AAECC
DOI 10.1007/s00200-007-0053-x

Weak quantifier elimination for the full linear theory of the integers

A uniform generalization of Presburger arithmetic

Aless Lasaruk · Thomas Sturm

Received: 9 April 2006 / Revised: 22 March 2007

Abstract We describe a weak quantifier elimination procedure for the full linear theory of the integers. This theory is a generalization of Presburger arithmetic, where the coefficients are arbitrary polynomials in non-quantified variables. The notion of weak quantifier elimination refers to the fact that the result possibly contains bounded quantifiers. For fixed choices of parameters these bounded quantifiers can be expanded into disjunctions or conjunctions. We furthermore give a corresponding extended quantifier elimination procedure, which delivers besides quantifier-free equivalents also sample values for quantified variables. Our methods are efficiently implemented within the computer logic system REDLOG, which is part of REDUCE. Various examples demonstrate the applicability of our methods. These examples include problems currently discussed in practical computer science.

Keywords Quantifier elimination · Integer constraint solving · Implementation

1 Introduction

After the fundamental work of Presburger [19] there has been considerable research on Presburger arithmetic, which is the additive theory of the integers with ordering and congruences. The largest part of this research was concerned with complexity issues and with decidability [1–3,11–14].

A. Lasaruk (✉)
FORWISS, Universität Passau, 94030 Passau, Germany
e-mail: lasaruk@uni-passau.de

T. Sturm
FIM, Universität Passau, 94030 Passau, Germany
e-mail: sturm@uni-passau.de

Weispfenning [28,29] was the first one who was explicitly interested in quantifier elimination as such in contrast to using it as a technique for decision. His quantifier elimination procedures are triply exponential, which is known to be optimal [13]. He managed, however, to optionally decrease that complexity by one exponential step to doubly exponential using the following technical trick: certain systematic disjunctions occurring during the elimination process are not written down explicitly. Instead one uses big $\bigvee$ (disjunction) and $\bigwedge$ (conjunction) operators with an index variable running over a finite range of integers. It is important to understand that at any time these big operators could be expanded such that one obtains a regular first-order formula at the price of considerably increasing the size.

In parallel Weispfenning and others have developed virtual substitution techniques for quantifier elimination in various theories starting with the reals and including also valued fields and Boolean algebras [17,20,24,27,30]. The present paper combines the two research areas by presenting for the first time integer quantifier elimination within the framework of virtual substitution. Moreover it extends that framework in order to cover a considerable generalization of Presburger arithmetic: We admit as coefficients arbitrary polynomials in the parameters, i.e., the unquantified variables. In other words, we use the language of rings together with ordering and ternary congruence relations with parametric moduli and impose the following restriction: Considering all terms as polynomials, the total degree with respect to the quantified variables must not exceed 1. We call this the full linear theory of the integers. It perfectly corresponds to what is referred to as linear quantifier elimination for the reals and for valued fields [17,24]. Recall that in regular Presburger arithmetic, in contrast, all coefficients must be numbers. The difference vanishes, however, when considering sentences and decision problems.

There is a price to pay for our new generality. We use big operators in the style of Weispfenning but with a crucial difference: The index variable no longer ranges over a fixed set of integers. Instead the range is determined by conditions possibly involving parameters. Consequently, our big operators cannot be expanded to regular first-order formulas unless one plugs in values for the parameters before. Strictly speaking, we thus do not really have a quantifier elimination procedure in the traditional sense. This would require to deliver equivalent quantifier-free *first-order* formulas. In terms of model theory our big operators are bounded quantifiers. To make this difference clear, we refer to our approach as *weak* quantifier elimination.

In comparison to other work [29] our approach implies various technical improvements, which turned out crucial for an applicable implementation: We can handle very liberal conditions on the range of the index variables of bounded quantifiers. In particular our ranges need not be centered around 0, and as a consequence, estimating their limits does not require considering the least common multiples of all leading coefficients in the formula. After fixing values for all parameters, this decreases the size of the obtained ranges by a factor that is exponential in the obtained values of the leading coefficients.

Devising quantifier elimination procedures by virtual substitution systematically reduces to constructing suitable finite elimination sets. Accordingly, correctness proofs for such procedures systematically reduce to proofs that certain sets are elimination sets, and there are well-established proof techniques for this [24,27]. Our step from

regular quantifier elimination to weak quantifier elimination corresponds on the technical side to generalizing the notion of an elimination set to that of a parametric elimination set. We consider it a major original result of the present article to present a technique for proving the parametric elimination set property. This technique does not at all depend on the theory of the integers but is generally applicable.

Further original contributions include an extended quantifier elimination procedure for the pure Presburger case and a corresponding extended weak quantifier elimination procedure for the full linear theory of the integers. The idea of extended quantifier elimination is to provide, besides quantifier-free equivalents, sample values for the eliminated quantified variables. We demonstrate that already for the pure Presburger case the usual data structure for the result as originally specified by Weispfenning for the reals [31] should be extended unless one accepts an exponential blowup.

To our knowledge we are describing here the only available implementation of a quantifier elimination procedure for the integers with parameters at all. Our implementation is part of the REDUCE package REDLOG [5,8]. REDLOG provides an extension of the computer algebra system REDUCE to a computer logic system implementing symbolic algorithms on first-order formulas with respect to temporarily fixed first-order languages and theories. Hence our implementation is embedded in an integrated environment that is widely accepted in the scientific community for more than 10 years now.

The plan of this paper is as follows. We introduce in Sect. 2 the notions of bounded quantifiers and parametric elimination sets. On this basis we give our central elimination theorem. Section 3 then turns to extended quantifier elimination procedures for the integers. In Sect. 4 we give an overview on our implementation. This description is not complete but sufficient to get a good impression about the implementation and the framework around it. In particular it enables the reader to reproduce the computation examples described in the following Sect. 5. Some of our examples emphasize the applicability of our methods in practical computer science. In Sect. 6 we finally summarize and evaluate our work.

2 Weak quantifier elimination

Already in 1990, Weispfenning has given an effective quantifier elimination procedure for conventional Presburger arithmetic [28]. This procedure was, however, not yet described in terms of the virtual substitution framework, which is meanwhile well-established for similar work mainly over the reals and the p-adic numbers [24,27,30]. It can, however, be reanalyzed as a procedure of that type. We start out doing so by means of an example. Next we are going to make precise definitions generalizing the corresponding notion of an elimination set to the more general parametric situation considered here. Then we establish the notion of weak quantifier elimination for this parametric situation, and prove a corresponding elimination theorem.

2.1 Reanalysis of conventional Presburger arithmetic

Consider the following formula in conventional Presburger arithmetic:

$$\exists x(5x = b). \tag{1}$$

For this, the following is a quantifier-free equivalent according to [28]:

$$\bigvee_{-5\le k\le 5} \big(b+k \equiv_5 0 \wedge 5(b+k) = 5b\big). \tag{2}$$

From the contained equation, one easily derives $k = 0$ and thus the equivalence of the entire elimination result to $b \equiv_5 0$. Anyway, let us concentrate on understanding the original result (2) in the right way for our purposes here.

In terms of virtual substitution, we have derived from formula (1) an *elimination set*

$$E = \left\{ \left(b+k \equiv_5 0,\, \tfrac{b+k}{5}\right) \;\middle|\; -5 \le k \le 5 \right\}. \tag{3}$$

Generally, a *regular elimination set* is a finite set

$$E = \big\{(\gamma_1, t_1), \ldots, (\gamma_n, t_n)\big\} \tag{4}$$

of *guarded points*. The t_i are pseudo-terms derived as solutions of equations corresponding to characteristic points of the solution sets of atomic formulas contained in the original formula. One example for such characteristic points are the boundaries of intervals given by inequalities. By pseudo-term, we mean that the t_i are possibly constructed in a language expanding the original one. For instance, the terms in our example contain division by 5. The γ_i are quantifier-free formulas over the original language.

The general idea is to eliminate an existential quantifier according to the following scheme:

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma,t)\in E} \big(\gamma \wedge \varphi[t /\!\!/ x]\big), \tag{5}$$

where $[t /\!\!/ x]$ denotes a generalized substitution of t for x in such a way that the substitution result is a formula in the original language. The corresponding guard γ serves as a filter deleting (by means of equivalence to false) substitutions that have no proper interpretation in the original structure. For instance, in our set E in (3) above the guards take care that the pseudo-terms actually describe integers. In fact, the range $-5 \le k \le 5$ in E has been chosen such that at least one pseudo-term describes an integer.

It remains to be clarified how the generalized substitution works. We give the substitution rules for atomic formulas with $=$, $\le$, and $\equiv_m$:

$$\begin{aligned}
(ax = b)\left[\tfrac{b'}{a'} /\!\!/ x\right] &:= (ab' = a'b),\\
(ax \le b)\left[\tfrac{b'}{a'} /\!\!/ x\right] &:= (aa'b' \le a'^2 b),\\
(ax \equiv_m b)\left[\tfrac{b'}{a'} /\!\!/ x\right] &:= (ab' \equiv_{ma'} a'b).
\end{aligned} \tag{6}$$

From these one can easily derive those for all other relations in our language L, which we are going to introduce right at the beginning of the next subsection. Furthermore, the substitution naturally generalizes to quantifier-free formulas.

In the framework of pure Presburger arithmetic a, b, a', b', and m are integers. We are going to use these rules also for our extended framework. There these objects are going to be terms. Our slightly more complicated substitution rule for inequalities reflects the fact that a' is possibly a negative number.

2.2 Bounded quantifiers

Let us now try to generalize the approach sketched in the previous subsection. We consider the language of ordered rings with congruences. For convenience, we add relation symbols, such that the language becomes closed under logical negation:

$$L = \left\{0^{(0)}, 1^{(0)}, -^{(1)}, +^{(2)}, \cdot^{(2)}; \neq^{(2)}, \leq^{(2)}, >^{(2)}, \geq^{(2)}, <^{(2)}, \equiv^{(3)}, \not\equiv^{(3)}\right\}. \tag{7}$$

Our domain are the integers $\mathbb{Z}$, which clarifies the semantics of the symbols in L. We assume without loss of generality that all terms with variables $a_1, \ldots, a_r, x_1, \ldots, x_s$ are distributive multivariate polynomials in $\mathbb{Z}[a_1, \ldots, a_r, x_1, \ldots, x_s]$.

The following is a generalized input formula corresponding to the example in (1).

$$\exists x(ax = b) \tag{8}$$

Following our ideas of the previous subsection, we would obtain the following elimination set for this:

$$E = \left\{ \left(a \neq 0 \wedge b + k \equiv_a 0, \tfrac{b+k}{a}\right) \;\middle|\; -|a| \leq k \leq |a| \right\} \cup \left\{(\text{true}, 0)\right\}. \tag{9}$$

We have added the guarded point (true, 0) in order to substitute at least one test point also for the case $a = 0$, when the equation becomes trivial.

Application of the substitution scheme discussed above yields the following output:

$$\bigvee_{-|a| \leq k \leq |a|} \left(a \neq 0 \wedge b + k \equiv_a 0 \wedge a(b + k) = ab\right) \vee \left(\text{true} \wedge a \cdot 0 = b\right). \tag{10}$$

Following the same idea as for the non-uniform variant and observing that $0 = b$ is a special case of $b \equiv_a 0$, this can be simplified to the quantifier-free formula $b \equiv_a 0$, which is equivalent to the input formula.

