}}\} \cup R$ and $\overline{b} \in \{0,1\}^r$, the transition function is defined as

$$\delta'(p, \overline{b}) := \begin{cases} -t(\overline{b}) & \text{if } p = q_{\mathrm{I}} \text{ and } -t(\overline{b}) \in R, \\ 2p + t(\overline{b}) & \text{if } p \in R \text{ and } 2p + t(\overline{b}) \in R, \\ q_{\mathrm{sink}} & \text{otherwise.} \end{cases}$$

Theorem 9. *The automaton $\mathcal{B}'$ is minimal.*

Proof. Since all states are reachable from the initial state q_{I}, it suffices to show that $p \not\sim_{\mathcal{B}'} q$, for all distinct $p, q \in \{q_{\mathrm{I}}, q_{\mathrm{sink}}\} \cup R$. Similar to Lemma 6(b), we can show that if $p \in R$ and $q \neq q_{\mathrm{I}}$ then $p \not\sim_{\mathcal{B}'} q$. Note that by construction of R, the state c is reachable from p. It is obvious that $q_{\mathrm{I}} \not\sim_{\mathcal{B}'} q_{\mathrm{sink}}$ since q_{sink} is a non-accepting sink state and c is reachable from q_{I}. It remains to show that $q_{\mathrm{I}} \not\sim_{\mathcal{B}'} q$, for all $q \in R$.

First, assume that $q \neq c$. There is a word $u := \overline{b}v \in (\{0,1\}^r)^+$ with $\delta'(q, \overline{b}v) = c$. Without loss of generality assume that $\delta'(q, \overline{b}) \neq q$. For the sake of absurdity, assume that $\delta'(q_{\mathrm{I}}, \overline{b}) \sim_{\mathcal{B}'} \delta'(q, \overline{b})$. Since $\delta'(q_{\mathrm{I}}, \overline{b}), \delta'(q, \overline{b}) \in R$ we have that $\delta'(q_{\mathrm{I}}, \overline{b}) = \delta'(q, \overline{b})$. By the definition of the transition function $\delta'(q, \overline{b}) = 2q + t(\overline{b}) = -t(\overline{b})$. It follows that $q = -t(\overline{b})$, and therefore, $\delta(q, \overline{b}) = 2q - q = q$ contradicting the assumption.

Second, assume that $q = c$. If $c \neq 0$ then it is obvious that $q_{\mathrm{I}} \not\sim_{\mathcal{B}'} c$ since c is an accepting state and q_{I} is a non-accepting state. Let $c = 0$. Note that we assume that $\gcd(t) = 1$ and the equation is neither $x = 0$, $-x = 0$, $x - y = 0$, nor $-x + y = 0$. There is a letter $\overline{b} \in \{0,1\}^r$ with $\delta'(q_{\mathrm{I}}, \overline{b}) \in R$ and $\delta'(q_{\mathrm{I}}, \overline{b}) \neq 0$. Note that $-t(\overline{b}) \neq t(\overline{b})$. For $\delta'(0, \overline{b}) \in R$ or $\delta'(0, \overline{b}) = q_{\mathrm{sink}}$, we already know that $\delta'(q_{\mathrm{I}}, \overline{b}) \not\sim_{\mathcal{B}'} \delta'(0, \overline{b})$. ⊣

Optimization #3. Similar to equations, we can optimize the automaton construction in [10, 26] for inequations.

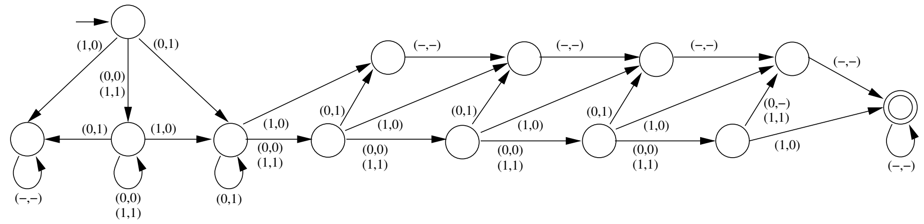

Fig. 3. Minimal automaton for the inequation $x - y > 32$.

Example 10. Consider the inequation $x - y > 32$. The construction in [10, 26] yields an automaton with the set of states $Q = \{q_{\mathrm{I}}, q_{\mathrm{small}}, -1, 0, \ldots, 32, q_{\mathrm{large}}\}$. But the minimal automaton for $x - y > 32$ (see Figure 3) has only 13. The reason for this gap is that several states in $Q' := \{q \in Q \cap \mathbb{Z} \mid q > 1\} = \{2, 3, ..., 32\}$ can be merged.

We can merge the states in $Q'' := \{q \in Q' \mid 2q + t(\bar{b}) > c, \text{ for all } \bar{b} \in \{0,1\}^2\} = \{17, \ldots, 32\}$ to a single state since all states in Q'' are non-accepting and all their transitions go to q_{large}. The state 16 cannot be merged with any other state since if we read the letter $(1, 0)$, we end up in the accepting state q_{large}, and if we read the letters $(0, 0)$, $(1, 1)$, or $(0, 1)$ we end up in the non-accepting states 32, or 31, respectively. The states in $\{9, \ldots, 15\}$ can be again merged to a single state since with every transition we reach a state in Q''. And so on.

We can also merge the states -1 and q_{small} since from both states only non-accepting states are reachable. Namely, -1 and q_{small}.

The reason for merging states in the cases where $\lessgtr \in \{<, >\}$ is that the transitions of the states q in $\mathcal{B}$ with $\|t\|_- < q \leq c$ lead only in one direction heading to the sink state q_{large}, or if $-\|t\|_+ > q \geq c$ then the transitions lead only in the direction heading to the sink state q_{small}. Intuitively speaking, a state measures the distance to the sink state q_{large} or q_{small}, respectively.

In the following, we give an optimized construction for an automaton $\mathcal{C}$ for an inequation $t \lessgtr c$ with $\lessgtr \in \{<, >\}$. We assume that $\lessgtr$ is $>$ and that $c \geq 0$. The other cases are analogous.

If $c < \|t\|_-$, then the automaton $\mathcal{C}$ is the automaton $\mathcal{B}$ from the construction in [10, 26] except that (i) the states $-\|t\|_+$ and q_{small} are merged, and (ii) the states $\|t\|_-$ and q_{large} are merged. The states $-\|t\|_+$ and q_{small} can be merged since both states are non-accepting and all transitions from these states either go to $-\|t\|_+$ or to q_{small}. Similarly for the states $\|t\|_-$ and q_{large}.

In the following, assume that $c \geq \|t\|_-$. The set of states of $\mathcal{C}$ is partitioned into two sets $Q_1 := \{q_{\mathrm{I}}, q_{\mathrm{sink}}, -\|t\|_+ + 1, \ldots, \|t\|_-\}$, and Q_2 is determined by a strict monotonically decreasing sequence $d_0 > d_1 > \cdots > d_n = \|t\|_- + 1$, for some $n \geq 1$. This sequence is recursively defined as follows. Let $d_0 := \infty$ and $d_1 := c + 1$. Assume that $d_0 > d_1 > \cdots > d_i > \|t\|_-$ for $i \geq 1$ are already defined.

– If $d_i = \|t\|_- + 1$ then we are done.

– If $d_i > \|t\|_- + 1$ then let $d_{i+1} < d_i$ be the smallest integer greater than $\|t\|_-$ such that for all $\overline{b} \in \{0,1\}^r$, there is an index j with $1 \leq j < i$ and

$$d_j \leq 2d_{i+1} + t(\overline{b}), 2(d_i - 1) + t(\overline{b}) < q_{j+1} \,. \tag{1}$$

Note that such a d_{i+1} always exists since $d_i - 1$ satisfies the condition (1), for all $\overline{b} \in \{0,1\}^r$.

The interval $[d, d')$ is the set of integers $\{d, \dots, d'-1\}$ if $d, d' \in \mathbb{Z}$, and if $d' = \infty$ then $[d, d')$ is the set of integers $\{d, d+1, \dots\}$. We define Q_2 as the set $\{[d_n, d_{n-1}), \dots, [d_2, d_1), [d_1, d_0)\}$. Note that we identify intervals with states.

The automaton $\mathcal{C}$ is defined as $\big(Q_1 \cup Q_2, \{0,1\}^r, q_{\mathrm{I}}, \eta, \{[d_1, d_0)\}\big)$ where the transition function is as follows

$$\eta(q, \overline{b}) := \begin{cases} 2q + t(b) & \text{if } q \in \mathbb{Z} \text{ and } -\|t\|_- < 2q + t(b) \leq \|t\|_-, \\ [q_i, q_{i-1}) & \text{if } q \in \mathbb{Z} \text{ and } i \text{ is the index with } 2q + t(b) \in [d_i, d_{i-1}), \\ -t(b) & \text{if } q = q_{\mathrm{I}} \text{ and } -t(b) > -\|t\|_+, \\ q_{\mathrm{sink}} & \text{otherwise,} \end{cases}$$

for $q \in Q_1$ and $\overline{b} \in \{0,1\}^r$, and for the index i with $n \geq i \geq 1$, we define

$$\eta([d_i, d_{i-1}), \overline{b}) := [d_j, d_{j-1}) \,,$$

where j is the index with $i \geq j \geq 1$ and $2d_i + t(b) \in [d_j, d_{j-1})$.

In the automaton $\mathcal{C}$ we have merged the states q_{small} and $-\|t\|_+$ of the automaton $\mathcal{B}$ to q_{sink}, and all the states $p, q \in [d_i, d_{i-1})$ of $\mathcal{B}$ have been merged in $\mathcal{C}$ to the state $[d_i, d_{i-1})$, for $1 \leq i \leq n$. For the states in Q of $\mathcal{B}$, it holds that $q_{\mathrm{small}} \sim_{\mathcal{B}} -\|t\|_+$, and $p \sim_{\mathcal{B}} q$, for all $p, q \in [d_i, d_{i-1})$ with $1 \leq i \leq n$. Therefore, $L(\mathcal{C}) = L(\mathcal{B})$.

Theorem 11. *The automaton $\mathcal{C}$ is minimal.*

Proof. We only give the prove if $c \geq \|t\|_-$. Let $\overline{b}_+ \in \{0,1\}^r$ be the letter that has a 1 in its ith coordinate iff the ith coefficient in t is positive, and let $\overline{b}_- \in \{0,1\}^r$ be the letter that has a 1 in its ith coordinate iff the ith coefficient in t is negative.

The state q_{sink} is not equivalent to any other state since q_{sink} is a non-accepting sink state and from all other states we can reach an accepting state. For every $q \in (Q_1 \cup Q_2) \setminus \{q_{\mathrm{I}}\}$, it holds that $q_{\mathrm{I}} \not\sim_{\mathcal{C}} q$, since $\eta(q_{\mathrm{I}}, \overline{b}_+) = q_{\mathrm{sink}} \neq \eta(q, \overline{b}_+)$.

It holds that $[d_i, d_{i-1}) \not\sim_{\mathcal{C}} q$, every for $q \in Q_1$, since $\eta([d_i, d_{i-1}), \overline{b}_-^n) = [d_1, d_0)$ and $\eta(q, \overline{b}_-^n) = q_{\mathrm{sink}}$. By induction, it is straightforward to show that $[d_i, d_{i-1}) \not\sim_{\mathcal{C}} [d_j, d_{j-1})$, for all $1 \leq i < j \leq n$.

For every $q \in S$, $\|t\|_- \not\sim_{\mathcal{C}} q$, since $\eta(\|t\|_-, \overline{b}_-^{\|t\|}) = \|t\|_-$ and $\eta(q, \overline{b}_-^{\|t\|}) = q_{\mathrm{sink}}$.