Unfortunately, we cannot expect such nice simplifications to be possible in general. Consequently, we have to expect objects like the one given as (10) to possibly establish final or intermediate elimination results. For the successive elimination of several quantifiers we possibly even have to process such objects in the input.

Let us have a closer look at that object. To start with, there occur symbols for the absolute value of a, which are not contained in our language. These can be considered as a short notation for a corresponding case distinction on the sign of a. There is, however, a substantial reason why (10) is by no means a valid first-order formula: The number of disjuncts in the big disjunction depends on a, and there is no restriction at

all on the possible choices for a. Hence it cannot be considered a short notation for a disjunction in the sense of first-order formulas.

Recall that uniform Presburger arithmetic in the form considered here is well-known to be undecidable [29]. From this it follows in turn that it does not admit quantifier elimination. From that point of view it is not too surprising that such problems crop up.

On the positive side, the present paper is going to demonstrate that quantifier elimination becomes possible when admitting formal objects corresponding to the questionable disjunction in (10). We are now going to make precise definitions for this.

We extend the language of logic by two additional quantifiers $\bigsqcup_{k:\beta}$ and $\bigsqcap_{k:\beta}$. Here k is a variable, and β is a formula in k and the parameters $a_1, \ldots, a_r$ not containing any quantifier. The semantics of the new quantifiers is defined as follows:

$$\bigsqcup_{k:\beta} \varphi \quad \text{iff} \quad \exists k(\beta \wedge \varphi), \qquad \bigsqcap_{k:\beta} \varphi \quad \text{iff} \quad \forall k(\beta \longrightarrow \varphi). \tag{11}$$

The quantifier $\bigsqcup_{k:\beta}$ is called an *existential bounded quantifier* if the solution set

$$S^k_\beta(z_1, \ldots, z_r) = \left\{ z \in \mathbb{Z} \;\middle|\; \beta(z, z_1, \ldots, z_r) \right\} \tag{12}$$

of β with respect to k is finite for all interpretations $z_1, \ldots, z_r \in \mathbb{Z}$ of $a_1, \ldots, a_r$. We call β a *k-bound* or sometimes more generally a *bound-condition*. We call k the *bound-variable*, and $a_1, \ldots, a_r$ the *bound-parameters*. Under the same condition $\bigsqcap_{k:\beta}$ is called a *universal bounded quantifier*. Formulas containing no quantifiers at all are called *strictly quantifier-free*. Formulas containing exclusively bounded quantifiers are called *weakly quantifier-free*.

The choice of notation obviously resembles Weispfenning's big disjunction and conjunction operators. In general, however, bounded quantifiers can be explicitly expanded only for fixed choices of all corresponding bound-parameters. The finite solution sets of all k-bounds can then be effectively enumerated. The idea is to start with the outermost bounded quantifier and to successively determine the finite solutions sets by means of DNF computations.

A term t is *linear* in $x_1, \ldots, x_s$ if it does not contain any products $x_i x_j$, in particular no powers of any x_i. A congruence $t \equiv_m t'$ is linear in $x_1, \ldots, x_s$ if so are both t and t', and additionally m does not contain $x_1, \ldots, x_s$ at all. The same definition applies to incongruences. All other atomic formulas are linear in $x_1, \ldots, x_s$ if so are both contained terms.

A formula is *linear*, if each contained atomic formula is linear in the set of variables that are bounded by some quantifier at the place of its occurrence and each occurrence of any of our new quantifiers is in fact a bounded quantifier.

Notice that linear formulas resemble formulas of Presburger arithmetic with the generalization that all coefficients are polynomials in some parameters $a_1, \ldots, a_r$. Therefore our framework has also been referred to as *uniform Presburger arithmetic* in the literature [29].

A *weak quantifier elimination* procedure is an algorithm assigning to any first-order formula φ a weakly quantifier-free formula φ^* such that

$$\varphi^* \longleftrightarrow \varphi. \tag{13}$$

Accordingly, a *strict quantifier elimination* procedure assigns to any first-order formula a strictly quantifier-free formula with property (13). That is, strict quantifier elimination corresponds to the standard definition of quantifier elimination.

It is now not hard to see that our questionable object (10) corresponds to a weakly quantifier-free formula

$$\bigsqcup_{k:\,-|a|\le k\le|a|} \big(a \neq 0 \wedge b + k \equiv_a 0 \wedge a(b+k) = ab\big) \vee \big(\text{true} \wedge a \cdot 0 = b\big). \tag{14}$$

Note that for any fixed choice of the parameter a, the bounded quantifier can be straightforwardly replaced by a finite disjunction such that the formula is then strictly quantifier-free.

To make the picture complete, it is quite instructive to state and briefly discuss the following result:

Theorem 1 (Undecidability of bounded quantifiers) *The question, whether a given quantifier* $\bigsqcup_{k:\,\beta}$ *is a bounded quantifier, is undecidable.*

Proof Consider an arbitrary multivariate polynomial $p \in \mathbb{Z}[a_1, \ldots, a_r]$. Obviously, p corresponds to a term in our language L. We define a strictly quantifier-free formula

$$\beta := p = 0 \vee (p \neq 0 \wedge k = 0).$$

The quantifier $\bigsqcup_{k:\,\beta}$ is a bounded quantifier if and only if β is a k-bound if and only if p has no integer root. So deciding whether a given quantifier is a bounded one is at least as hard as deciding whether a given Diophantine equation $p = 0$ has a solution or not. This is essentially Hilbert's 10th problem, which is known to be undecidable [4,18]. □

At first glance this result appears a bit discouraging with respect to the framework that we are proposing here: It is undecidable, whether a given formula is linear and thus an admissible input formula. There is, however, a substantial fact, which makes clear that there is not really a problem: In our extended logic there are the same sets definable as in first-order logic. This allows the user to restrict to first-order logic, for which linearity if obviously decidable. In addition, there are many natural applications of bounded quantifiers, where their boundedness is obvious. One example is the description of intervals.

Finally, it is going to turn out that our elimination procedure delivers linear output for linear input. This has the important consequence that the procedure can be iterated after the elimination of some innermost quantifier.

2.3 Parametric elimination sets

Recall the general idea for the elimination of an existential quantifier by means of an elimination set E, which we had discussed as Eq. (5):

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma,t)\in E} \big(\gamma \wedge \varphi[t /\!\!/ x]\big). \tag{15}$$

For our generalized approach here it is important to understand that there is *not* the entire big disjunction simply replaced by a bounded quantifier. Instead an elimination set still contains finitely many guarded points, which are substituted by means of a regular disjunction. Some of these guarded points however, introduce bounded quantifiers within their particular disjunct. In our example elimination set (9) there are two guarded points, the first of which introduces one bounded quantifier in the respective weak quantifier elimination result (14). The top-level $\vee$ there actually corresponds to the big disjunction above. It should now be obvious that the notion of an elimination set has to be generalized in such a way that it encodes as well the bounded quantifiers introduced by each guarded point.

Let φ be a weakly quantifier-free formula. A *parametric pre-elimination set* for $\exists x \varphi$ is a finite set

$$E = \big\{ (\gamma_i, t_i, B_i) \;\big|\; 1 \le i \le n \big\}, \quad \text{where } B_i = \big((k_{ij}, \beta_{ij}) \;\big|\; 1 \le j \le m_i \big). \tag{16}$$

The *guards* γ_i are strictly quantifier-free formulas, the *test points* t_i are pseudo-terms possibly involving division, the k_{ij} are variables, and the β_{ij} are k_{ij}-bounds. The variables possibly occurring in γ_i and t_i are restricted to the free variables of φ together with $k_{i1}, \ldots, k_{im_i}$. The variables possibly occurring in β_{ij} are restricted to the free variables of φ together with $k_{i1}, \ldots, k_{ij-1}$.

A *parametric elimination set* E for $\exists x \varphi$ is a parametric pre-elimination set such that using the definition of $\varphi[t_i /\!\!/ x]$ from (6), E satisfies

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma_i,t_i,B_i)\in E} \bigsqcup_{k_{i1}:\beta_{i1}} \cdots \bigsqcup_{k_{im_i}:\beta_{im_i}} \big(\gamma_i \wedge \varphi[t_i /\!\!/ x]\big). \tag{17}$$

Thus a parametric pre-elimination set for $\exists x \varphi$ is a set of a particular format, while a parametric elimination set is suitable for weak quantifier elimination of $\exists x$ from $\exists x \varphi$.

Although the definition is absolutely precise, it might support understanding if we mention two additional properties, which all our parametric elimination sets are going to have: First, the variables possibly occurring in γ_i and t_i are going to be even restricted to the free variables of φ together with k_{im_i}. Second, for every interpretation of variables satisfying γ_i there are no zero denominators in t_i, and t_i is an integer.

As an example for a parametric elimination set, we formally write down the elimination set for $\exists x (ax = b)$, which we had informally given as Eq. (9):

$$E = \Big\{ \big(a \neq 0 \wedge b + k \equiv_a 0,\ \tfrac{b+k}{a},\ \big((k, -|a| \le k \le |a|)\big)\big),\ \big(\text{true}, 0, \varnothing\big) \Big\}. \tag{18}$$

Note that the elimination result discussed as (14) is exactly the result of applying E in the sense of (17).

We have already mentioned that for fixed choices of all parameters, all bounded quantifiers inside a weakly quantifier-free formula can be equivalently replaced by regular finite disjunctions. Parametric elimination sets have a very similar property: For a fixed choice of all parameters, they can be equivalently replaced by regular elimination sets in the sense of Eq. (4). Aside from justifying the term *parametric elimination set*, this is a crucial technical concept for the proof of our elimination theorem in the following subsection. The remainder of the present subsection is devoted to making precise this concept and to proving the relevant results. We start by giving a construction of the regular elimination sets mentioned above. Notice that the computation of such sets is not going to be part of the weak quantifier elimination algorithm.

Consider a parametric pre-elimination set E for $\exists x\varphi$ as in (16). Let $a_1, \ldots, a_r$ be the parameters of φ, and let $z_1, \ldots, z_r \in \mathbb{Z}$. Substitution of terms for variables on the guards γ_i, the test points t_i, and the bounds β_{ij} naturally induces a substitution on parametric pre-elimination sets. Since all integers are definable in our language L, we may simplify notation and substitute integers instead of the corresponding terms. We define

$$E_0 := E[z_1/a_1, \ldots, z_r/a_r] = \big\{(\gamma_i', t_i', B_i') \mid 1 \le i \le n\big\}, \tag{19}$$

where $B_i' = (\,(k_{ij}, \beta_{ij}') \mid 1 \le j \le m_i\,)$.

The number of bounded quantifiers introduced by the application of E_0 in the sense of (17) is exactly the sum of all the m_i. For technical reasons, we are rather interested in the maximal number $\mu(E_0)$ of bounded quantifiers introduced by a single element of E_0, and in the number $\nu(E_0)$ of elements of E_0 actually introducing that maximal number of bounded quantifiers. Formally, we define for parametric elimination sets F:

$$\mu(F) = \max_{1\le i\le n} m_i \quad \text{and} \quad \nu(F) = \big|\big\{\, i \in \{1,\ldots,n\} \;\big|\; m_i = \mu(F)\,\big\}\big|. \tag{20}$$

By these definitions we obviously have $\mu(F) \neq 0$ if and only if F actually introduces bounded quantifiers. In the positive case, we may assume without loss of generality that $m_1 = \mu(F)$. Let us consider separately that first triplet:

$$E_0 = \big\{\big(\gamma_1', t_1', \big((k_{11}, \beta_{11}'), (k_{12}, \beta_{12}'), \ldots, (k_{1m_1}, \beta_{1m_1}')\big)\big)\big\} \cup \bar{E}_0, \tag{21}$$

where $\bar{E}_0 = \{(\gamma_i', t_i', B_i') \mid 2 \le i \le n\}$. Due to our substitution $[z_1/a_1, \ldots, z_r/a_r]$, β_{11}' contains only one variable, which is k_{11}. Since by definition β_{11} is a k_{11}-bound, the solution set $S^{k_{11}}_{\beta_{11}'} = S^{k_{11}}_{\beta_{11}}(z_1, \ldots, z_r)$ is finite. We define

$$E_1 = \bigcup_{y \in S^{k_{11}}_{\beta_{11}'}} \big\{\big(\gamma_1', t_1', \big((k_{12}, \beta_{12}'), \ldots, (k_{1m_1}, \beta_{1m_1}')\big)\big)\big\}[y/k_{11}] \cup \bar{E}_0. \tag{22}$$

The step from E_0 to E_1 has possibly even increased the overall number of bounded quantifiers. It is, however, not hard to see that with respect to the lexicographic order

on $\mathbb{N}^2$ we have achieved that

$$\big(\mu(E_1), \nu(E_1)\big) < \big(\mu(E_0), \nu(E_0)\big). \tag{23}$$

Thus after finitely many iterations l of the described process, one finally obtains a uniquely determined parametric elimination set E_l with $\mu(E_l) = 0$. In E_l we have $B_i = \varnothing$ for all i. Thus E_l can be straightforwardly transformed into a regular elimination set $\Pi(E, z_1, \ldots, z_r)$ by simply dropping all these empty lists, which turns the contained triplets into pairs. We call $\Pi(E, z_1, \ldots, z_r)$ the $(z_1, \ldots, z_r)$-*projection* of E.

Lemma 2 *Let φ be a weakly quantifier-free formula with parameters $a_1, \ldots, a_r$. Let E be a parametric pre-elimination set for $\exists x\varphi$. Let $z_1, \ldots, z_r$ be an interpretation of the parameters $a_1, \ldots, a_r$. Then the following holds*:

(i) $(\gamma', t') \in \Pi(E, z_1, \ldots, z_r)$ *if and only if there exist* $(\gamma_i, t_i, B_i) \in E$ *and*

$$y_1 \in S^{k_{i1}}_{\beta_{i1}}(z_1, \ldots, z_r), \ldots, y_{m_i} \in S^{k_{im_i}}_{\beta_{im_i}}(z_1, \ldots, z_r, y_1, \ldots, y_{m_i-1})$$

such that

$$\begin{aligned} \gamma' &= \gamma_i[z_1/a_1, \ldots, z_r/a_r, y_1/k_{i1}, \ldots, y_{m_i}/k_{im_i}] \quad \textit{and} \\ t' &= t_i[z_1/a_1, \ldots, z_r/a_r, y_1/k_{i1}, \ldots, y_{m_i}/k_{im_i}]. \end{aligned}$$

(ii) *If E is even a parametric elimination set for $\exists x\varphi$, then $\Pi(E, z_1, \ldots, z_r)$ is a regular elimination set for $\exists x\varphi'$, where $\varphi' = \varphi[z_1/a_1, \ldots, z_r/a_r]$*, i.e.,

$$\exists x\varphi' \longleftrightarrow \bigvee_{(\gamma', t') \in \Pi(E, z_1, \ldots, z_r)} \big(\gamma' \wedge \varphi'[t' /\!\!/ x]\big).$$

Proof (i) Inspection of the construction of $\Pi(E, z_1, \ldots, z_r)$ above shows that each element of this projection is obtained by finitely many substitutions into some element of E. It is straightforward to verify that these substitutions are exactly those given in the lemma.

(ii) Since E is a parametric elimination set for $\exists x\varphi$, it follows that E_0 is a parametric elimination set for $\exists x\varphi'$. That is

$$\exists x\varphi' \longleftrightarrow \bigvee_{(\gamma_i', t_i', B_i') \in E_0} \bigsqcup_{k_{i1} : \beta'_{i1}} \cdots \bigsqcup_{k_{im_i} : \beta'_{im_i}} \big(\gamma_i' \wedge \varphi'[t_i' /\!\!/ x]\big).$$

Considering the construction step for obtaining E_1 from E_0 it is easy to see that

$$\bigsqcup_{k_{11}:\beta'_{11}} \bigsqcup_{k_{12}:\beta'_{12}} \cdots \bigsqcup_{k_{1m_1}:\beta'_{1m_1}} \big(\gamma'_1 \wedge \varphi'[t'_1/\!\!/x]\big)$$
$$\longleftrightarrow \bigvee_{y \in S^{k_{11}}_{\beta'_{11}}} \bigsqcup_{k_{12}:\beta'_{12}[y/k_{11}]} \cdots \bigsqcup_{k_{1m_1}:\beta'_{1m_1}[y/k_{11}]} \big(\gamma'_1 \wedge \varphi'[t'_1/\!\!/x]\big)\big[y/k_{11}\big],$$

and consequently

$$\exists x \varphi' \longleftrightarrow \bigvee_{(\gamma'_i,t'_i,B'_i)\in E_0} \bigsqcup_{k_{i1}:\beta'_{i1}} \cdots \bigsqcup_{k_{im_i}:\beta'_{im_i}} \big(\gamma'_i \wedge \varphi'[t'_i/\!\!/x]\big)$$
$$\longleftrightarrow \bigvee_{y \in S^{k_{11}}_{\beta'_{11}}} \bigsqcup_{k_{12}:\beta'_{12}[y/k_{11}]} \cdots \bigsqcup_{k_{1m_1}:\beta'_{1m_1}[y/k_{11}]} \big(\gamma'_1 \wedge \varphi'[t'_1/\!\!/x]\big)\big[y/k_{11}\big]$$
$$\vee \bigvee_{\substack{(\gamma'_i,t'_i,B'_i)\in E_0 \\ i>1}} \bigsqcup_{k_{i1}:\beta'_{i1}} \cdots \bigsqcup_{k_{im_i}:\beta'_{im_i}} \big(\gamma'_i \wedge \varphi'[t'_i/\!\!/x]\big)$$
$$\longleftrightarrow \bigvee_{(\gamma',t',((k_i,\beta'_i)|1\le i\le m))\in E_1} \bigsqcup_{k_1:\beta'_1} \cdots \bigsqcup_{k_m:\beta'_m} \big(\gamma' \wedge \varphi'[t'/\!\!/x]\big).$$

Corresponding equivalences obviously hold for all further steps of the construction such that we finally obtain the assertion for $\Pi(E, z_1, \ldots, z_r)$. □

Next, we are going to show the converse of the previous Lemma 2: A parametric pre-elimination set for $\exists x\varphi$ is actually a parametric elimination set for $\exists x\varphi$ provided that every $(z_1, \ldots, z_r)$-projection $\Pi(E, z_1, \ldots, z_r)$ is a elimination set for $\varphi' = \varphi[z_1/a_1, \ldots, z_r/a_r]$.

Lemma 3 *Let φ be a weakly quantifier-free formula with parameters $a_1, \ldots, a_r$. Let E be a parametric pre-elimination set for $\exists x\varphi$ with the following property*: *For each interpretation $z_1, \ldots, z_r \in \mathbb{Z}$ of the parameters $a_1, \ldots, a_r$, we have*

$$\exists x \varphi' \longleftrightarrow \bigvee_{(\gamma',t')\in\Pi(E,z_1,\ldots,z_r)} \big(\gamma' \wedge \varphi'[t'/\!\!/x]\big),$$

where $\varphi' = \varphi[z_1/a_1, \ldots, z_r/a_r]$. Then E is a parametric elimination set for $\exists x\varphi$.

Proof According to the definition of a parametric elimination set we have to show that

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma_i,t_i,B_i)\in E} \bigsqcup_{k_{i1}:\beta_{i1}} \cdots \bigsqcup_{k_{im_i}:\beta_{im_i}} \big(\gamma_i \wedge \varphi[t_i/\!\!/x]\big).$$

The direction from the right to the left is trivial. Let vice versa $z_1, \ldots, z_r \in \mathbb{Z}$ be an interpretation of $a_1, \ldots, a_r$ such that $\exists x \varphi$ holds. It follows immediately that also $\exists x \varphi'$ holds. By the assumption in the lemma it follows that the following application of $\Pi(E, z_1, \ldots, z_r)$ holds:

$$\bigvee_{(\gamma', t') \in \Pi(E, z_1, \ldots, z_r)} \big(\gamma' \wedge \varphi'[t' /\!\!/ x]\big).$$

That is, there is at least one $(\gamma', t') \in \Pi(E, z_1, \ldots, z_r)$ such that $\gamma' \wedge \varphi'[t' /\!\!/ x]$ holds. According to Lemma 2(i) there exist $(\gamma_i, t_i, B_i) \in E$ and $y_1, \ldots, y_{m_i}$ such that

$$\begin{aligned} \gamma' &= \gamma_i[z_1/a_1, \ldots, z_r/a_r, y_1/k_{i1}, \ldots, y_{m_i}/k_{im_i}] \quad \text{and} \\ t' &= t_i[z_1/a_1, \ldots, z_r/a_r, y_1/k_{i1}, \ldots, y_{m_i}/k_{im_i}]. \end{aligned}$$

It follows that

$$\bigsqcup_{k_{i1} : \beta'_{i1}} \cdots \bigsqcup_{k_{im_i} : \beta'_{im_i}} \big(\gamma_i[z_1/a_1, \ldots, z_r/a_r] \wedge \varphi'[t_i[z_1/a_1, \ldots, z_r/a_r] /\!\!/ x]\big),$$

where $\beta'_{ik} = \beta_{ik}[z_1/a_1, \ldots, z_r/a_r]$. With respect to our fixed interpretation of $a_1, \ldots, a_r$ by $z_1, \ldots, z_r \in \mathbb{Z}$, it follows that the following parametric formula holds:

$$\bigsqcup_{k_{i1} : \beta_{i1}} \cdots \bigsqcup_{k_{im_i} : \beta_{im_i}} \big(\gamma_i \wedge \varphi[t_i /\!\!/ x]\big).$$

It finally follows that $\bigvee_{(\gamma_i, t_i, B_i) \in E} \bigsqcup_{k_{i1} : \beta_{i1}} \cdots \bigsqcup_{k_{im_i} : \beta_{im_i}} \big(\gamma_i \wedge \varphi[t_i /\!\!/ x]\big)$ holds. □

The next lemma states that for the elimination of an existential quantifier in front of two alternative equivalent weakly quantifier-free formulas the same parametric elimination set can be used.