The states of $\mathcal{C}$ in $\{-\|t\|_+ + 1, \dots, \|t\|_- - 1\}$ are not equivalent since these states are not equivalent in $\mathcal{B}$ (see Lemma 6(b)). ⊣

3.3 Divisibility

In this subsection we give an upper bound of the size of the minimal automaton for a formula of the form $d|t+c$, where $d \geq 1$, t is a homogeneous term, and $c \in \mathbb{Z}$. The theorem below shows that the size of the minimal automaton for $d|t+c$ depends only *linearly* on d.

Theorem 12. *The minimal automaton for the atomic formula $d|t+c$ has at most $d+1$ states.*

Proof. The case for $d=1$ is trivial. We assume in the remainder of the proof that $d > 2$. We prove the claim by constructing an automaton $\mathcal{A}$ with the set of states $Q := \{q_{\mathrm{I}}, 0, 1, \dots, d-1\}$, where a state $q \in Q \cap \mathbb{Z}$ has the following intuitive interpretation. If we reach the state q with a word in $(\{0,1\}^r)^*$ representing $\overline{v} \in \mathbb{Z}^r$ then the remainder of the division of $t(\overline{v}) + c$ by d equals q. We denote by $\mathrm{rem}(q,d)$ the remainder of $q \in \mathbb{Z}$ divided by d.

Let $\mathcal{A} := (Q, \{0,1\}^r, q_{\mathrm{I}}, \delta, F)$ with

$$F := \{q \in Q \cap \mathbb{Z} \mid d|q+c\} \cup \begin{cases} \{q_{\mathrm{I}}\} & \text{if } d|c, \\ \emptyset & \text{otherwise,} \end{cases}$$

and $\delta : Q \times \{0,1\}^r \to Q$ is defined as

$$\delta(q, \overline{b}) := \begin{cases} \mathrm{rem}(-t(\overline{b}), d) & \text{if } q = q_{\mathrm{I}}, \\ \mathrm{rem}(2q + t(\overline{b}), d) & \text{otherwise,} \end{cases}$$

where $q \in Q$ and $\overline{b} \in \{0,1\}^r$.

We need the following two facts for the correctness prove of our construction.

(a) For all $n \in \mathbb{Z}$, $d|n+c$ iff $d|\,\mathrm{rem}(n,d) + c$.
(b) If $w \in (\{0,1\}^r)^+$ represents $\overline{v} \in \mathbb{Z}^r$ then $\delta(q_{\mathrm{I}}, w) = \mathrm{rem}(t(\overline{v}), d)$.

With (a) and (b) the correctness of the construction follows immediately from the definition of F. Note that the empty word represents $\overline{0}$ and $t(\overline{0}) = 0$.

The proof of (a) is straightforward. There are $p, q \in \mathbb{Z}$ such that $pd + q = n$ and $0 \leq q < d$. Note that $q = \mathrm{rem}(n,d)$. By definition, $d|n+c$ iff there is a $k \in \mathbb{Z}$ with $dk = n + c = pd + q + c$. The equality can be rewritten to $d(k-p) = q + c$, i.e., $d|\,\mathrm{rem}(n,d) + c$.

We prove (b) by induction over the length of w. For the base case, let $w = \overline{b} \in \{0,1\}^r$. Since we use the 2's complement representation the letter $\overline{b}$ represents $-\overline{v} \in \mathbb{Z}^r$. Therefore, $t(\overline{v}) = -t(\overline{b})$. By definition,

$$\delta(q_{\mathrm{I}}, \overline{b}) = \mathrm{rem}(-t(\overline{b}), d) = \mathrm{rem}(t(\overline{v}), d)\,.$$

For the step case, assume $\delta(q_{\mathrm{I}}, w) = \mathrm{rem}(t(\overline{v}), d)$. Let $\overline{b} \in \{0,1\}^r$ and assume that $w\overline{b}$ represents $\overline{v}' \in \mathbb{Z}^r$. There are $p, q \in \mathbb{Z}$ with $t(\overline{v}) = pd + q$ and $0 \leq q < d$.

Note that $q = \mathrm{rem}(t(\overline{v}), d)$ and $t(\overline{v}') = 2t(\overline{v}) + t(\overline{b}) = 2pd + 2q + t(\overline{b})$. We have that

$$\begin{aligned}\mathrm{rem}(t(\overline{v}'), d) &= \mathrm{rem}(2pd + 2q + t(\overline{b}), d) = \mathrm{rem}(2q + t(\overline{b}), d) = \delta(q, \overline{b})\\ &\overset{\mathrm{IH}}{=} \delta(\delta(q_{\mathrm{I}}, w), \overline{b}) = \delta(q_{\mathrm{I}}, w\overline{b})\,.\end{aligned}$$

$\dashv$

An optimization is to filter out the states that are not a multiple of $\gcd(\gcd(t), d)$. These states are not reachable from the initial state since $\mathrm{rem}(t(\overline{v}), d)$ is a multiple of $\gcd(\gcd(t), d)$, for every $\overline{v} \in \mathbb{Z}^r$.

In the next subsection we need the following definition. The pre-automaton $\mathcal{B}^{t,d}$ for a homogeneous term t and $d \geq 1$ is the pre-automaton of the automaton for an atomic formula $d \big| t$ given in the proof of Theorem 12.

3.4 Quantifier-Free Formulas

In this section, we give an upper bound on the size of the minimal automaton for a quantifier-free formula. This upper bound depends on the number of different terms that occur in the formula, the maximal absolute integer, and the number of distinct ds in atomic formulas of the form $d \big| t + c$. It does *not* depend on the number of atomic formulas and the Boolean combination of these atomic formulas. This is not obvious since the Boolean connectives $\vee$ and $\wedge$ are handled by the automata operations for union and intersection, respectively, if we construct the automaton recursively over the structure of the quantifier-free formula. Note that the automata operations for union and intersection are based on the so-called product construction. The size of the resultant automaton is in the worst-case the product of the number of states of the two automata. Furthermore, the connective $\leftrightarrow$ is allowed. Eliminating this connective can lead to exponentially larger formulas. But the above mentioned parameters on which the size of the minimal automaton depends are not altered.

Let T be a finite nonempty set of homogeneous terms, and let D be a finite nonempty set of naturals numbers greater than 0. Furthermore, let c_{min}, c_{max} be natural numbers such that

$$c_{min} > \max\big\{\|t\|_+ \,\big|\, t \in T\big\} \qquad \text{and} \qquad c_{max} > \max\big\{\|t\|_- \,\big|\, t \in T\big\}\,.$$

Theorem 13. *Let ψ be quantifier-free formula containing only atomic formulas of the form $t \lessgtr c$ and $d \big| t + c'$, with $t \in T$, $d \in D$, $-c_{min} < c < c_{max}$, $c' \in \mathbb{Z}$, and $\lessgtr \in \{=, \neq, <, \leq, >, \geq\}$. The number of states of the minimal automaton for ψ is at most*

$$1 + (c_{min} + c_{max})^{|T|} \cdot (\max D)^{|T| \cdot |D|}\,.$$

Proof. Assume that the variables occurring in the terms in T are at most $y_1, \ldots, y_n$. The (pre-)automata we are going to define will have always the alphabet $\{0,1\}^n$. Recall that a pre-automaton is an automaton without specifying the set of accepting states.

For $t \in T$ and $d \in D$, the pre-automaton $\mathcal{A}^t_{[-c_{min},c_{max}]}$ has $1+c_{min}+c_{max}$, and the pre-automaton $\mathcal{B}_{t,d}$ has $1+d$ states. Recall that the definition of the pre-automaton $\mathcal{A}^t_{[-c_{min},c_{max}]}$ is given in §3.2, and the definition of the pre-automaton $\mathcal{B}^{t+d}$ is given in §3.3. To simplify notation, we omit the subscripts $[-c_{min}, c_{max}]$ wherever they occur, and assume that $T = \{t_1, \dots, t_r\}$, and $D = \{d_1, \dots, d_s\}$. Let $\mathcal{C}$ be the product of all the $\mathcal{A}^{t_i}$s and $\mathcal{B}^{t_i,d_j}$s. That is,

$$\mathcal{C} := \mathcal{A}^{t_1} \times \cdots \times \mathcal{A}^{t_r} \times \mathcal{B}^{t_1,d_1} \times \cdots \times \mathcal{B}^{t_1,d_s} \times \cdots \times \mathcal{B}^{t_r,d_s} .$$

Without loss of generality, we assume that all states of $\mathcal{C}$ are reachable from the initial state $q_{\mathrm{I}} = (q_{\mathrm{I}}^{t_1}, \dots, q_{\mathrm{I}}^{t_r}, q_{\mathrm{I}}^{t_1,d_1}, \dots, q_{\mathrm{I}}^{t_1,d_s}, \dots, q_{\mathrm{I}}^{t_r,d_r})$, where $q_{\mathrm{I}}^{t_i}$ is the initial state of $\mathcal{A}^{t_i}$, and $q_{\mathrm{I}}^{t_i,d_j}$ is the initial state of $\mathcal{B}^{t_i,d_j}$, for $1 \leq i \leq r$ and $1 \leq j \leq s$. An initial state of a pre-automaton $\mathcal{A}^{t_i}$ or $\mathcal{B}^{t_i,d_j}$ occurs only in q_{I}, since the every initial state of the pre-automaton $\mathcal{A}^{t_i}$ or $\mathcal{B}^{t_i,d_j}$ is not reachable from another state. Therefore, the number of states of $\mathcal{C}$ is at most

$$1 + \prod_{t \in T}(c_{min} + c_{max}) \cdot \prod_{t \in T} \prod_{d \in D} d \leq 1 + (c_{min} + c_{max})^r \cdot (\max D)^{rs} .$$

The automaton for the formula ψ is $(\mathcal{C}, F)$, where the set of final states F is defined as follows. We identify an initial state of a pre-automaton $\mathcal{A}^{t_i}$ or $\mathcal{B}^{t_i,d_j}$ with the integer 0, a state $q^{t_i}_{\mathrm{small}}$ of the automaton $\mathcal{A}^{t_i}$ is identified with the integer $-c_{min}$, and a state $q^{t_i}_{\mathrm{large}}$ of the automaton $\mathcal{A}^{t_i}$ is identified with c_{max}. A state $(q^{t_1}, \dots, q^{t_r}, q^{t_1,d_1}, \dots, q^{t_r,d_s})$ of $\mathcal{C}$ is in F iff $\mathfrak{Z} \models \psi'$, where ψ' is the formula that we obtain by substituting

- the integer q^{t_i} for the term t_i in the atomic formulas of the form $t_i \lessgtr c$, and
- the integer q^{t_i,d_j} for the term t_i in the atomic formulas of the form $d_j \big| t_i + c$.

Note that ψ' is a sentence, and therefore it is either true or not in $\mathfrak{Z}$.