Lemma 4 *Let φ and $\bar{\varphi}$ be weakly quantifier-free formulas, such that $\varphi \longleftrightarrow \bar{\varphi}$. Let E be a parametric elimination set for $\exists x \varphi$. Then E is also a parametric elimination set for $\exists x \bar{\varphi}$.*

Proof Let E be of the form

$$E = \big\{(\gamma_i, t_i, B_i) \mid 1 \le i \le n\big\}, \quad \text{where} \quad B_i = \big((k_{ij}, \beta_{ij}) \mid 1 \le j \le m_i\big).$$

From the assumption $\varphi \longleftrightarrow \bar{\varphi}$ it follows that $\exists x \varphi \longleftrightarrow \exists x \bar{\varphi}$ holds as well. Consequently

$$\exists x\bar{\varphi} \longleftrightarrow \exists x\varphi \longleftrightarrow \bigvee_{(\gamma_i,t_i,B_i)\in E} \bigsqcup_{k_{i1}:\beta_{i1}} \cdots \bigsqcup_{k_{im_i}:\beta_{im_i}} \big(\gamma_i \wedge \varphi[t_i /\!\!/ x]\big)$$
$$\longleftrightarrow \bigvee_{(\gamma_i,t_i,B_i)\in E} \bigsqcup_{k_{i1}:\beta_{i1}} \cdots \bigsqcup_{k_{im_i}:\beta_{im_i}} \big(\gamma_i \wedge \bar{\varphi}[t_i /\!\!/ x]\big).$$

□

2.4 The elimination theorem

In this subsection we give a weak quantifier elimination procedure for the full linear theory of the integers.

We call a formula φ *positive* if φ does not contain the symbol $\neg$. We call a weakly quantifier-free formula φ in *prenex normal form* if φ is of the form

$$\underset{k_1:\beta_1}{Q_1} \cdots \underset{k_n:\beta_n}{Q_n}\, \psi, \tag{24}$$

where the Q_i are bounded quantifiers, the β_i are k_i-bounds, and ψ is strictly quantifier-free. The following lemma summarizes some properties of these normal forms, which we are going to use.

Lemma 5 (i) *Let φ be a weakly quantifier-free formula. Then there exists a weakly quantifier-free formula $\bar{\varphi}$ in prenex normal form such that $\bar{\varphi} \longleftrightarrow \varphi$.*

(ii) *Let φ be a strictly quantifier-free formula. Then there is a strictly quantifier-free positive formula $\bar{\varphi}$ such that $\bar{\varphi} \longleftrightarrow \varphi$.* □

We first restrict our attention to giving an elimination procedure for $\exists x$ in front of a positive prenex weakly quantifier-free formula. In the case that x occurs in some β_i the bounded quantifier should be transformed into a regular quantifier and eliminated first. From now on we may assume that x does not occur in any of the bounds β_i.

Consider a subset $S \subseteq \mathbb{Z}$. We call $z \in S$ an *interval boundary* in S if $z - 1 \notin S$ or $z + 1 \notin S$. In the former case, z is called a *lower* interval boundary. In the latter case, z is called an *upper* interval boundary.

Lemma 6 (Representation Lemma) *Let φ be a formula of the form $\bigwedge_{i\in I_1} n_i x \ \varrho_i \ r_i$, where $n_i, r_i \in \mathbb{Z}$ with $\gcd\{n_i, r_i\} = 1$ and $\varrho_i \in \{=, \neq, \leq, >, \geq, <\}$ for all $i \in I_1$. Then the set S^x_φ is a finite union of intervals. Furthermore, for each interval boundary $z \in S^x_\varphi$ we have*

$$z = \frac{r_i + k}{n_i} \in \mathbb{Z}$$

for some $i \in I_1$ and $k \in \mathbb{Z}$, where $0 \leq k \leq |n_i|$ if z is a lower boundary and $-|n_i| \leq k \leq 0$ if z is an upper boundary. Notice that for this choice of i and k we have in particular $n_i \mid r_i + k$.

Proof We prove the first part of our assertion by induction on the number s of occurrences of the relation symbol $\neq$ in φ. If $s = 0$, then according to the interpretation of Presburger formulas we have

$$S_\varphi^x = \bigcap_{i \in I_1} S_{n_i x \varrho_i r_i}^x,$$

which is an intersection of intervals and thus an interval itself. For $s > 0$, say $\varrho_{\hat{\imath}}$ equals $\neq$, we have

$$\varphi \longleftrightarrow \left(\bigwedge_{\substack{i \in I_1 \\ i \neq \hat{\imath}}} n_i x \ \varrho_i \ r_i \wedge n_{\hat{\imath}} x < r_{\hat{\imath}} \right) \vee \left(\bigwedge_{\substack{i \in I_1 \\ i \neq \hat{\imath}}} n_i x \ \varrho_i \ r_i \wedge n_{\hat{\imath}} x > r_{\hat{\imath}} \right).$$

By our induction hypothesis, the solution set of each of the two conjunctions above is a finite union of intervals, and S_φ^x is in turn their union.

For the second part, we consider the case that z is a lower boundary. The other one is analogous. By multiplication with -1 of atomic formulas in φ and adapting the relation symbols accordingly, we may without loss of generality assume that all $n_i > 0$. Since $z \in S$ and $z - 1 \notin S$, there is a corresponding atomic formula $n_{\bar{\imath}} x \ \varrho_{\bar{\imath}} \ r_{\bar{\imath}}$ with that very property. It follows from our assumptions made so far that $\varrho_{\bar{\imath}} \notin \{\leq, <\}$. If $n = 1$ and in addition $\varrho_{\bar{\imath}} \in \{=, \geq\}$, then we choose $k = 0$. Otherwise, we choose the unique $k \in \{1, \ldots, |n_{\bar{\imath}}|\}$, such that $n_{\bar{\imath}} \mid r_{\bar{\imath}} + k$. Note that $r_{\bar{\imath}}/n_{\bar{\imath}}$ is not an integer then. □

We call a subset $S \subseteq \mathbb{Z}$ an *m-periodic set* if $z \in S$ implies $z \pm m \in S$. It is not hard to see that intersection of an m_1-periodic set with an m_2-periodic is an m-periodic set, where $m = \operatorname{lcm}\{m_1, m_2\}$. This can be transferred to the solution sets of congruences and incongruences both of which are periodic sets:

Lemma 7 *Let φ be a formula of the form $\bigwedge_{i \in I_2} n_i x \ \varrho_i \ r_i$, where $n_i, r_i \in \mathbb{Z}$ and ϱ_i is a congruence or an incongruence with modulus $0 \neq m_i \in \mathbb{Z}$ for all $i \in I_2$. Then the solution set S_φ^x is an m-periodic set with $m = \operatorname{lcm}\{ m_i \mid i \in I_2 \}$.*

We now can state the central technical result of this article:

Lemma 8 (Weak elimination of one quantifier) *Consider a linear formula $\exists x \varphi$ with parameters $a_1, \ldots, a_r$, where φ is weakly quantifier-free, positive, and in prenex normal form*

$$\varphi = \underset{k_1 : \beta_1}{Q_1} \ \ldots \ \underset{k_n : \beta_n}{Q_n} \ \psi.$$

Let the set of all atomic formulas of ψ that contain x be

$$\{ n_i x \ \varrho_i \ s_i + r_i \mid i \in I_1 \dot{\cup} I_2 \}.$$

Here, the n_i and r_i are polynomials in the parameters $a_1, \ldots, a_r$. The s_i are polynomials in both the parameters $a_1, \ldots, a_r$ and the bound-variables $k_1, \ldots, k_n$. For $i \in I_1$, we have $\varrho_i \in \{=, \neq, \leq, >, \geq, <\}$. For $i \in I_2$, we have that ϱ_i is either $\equiv_{m_i}$ or $\not\equiv_{m_i}$, where m_i is a polynomial in $a_1, \ldots, a_r$.

Let $k, k_1^*, \ldots, k_n^*$ denote new variables. Define

$$\beta_1^* = \beta_1[k_1^*/k_1, \ldots, k_n^*/k_n], \quad \ldots, \quad \beta_n^* = \beta_n[k_1^*/k_1, \ldots, k_n^*/k_n].$$

Define $m = \operatorname{lcm}\{ m_i^2 + 1 \mid i \in I_2 \}$. For $i \in I_1 \cup I_2$ define

$$s_i^* = s_i[k_1^*/k_1, \ldots, k_n^*/k_n] \quad \textit{and} \quad \delta_i = -|n_i|m \leq k - s_i^* \leq |n_i|m.$$

Then $E = \{ (\gamma_i, t_i, B_i) \mid i \in I_1 \cup I_2 \} \cup \{(\text{true}, 0, \varnothing)\}$, where

$$\gamma_i = (n_i \neq 0 \wedge r_i + k \equiv_{n_i} 0), \quad t_i = \frac{r_i + k}{n_i}, \quad B_i = \big((k_1^*, \beta_1^*), \ldots, (k_n^*, \beta_n^*), (k, \delta_i)\big),$$

is a parametric elimination set for $\exists x \varphi$.

Proof Fixing an interpretation $z_1, \ldots, z_r \in \mathbb{Z}$ of the parameters $a_1, \ldots, a_r$ of φ, we set $\varphi' = \varphi[z_1/a_1, \ldots, z_r/a_r]$. In the same fashion, we denote

$$\begin{aligned} n_i' &= n_i[z_1/a_1, \ldots, z_r/a_r], m_i' = m_i[z_1/a_1, \ldots, z_r/a_r], \\ s_i' &= s_i[z_1/a_1, \ldots, z_r/a_r], s_i^{*\prime} = s_i^*[z_1/a_1, \ldots, z_r/a_r], \\ r_i' &= r_i[z_1/a_1, \ldots, z_r/a_r]. \end{aligned}$$

According to Lemma 3 it now suffices to show that

$$\exists x \varphi' \longleftrightarrow \bigvee_{(\gamma', t') \in \Pi(E, z_1, \ldots, z_r)} \big(\gamma' \wedge \varphi'[t' /\!\!/ x]\big),$$

where $\Pi(E, z_1, \ldots, z_r)$ is the $(z_1, \ldots, z_r)$-projection of E. The implication from the right hand side to the left hand side is trivial.

For the other direction, assume that $\exists x \varphi'$ holds. We are now going to take three steps to transform φ' into an equivalent formula φ'', which is syntactically much more convenient for our purposes here:

1. We equivalently replace all atomic formulas of the form $n_i' x \equiv_0 s_i' + t_i'$ with corresponding equations $n_i' x = s_i' + t_i'$. In the same way, we equivalently replace $n_i' x \not\equiv_0 s_i' + t_i'$ with $n_i' x \neq s_i' + t_i'$. Let $J \subseteq I_2$ be the set of all indices affected by this step.
2. Since all the parameters have been substituted with integers, we can expand all bounded quantifiers into finite disjunctions or conjunctions. After this, all atomic formulas are of the form $n_i' x \; \varrho_i' \; s_{i,\mathbf{y}}'' + r_i'$, where

$$s_{i,\mathbf{y}}'' = s_{i,y_1,\ldots,y_n}'' = s_i'[y_1/k_1, \ldots, y_n/k_n] \quad \text{with} \quad \mathbf{y} = (y_1, \ldots, y_n) \in \mathbb{Z}^n.$$

More precisely, for $j \in \{1, \ldots, n\}$ we have

$$y_j \in S^{k_i}_{\beta_i}(z_1, \ldots, z_r, y_1, \ldots, y_{j-1}) = S^{k^*_i}_{\beta^*_i}(z_1, \ldots, z_r, y_1, \ldots, y_{j-1}).$$

Defining ${s^*_{i,y_1,\ldots,y_n}}'' = {s^*_i}'[y_1/k^*_1, \ldots, y_n/k^*_n]$ we obviously obtain $s''_{i,\mathbf{y}} = {s^*_{i,\mathbf{y}}}''$.

3. We finally obtain a disjunctive normal form by finitely many applications of the laws of distributivity. This does not change the set of contained atomic formulas.

If $\varphi'' \longleftrightarrow$ true, then also $\varphi' \longleftrightarrow$ true, and $(\text{true}, 0) \in \Pi(E, z_1, \ldots, z_r)$ is a suitable test point. We may from now on assume that not $\varphi'' \longleftrightarrow$ true.

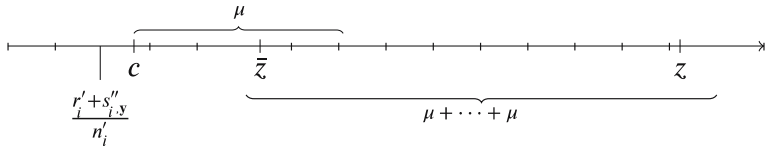
We had assumed that $\exists x \varphi'$ holds. It follows that the equivalent formula $\exists x \varphi''$ holds as well. From this it follows in turn that $\exists x \omega$ holds for at least one conjunction ω in the disjunctive normal form φ''. We may assume without loss of generality that all atomic formulas in ω actually contain x. We can split ω into the conjunction ω_2 containing all the congruences and incongruences and the conjunction ω_1 containing all other atomic formulas. Formally, $\omega = \omega_1 \wedge \omega_2$, where

$$\omega_1 = \bigwedge_{i \in J_1} \bigwedge_{\mathbf{y} \in Y_i} n'_i x \; \varrho'_i \; s''_{i,\mathbf{y}} + r'_i \quad \text{and} \quad \omega_2 = \bigwedge_{i \in J_2} \bigwedge_{\mathbf{y} \in Y_i} n'_i x \; \varrho'_i \; s''_{i,\mathbf{y}} + r'_i$$

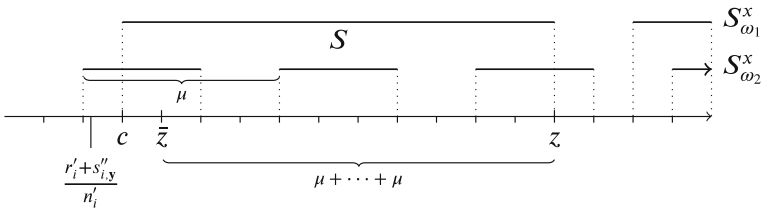
for suitable $J_1 \subseteq I_1 \cup J$ and $J_2 \subseteq I_2 \backslash J$. We assume without loss of generality that $n'_i > 0$ and $\gcd\{n'_i, s''_{i,\mathbf{y}} + r'_i\} = 1$ for all $i \in J_1$.

Consider the solution sets $S^x_{\omega_1}$ and $S^x_{\omega_2}$. On the assumptions made so far, it is clear that $S^x_{\omega_1} \neq \mathbb{Z}$ or $S^x_{\omega_2} \neq \mathbb{Z}$. It is helpful to distinguish cases on the form of $S^x_{\omega_1}$ for a moment. Recall that $S^x_{\omega_2}$ is periodic by Lemma 7. Denoting by μ its minimal positive period, we have $1 \leq \mu \leq m' = m[z_1/a_1, \ldots, z_r/a_r]$:

(a) If $S^x_{\omega_1} = \mathbb{Z}$, then $S^x_{\omega_2} \neq \mathbb{Z}$ and thus $J_2 \neq \varnothing$. Choose an arbitrary $i \in J_2$ and $\mathbf{y} \in Y_i$. Then there exists a $\Delta \in \mathbb{Z}$ with $0 \leq \Delta < n'_i$ with $n'_i \mid r'_i + s''_{i,\mathbf{y}} + \Delta$. Set $c = (r'_i + s''_{i,\mathbf{y}} + \Delta)/n'_i$. Due to the periodicity of $S^x_{\omega_2}$ there exists some $\bar{z} \in S^x_{\omega_2} = S^x_{\omega}$ with $c \leq \bar{z} < c + \mu \leq c + m'$; in other words, $0 \leq \bar{z} - c \leq m' - 1$.



(b) If $S^x_{\omega_1} \neq \mathbb{Z}$, then still $z \in S^x_{\omega_1}$. According to Lemma 6, z lies an interval S that is maximal with the property $z \in S \subseteq S^x_{\omega_1}$. Without loss of generality this interval is bounded from below. Let $c \in S$ with $c - 1 \notin S$ be a corresponding interval boundary. Then Lemma 6 furthermore guarantees that there is $i \in J_1$, $\mathbf{y} \in Y_i$, and $\Delta \in \mathbb{Z}$ with $0 \leq \Delta \leq n'_i$ such that $c = (r'_i + s''_{i,\mathbf{y}} + \Delta)/n'_i$. Due to periodicity there is some $\bar{z} \in S^x_{\omega_2}$ with $\bar{z} \leq z$ and $c \leq \bar{z} < c + \mu \leq c + m'$; in other words, $0 \leq \bar{z} - c \leq m' - 1$. It follows that $c \leq \bar{z} \leq z$, thus $\bar{z} \in S \subseteq S^x_{\omega_1}$, and furthermore $\bar{z} \in S^x_{\omega_1} \cap S^x_{\omega_2} = S^x_{\omega}$.



We proceed using the values for i, $\mathbf{y}$, Δ, c, and $\bar{z}$ from case (a) or (b), respectively. Set $y = \Delta + n'_i(\bar{z} - c) + s''_{i,\mathbf{y}} \in \mathbb{Z}$. Then y satisfies the bound δ_i since

$$0 \leq y - {s^*_{i,\mathbf{y}}}'' = y - s''_{i,\mathbf{y}} = \Delta + n'_i(\bar{z} - c) \leq n'_i + n'_i(m' - 1) = n'_i m'.$$

By Lemma 2(i) there exists exactly one guarded point $(\gamma', t') \in \Pi(E, z_1, \ldots, z_r)$ originating from $(\gamma_i, t_i, B_i) \in E$ for our choice of $\mathbf{y}$ and y. We can write γ' and t' as follows:

$$\gamma' = (n'_i \neq 0 \wedge r'_i + y \equiv_{n'_i} 0) \quad \text{and} \quad t' = \frac{r'_i + y}{n'_i}.$$

For the term $r'_i + y$ we obtain the identity

$$r'_i + y = r'_i + \Delta + n'_i(\bar{z} - c) + s''_{i,\mathbf{y}} = (r'_i + s''_{i,\mathbf{y}} + \Delta) + n'_i(\bar{z} - c).$$

This shows that $n'_i \mid r'_i + y$. Together with our assumption that $n'_i > 0$ it follows that γ' holds. For t' it follows that

$$t' = \frac{r'_i + y}{n'_i} = \frac{r'_i + s''_{i,\mathbf{y}} + \Delta}{n'_i} + \frac{n'_i(\bar{z} - c)}{n'_i} = c + (\bar{z} - c) = \bar{z}.$$

Consequently $\omega[t'/\!\!/x]$ is equivalent to $\omega[\bar{z}/x]$, which in turn holds due to $\bar{z} \in S^x_\omega$. It further follows that $\varphi''[t'/\!\!/x]$ holds, and in turn $\varphi'[t'/\!\!/x]$ holds as well. This proves that the entire disjunction

$$\bigvee_{(\gamma', t') \in \Pi(E, z_1, \ldots, z_r)} \left(\gamma' \wedge \varphi'[t'/\!\!/x]\right)$$

holds, which is what had to be shown. □

Theorem 9 (Uniform weak elimination theorem) *The full linear theory of the integers admits weak quantifier elimination.*

Proof Let $\hat{\varphi}$ be a linear formula. We proceed by induction on the number n of regular quantifiers in $\hat{\varphi}$. If $n = 0$, then $\hat{\varphi}$ is already weakly quantifier-free. So there is nothing to do. Consider now the case $n > 0$. There is then a subformula of $\hat{\varphi}$ of one of the forms $\exists x\varphi$ or $\forall x\varphi$, where φ is weakly quantifier-free. The latter case can be reduced to the

former one by means of the equivalence $\forall x\varphi \longleftrightarrow \neg\exists x\neg\varphi$. According to Lemma 5 we may assume that φ is in prenex normal form and positive. By Lemma 8, there exists a parametric elimination set E for $\exists x\varphi$. That is, $\exists x\varphi$ is equivalent to

$$\varphi^* = \bigvee_{(\gamma_i,t_i,B_i)\in E} \bigsqcup_{k_{i1}:\beta_{i1}} \dots \bigsqcup_{k_{im_i}:\beta_{im_i}} \big(\gamma_i \wedge \varphi[t_i /\!\!/ x]\big), \quad B_i = \big((k_{ij},\beta_{ij}) \mid 1 \le j \le m_i\big).$$

We obtain $\hat{\varphi}^*$ from $\hat{\varphi}$ by equivalently replacing $\exists x\varphi$ with φ^*. Inspection of the definition of E in Lemma 8 shows that $\hat{\varphi}^*$ is again linear. Hence we can eliminate the remaining quantifiers from $\hat{\varphi}^*$ by our induction hypothesis. □

Note that due to our linearity conditions, sentences in our framework are exactly Presburger sentences. So our generalized quantifier elimination procedure does not amount to a generalized decision procedure. It might be noteworthy, however, that according to our theorem, Presburger arithmetic remains decidable when admitting bounded quantifiers in the input formulas.

3 Extended weak quantifier elimination

Let us return to our introductory example for conventional elimination sets from Sect. 2.1: For the conventional Presburger formula $\exists x(5x = b)$, there is a regular elimination set

$$E = \left\{\left(b + k \equiv_5 0,\ \tfrac{b+k}{5}\right) \;\middle|\; -5 \le k \le 5\right\}. \tag{25}$$

The application of this regular elimination set leads to the quantifier-free equivalent

$$\bigvee_{-5\le k\le 5} \big(b + k \equiv_5 0 \wedge 5(b + k) = 5b\big), \tag{26}$$

which is in turn obviously equivalent to $b \equiv_5 0$. It is easy to derive from the original quantified formula that for any choice of b that satisfies our quantifier-free equivalent, $b/5$ is a suitable choice for the existing x. It is well-known that with virtual substitution methods over other domains such information can be systematically derived during the elimination process [7,20,30]. A corresponding procedure has been originally introduced for the reals by Weispfenning, who referred to the problem as the *extended linear elimination problem* [31]. Instead of forming a disjunction, we write down the following data:

$$\left[\begin{array}{l|l} b + (-5) \equiv_5 0 \wedge 5(b + (-5)) = 5b & \left\{x = \frac{b+(-5)}{5}\right\} \\ \vdots & \vdots \\ b + 0 \equiv_5 0 \wedge 5(b + 0) = 5b & \left\{x = \frac{b+0}{5}\right\} \\ \vdots & \vdots \\ b + 5 \equiv_5 0 \wedge 5(b + 5) = 5b & \left\{x = \frac{b+5}{5}\right\} \end{array}\right]. \tag{27}$$

For any choice of b, the left hand side formula of exactly the row displayed in the middle, which origins from $k = 0$, becomes true. The corresponding right hand side assignment is our above-mentioned suitable choice for x.