It remains to prove that the automaton $(\mathcal{C}, F)$ represents $\{(u_1, \dots, u_n) \in \mathbb{Z}^n \,|\, \mathfrak{Z} \models \psi[u_1, \dots, u_n]\}$. Let $w \in (\{0,1\}^n)^*$ be a word representing $(u_1, \dots, u_n) \in \mathbb{Z}^n$. For a term $t \in T$, the value $t(u_1, \dots, u_n)$ can be replaced by c_{max} if $t(u_1, \dots, u_n) \geq c_{max}$ and by c_{-min} if $t(u_1, \dots, u_n) \leq c_{min}$ in every atomic formula of the form $t \lessgtr c$ without changing its truth value since $-c_{min} < c < c_{max}$. This modified value corresponds to the state reached by the pre-automaton $\mathcal{A}^t$ after reading the word w. In an atomic formula of the form $d \big| t + c$ we can replace $t(u_1, \dots, u_n) + c$ by $\mathrm{rem}(t(u_1, \dots, u_n) + c, d) < d$ without changing the truth value. Again, the adjusted value corresponds to the state reached by the pre-automaton $\mathcal{B}^{t,d}$ after reading the word w. From the definition of F, it follows that $w \in L(\mathcal{C}, F)$ iff $\mathfrak{Z} \models \psi[u_1, \dots, u_n]$. $\dashv$

4 Upper Bound on the Automata Size

In this section we provide an upper bound on the size of the minimal automaton for an arbitrary formula. We obtain this bound by examining the quantifier-free formula constructed by the quantifier elimination method from Reddy and Loveland [18] which improves Cooper's quantifier elimination method [7].

For the sake of completeness, we briefly recall in §4.1 Reddy and Loveland's quantifier elimination method. We analyze it in §4.2 by giving upper bounds on the number of distinct coefficients in the constructed quantifier-free formula, etc. Finally, in §4.4, we use these upper bounds to give an upper bound on the size of the minimal automaton for a formula.

4.1 Quantifier Elimination in Presburger Arithmetic

Consider the formula $Qx\varphi$ where $Q \in \{\forall, \exists\}$ and $\varphi(x, y_1, \ldots, y_n)$ is quantifier-free. The construction of the quantifier-free formula $\psi(y_1, \ldots, y_n)$ proceeds in four steps.

Step 1: If $Q = \forall$ then replace $\forall x\varphi$ by $\neg\exists x\neg\varphi$. In the following, we assume $Q = \exists$.

Step 2: First, eliminate the connectives $\rightarrow$ and $\leftrightarrow$ in φ by the standard rules. Second, push all negations symbols that are to the right of the quantifier Q inwards (using De Morgan's laws, etc.) until they only occur directly in front of the atomic formulas. Third, eliminate the negation symbols and adjust the atomic formulas in which x occurs such that they are only of one of the forms

$$k \cdot x \le t(y_1, \ldots, y_n) + c\,, \qquad \text{or} \tag{A}$$

$$t(y_1, \ldots, y_n) + c \le k \cdot x\,, \qquad \text{or} \tag{B}$$

$$d \big| k \cdot x + t(y_1, \ldots, y_n) + c\,, \tag{C}$$

with $k > 0$. Note that we can always assume $d > 0$ since $d|u$ iff $d| - u$, for all $u \in \mathbb{Z}$. For instance, $\neg k \cdot x + k' \cdot y \le c$ is logically equivalent to $-k' \cdot y + c - 1 \le k \cdot x$ if $k > 0$, and $-k \cdot x \le k' \cdot y - c + 1$ otherwise. Let $\exists x\varphi'$ be the resultant formula.

Step 3: Let $\psi_{-\infty}$ be the formula where all the literals of type (A) in φ' are replaced by "true", i. e. $0 < 1$, and all literals of type (B) are replaced by "false", i. e. $1 < 0$. Let B be the set of the atomic formulas in φ' of type (B). The formula $\exists x\varphi'$ is logically equivalent to

$$\exists j \psi_{-\infty}[j/x] \vee \bigvee_{k\cdot x \le t+c \in B} \exists j \big(0 \le j \wedge k \big| t+c+j \wedge \varphi'[t+c+j/k\cdot x]\big)\,. \tag{2}$$

We have to explain the notation $\varphi'[t + c + j/k \cdot x]$. It means that an atomic formula α in φ in which x occurs is multiplied by k and $k \cdot x$ is substituted by $t + c + j$, i. e.

$$\alpha[t+c+j/k\cdot x] := \begin{cases} k' \cdot (t + c + j) \le k \cdot (t' + c') & \text{if } \alpha = k' \cdot x \le t' + c', \\ k \cdot (t' + c') \le k' \cdot (t + c + j) & \text{if } \alpha = t' + c' \le k' \cdot x, \\ kd \big| k' \cdot (t + c + j) + k \cdot (t' + c') & \text{if } \alpha = d \big| k' \cdot x + t' + c. \end{cases}$$

Moreover, we assume that $\alpha[t + c + j/k \cdot x]$ is put into normal form. Note that it suffices to multiply α with the least common multiple of k and k'. But we multiply α always with k since it simplifies matters.

Step 4: Let D be the least common multiple of all d's of the atomic formulas of type (C) occurring in the formula (2). Let $\psi(y_1, \ldots, y_n)$ be the formula

$$\bigvee_{0 \leq j \leq D-1} \psi_{-\infty}[j/x] \vee \bigvee_{t+c \leq k \cdot x \in B} \bigvee_{0 \leq j \leq D-1} \big(k \big| t+c+j \wedge \varphi'[t+c+j/k \cdot x]\big) .$$

The formula ψ is logically equivalent to the formula (2), and therefore logically equivalent to $\exists x \varphi$.

4.2 Analysis of the Quantifier Elimination Method

Oppen analyzed in [13] Cooper's quantifier elimination method by relating the growth in the number of atomic formulas and the size of the constants appearing in these atomic formulas essentially to the number of distinct coefficients that may appear. Reddy and Loveland analyzed their improved version of Cooper's quantifier elimination method in a similar way [18]. We refine their analysis. These kinds of analysis are technical and subtle. Oppen's analysis [12,13] as well as Reddy and Loveland's [18] analysis are flawed. We will point out the incorrect arguments used in their analysis. Fortunately, their theorems are nevertheless correct.

We give upper bounds on the following parameters. Let φ be a quantifier-free formula, and let $\mathrm{Atoms}(\varphi)$ be the set of atomic formulas of φ. The parameters are

$$\begin{aligned}
\#\mathit{terms}(\varphi) &:= \big|\big\{t \,\big|\, t \lessgtr c \in \mathrm{Atoms}(\varphi) \text{ with } \lessgtr \in \{=, \neq, <, \leq, >, \geq\} \text{ or } \\
&\qquad\qquad d \big| t+c \in \mathrm{Atoms}(\varphi)\big\}\big| \\
\#\mathit{coefs}(\varphi) &:= \big|\big\{k \,\big|\, k \text{ is a coefficient in some } \alpha \in \mathrm{Atoms}(\varphi)\big\}\big| \\
\#\mathit{divs}(\varphi) &:= \big|\big\{d \,\big|\, d \big| t+c \in \mathrm{Atoms}(\varphi)\big\}\big|
\end{aligned}$$

and

$$\begin{aligned}
\mathit{max_coef}(\varphi) &:= \max\{1\} \cup \big\{|k| \,\big|\, k \text{ is a coefficient in some } \alpha \in \mathrm{Atoms}(\varphi)\big\} \\
\mathit{max_const}(\varphi) &:= \max\{1\} \cup \big\{|c| \,\big|\, t \lessgtr c \in \mathrm{Atoms}(\varphi) \text{ with } \lessgtr \in \{=, \neq, <, \leq, >, \geq\}\big\} \\
\mathit{max_div}(\varphi) &:= \max\{1\} \cup \big\{d \,\big|\, d \big| t+c \in \mathrm{Atoms}(\varphi)\big\} .
\end{aligned}$$

The following theorem states that the absolute values of the coefficients grow significantly slower than the absolute values of the constants. Another point that we are going to exploit in §4.4 is that the number of distinct terms only grows quadratically. Note that the number of atomic formulas grows faster than quadratic.

Theorem 14. *Let $\varphi(x, y_1, \ldots, y_n)$ be a quantifier-free formula. For the quantifier-free formula $\psi(y_1, \ldots, y_n)$ produced by the previously described quantifier elimination method applied to $\exists x \varphi$, it holds that*

$$\begin{aligned}
&\#\mathit{terms}(\psi) \leq \big(2\#\mathit{terms}(\varphi)\big)^2 \\
&\#\mathit{coefs}(\psi) \leq 16\#\mathit{coefs}(\varphi)^4 + 2\#\mathit{coefs}(\varphi) \\
&\#\mathit{divs}(\psi) \leq \#\mathit{divs}(\varphi) \cdot \big(\#\mathit{terms}(\varphi) + 1\big) + \#\mathit{terms}(\varphi)
\end{aligned}$$

and

$$\begin{aligned}
&\mathit{max_coef}(\psi) \leq 2\mathit{max_coef}(\varphi)^2\\
&\mathit{max_const}(\psi) \leq 2\mathit{max_coef}(\varphi) \cdot \big(\mathit{max_const}(\varphi) + 1\big) + \\
&\qquad\qquad\qquad\qquad \mathit{max_coef}(\varphi)^{\#\mathit{coefs}(\varphi)+1} \cdot \mathit{max_div}(\varphi)^{\#\mathit{divs}(\varphi)}\\
&\mathit{max_div}(\psi) \leq \mathit{max_coef}(\varphi) \cdot \mathit{max_div}(\varphi)\,.
\end{aligned}$$

Before we launch into the proof, we want to point out an incorrect argument in Reddy and Loveland's analysis. They claim that Step 2 does not alter the number of distinct coefficients. Actually, Step 2 may double the number of distinct coefficients since a coefficient k occurring in φ may occur in φ' and also its opposite $-k$. To patch this, one has to work on the equivalence classes of logically equivalent atomic formulas and their negations, or enlarge the upper bounds accordingly. We will do the latter.[2] Fortunately, this growth in the number of coefficients in Step 2 does not affect the size of the least common multiple D the ds occurring in the atomic formulas of (2) that is used in Step 4. It holds that

$$D \leq \mathit{max\_coef}(\varphi)^{\#\mathit{coefs}(\varphi)} \cdot \mathit{max\_div}(\varphi)^{\#\mathit{divs}(\varphi)}\,.$$

The remainder of this subsection contains the proof of Theorem 14. For the sake of brevity, we write $\#\mathit{terms}$ instead of $\#\mathit{terms}(\varphi)$, and analogously for the other parameters.

Maximal Absolute Values. We first give upper bounds on the maximal value of the ds in the atomic formulas of type (C), and upper bounds on the maximal absolute value of the coefficients and constants in the formula ψ. Step 1 does not alter any of these parameters. Step 2 may increase the maximal absolute value of a constant by 1; the other two parameters are not changed in Step 2. We do not analyze the formula produced by Step 3 but only the formula ψ in Step 4.

To carry out the substitution for an atomic formula $t + c \leq k \cdot x \in B$ in Step 4 we have to multiply each atomic formula α of φ' in which x occurs by k. It holds that

$$\alpha[t{+}c{+}j/k{\cdot}x] = \begin{cases} k' \cdot (t + c + j) \leq k \cdot (t' + c') & \text{if } \alpha = k' \cdot x \leq t' + c',\\ k \cdot (t' + c') \leq k' \cdot (t + c + j) & \text{if } \alpha = t' + c' \leq k' \cdot x,\\ kd \big| k' \cdot (t + c + j) + k \cdot (t' + c') & \text{if } \alpha = d \big| k' \cdot x + t' + c. \end{cases}$$

In the worst case, we have that $j = D - 1$, $|c| = |c'| = \mathit{max\_const} + 1$, and $d = \mathit{max\_div}$. Moreover, it can be the case that there are coefficients ℓ and ℓ' of a variable y_i in t and t', respectively, with $|\ell| = |\ell'| = \mathit{max\_coef}$. The signs of c, c', ℓ, and ℓ' depend on the type of the atomic formula α. Since $k, k' \leq \mathit{max\_coefs}$, we obtain the upper bounds

$$\begin{aligned}
&\mathit{max_coef}(\psi) \leq 2\mathit{max_coef}^2\,,\\
&\mathit{max_div}(\psi) \leq \mathit{max_coef} \cdot \mathit{max_div}\,,
\end{aligned}$$

[2] Oppen's analysis [12, 13] has a similar error in reasoning, where he analysis Step 2 of Cooper's quantifier elimination method. See page 328 in [13].

and

$$\begin{aligned} \mathit{max\_const}(\psi) &\leq 2\mathit{max\_coef} \cdot (\mathit{max\_const} + 1) + D \cdot \mathit{max\_coef} \\ &\leq 2\mathit{max\_coef} \cdot (\mathit{max\_const} + 1) + \mathit{max\_coef}^{\#\mathit{coefs}+1} \cdot \mathit{max\_div}^{\#\mathit{divs}} \end{aligned}$$

Note that the values of the parameters in the subformula $\bigvee_{0\leq j\leq D-1} \psi_{-\infty}[j/x]$ of ψ cannot get larger than the bounds given above. This is easily checked.

Number of Distinct Objects. For the following, let $B_0 := \{t \leq k \cdot x \mid t + c \leq k \cdot x \in B \text{ for some } c \in \mathbb{Z}\}$. Note that $|B_0| \leq \#\mathit{terms}$. It holds that

$$\#\mathit{divs}(\psi) \leq \#\mathit{divs}\big(\psi_{-\infty}[0/x] \vee \bigvee_{t\leq k\cdot x\in B_0} (k \big| t \wedge \varphi'[t/k\cdot x])\big)$$

since for an atomic formula of the form $d\big|t+c$ in ψ, we have an atomic formula of the form $d\big|t+c'$ in the latter formula. Similar upper bounds hold for $\#\mathit{coefs}(\psi)$ and $\#\mathit{terms}(\psi)$.

Step 1 and Step 2 do not alter the number of distinct ds occurring in atomic formulas of type (C), i. e. $\#\mathit{divs}(\varphi) = \#\mathit{divs}(\varphi')$. It holds that

$$\begin{aligned} \#\mathit{divs}(\psi) &\leq \#\mathit{divs}\big(\psi_{-\infty}[0/x] \vee \textstyle\bigvee_{t\leq k\cdot x\in B_0} (k \big| t \wedge \varphi'[t/k\cdot x])\big) \\ &\leq \#\mathit{divs} + |B_0| + |B_0| \cdot \#\mathit{divs} \\ &\leq \#\mathit{divs} \cdot (\#\mathit{terms} + 1) + \#\mathit{terms}\,. \end{aligned}$$

The number of distinct terms does not change in the first two steps. Note that we rewrite terms in Step 2 but their normal forms do not change. For the number of distinct terms in the formula ψ produced by Step 4, we have that

$$\begin{aligned} \#\mathit{terms}(\psi) &\leq \#\mathit{terms}\big(\psi_{-\infty}[0/x] \vee \textstyle\bigvee_{t\leq k\cdot x\in B_0} (k \big| t \wedge \varphi'[t/k\cdot x])\big) \\ &\leq \#\mathit{terms} + \#\mathit{terms} + |B_0| + |B_0| \cdot \#\mathit{terms} \\ &\leq 4\#\mathit{terms}^2\,. \end{aligned}$$

It remains to give an upper bound on the number of distinct coefficients occurring in the formula ψ. Step 1 does change the coefficients. As already said Step 2 might double the number of distinct coefficients since by rewriting an atomic formula in one of the forms (A), (B), or (C) a coefficient k may become its opposite $-k$. Therefore, $\#\mathit{coefs}(\varphi') \leq 2\#\mathit{coefs}$.

Let $t \leq k \cdot x \in B_0$. To carry out the substitution, we have to multiply an atomic formula α of φ' in which x occurs by the factor k. α is of one of the forms $k' \cdot x \leq t' + c'$, $t' + c' \leq k' \cdot x$, or $d\big|k' \cdot x + t' + c'$. Let ℓ be a coefficient in the term t for the variable y_i, and let ℓ' be a coefficient in the term t' for the variable y_i. In the worst case, the coefficient $\pm k'\ell \pm k\ell'$ for the variable y_i in the atomic formula $\alpha[t/k \cdot x]$ is different for any possible value of k, k', ℓ, and ℓ'. Here, $\pm$ stands for either $+$ or $-$ depending on the type of α. We do not have to consider if $\pm$ is either $+$ or $-$ since we assume that if $k \in \mathbb{Z}$ is a coefficient in φ' then

$-k$ may also occurs as a coefficient in φ'. Since k, k', ℓ, ℓ' are coefficients we get at most $\#\mathit{coefs}(\varphi')^4$ distinct coefficients. Moreover, all these coefficients can be different to the coefficients in φ' which may still occur in the formula $\psi_{-\infty}[0/x]$, in the atomic formulas without the variable x, or in the added atomic formulas $k|t$.[3] Therefore, we have that

$$\#\mathit{coefs}(\psi) \leq 16\#\mathit{coefs}^4 + 2\#\mathit{coefs}\,.$$

This completes the proof of Theorem 14.

4.3 Quantifier Elimination for Arbitrary Formulas

The common way to extend quantifier elimination methods to arbitrary formulas is (a) to transform the formula into prenex normal form, and (b) to successively iterate the quantifier elimination method from the innermost quantifier. The costs for doing (a) can be very high as the following examples demonstrates.

Example 15. Consider the formula $(\forall x\varphi) \leftrightarrow \psi$. Even if we assume that x does not occur in ψ, the quantifier $\forall$ does not distribute in general over $\leftrightarrow$. Therefore, we eliminate the connective $\leftrightarrow$ first. We obtain the formula $((\forall x\varphi) \rightarrow \psi) \wedge (\psi \rightarrow \forall x\varphi)$. The quantifiers do not distribute in general over $\rightarrow$. Eliminating $\rightarrow$ yields $((\neg\forall x\varphi) \vee \psi) \wedge (\neg\psi \wedge \forall x\varphi)$. To move the quantifiers at the very beginning, we have to push the first negation inwards. Finally, we obtain the formula $\exists x\forall x'((\neg\varphi \vee \psi) \wedge (\neg\psi \wedge \varphi[x'/x]))$. We have not only at least doubled the length of the formula but we have also doubled the number of quantifiers. We want to *eliminate* quantifiers. Thus, we have doubled our work that we have to do. Even worse, we have introduced a quantifier alternation.[4]

---

[3] Oppen's analysis [12, 13] of Cooper's quantifier elimination method is flawed at a similar point. Oppen gives in [13] in the proof of Theorem 1 on page 328 (Step 2) an "upper bound" on the number of distinct coefficients plus the distinct ds in the atomic formulas of type (C). But the given "upper bound" is too small since it does not count the coefficients in atomic formulas in which the variable that one wants to eliminate does not occur and it does not take into account that for different types of atomic formulas we can obtain different coefficients. A counter-example for Oppen's analysis on Step 2 of Cooper's quantifier elimination method is the formula $2x < 2y + 5 \wedge 2y + 1 < 3x \wedge 3|2y \wedge 2|3y$, where x is the variable that one wants to eliminate.

[4] Oppen claims on page 329 in [13] that the bounds also hold if the connectives $\rightarrow$ and $\leftrightarrow$ are permitted (he restricts himself to the connectives $\neg$, $\vee$, and $\wedge$ in the preceding investigations). His justification is that all the occurrences of $\rightarrow$ and $\leftrightarrow$ can be eliminated in the very beginning. The number of atomic formulas can exponentially grow by this step—but this does not matter—but none of the other relevant quantities changes. This is only true in general under the assumption that the formula is in prenex normal form. It is not clear from the context if Oppen makes this assumption. Oppen's argumentation is not correct for arbitrary formulas containing the connective $\leftrightarrow$. As we have seen constructing a prenex normal form can increase the number of quantifiers.

The example above illustrates that building the prenex normal form is not a good idea.[5] Fortunately, we can do better by successively (i) applying the quantifier elimination method to subformulas of the form $Qx\psi$ where ψ is quantifier-free and $Q \in \{\forall, \exists\}$, and (ii) replacing the subformula $Qx\psi$ by the obtained logically equivalent formula. We get the following upper bounds on the parameters using Theorem 14.

Corollary 16. *Let $\varphi(y_1, \dots, y_n)$ be an arbitrary formula containing m quantifiers. Let $a \geq 2$ be greater than or equal to the number of atomic formulas in φ, $s \geq 2$ be greater than or equal to the maximal absolute integer of φ, and let $c \geq 3$ be greater than or equal to the number of distinct coefficients in φ. There is a quantifier-free formula $\psi(y_1, \dots, y_n)$ that is logically equivalent to φ and it holds that*

$$\begin{aligned} &\#\mathit{terms}(\psi) \leq a^{2^{3m}} \\ &\#\mathit{coefs}(\psi) \leq c^{2^{3m}} \\ &\#\mathit{divs}(\psi) \leq a^{2^{3m}} \end{aligned}$$

and

$$\begin{aligned} &\mathit{max\_coef}(\psi) \leq s^{2^{m+1}} \\ &\mathit{max\_const}(\psi) \leq 2s^{\left(c^{2^{3m}} + a^{2^{3m}} + 2^m\right)} \\ &\mathit{max\_div}(\psi) \leq s^{2^{m+1}} . \end{aligned}$$

Furthermore, ψ is effectively constructible from φ.

Proof. Without loss of generality, we assume that the bound variables are pairwise distinct, and they are different from the free variables $y_1, \dots, y_n$.

There is another inaccuracy in [13] concerning the length of a formula. Oppen claims on page 329 that the prenex normal form can be obtained in linear time and the space is at most twice of the original formula (the original formula contains only the connectives $\neg$, $\vee$, and $\wedge$). Due to the fact that variable reuse is heavily used to encode coefficients and constants in logarithmic space, it is not accurate to consider the length of a variable as 1. For instance, for the formula

$$\exists y\bigl(y = 0 \wedge \exists z(y = z + z \wedge \cdots \wedge \exists z(z = y + y \wedge \exists y(z = y + y \wedge y = x + x)) \dots)\bigr) ,$$

encoding $2^n \cdot x = 0$, we obtain the prenex normal form $\exists y_n \exists y_{n-1} \dots \exists y_2 \exists y_1 (y_n = 0 \wedge y_n = y_{n-1} + y_{n-1} \wedge \cdots \wedge y_2 = y_1 + y_1 \wedge y_1 = x + x)$.

A better measure of the length of a variable is the ceiling of the logarithm of the index of the variable plus a constant. The length of the constructed prenex normal form is then $\mathrm{O}(n \lg n)$ where n is the length of the original formula. This change does not make Oppen's upper bound invalid, but the argumentation is not that straightforward anymore.

[5] The decision procedures based on quantifier elimination of which the author is aware of do not handle the connective $\leftrightarrow$ efficiently.

We define $\varphi_0 := \varphi$. φ_k is the formula *after* the kth iteration of (i) applying the quantifier elimination method to a subformula of φ_{k-1} that is of the form $Q_k x \psi_k$, where ψ_k is quantifier-free, and (ii) replacing $Q_k x \psi_k$ by the constructed quantifier-free formula. We denote by χ_k the outcome of the quantifier elimination method to $Q_k x \psi_k$. We iterate (i) and (ii) m times. The sketched algorithm produces a quantifier-free formula φ_m that is logically equivalent to φ. We define $\psi := \varphi_m$.