Generally, given a regular elimination set as in (4) the corresponding substitution scheme (5) can be replaced by the following *extended quantifier elimination scheme*:

$$\exists x \varphi \rightsquigarrow \left[\begin{array}{c|c} \gamma_1 \wedge \varphi[t_1 /\!\!/ x] & \{x = t_1\} \\ \vdots & \vdots \\ \gamma_n \wedge \varphi[t_n /\!\!/ x] & \{x = t_n\} \end{array}\right]. \tag{28}$$

We have to clarify the relation between the left hand side and the right hand side in (28). For obvious syntactic reasons there cannot hold equivalence. Nevertheless, there are absolutely precise semantics for extended quantifier elimination:

Lemma 10 (Semantics of extended existential quantifier elimination) *Fix arbitrary values for all parameters. The quantified input formula holds if and only if at least one of the quantifier-free left hand side formulas in the extended quantifier elimination result holds. In the positive case, for each left hand side formula that holds, the corresponding right hand side assignment describes one possible choice for the existentially quantified variable.*

Extended quantifier elimination straightforwardly generalizes to an entire block of existential quantifiers. Such as disjunctions of disjunctions can be equivalently replaced by one disjunction inside regular elimination results, there is also no need for any nested structures with extended quantifier elimination. On the right hand side one then obtains sets of several assignments, one for each quantified variable.

For one universal quantifier, we obtain dual semantics:

Lemma 11 (Semantics of extended universal quantifier elimination) *Fix arbitrary values for all parameters. The quantified input formula holds if and only if all of the quantifier-free left hand side formulas in the extended quantifier elimination result hold. In the negative case, for each left hand side formula that does not hold, the corresponding right hand side assignment describes one possible bad choice for the universally quantified variable.* □

Again, this straightforwardly generalizes to blocks of universal quantifiers. Finally, if there are several alternating quantifier blocks, extended quantifier elimination is defined by first applying regular quantifier elimination for all inner blocks. After this there is extended quantifier elimination applied to the outermost block.

3.1 Extended pure weak quantifier elimination

We restrict for a moment to *pure* weak quantifier elimination, which means we do not admit any simplifications during or after the elimination process. Let us switch to the slightly more general example $\exists x(ax = b)$. For this we had determined the parametric elimination set

$$E = \left\{\left(a \neq 0 \wedge b + k \equiv_a 0, \tfrac{b+k}{a}, \left((k, -|a| \leq k \leq |a|)\right)\right), \left(\text{true}, 0, \varnothing\right)\right\}. \tag{29}$$

To start with, we copy the idea of our extended quantifier elimination scheme (28). That is, we produce one row for each formally substituted term, where an assignment of the respective term appears guarded by the subformula that it would produce with non-extended quantifier elimination:

$$\left[\begin{array}{l|l} \bigsqcup_{k:\,-|a|\le k\le|a|} \big(a\neq 0\wedge b+k\equiv_a 0\wedge a(b+k)=ab\big) & \left\{x=\frac{b+k}{a}\right\} \\ \text{true}\wedge a\cdot 0=b & \{x=0\} \end{array}\right]. \tag{30}$$

It is going to turn out that for any interpretation of a, b with $a\neq 0$ and $b\equiv_a 0$ the left hand side formula of the first row becomes true. In order to make use of the corresponding assignment on the right hand side we do, however, have to take a closer look at the bounded quantifier. By successively plugging in the at most $2|a|+1$ possible choices for k, we can find out that the formula holds due to the possible choice $k=0$ provided that a and b are chosen as mentioned above. This possible choice for k yields the possible choice for x, which is b/a. In general, there can be several possible choices for bound-variables, each of which leads to a possible choice for x. Recall that generally the assignments do not describe solution spaces but yield only sample solutions. From that point of view it is a reasonable strategy to gain some efficiency by restricting to finding only one possible choice for the bound-variables.

For $a=b=0$ we would use the second row exactly as described for regular elimination sets above. For all other choices of a and b the original quantified formula does not hold.

With pure weak quantifier elimination the procedure described above by means of an example is generally applicable. There is, however, one subtle point about the syntactical form of our extended quantifier elimination result (30). Within the left hand side formulas, the bound-variables do not occur freely. For instance, renaming k to l in the formula in the first row would not at all change its semantics. In the corresponding assignments, in contrast, the bound-variables apparently do occur freely. This observation is in fact evidence for a much deeper problem, which becomes apparent when simplification gets involved.

3.2 Extended weak quantifier elimination in general

When applying our methods described here in practice, it is crucial for the success to use powerful simplification techniques for quantifier-free intermediate results. Our implementation comprises a variant of the *standard simplifier* described for the reals by the second author and others [6,16].

For examining the possible effects of simplification on extended weak quantifier elimination let us return once more to our simpler example $\exists x(5x=b)$, which is even a regular Presburger formula. Using parametric elimination sets, the extended quantifier result is the following instance of (30):

$$\left[\begin{array}{l|l} \bigsqcup_{k:\,-5\le k\le 5} \big(5\neq 0\wedge b+k\equiv_5 0\wedge 5(b+k)=5b\big) & \left\{x=\frac{b+k}{5}\right\} \\ \text{true}\wedge 5\cdot 0=b & \{x=0\} \end{array}\right]. \tag{31}$$

Automatic simplification replaces the formulas in the first and in the second row by $b \equiv_5 0$ and $b = 0$ respectively. Since the latter implies the former, we may entirely drop the second row. This way, (31) simplifies as follows:

$$\left[\, b \equiv_5 0 \,\middle|\, \left\{x = \tfrac{b+k}{5}\right\} \right]. \tag{32}$$

At that point the bound-variable k occurs in the sample point but no longer in the guarding formula. It is really tempting to believe that in such situations the choice of k does not matter at all. Our example, however, demonstrates that this belief is wrong. In fact, we have lost the information contained in (31) that we have to consider for k the finite range $\{-5, \ldots, 5\}$. So we have to learn that here the classical scheme for extended quantifier elimination does not work in combination with simplification, which is in turn inevitable.

Notice that our counter-example is even a regular Presburger formula. For such formulas, the problem can obviously be resolved by using in the elimination set instead of bounded quantifiers a fixed number of corresponding regular test points. This, however, increases both time complexity and the size of the output by one exponential step from doubly exponential to triply exponential in the worst case.

Our general solution is to modify the extended quantifier elimination scheme (28) such that we additionally store the bound-conditions contained in the parametric elimination sets. With respect to a parametric elimination set

$$E = \big\{(\gamma_1, t_1, B_1), \ldots, (\gamma_n, t_n, B_n)\big\}, \quad \text{where} \quad B_i = \big((k_{i1}, \beta_{i1}), \ldots, (k_{im_i}, \beta_{im_i})\big),$$

this leads to the following *extended parametric quantifier elimination scheme*:

$$\exists x \varphi \rightsquigarrow \left[\begin{array}{c|c|c} \bigsqcup_{k_{11}:\beta_{11}} \cdots \bigsqcup_{k_{1m_1}:\beta_{1m_1}} \big(\gamma_1 \wedge \varphi[t_1 /\!\!/ x]\big) & \{\beta_{11}, \ldots, \beta_{1m_1}\} & \{x = t_1\} \\ \vdots & \vdots & \vdots \\ \bigsqcup_{k_{n1}:\beta_{n1}} \cdots \bigsqcup_{k_{nm_n}:\beta_{nm_n}} \big(\gamma_n \wedge \varphi[t_n /\!\!/ x]\big) & \{\beta_{n1}, \ldots, \beta_{nm_n}\} & \{x = t_n\} \end{array}\right]. \tag{33}$$

We make precise the semantics for existentially quantified formulas. The semantics for universal formulas can be derived in analogy to the non-parametric case in Lemma 11.

Lemma 12 (Semantics of extended weak quantifier elimination) *Fix arbitrary values for all parameters. The existentially quantified input formula holds if and only if at least one of the quantifier-free left hand side formulas in the extended parametric quantifier elimination result holds. In the positive case, for each left hand side formula that holds, the corresponding conditions in the center column describe a finite range for the bound-variables $k_{i1}, \ldots, k_{im_i}$. Within this range there is at least one choice for the bound-variables such that the corresponding right hand side assignment describes* one *possible choice for the existentially quantified variable.* □

Given a quantified formula $\exists x \varphi$, for fixed parameters good choices for $k_{i1}, \ldots, k_{im_i}$ can always be determined: One successively plugs into φ the values of the terms t_i for

the finitely many possible choices for $k_{i1}, \ldots, k_{im_i}$ within the finite range described by $\beta_{i1}, \ldots, \beta_{im_i}$.

In general, simplification will not remove all the bounded quantifiers from the left hand side formulas. Whenever there are any remaining bounded quantifiers in some left hand side formula, the test whether this formula holds for the fixed parameters yields in the positive case suitable choices for the corresponding bound-variables. These choices are, of course, also good ones for the corresponding right hand side assignment. This considerably reduces the search space.

Similarly to regular extended quantifier elimination, extended weak quantifier elimination generalizes to the entire outermost quantifier block of a prenex input formula after non-extended weak quantifier elimination of all other quantifiers.

We conclude this section by returning to our example $\exists x(5x = b)$ once more. Extended weak quantifier elimination yields

$$\left[b \equiv_5 0 \,\middle|\, \{-5 \leq k \leq 5\} \,\middle|\, \left\{ x = \tfrac{b+k}{5} \right\} \right]. \tag{34}$$

For the choice $b = 15$, we plug into $5x = 15$ the terms $(15 - 5)/5$, $(15 - 4)/5, \ldots$ until we find the sample solution $(15 - 0)/5$, which happens to be the only one in this special case.

4 Implementation

The procedures described in this paper have been implemented in REDLOG,[1] which stands for REDUCE *logic system* [5,8]. It provides an extension of the computer algebra system REDUCE to a computer logic system implementing symbolic algorithms on first-order formulas with respect to temporarily fixed first-order languages and theories. Such a choice of language and theory is called a *domain* or, alternatively, *context*.

Before turning to the integer context relevant for our work here, we briefly summarize the other existing domains together with short names and alternative names, which are supported for backward compatibility:

BOOLEAN, B, IBALP The class of Boolean algebras with two elements. These algebras are uniquely determined up to isomorphisms. BOOLEAN comprises quantified propositional calculus [20].

COMPLEX, C, ACFSF The class of algebraically closed fields such as the complex numbers over the language of rings.