Number of Distinct Objects. We prove by induction over $k \geq 0$ that the number of distinct terms t_k in φ_k is at most $2^{2^{2k}-1} a^{2^k}$. The base case holds for $k = 0$ due to the choice of a. Let $k \geq 1$. We apply the quantifier elimination method to the subformula $Q_k x \psi_k$ of the formula φ_{k-1}. The worst case is that all distinct terms already occur in ψ_k and the terms in χ_k are all different from the terms occurring in φ_{k-1}. Clearly, it holds that $\#\mathit{terms}(\psi_k) \leq t_{k-1}$, and by Theorem 14, it holds that $\#\mathit{terms}(\chi_k) \leq (2\#\mathit{terms}(\psi_k))^2$. Therefore,

$$\begin{aligned} t_k &\leq t_{k-1} + (2t_{k-1})^2 \overset{\text{IH}}{\leq} 2^{2^{2(k-1)}-1} a^{2^{k-1}} + \left(2 \cdot 2^{2^{2(k-1)}-1} a^{2^{k-1}}\right)^2 \\ &\leq 8 \cdot 2^{2 \cdot 2^{2(k-1)}-2} a^{2^k} = 2^{2^{2k-1}+1} a^{2^k} \leq 2^{2^{2k}-1} a^{2^k} . \end{aligned}$$

The last inequality holds since $2^{2k-1} + 1 \leq 2^{2k} - 1$, for all $k \geq 1$.

From t_m we obtain the claimed upper bound on the distinct number of terms in ψ:

$$\#\mathit{terms}(\psi) \leq t_m \leq 2^{2^{2m}-1} a^{2^m} \leq a^{2^{2m}-1+2^m} \leq a^{2^{3m}} .$$

We prove by induction over $k \geq 0$ that the number d_k of distinct d's occurring in atomic formulas $d \mid t + c$ in φ_k is at most $2^{2^{2k}-1} a^{2^k}$. The base case for $k = 0$ is obvious. Let $k \geq 1$. We apply the quantifier elimination method to the subformula $Q_k x \psi_k$ in the formula φ_{k-1}. An upper bound on d_k is

$$\begin{aligned} d_k &\leq d_{k-1} + \#\mathit{divs}(\chi_k) \\ &\leq d_{k-1} + \#\mathit{divs}(\psi_k) \cdot (\#\mathit{terms}(\psi_k) + 1) + \#\mathit{terms}(\psi_k) \\ &\leq d_{k-1} + d_{k-1}(t_{k-1} + 1) + t_{k-1} \leq t_{k-1} + (d_{k-1} t_{k-1} + 2d_{k-1} + 1) \\ &\overset{\text{IH}}{\leq} 2^{2^{2(k-1)}-1} a^{2^{k-1}} + (2^{2^{2(k-1)}-1} a^{2^{k-1}} + 1)^2 \\ &\leq 2^{2^{2(k-1)}-1} a^{2^{k-1}} + (2 \cdot 2^{2^{2(k-1)}-1} a^{2^{k-1}})^2 \leq 2^{2^{2k}-1} a^{2^k} . \end{aligned}$$

Analogously to the bound on $\#\mathit{terms}(\psi)$ we get the bound $\#\mathit{divs}(\psi) \leq a^{2^{3m}}$.

Let c_k be the number of distinct coefficients in φ_k. We show by induction over $k \geq 0$ that $c_k \leq 3^{4^k-1} c^{4^k}$. This bound obviously holds for $k = 0$. Let $k \geq 0$. We apply the quantifier elimination method to the subformula $Q_k x \psi_k$ in the formula φ_{k-1}. An upper bound on c_k is

$$\begin{aligned} c_k &\leq c_{k-1} + \#\mathit{coefs}(\chi_k) \leq c_{k-1} + 16\#\mathit{coefs}(\psi_k)^4 + 2\#\mathit{coefs}(\psi_k) \\ &\leq 16 c_{k-1}^4 + 2c_{k-1} \leq 18 c_{k-1}^4 \\ &\overset{\text{IH}}{\leq} 18(3^{4^{k-1}-1} c^{4^{k-1}})^4 \leq 18 \cdot 3^{4^k-4} c^{4^k} \leq 3^{4^k-1} c^{4^k} . \end{aligned}$$

The claimed bound on $\#coefs(\psi)$ holds since

$$\#coefs(\psi) \leq c_m \leq 3^{4^m-1}c^{4^m} \leq c^{4^m-1+4^m} = c^{2^{2m}+2^{2m}-1} \leq c^{2^{3m}} .$$

Maximal Absolute Values. The maximal absolute coefficient in the formula φ_k, for every $k \geq 1$, is equal to the maximum of the maximal absolute coefficient of φ_{k-1} and the maximal absolute coefficient that occurs in χ_k. Recall that χ_k is the constructed formula from the quantifier elimination method applied to the subformula $Q_k x\psi_k$ of φ_{k-1}. Analogously for the maximal absolute constant, and the maximal "divider" in the atomic formulas of the form $d|t+c$ of φ_k.

We prove by induction over $k \geq 0$ that $max\_coef(\varphi_k) \leq 2^{2^k-1}s^{2^k}$. The base case $k=0$ is obvious. For the step case $k \geq 1$,

$$\begin{aligned} max\_coef(\varphi_k) &= \max\{max\_coef(\varphi_{k-1}), max\_coef(\chi_k)\} \\ &\leq \max\{max\_coef(\varphi_{k-1}), 2max\_coef(\psi_k)^2\} \\ &\leq 2max\_coef(\varphi_{k-1})^2 \overset{\text{IH}}{\leq} 2\big(2^{2^{k-1}-1}s^{2^{k-1}}\big)^2 \leq 2^{2^k-1}s^{2^k} . \end{aligned}$$

Since $s \geq 2$ it holds that

$$max\_coef(\psi) = max\_coef(\varphi_m) \leq 2^{2^m-1}s^{2^m} \leq s^{2^m-1+2^m} \leq s^{2^{m+1}} . \tag{3}$$

We prove by induction over $k \geq 0$ that $max\_div(\varphi_k) \leq 2^{2^k-1}s^{2^k}$. The base case $k=0$ is obvious. For the step case $k \geq 1$, we have that

$$\begin{aligned} max\_div(\varphi_k) &= \max\{max\_div(\varphi_{k-1}), max\_div(\chi_k)\} \\ &\leq \max\{max\_div(\varphi_{k-1}), max\_coef(\psi_k)\cdot max\_div(\psi_k)\} \\ &\leq max\_coef(\varphi_{k-1})\cdot max\_div(\varphi_{k-1}) \\ &\overset{\text{IH}}{\leq} 2^{2^{k-1}-1}s^{2^{k-1}}\cdot 2^{2^{k-1}-1}s^{2^{k-1}} = 2^{2^k-1}s^{2^k} . \end{aligned}$$

Analogously to (3) we obtain the desired upper bound on $max\_div(\psi)$.

We prove by induction over $k \geq 0$ that $max\_const(\varphi_k) \leq 2s^{\left(c^{2^{3k}}+a^{2^{3k}}+2^k\right)}$. The base case $k=0$ is obvious. For the step case, assume $k \geq 1$. We first give an upper bound on $N := max\_coef(\psi_k)^{\#coefs(\psi_k)+1}\cdot max\_div(\psi_k)^{\#divs(\psi_k)}$. Note that for every subformula ξ of φ_{k-1}, we have that $max\_coef(\xi) \leq max\_coef(\varphi_{k-1})$. Similar inequalities hold for the other parameters. Using our yet established upper bounds, we get that

$$\begin{aligned} N &\leq \big(s^{2^k}\big)^{\#coefs(\psi_k)+\#divs(\psi_k)+1} \leq \big(s^{2^k}\big)^{c^{2^{3(k-1)}}+a^{2^{3(k-1)}}+1} \\ &= s^{\left(2^k c^{2^{3(k-1)}}+2^k a^{2^{3(k-1)}}+2^k\right)} \leq s^{\left(c^{k+2^{3(k-1)}}+a^{k+2^{3(k-1)}}+2^k\right)} \\ &\leq s^{\left(c^{2^{3k}}+a^{2^{3k}}+2^k\right)} . \end{aligned} \tag{4}$$

An upper bound on $2max\_coef(\psi_k)$ is

$$2max\_coef(\psi_k) \leq 2\cdot 2^{2^{k-1}-1}s^{2^{k-1}} \leq 2^{2^{k-1}}s^{2^{k-1}} \leq s^{2^k} . \tag{5}$$

Using Theorem 14, it holds that

$$\begin{array}{rcl}
max_const(\varphi_k) & = & \max\big\{ max_const(\varphi_{k-1}), max_const(\chi_k)\big\} \\
& \leq & \max\big\{ max_const(\varphi_{k-1}), \\
& & \qquad 2max_coef(\psi_k)(max_const(\psi_k)+1)+N\big\} \\
& \leq & 2max_coef(\psi_k)(max_const(\varphi_{k-1})+1)+N \\
& \stackrel{(4)\text{ and }(5)}{\leq} & s^{2^k}\big(2max_const(\varphi_{k-1})\big) + s^{\left(c^{2^{3k}}+a^{2^{3k}}+2^k\right)} \\
& \stackrel{\text{IH}}{\leq} & s^{2^k}\left(4s^{\left(c^{2^{3(k-1)}}+a^{2^{3(k-1)}}+2^{k-1}\right)}\right) + s^{\left(c^{2^{3k}}+a^{2^{3k}}+2^k\right)} \\
& \leq & s^{\left(c^{2^{3(k-1)}}+a^{2^{3(k-1)}}+2^{k-1}+2^k+2\right)} + s^{\left(c^{2^{3k}}+a^{2^{3k}}+2^k\right)} \\
& \leq & 2s^{\left(c^{2^{3k}}+a^{2^{3k}}+2^k\right)} .
\end{array}$$

For $k = m$, we obtain the claimed bound on $max\_const(\psi)$. ⊣

4.4 Putting It All Together

We now prove our main result: the number of states of the minimal automaton for a formula is at most triple exponential in the length of the formula.

Let $\varphi(y_1, \ldots, y_n)$ be a formula of length m. Note that m is at least 3. The numbers of quantifiers, atomic formulas, and the maximal absolute integer in φ are at most m. Form Corollary 16 it follows that there is a logically equivalent, quantifier-free formula $\psi(y_1, \ldots, y_n)$ with

$$\#terms(\psi), \#divs(\psi) \leq m^{2^{3m}} = 2^{\lg m^{2^{3m}}} = 2^{2^{3m}\lg m} \leq 2^{2^{4m}}$$

and

$$max\_coef(\psi), max\_div(\psi) \leq m^{2^{m+1}} = 2^{\lg m^{2^{m+1}}} = 2^{2^{m+1}\lg m} \leq 2^{2^{2m}} .$$

Moreover, a strict upper bound on the maximal absolute value of a constant is

$$\begin{array}{l}
max_const(\psi) \leq 2m^{\left(m^{2^{3m}}+m^{2^{3m}}+2^m\right)} \leq 2m^{3m^{2^{3m}}} = 2^{1+3m^{2^{3m}}\lg m} \\
\qquad\qquad < 2^{2^m \cdot m^{2^{3m}}} \leq 2^{2^m \cdot 2^{2^{3m}\lg m}} \leq 2^{2^m \cdot 2^{2^{4m}}} \leq 2^{2^{2^{5m}}} .
\end{array} \tag{6}$$

We used the fact that $3x \lg y < 2^y x - 1$, for every $x, y \geq 2$.

We want to apply Theorem 13 to $\psi(y_1, \ldots, y_n)$, which is a Boolean combination of at most $\#terms(\psi)$ (in)equations and $max\_coef(\psi) \cdot max\_div(\psi)$ different formulas of the form $d \big| t + c$. Let $c_{min}, c_{max} := 2^{2^{2^{5m}}}$. From (6) we know that $c_{min}, c_{max} < max\_const(\psi)$. Since every term in ψ contains at most the variables $y_1, \ldots, y_n$, the sum of the absolute values of the coefficients in a term is bounded by

$$n2^{2^{2m}} \leq m2^{2^{2m}} < 2^{2^{2^{5m}}} .$$

From Theorem 13 we know that the minimal automaton for ψ has at most

$$
\begin{aligned}
1 + \left(2 \cdot 2^{2^{2^{5m}}}\right)^{2^{2^{4m}}} \cdot \left(2^{2^{4m}}\right)^{(2^{2^{4m}})^2} &\leq 1 + 2^{\left(1+2^{2^{5m}}\right)\cdot 2^{2^{4m}}} \cdot \left(2^{2^{4m}}\right)^{2^{2^{4m+1}}} \\
&\leq 1 + 2^{2^{4m}+2^{2^{5m}+2^{4m}}} \cdot 2^{2^{4m}\cdot 2^{2^{4m+1}}} \\
&\leq 1 + 2^{2\cdot 2^{2^{5m+1}}} \cdot 2^{2^{4m+2^{4m+1}}} \\
&\leq 1 + 2^{2^{2^{5m+2}}} \cdot 2^{2^{2^{4m+2}}} \leq 1 + 2^{2^{2^{5m+2}+1}} \\
&\leq 2^{2^{2^{6m}}}
\end{aligned}
$$

states. This proves our main result that the size of the minimal automaton is triple exponentially bounded by the length of the formula.

Theorem 17. *There is a constant $c \geq 0$ such that the minimal automaton for a formula of length m has at most $2^{2^{2^{cm}}}$ states.*

If we would have measured the length of the integers logarithmically and not unarily, the statement of the above theorem does not change (the concrete c might become larger). The only change is that the maximal absolute integer of φ is not bounded by m but by 2^m. We have to adjust the bounds on $max\_coef(\psi)$, $max\_div(\psi)$, and $max\_const(\psi)$. For instance $max\_coef(\psi)$ is now bounded by $(2^m)^{2^{m+1}} = 2^{m2^{m+1}}$ which is still less than $2^{2^{pm}}$ for some $p \geq 1$. Analogously for $max\_div(\psi)$ and $max\_const(\psi)$.

4.5 Automata-based Decision Procedures

We discuss some aspects of an automata-based decision procedures for Presburger arithmetic in regard to the newly gained insights.

The basic idea of an automata-based decision procedure for Presburger arithmetic is the following. For a formula $\varphi(x_1, \dots, x_r)$, we construct an automaton $\mathcal{A}$ representing the set $\{(u_1, \dots, u_r) \in \mathbb{Z}^r \mid \mathfrak{Z} \models \varphi[u_1, \dots, u_r]\}$. The automaton is constructed recursively over the structure of the formula, where, e. g., the logical connective $\wedge$ is handled by the product construction for automata for intersection, and the quantifier $\exists$ is mainly handled by projecting away (guessing) the corresponding track of the quantified variable (in addition to the projection we have to do some extra work for the first letter—the sign bit). The formula φ is valid iff $\mathcal{A}$ accepts all words over $\{0, 1\}^r$. More details can be found in [26].

Note that Theorem 17 *only* shows that the size of the minimal automaton is at most triple exponential in the length of the formula. (In the next section we show that this bound is tight.) We have *not* shown how to construct an automaton for a formula in triple exponential time or space. Handling the Boolean connectives and the quantifiers by standard automata constructions, like product and subset construction, yields a construction that uses at least one exponent more space. The reason is that there can occur an exponential blow-up each time we apply the subset construction. However, we know that the minimized automaton after this (possibly) expensive constructions has at most triple exponential many states. It is open if there are more efficient constructions than the standard ones.

Already the triple exponential bound might indicate that an automata-based decision procedure is far away from practical usage. But the insights obtained in §3 on the structure of the automata for Presburger arithmetic formulas give some intuitive reasons why—despite this daunting bound—existing implementations of automata-based decision procedures have competitive performance in practice [1, 10, 17].

One intuitive reason of the high complexity of Presburger arithmetic is the growth of the number of atomic formulas and the growth of the absolute values of integers while eliminating a quantifier. Every decision procedure has to tackle these large numbers. On the one hand, large absolute values of the coefficients in linear (in)equations are an intrinsic problem for automata. Corollary 7 shows that the minimal automaton for an (in)equation $t \lessgtr c$ has at least $\|t\| - 1$ states. This was already illustrated by experimental results in [10]. But on the other hand, Theorem 13 reveals the fact that automata "optimally share the homogeneous terms in the (in)equations" occurring in quantifier-free formulas.

A desirable data-structure for deciding Presburger arithmetic formulas would be a data-structure that could handle large coefficients *and* share the terms of the (in)equations in a similar optimal way as automata do. Moreover, this data-structure should be efficiently minimizable. A property that automata have: they can be minimized in time $\mathrm{O}(n \lg n)$ where n is the size of the automaton. One can think of minimizing the automaton corresponds to finding the shortest logically equivalent formula. An intuitive explanation why automata-based decision procedures have competitive performance is that automata have two out of these three properties.

5 A Worst Case Example

We give an example showing that the upper bound in Theorem 17 in §4.4 is tight. We are going, to use the formulas $Prod_n(x, y, z)$ with $n \geq 0$ defined in [9]. Fischer and Rabin looked at the structure $(\mathbb{N}, +)$ in [9], and not at $\mathfrak{Z}$, but it is straightforward to adopt the definition of $Prod_n(x, y, z)$ to $\mathfrak{Z}$. The formula $Prod_n(x, y, z)$ has linear length in $n \geq 0$, and it is true iff $xy = z$ and $0 \leq x, y, z < \prod_{p < f(n+2) \text{ and } p \text{ prime}} p$, where $f(n) := 2^{2^n}$. It follows immediately from the Prime Number Theorem that $\prod_{p < f(n+2) \text{ and } p \text{ prime}} p \geq 2^{f(n)^2} = 2^{2^{2^{n+1}}}$.

Theorem 18. *For every $n \geq 0$, the minimal automaton representing the set $\{(a, b, c) \in \mathbb{Z}^3 \,|\, \mathfrak{Z} \models Prod_n[a, b, c]\}$ has at least $2^{f(n)^2}$ states.*

We need the following fact to prove the above theorem.

Fact 19. Let $\ell \geq 1$. For all $z \in \{2^{\ell-1}, \dots, 2^\ell - 2\}$, there are $x, y, z' \in \{0, \dots, 2^\ell - 1\}$ such that $xy = 2^\ell z + z'$.

Proof. If $\ell = 1$ then there is nothing to prove since $\{2^{\ell-1}, \dots, 2^\ell - 2\} = \emptyset$. In the following, assume that $\ell > 1$ and $z \in \{2^{\ell-1}, \dots, 2^\ell - 2\}$. Let $x, y \in \{0, \dots, 2^\ell - 1\}$

with $xy \geq 2^\ell z$ and $z' := xy - 2^\ell z$ is minimal. Note that it is always possible to find x, y with $xy \geq 2^\ell z$, since for $x = y = 2^\ell - 1$, we have that

$$xy = (2^\ell - 1)^2 = 2^{2\ell} - 2^{\ell+1} + 1 \geq 2^{2\ell} - 2^{\ell+1} = 2^\ell(2^\ell - 2) \geq 2^\ell z\,.$$

For the sake of absurdity, assume that $z' \geq 2^\ell$. It follows that

$$(x-1)y = xy - y = 2^\ell z + z' - y \geq 2^\ell z$$

since $y < 2^\ell$ and $z' \geq 2^\ell$. This contradicts the minimality of z' since $xy > (x-1)y$. ⊣

Proof (Theorem 18). It suffices to show that an automaton representing the $n \geq 1$ bit multiplication has at least 2^n states. More precisely, we show that for every $n \geq 1$, the automaton $\mathcal{A}$ has at least 2^n states, where $\mathcal{A}$ represents the set $\{(x, y, z) \in \mathbb{Z}^3 \,|\, 0 \leq x, y \leq 2^n - 1 \text{ and } xy = z\}$.

In the following, $\sim$ denotes the equivalence relation on words over $\{0,1\}^3$ that is defined as $u \sim u'$ iff $uv \in L(\mathcal{A}) \Leftrightarrow u'v \in L(\mathcal{A})$ for all $v \in (\{0,1\}^3)^*$. Let K be the set of words $u \in (\{0\}^2 \times \{0,1\})^*$ with $2 \leq |u| \leq n+1$ and u starts with the letters $(0,0,0)(0,0,1)$. We show that $w \not\sim w'$, for distinct words $w, w' \in K$. From this it follows that $\mathcal{A}$ has at least 2^n states since the states p, q that are reached after reading words $u, u' \in K \cup \{(0,0,0)\}$ are identical only if $u = u'$. Note that cardinality of K is $2^0 + 2^1 + \cdots + 2^{n-1} = 2^n - 1$, and that for every $u \in K$, $(0,0,0) \not\sim u$ since $(0,0,0) \in L(\mathcal{A})$ and $u \notin L(\mathcal{A})$.

Let w be a word in $K \setminus \{(0,0,0)(0,0,1)^n\}$. Let $u := w$ if $w \neq (0,0,0)(0,0,1)^\ell$ and $u := w(0,0,0)$ otherwise, and let $\ell+1$ be the length of u. Note that $2 \leq \ell \leq n$. The third track of u encodes a natural number $z \in \{2^{\ell-1}, \ldots, 2^\ell - 2\}$, since the first letter of u is $(0,0,0)$, the second letter is $(0,0,1)$, and $u \neq (0,0,0)(0,0,1)^\ell$. From Fact 19, it follows that there are natural numbers $x, y, z' \in \{0, \ldots, 2^\ell - 1\}$ such that $xy = 2^\ell z + z'$. Let $\overline{b}_\ell, \ldots, \overline{b}_1 \in \{0,1\}^3$ be the binary representation of length ℓ of (x, y, z'), i.e. $(x, y, z') = \sum_{1 \leq i \leq \ell} 2^{i-1}\overline{b}_i$. We have that $u\overline{b}_\ell \ldots \overline{b}_1 \in L(\mathcal{A})$. Let $v := \overline{b}_\ell \ldots \overline{b}_1$ if $w = u$ and $v := (0,0,0)\overline{b}_\ell \ldots \overline{b}_1$ otherwise. It holds that $wv = u\overline{b}_\ell \ldots \overline{b}_1$ and v is also a binary representation of (x, y, z').

Now, let $w' \in K \setminus \{w\}$. Because the first and second track of wv and $w'v$ encode both the pair (x, y), and the third track of $w'v$ does not encode $2^\ell z + z'$, it cannot be the case that $w'v \in L(\mathcal{A})$. ⊣

6 Conclusion

The main result of this paper is that the size of the minimal automaton for a Presburger arithmetic formula is triple exponentially bounded in the length of the formula. The main technique to prove this bound was to relate the automata with the formulas constructed by quantifier elimination methods. This technique can also be used for finding bounds on the size of the minimal automata for other logics which are automatic [2, 11] and admit quantifier elimination. Like the mixed first-order theory over the structure $(\mathbb{R}, \mathbb{Z}, +, <)$ [4, 25], or the first-order theory of queues [20, 21]. Our analysis has also given some insights into the structure of the automata for Presburger formulas. As future work, we want to exploit these insights to develop automata constructions that manipulate these automata more efficiently, like for instance, specific constructions that handle quantification more efficiently than it is done by the standard subset construction.