DIFFERENTIAL, DCFSF A domain for computing over differentially closed fields. There is no natural example for such a field, but the methods can well be used for obtaining relevant and interpretable results also for reasonable differential fields [9].

PADICS, DVFSF One prominent example for discretely valued fields are p-adic numbers for some prime p with abstract divisibility relations encoding order between values. All PADICS algorithms are optionally uniform in p [24].

QUEUES, QQE A (two-sided) queue is a finite sequence of elements of some basic type. There are two sorts of variables, one for the basic type and one for the queue type.

[1] www.redlog.eu

Accordingly, there is first-order quantification possible for both sorts. So far, the implementation is restricted to the reals as basic type [23].

REALS, R, OFSF The class of real closed fields such as the real numbers with ordering. This context was the original motivation for REDLOG. It is still the most important and most comprehensive one [10].

TERMS, TALP Free Malcev-type term algebras. The available function symbols and their arity can be freely chosen. [25].

The work discussed here establishes another such context:

INTEGERS, Z, PASF The full linear theory of the integers.

The idea of REDLOG is to combine methods from computer algebra with first-order logic thus extending the computer algebra system REDUCE to a computer logic system. In this extended system both the algebraic side and the logic side greatly benefit from each other in numerous ways. The current version REDLOG 3.0 is an integral part of the computer algebra system REDUCE 3.8. The implementation of our methods described here is part of the current development version of REDLOG. It is going to be distributed with REDUCE 3.9. Until then it is freely available on the REDLOG homepage. To our knowledge our implementation is the only existing implementation of quantifier elimination for the integers with multiplicative parameters.

Besides the core elimination methods described in the present paper considerable further theoretical research has entered the implementation. This comprises at the first place sophisticated simplification methods for quantifier-free formulas over the integers in the style of previous work of the second author and others on *smart simplification* for the reals [6]. The resulting simplification theory for the integers has so far been described only in the diploma thesis of the first author [16].

The remainder of this section is devoted to a very short *primer* on REDLOG and the INTEGERS domain. This is by no means a full documentation. The system provides much more functionally for handling and manipulating first-order functions than can be described here. Instead, the idea is to make the present paper self-contained in the sense that it provides sufficient information such that readers can reproduce the results discussed in the following section on applications.

After starting REDUCE, the following commands load the REDLOG package and switch to INTEGERS:

```
▷ load_package redlog;
▷ rlset integers;
```

Always use either `;` or `$` to terminate commands. The latter suppresses the output. Parentheses may be omitted when functions have only one argument. Turn on the switch `time` for displaying the computation times after each command:

```
▷ on time;
```

These times are CPU times, which include the time for printing the results of the commands. So for longer results it makes a difference for the time whether they are terminated by `;` or `$`. A formula like for instance

$$\forall x \exists y (y \equiv_{a^2} x \wedge x < y \wedge y \leq ax + b + 1)$$

can be input and assigned to a variable as follows:

```
▷ phi := all(x,ex(y,cong(y,x,a^2) and x<y<=a*x+b+1));
```

REDLOG automatically transforms all such inputs into some more canonical representation. This transformation includes subtracting all left hand sides to the corresponding right hand sides and then canonically ordering the right hand side polynomials. Use `part` to select the subformula starting with the existential quantifier:

```
▷ phi0 := part(phi,2);
```

Weak quantifier elimination can be applied as follows:

```
▷ sol := rlwqe phi0;
```

REDLOG functions are generally prefixed with `rl`. The size of the solution is determined as follows:

```
▷ rlatnum sol;
```

The name of this function is derived from *atomic formula number*. After plugging in a value for the multiplicative parameter a, the bounded quantifier can be expanded:

```
▷ rlexpand sub(a=5,sol);
```

When there are no multiplicative parameters, weak quantifier elimination and successive expansion can be performed in one step:

```
▷ rlqe sub(a=5,phi0);
```

In this case, which is the pure Presburger case, we have in fact regular quantifier elimination. Hence the name `rlqe`. *Extended* weak quantifier elimination is applied as follows:

```
▷ sola := rlwqea phi0;
```

The `a` in `rlwqea` stands for *answer*. Extended quantifier elimination is generally also referred to as *quantifier elimination with answer*. Here is how to expand an answer after substitution for the multiplicative parameter `a`:

```
▷ rlexpanda sub(a=5,sola);
```

Finally terminate your REDUCE session:

```
▷ quit;
```

5 Benchmarks and application examples

In this section we collect some practical computations with our implementation. On the one hand, this gives a good impression of the *current* practical applicability of the methods described here. On the other hand, we want to give an idea of the possible application range. The REDUCE input for all these examples is available online in the REMIS database on the REDLOG homepage. REMIS is the REDLOG *example management and information system*. All computations have been performed on a 2 GHz Pentium 4 using 128 MB of RAM. All times given are CPU times.

5.1 Planar transportation problems

Multidimensional planar transportation problems have been a challenging benchmark for real quantifier elimination for many years [17,26]. These are integer programming problems, which are tractable by real methods because the corresponding matrix is totally unimodular. Since real methods are certainly more efficient, our methods developed here are not really a reasonable tool for solving planar transportation problems in practice. Anyway, they provide excellent benchmark examples, which even allow to directly compare the performance of our weak quantifier elimination with linear real quantifier elimination.

We adopt the formulations and notations used by Loos and Weispfenning [17]. All formulas are going to be regular Presburger formulas. In dimension 1, the input formulas are of the form

$$T_{1,m} := \exists x_1 \ldots \exists x_m \left(\sum_{i=1}^{m} x_i = a \wedge \bigwedge_{i=1}^{m} x_i \geq 0 \right).$$

For explaining the entries in the tables below, let us discuss $T_{1,2}$ in detail. The result of our weak quantifier elimination procedure (`rlwqe`) is the following:

$$\bigsqcup_{k_1 : k_1+1 \geq 0 \wedge k_1 \leq 0} a + k_1 \geq 0 \vee \bigsqcup_{k_1 : k_1-1 \leq 0 \wedge k_1 \geq 0} a - k_1 \geq 0.$$

The *size* of this result is 6, which is the number of atomic formulas counting the ones in the bound-conditions as well. Since we are dealing with regular Presburger formulas, the bound-conditions do not contain any parameters. They can thus be expanded to disjunctions (`rlexpand`). The result of this expansion is $a \geq 0$, which has size 1. In general, one has to expect the size to grow with expansion. The total time is simply the sum of the two other times. For comparison, the current standard real quantifier

m	Time (`rlwqe`) (s)	Size	Time (`rlexpand`) (s)	Result	Time (total) (s)
5	0.06	192	0.02	$a \geq 0$	0.08
6	0.14	480	0.06	$a \geq 0$	0.20
7	0.34	1,152	0.24	$a \geq 0$	0.58
8	0.80	2,688	0.74	$a \geq 0$	1.54
9	1.89	6,144	2.39	$a \geq 0$	4.28
10	4.43	13,824	10.05	$a \geq 0$	14.48
11	11.10	30,720	44.43	$a \geq 0$	55.53

elimination procedure of REDLOG computes the instances $T_{1,100}$ and $T_{1,500}$ in 0.16 s and 19.76 s, respectively. The result there is also $a \geq 0$ in all cases, which is obviously optimal with respect to redundancy.

In dimension 2, we have the following family of input formulas, where the number of quantifiers for the instance $T_{2,m}$ is m^2:

$$T_{2,m} := \exists x_{11} \ldots \exists x_{mm} \left(\bigwedge_{i=1}^{m} \sum_{j=1}^{m} x_{ij} = a_i \wedge \bigwedge_{j=1}^{m} \sum_{i=1}^{m} x_{ij} = b_j \wedge \bigwedge_{i=1}^{m} \bigwedge_{j=1}^{m} x_{ij} \geq 0 \right).$$

In contrast to the situation with dimension 1, our final results here are mostly so large that it is not reasonable to print them. We instead give the sizes. For $m = 1$ there do not any bounded quantifiers come into existence. For comparison, real quantifier

m	Time (`rlwqe`) (s)	Size	Time (`rlexpand`) (s)	Size	Time (total) (s)
1	< 0.01	2	–	2	< 0.01
2	0.01	24	< 0.01	32	< 0.02
3	0.94	5,032	39.62	32,766	40.56

elimination in REDLOG applied to $T_{2,3}$, $T_{2,7}$, and $T_{2,8}$ yields 36 (in 0.01 s), 12,936 (in 8.85 s), and 54,912 (in 75.11 s) atomic formulas, respectively. In fact, each $T_{2,m}$ is known to be equivalent to

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{m} b_j \wedge \bigwedge_{i=1}^{m} a_i \geq 0 \wedge \bigwedge_{j=1}^{m} b_j \geq 0,$$

which contains $2m + 1$ atomic formulas. For $m = 1$ we can obtain in our table size 2 instead of 3, because the condition $b_1 \geq 0$ follows from $a_1 = b_1 \wedge a_1 \geq 0$.

5.2 Dependency analysis for automatic parallelization

For executing the following nested `for`-loops in parallel it is crucial that there is no data dependency. Generally, a data dependency can occur if there is some array cell accessed twice where at least one of the accesses is a writing access [15]. In our piece of code a critical data dependency occurs if there is some array cell written twice and with different values:

```
for i := 0 to m do
   for j := 0 to m do
      A[n*i+j] := i+j
   od
od.
```

In this example the dependency condition involves parameters m and n. Consequently, it is *not* a regular Presburger formula. It can, however, well be expressed

within our framework. Here is our formulation:

$$\exists i \exists j \exists i' \exists j' \big(0 \le i \le m \wedge 0 \le j \le m \wedge 0 \le i' \le m \wedge 0 \le j' \le m \wedge (i \ne i' \vee j \ne j') \wedge i + j \ne i' + j' \wedge ni + j = ni' + j' \big).$$

To start with, note that real quantifier elimination cannot provide any useful information here: It yields 24 atomic formulas after 0.01 s. The result is equivalent to $n \ne 1$ on the reasonable assumption that both m and n are greater than 0.

Let us now turn to weak quantifier elimination over the integers: `rlwqe` yields after 3.53 s a quantifier-free equivalent containing 25441 atomic formulas. This is a uniform result for all possible m and n, which occur there as parameters. It can, however, not be expanded, because these parameters occur in particular within bound-conditions. After plugging in $m = n = 4$ expansion becomes possible. It yields "true" after 1.4 s, which means there is a dependency.

A bit surprising, it is much more efficient in this example to first plug in values for the parameters m and n, and then to perform quantifier elimination plus expansion. This way we obtain "true" for $m = n = 4$ in only 0.28 s, and we obtain "false" for $m = 4$ and $n = 5$ in 0.29 s.

For understanding why the two approaches are so much out of proportion at present consider Lemma 8. In the non-parametric case we can and in fact do replace the definition of m by $m = \mathrm{lcm}\{ m_i \mid i \in I_2 \}$ because $m_i > 0$ for all i. That problem can be overcome in future versions by adding to the language a function symbol for the absolute value. In addition, for the definition of E one does not have to consider I_2 unless $I_1 = \varnothing$.