References

1. C. Bartzis and T. Bultan, *Automata-based representations for arithmetic constraints in automated verification*, in CIAA'02, vol. 2608 of LNCS, pp. 282–288.
2. A. Blumensath and E. Grädel, *Automatic structures*, in LICS'00, pp. 51–62.
3. B. Boigelot, *Symbolic Methods for Exploring Infinite State Spaces*, PhD thesis, Faculté des Sciences Appliquées de l'Université de Liège, Liège, Belgium, 1999.
4. B. Boigelot, S. Jodogne, and P. Wolper, *On the use of weak automata for deciding linear arithmetic with integer and real veriables*, in IJCAR'01, vol. 2083 of LNCS, pp. 611–625.
5. A. Boudet and H. Comon, *Diophantine equations, Presburger arithmetic and finite automata*, in CAAP'96, vol. 1059 of LNCS, pp. 30–43.
6. J. Büchi, *Weak second-order arithmetic and finite automata*, Zeitschrift der mathematischen Logik und Grundlagen der Mathematik, 6 (1960), pp. 66–92.
7. D. Cooper, *Theorem proving in arithmetic without multiplication*, Machine Intelligence, 7 (1972), pp. 91–99.
8. J. Dixmier, *Proof of a conjecture by Erdös and Graham concerning the problem of Frobenius*, Journal of Number Theory, 34 (1990), pp. 198–209.
9. M. Fischer and M. Rabin, *Super-exponential complexity of Presburger arithmetic*, in Symposium on Applied Mathematics, vol. VII of SIAM-AMS Proceedings, New York, USA, 1974, pp. 27–41.
10. V. Ganesh, S. Berezin, and D. Dill, *Deciding Presburger arithmetic by model checking and comparisons with other methods*, in FMCAD'02, vol. 2517 of LNCS, pp. 171–186.
11. B. Khoussainov and A. Nerode, *Automatic presentations of structures*, in LCC'94, vol. 960 of LNCS, pp. 367–392.
12. D. Oppen, *Elementary bounds for Presburger arithmetic*, in STOC'73, pp. 34–38.
13. ———, *A $2^{2^{2^{pn}}}$ upper bound on the complexity of Presburger arithmetic*, Journal of Computer and System Sciences, 16 (1978), pp. 323–332.
14. M. Presburger, *Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einizge Operation hervortritt*, in Comptes Rendus du premier congrès des Mathématiciens des Pays Slaves, Warsaw, 1929, pp. 92–101.
15. ———, *On the completeness of a certain system of arithmetic of whole numbers in which addition occurs as the only operation*, History and Philosophy of Logic, 12 (1991), pp. 225–233. English translation of the article [14] by D. Jacquette.
16. W. Pugh, *The Omega test: a fast and practical integer programming algorithm for dependence analysis*, in Supercomputing '91, pp. 4–13.
17. R. Ranhan, T. Shiple, and J. Kukula, *A comparison of Presburger engines for EFSM reachability*, in CAV'98, vol. 1427 of LNCS, pp. 280–292.
18. C. Reddy and D. Loveland, *Presburger arithmetic with bounded quantifier alternation*, in STOC'78, pp. 320–325.

19. K. Reinhardt, *The complexity of translating logic to finite automata*, in Automata, Logics, and Infinite Games, E. Grädel, W. Thomas, and T. Wilke, eds., vol. 2500 of LNCS, 2002, ch. 13, pp. 231–238.
20. T. Rybina and A. Voronkov, *Upper bounds for a theory of queues*, in ICALP'03, vol. 2719 of LNCS, pp. 714–724.
21. ———, *A decision procedure for term algebras with queues*, ACM Transactions on Computational Logic, 2 (2001), pp. 155–181.
22. T. Skolem, *Über einige Satzfunktionen in der Arithmetik*, in Skrifter utgitt av Det Norske Videnskaps-Akademi i Oslo, I. Matematisk naturvidenskapelig klasse, vol. 7, Oslo, 1931, pp. 1–28.
23. ———, *Über einige Satzfunktionen in der Arithmetik*, in Selected Works in Logic, J. Fenstad, ed., Universitetsforlaget, Oslo, 1970, pp. 281–306. Reprint of the article [22].
24. L. Stockmeyer, *The complexity of decision problems in automata theory and logic*, PhD thesis, Departement of Electrical Engineering, MIT, Boston, MA, USA, 1974.
25. V. Weispfenning, *Mixed real-integer linear quantifier elimination*, in ISSAC'99, pp. 129–136.
26. P. Wolper and B. Boigelot, *On the construction of automata from linear arithmetic constraints*, in TACAS'00, vol. 1785 of LNCS, pp. 1–19.

A Additional Proof Details to Lemma 6

In the appendix, we prove Lemma 6(a). Let $p \in S$ and let $q \in Q \setminus \{q_\mathrm{I}\}$. Assume that t has the form $k_1 \cdot x_1 + \cdots + k_r \cdot x_r$. The case where $r = 1$ is trivial. It holds that $k_1 = 1$ or $k_1 = -1$ since we always assume $\gcd(t) = 1$. It follows that $S = \emptyset$, for $r = 1$. In the remainder of the proof, assume that $r \geq 2$, and let $i_1, \ldots, i_\mu$ be the indices of the positive coefficients, i.e., $k_{i_\xi} > 0$, for all $1 \leq \xi \leq \mu$. And, let $j_1, \ldots, j_\nu$ be the indices of the negative coefficients, i.e., $k_{i_\xi} < 0$, for all $1 \leq \xi \leq \nu$.

It is straightforward to show that the states q_small and q_large are reachable from p. For instance, $\delta(p, \overline{b}^N) = q_\mathrm{large}$, where N is the number of states of $\mathcal{B}$ and $\overline{b} \in \{0,1\}^r$ is the letter in which the ith coordinate of $\overline{b}$ is 1 iff the coefficient of the variable x_i is positive.

In the following, let $q \in Q \cap \mathbb{Z}$. The main idea of the proof is to translate the question, whether q is reachable from p to an equivalent number theoretic problem. For answering this number theoretic question, we need the following notions and facts.

The **Frobenius number** of $0 < c_1 < \cdots < c_n$ with $\gcd(c_1, \ldots, c_n) = 1$, denoted by $G(c_1, \ldots, c_n)$, is the greatest integer z for which the linear Diophantine equation $c_1 \cdot x_1 + \cdots + c_n \cdot x_n = z$ has *not* a solution in the natural numbers. For $n = 1$, trivially it holds that $G(c_1) = -1$. The upper bound $G(c_1, \ldots, c_n) \leq \frac{c_n^2}{n-1}$, for $n \geq 2$ was proved by Dixmier [8].

Proposition 20. *For $0 < c_1 < \cdots < c_n$ with $\gcd(c_1, \ldots, c_n) = 1$, we have that $G(c_1, \ldots, c_n) < 2^{c_1 + \cdots + c_n} - (c_1 + \cdots + c_n)$.*

Proof. The proof for $n = 1$ is trivial since $c_1 = 1$. Let $n \geq 2$. It is straightforward to check the claim for $c_n \leq 3$ since there are only finitely many cases. Namely, $c_1 = 1$ and $c_2 = 2$, or $c_1 = 1$ and $c_2 = 3$, or $c_1 = 2$ and $c_2 = 3$, or $c_1 = 1$ and $c_2 = 2$ and $c_3 = 3$. Assume that $c_n > 3$. We have that

$$2^{c_1 + \cdots + c_n} - (c_1 + \cdots + c_n) > 2^{c_n} 2^{c_1 + \cdots + c_{n-1}} - n \cdot c_n \geq 2^{c_n} 2^{n-1} - c_n^2 .$$

It suffices to check that $\frac{c_n^2}{n-1} \leq 2^{c_n} 2^{n-1} - c_n^2$. This inequality is equivalent to

$$\frac{c_n^2}{2^{c_n}(n-1)} \leq 2^{n-1} - \frac{c_n^2}{2^{c_n}} .$$

Since $\frac{c_n^2}{2^{c_n}} \leq 1$ for $c_n > 3$ (which can be easily checked by induction over c_n), the previous inequality can be simplified to $\frac{1}{n-1} \leq 2^{n-1} - 1$. This inequality can be easily checked by induction over $n \geq 2$. ⊣

Furthermore, we need the following inequalities.

Proposition 21. *The following inequalities are true.*

(a) $z \leq 2^z$, for all $z \geq 0$.

(b) $y + z + (y-1)^2 \leq 2^{y+z}$, *for all* $y, z \geq 0$.
(c) $xy + xz + x(y-1)^2 \leq 2^{xy+z}$, *for all* $x, y \geq 1$ *and* $z \geq 0$.

Proof. (a). The inequality $z \leq 2^z$ is easily checked by induction over z.

(b). We prove the inequality by induction over y. The base case $z \leq 2^z$ follows directly from (a). For the step case, assume $2y + z - 1 \leq 2^{y+z}$. It holds,

$$\begin{aligned} y + 1 + z + y^2 &= y + z + (y^2 - 2y + 1) + 2y = y + z + (y-1)^2 + 2y \\ &\leq 2\big(y + z + (y-1)^2\big) \overset{\text{IH}}{\leq} 2 \cdot 2^{y+z} = 2^{y+1+z} . \end{aligned}$$

(c). We prove the inequality by induction over x. The base case $y+z+(y-1)^2 \leq 2^{y+z}$ follows directly from (b). For the step case, assume $xy + xz + x(y-1)^2 \leq 2^{xy+z}$ where $x \geq 1$. It holds,

$$\begin{aligned} (x+1)y + (x+1)z + (x+1)(y-1)^2 &= xy + xz + x(y-1)^2 + y + z + (y-1)^2 \\ &\overset{x \geq 1}{\leq} 2\big(xy + xz + x(y-1)^2\big) \\ &\overset{\text{IH}}{\leq} 2 \cdot 2^{xy+z} \\ &\overset{y \geq 1}{\leq} 2^{xy+z+y} = 2^{(x+1)y+z} . \end{aligned}$$

⊣

Now, we are ready to show that state $q \in Q \cap \mathbb{Z}$ is reachable from state $p \in S$. Without loss of generality, we assume $p \neq q$. The question, whether q is reachable from p is translated by the following fact to a number theoretic question.

Fact 22. State q is reachable from state $p \neq q$ iff there are $a_1, \dots, a_r \geq 0$ and $n \geq \lceil \lg(1 + \max\{a_1, \dots, a_r\}) \rceil$ with $t(a_1, \dots, a_r) + p2^n = q$.

Proof. ($\Rightarrow$). Assume $\delta(p, \overline{b}_n \dots \overline{b}_1) = q$ for some $\overline{b}_1, \dots, \overline{b}_n \in \{0,1\}^r$. We have, $n \geq \lceil \lg(1+a) \rceil$ where a is the greatest number encoded in $\overline{b}_n \dots \overline{b}_1$. It follows from Lemma 3(a) that

$$q = \delta(p, \overline{b}_n \dots \overline{b}_1) = p2^n + t(\textstyle\sum_{1 \leq i \leq n} 2^{i-1} \overline{b}_i) .$$

($\Leftarrow$). Let $\overline{b}_m \dots \overline{b}_1 \in (\{0,1\}^r)^*$ be the shortest word encoding the tuple $(a_1, \dots, a_r)$, i. e., $\overline{b}_m \neq \overline{0}$ and $\sum_{1 \leq i \leq m} 2^{i-1} \overline{b}_i = (a_1, \dots, a_r)$. Again, by Lemma 3(a), we have that

$$\delta(p, \overline{0}^{n-m} \overline{b}_m \dots \overline{b}_1) = p2^n + t(a_1, \dots, a_r) = q .$$

⊣

By Fact 22 it suffices to prove that there are $a_1, \dots, a_r \in \{0, \dots, 2^N - 1\}$ such that $t(a_1, \dots, a_r) + p2^N = q$. Recall that N is the number of states of the automaton $\mathcal{B}$, i. e., $N = 3 + \max\{|c|, \|t\|_+\} + \max\{c, \|t\|_-\}$.