5.3 Information flow control

Program dependency graphs in combination with constraint solving are a recent approach to information flow control. The aim is to automatically check source code for security problems such as external manipulation of critical computations. Existing software tools automatically generate *path conditions* from source code. These are first-order sentences, which are necessary conditions for information flow on the path between two program statements. This works, e.g, for C programs up to 10000 LOC [21,22].

Consider for instance the following piece of program code, where all variables are of type integer:

```
if (a < b) then
   if (a+b mod 2 = 0) then
      n := (a+b)/2
   else
      n := (a+b+1)/2
   fi
   A[n] := get_sensitive_data(x)
   send_sensitive_data(trusted_receiver,A[n])
fi
y := A[abs(b-a)].
```

We say that there is an information flow within this piece of code, if the value of some variable set in there is transferred to another one. To be concrete: Can `y` get the value of `A[n]`? If so, `A[n]`, which is supposed to be sensitive data, could become accessible outside our piece of code.

The following sentence is a straightforward first-order formulation of a corresponding path condition, which could be automatically generated:

$$\begin{aligned}\exists a \exists b \exists n \big(& (a < b \wedge a + b \equiv_2 0 \wedge 2n = a + b \wedge \\ & ((a < b \wedge b - a = n) \vee (a \geq b \wedge a - b = n))) \vee \\ & (a < b \wedge a + b \not\equiv_2 0 \wedge 2n = a + b + 1 \wedge \\ & ((a < b \wedge b - a = n) \vee (a \geq b \wedge a - b = n)))\big).\end{aligned}$$

If this sentence is equivalent to "false," then there is definitely no information flow. If it is in contrast equivalent to "true," then information flow might take place. Our quantifier elimination delivers "true" in less than 0.01 s.

Extended quantifier elimination can do even more. Besides the result "true" it delivers in less than 0.01 s possible values for the external variables a and b such that information flow might take place:

$$\left[\begin{array}{l|l|l} \text{true} & \{-2 \leq k_1 \leq 2\} & \left\{n = k_1, b = \frac{3k_1 - 1}{2}, a = \frac{k_1 - 1}{2}\right\} \\ \text{true} & \{-2 \leq k_2 \leq 2\} & \left\{n = k_2, b = \frac{3k_2}{2}, a = \frac{k_2}{2}\right\} \end{array}\right].$$

One easily verifies that $k_1 = 1$ and $k_2 = 2$ deliver suitable values.

While the sample values obtained by extended quantifier elimination are useful for an attacker, the programmer would rather be interested in an exact description of all values for a and b possibly causing problems. This can be obtained in less than 0.01 s by dropping the quantifiers $\exists a \exists b$ from our path condition thus only eliminating $\exists n$:

$$(3a - b + 1 = 0 \wedge a + b \equiv_2 0 \wedge a < b) \vee (3a - b = 0 \wedge a + b \not\equiv_2 0 \wedge a < b).$$

5.4 Feasibility of parametric constraints

As another example for extended weak quantifier elimination we consider the following question on the feasibility of a system of two parametric constraints:

$$\exists x (ax \geq b \wedge cx \leq d).$$

Extended weak quantifier elimination yields a result after less than 0.01 s. Using the absolute value within bound-conditions in order to keep our notation here short this

result looks as follows:

$$\left[\begin{array}{l|l|l} \bigsqcup\limits_{k_1:\,-|a|\le k_1\le|a|} (b+k_1 \equiv_a 0 \wedge a\neq 0 \wedge a^2k_1 \ge 0 \wedge a^2d-abc-ack_1 \ge 0) & \{-|a| \le k_1 \le |a|\} & \left\{x=\frac{b+k_1}{a}\right\} \\ \bigsqcup\limits_{k_1:\,-|c|\le k_1\le|c|} (d+k_1 \equiv_c 0 \wedge c\neq 0 \wedge c^2k_1 \le 0 \wedge acd+ack_1-bc^2 \ge 0) & \{-|c| \le k_1 \le |c|\} & \left\{x=\frac{d+k_1}{c}\right\} \\ b\le 0 \wedge d \ge 0 & \{\} & \{x=0\} \end{array}\right].$$

Let us now choose values $a = 5$, $b = 1$, $c = 0$, and $d = 1$ for the parameters. Then it is easy to see that the left hand side formulas in both the second and the third row do *not* hold. It is also not hard to see that the formula in the first row holds, where $-5 \le k_1 = 4 \le 5$ is the only choice satisfying both the congruence and the inequality $a^2k_1 \ge 0$. Accordingly, $x = 1$ is one solution of the given constraint system for our fixed choice of parameters.

6 Conclusions

We have given a weak quantifier elimination procedure for the full linear theory of the integers. For this, we have used a proof technique that we expect to be applicable also to other theories whenever parametric elimination sets become relevant. Next, we have modified the notion of extended quantifier elimination such that it becomes applicable to quantifier elimination for Presburger arithmetic and even more generally for the the full linear theory of the integers. We have given corresponding extended quantifier elimination schemes. All our methods are efficiently implemented in REDLOG. We have given various examples computed with this implementation. Our examples demonstrate that our methods are of considerable relevance for problems in practical computer science.

References

1. Berman, L.: Precise bounds for Presburger arithmetic and the reals with addition. In: 18th Annual Symposium on Foundations of Computer Science (FOCS 1977), 31 October–2 November, Providence, RI, USA, pp. 95–99. IEEE (1977)
2. Berman, L.: The complexity of logical theories. Theoret. Comput. Sci. **11**, 71–77 (1980)
3. Cooper, D.C.: Theorem proving in arithmetic without multiplication. In: Machine Intelligence, vol. 7, pp. 91–99. American Elsevier, New York (1972)
4. Davis, M.: Hilbert's tenth problem is unsolvable. Am. Math. Monthly **80**, 233–269 (1973)
5. Dolzmann, A., Sturm, T.: Redlog: computer algebra meets computer logic. ACM SIGSAM Bull. **31**(2), 2–9 (1997)
6. Dolzmann, A., Sturm, T.: Simplification of quantifier-free formulae over ordered fields. J Symbolic Comput. **24**(2), 209–231 (1997)
7. Dolzmann, A., Sturm, T.: P-adic constraint solving. In: Dooley S. (ed.) Proceedings of the 1999 International Symposium on Symbolic and Algebraic Computation (ISSAC 99), Vancouver, BC, pp. 151–158. ACM Press, New York (1999)
8. Dolzmann, A., Sturm, T.: Redlog user manual. Technical Report MIP-9905, FMI, Universität Passau, D-94030 Passau, Germany (1999). Edition 2.0 for Version 2.0 (1999)

9. Dolzmann, A., Sturm, T.: Generalized constraint solving over differential algebras. In: Ganzha, V.G., Mayr, E.W., Vorozhtsov, E.V. (eds.) Computer Algebra in Scientific Computing. Proceedings of the CASC 2004, pp. 111–125. Institut für Informatik, Technische Universität München, München, Germany (2004)
10. Dolzmann, A., Sturm, T., Weispfenning, V.: Real quantifier elimination in practice. In: Matzat, B.H., Greuel, G.M., Hiss, G. (eds.) Algorithmic Algebra and Number Theory, pp. 221–247. Springer, Berlin (1998)
11. Ferrante, J., Rackoff, C.W.: A decision procedure for the first-order theory of real addition with order. SIAM J. Comput. **4**, 69–77 (1975)
12. Ferrante, J., Rackoff, C.W.: The computational complexity of logical theories. No. 718 in Lecture Notes in Mathematics. Springer, Berlin (1979)
13. Fischer, M., Rabin, M.: Super-exponential complexity of Presburger arithmetic. SIAM-AMS Proc **7**, 27–41 (1974)
14. von zur Gathen, J., Sieveking, M.: A bound on solutions of linear integer equalities and inequalities. Proc AMS **72**, 155–158 (1978)
15. Griebl, M.: Automatic parallelization of loop programs for distributed memory architectures. Habilitationsschrift, University of Passau, 94030 Passau, Germany (2004). http://staff.uni-passau.de/~griebl/habilitation.html
16. Lasaruk, A.: Parametrisches integer-solving. Diploma Thesis, Universität Passau, D-94030 Passau, Germany (in German, 2005)
17. Loos, R., Weispfenning, V.: Applying linear quantifier elimination. Comput. J. **36**(5), 450–462 Special issue on computational quantifier elimination (1993)
18. Matiyasevich, Y.V.: Enumerable sets are Diophantine. Soviet Mathematics. Doklady **11**(2), 354–358 (1970). Translation of Doklady Akademii Nauk SSSR 191, 279–282 (1970)
19. Presburger, M.: Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. In: Comptes Rendus du premier congres de Mathematiciens des Pays Slaves, pp. 92–101. Warsaw, Poland (1929)
20. Seidl, A.M., Sturm, T.: Boolean quantification in a first-order context. In: Ganzha, V.G., Mayr, E.W., Vorozhtsov, E.V. (eds.) Computer Algebra in Scientific Computing. In: Proceedings of the CASC 2003, pp. 329–345. Institut für Informatik, Technische Universität München, München, Germany (2003)
21. Snelting, G.: Quantifier elimination and information flow control for software security. In: Dolzmann, A., Seidl, A., Sturm, A. (eds.) Algorithmic Algebra and Logic. Proceedings of the A3L 2005, pp. 237–242. BoD, Germany, Norderstedt (2005)
22. Snelting, G., Robschink, T., Krinke, J.: Efficient path conditions in dependence graphs for software safety analysis. ACM Trans Softw Eng Methodol **15**(4), 410–457 (2006)
23. Straßer, C.: Quantifier elimination for queues. In: Draisma, J., Kraft, H. (eds.) Rhine Workshop on Computer Algebra. Proceedings of the RWCA 2006, pp. 239–248. Universität Basel, Basel (2006)
24. Sturm, T.: Linear problems in valued fields. J. Symbolic Comput. **30**(2), 207–219 (2000)
25. Sturm, T., Weispfenning, V.: Quantifier elimination in term algebras. The case of finite languages. In: Ganzha, V.G., Mayr, E.W., Vorozhtsov, E.V. (eds.) Computer Algebra in Scientific Computing. Proceedings of the CASC 2002, pp. 285–300. Institut für Informatik, Technische Universität München, München, Germany (2002)
26. Vlach, M.: Conditions for the existence of solutions of the three-dimensional planar transportation problem. Discr. Appl. Math. **13**, 61–78 (1986)
27. Weispfenning, V.: The complexity of linear problems in fields. J. Symbolic Comput. **5**(1&2), 3–27 (1988)
28. Weispfenning, V.: The complexity of almost linear diophantine problems. J. Symbolic Comput. **10**(5), 395–403 (1990)
29. Weispfenning, V.: Complexity and uniformity of elimination in Presburger arithmetic. In: Küchlin, W.W. (ed.) Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation, Maui, HI (ISSAC 97), pp. 48–53. ACM Press, New York (1997)
30. Weispfenning, V.: Quantifier elimination for real algebra—the quadratic case and beyond. Appl. Algebra Eng. Commun. Comput. **8**(2), 85–101 (1997)
31. Weispfenning, V.: Simulation and optimization by quantifier elimination. J. Symbolic Comput. **24**(2), 189–208. Special issue on applications of quantifier elimination (1997)