Case I: $p = 0$. Since $p = 0$ there are positive and negative coefficients in t. Therefore, the equation $t(x_1, \dots, x_r) = q$ has infinitely many positive solutions since $\gcd(t) = 1$, by assumption. In particular, there are $a_1, \dots, a_r \geq 0$ with $t(a_1, \dots, a_r) = q$.

Case II: $p > 0$ and $q \geq 0$. Note that $\nu > 0$ since $p > 0$. We rewrite $t(x_1, \dots, x_r) + p2^N = q$ to the linear Diophantine equation

$$p2^N - q + k_{i_1} \cdot x_{i_1} + \dots + k_{i_\mu} \cdot x_{i_\mu} = |k_{j_1}| \cdot x_{j_1} + \dots + |k_{j_\nu}| \cdot x_{j_\nu} \, . \tag{7}$$

Note that $p2^N - q \geq 0$ since $p > 0$ and $2^N \geq q$. For showing the existence of a solution $a_1, \dots, a_r \in \{0, \dots, 2^N - 1\}$ of the linear Diophantine equation (7), we proceed in two steps:

Step 1: There are $a_{i_1}, \dots, a_{i_\mu} \in \{0, \dots, \gcd(|k_{j_1}|, \dots, |k_{j_\nu}|) - 1\}$ such that

$$\gcd(|k_{j_1}|, \dots, |k_{j_\nu}|) \big| p2^N - q + k_{i_1} a_{i_1} + \dots + k_{i_\mu} a_{i_\mu} \, .$$

Step 2: There are $a_{j_1}, \dots, a_{j_\nu} \in \{0, \dots, 2^N - 1\}$ such that

$$p2^N - q + k_{i_1} a_{i_1} + \dots + k_{i_\mu} a_{i_\mu} = |k_{j_1}| a_{j_1} + \dots + |k_{j_\nu}| a_{j_\nu} \, ,$$

where $a_{i_1}, \dots, a_{i_\mu}$ are from Step 1.

The existence of a solution in $\{0, \dots, 2^N - 1\}$ for the linear Diophantine equation (7) follows from Step 1 and Step 2.

Proof of Step 1: If $\mu = 0$ then there is nothing to prove. We assume in the remainder of the proof of Step 1 that $\mu > 0$.

Let $D := \gcd(|k_{j_1}|, \dots, |k_{j_\nu}|)$. There are $K, R \geq 0$ such that $p2^N - q = DK + R$ with $R < D$. It suffices to show that there are $0 \leq a_{i_1}, \dots, a_{i_\mu} < D$ and $K' \geq 0$ with $DK' = R + k_{i_1} a_{i_1} + \dots + k_{i_\mu} a_{i_\mu}$, since then

$$\begin{aligned} p2^N - q + k_{i_1} a_{i_1} + \dots + k_{i_\mu} a_{i_\mu} &= DK + R + k_{i_1} a_{i_1} + \dots + k_{i_\mu} a_{i_\mu} \\ &= DK + DK' = D(K + K') \, , \end{aligned}$$

this means that $D \big| p2^N - q + k_{i_1} a_{i_1} + \dots + k_{i_\mu} a_{i_\mu}$.

First, assume the existence of $a_{i_1}, \dots, a_{i_\mu} \geq 0$ with $D \big| R + k_{i_1} a_{i_1} + \dots + k_{i_\mu} a_{i_\mu}$, where $a_{i_\xi} > D$, for some $1 \leq \xi \leq \mu$. To simplify matters, we can assume without loss of generality that $\xi = 1$. There is a $a \geq 0$ with $a_{i_1} = D + a$. Further, assume that there is not a $b < a_{i_1}$ with $D \big| R + k_{i_1} b + k_{i_2} a_{i_2} + \dots + k_{i_\mu} a_{i_\mu}$. For some $K' \geq 0$, we have

$$DK' = R + k_{i_1}(D + a) + k_{i_2} a_{i_2} + \dots + k_{i_\mu} a_{i_\mu} k_{i_1} D + R + k_{i_1} a + k_{i_2} a_{i_2} + \dots + k_{i_\mu} a_{i_\mu}$$

Therefore, $D(K' - k_{i_1}) = R + k_{i_1} a + k_{i_2} a_{i_2} + \dots + k_{i_\mu} a_{i_\mu}$, i.e. $D \big| R + k_{i_1} a + k_{i_2} a_{i_2} + \dots + k_{i_\mu} a_{i_\mu}$. This is a contradiction to the minimality of a_{i_1}.

It remains to show the existence of $a_{i_1}, \dots, a_{i_\mu} \geq 0$ with $D \big| R + k_{i_1} a_{i_1} + \cdots + k_{i_\mu} a_{i_\mu}$. The existence reduces to the problem whether the linear Diophantine equation

$$D \cdot y - k_{i_1} a_{i_1} - \cdots - k_{i_\mu} a_{i_\mu} = R$$

has a solution in the natural numbers. This is always the case since $\gcd(D, k_{i_1}, \dots, k_{i_\mu}) = 1$, by assumption.

Proof of Step 2: Assume that there are $\eta \geq 1$ distinct coefficients in the linear Diophantine equation (7). Without loss of generality assume that $0 < |k_{j_1}| < \cdots < |k_{j_\eta}|$.

Let $W := \frac{p2^N - q + k_{i_1} a_{i_1} + \cdots + k_{i_\mu} a_{i_\mu}}{D}$, $L := \frac{\|t\|_-}{D}$, and $\ell_\xi := \frac{|k_{j_\xi}|}{D}$, for $1 \leq \xi \leq \nu$. Note that $\ell_1 < \cdots < \ell_\eta$ and that $\gcd(\ell_1, \dots, \ell_\eta) = 1$. The linear Diophantine equation (7) simplifies with the a_is from Step 1 to

$$\ell_1 \cdot x_{j_1} + \cdots + \ell_\nu \cdot x_{j_\nu} = W \,. \tag{8}$$

It suffices to show that the linear Diophantine equation (8) has a solution in $\{0, \dots, 2^N - 1\}$. An upper bound on W is

$$W \leq \frac{p2^N - q + (D-1)\|t\|_+}{D} \tag{9}$$

and a lower bound on W is

$$\begin{aligned} W &\geq \frac{2^N - q}{D} \geq \frac{2^N - \max\{c, \|t\|_-\}}{D} \geq \frac{2^{D(\ell_1 + \cdots + \ell_\nu)} - D(\ell_1 + \cdots + \ell_\nu)}{D} \\ &\geq 2^{\ell_1 + \cdots + \ell_\eta} - (\ell_1 + \cdots + \ell_\eta) \,. \end{aligned}$$

From Proposition 20, it follows that there are $a_1, \dots, a_\eta \geq 0$ with

$$\kappa L + \ell_1 a_1 + \cdots + \ell_\eta a_\eta = W \,, \tag{10}$$

for some $\kappa \geq 0$. Let $\kappa \geq 0$ be maximal such that there are $a_1, \dots, a_\eta \geq 0$ satisfying (10). We have that $a_1, \dots, a_\eta < L$. For the sake of contradiction, assume that there is a $1 \leq \xi \leq \eta$ with $a_\xi = L + a$, for some $a \geq 0$. Without loss of generality assume that $\xi = 1$. This contradicts the assumption that κ is maximal:

$$\begin{aligned} W &= \kappa L + u = \kappa L + \ell_1(L + a) + \ell_2 a_2 + \cdots + \ell_\eta + a_\eta \\ &= (\kappa + \ell_1)L + \ell_1 a + \ell_2 a_2 + \cdots + \ell_\eta + a_\eta \,. \end{aligned}$$

From κ and $a_1, \dots, a_\eta$, we obtain a solution for the Diophantine equation (8) in the natural numbers, namely

$$\begin{aligned} W &= \kappa L + \ell_1 a_1 + \cdots + \ell_\eta a_\eta = \kappa(\ell_1 + \cdots + \ell_\nu) + \ell_1 a_1 + \cdots + \ell_\eta a_\eta \\ &= \ell_1(\kappa + a_1) + \cdots + \ell_\eta(\kappa + a_\eta) + \ell_{\eta+1}\kappa + \cdots + \ell_\nu \kappa \,. \end{aligned}$$

It suffices to show that $\kappa \leq 2^N - 1 - \max\{a_1, \dots, a_\eta\}$. An upper bound on κ is

$$\begin{aligned}
\kappa &\overset{(10)}{=} \tfrac{W-(\ell_1 a_1+\dots+\ell_\eta a_\eta)}{L} \leq \tfrac{W-\max\{a_1,\dots,a_\eta\}}{L} \\
&\overset{(9)}{\leq} \tfrac{p2^N-q+(D-1)\|t\|_+}{DL} - \tfrac{\max\{a_1,\dots,a_\eta\}}{L} \\
&\leq \tfrac{(\|t\|_- -1)2^N}{DL} + \tfrac{D-1}{DL}\|t\|_+ - \tfrac{\max\{a_1,\dots,a_\eta\}}{L} \\
&\leq 2^N - \tfrac{2^N}{DL} + \tfrac{\|t\|_+ - \max\{a_1,\dots,a_\eta\}}{L}\,.
\end{aligned}$$

We have to check if the following inequality is valid

$$2^N - \frac{2^N}{DL} + \frac{\|t\|_+ - \max\{a_1,\dots,a_\eta\}}{L} \leq 2^N - 1 - \max\{a_1,\dots,a_\eta\}\,.$$

The previous inequality rewrites to

$$1 + \frac{\|t\|_+ + \max\{a_1,\dots,a_\eta\}(L-1)}{L} \leq \frac{2^N}{DL}\,.$$

Multiplying with the common denominator DL, the inequality rewrites further to

$$DL + D\|t\|_+ + D\max\{a_1,\dots,a_\eta\}(L-1) \leq 2^N\,.$$

Since $\max\{a_1,\dots,a_\eta\} \leq L-1$ and $N \geq \|t\| = DL + \|t\|_+$, it suffices to show the validity of the inequality

$$DL + D\|t\|_+ + D(L-1)^2 \leq 2^{DL+\|t\|_+}\,. \tag{11}$$

By Proposition 21(c), the inequality (11) is true for all $D, L \geq 1$ and $\|t\|_+ \geq 0$.

Case III: $p < 0$ and $q \leq 0$. Note that $\mu > 0$ since $p < 0$. This case reduces to Case II since it suffices to show the existence of a solution $a_1, \dots, a_r \in \{0, \dots, 2^{\|t\|} - 1\}$ for

$$k_{i_1}\cdot x_{i_1} + \dots + k_{i_\mu}\cdot x_{i_\mu} = |p|2^{\|t\|} - |q| + |k_{j_1}|\cdot x_{j_1} + \dots + |k_{j_\mu}|\cdot x_{j_\nu}\,. \tag{12}$$

(12) is symmetric to (7).

Case IV: $p > 0$ and $q \leq 0$. This case can be solved with Case I and Case II. Since $p > 0$ and $q \leq 0$, we have that $0 \in S$. By Case II, the state 0 is reachable from p, and by Case I, q is reachable from state 0.

Case V: $p < 0$ and $q \geq 0$. Analogously, this case can be solved by Case I and Case III. ⊣