

# Weak Quantifier Elimination for the Full Linear Theory of the Integers

## A Uniform Generalization of Presburger Arithmetic

Aless Lasaruk\*

FORWISS

Universität Passau

Germany

Thomas Sturm†

FMI

Universität Passau

Germany

MIP-0604  
April 9, 2006

### Abstract

We describe a weak quantifier elimination procedure for the full linear theory of the integers. That 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. 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 application examples demonstrate the applicability of our methods. These examples include problems currently discussed in practical computer science.

---

\*[lasaruk@uni-passau.de](mailto:lasaruk@uni-passau.de), [www.fmi.uni-passau.de/~lasaruk](http://www.fmi.uni-passau.de/~lasaruk)

†[sturm@uni-passau.de](mailto:sturm@uni-passau.de), [www.fmi.uni-passau.de/~sturm](http://www.fmi.uni-passau.de/~sturm)

# 1 Introduction

After the fundamental work of Presburger [Pre29] 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 [Coo72, FM74, FR75, FR79, vzGS78, Ber77, Ber80].

Weispfenning [Wei90, Wei97a] 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 [FM74]. 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 [Wei88, LW93, Wei97b, Stu00, SS03]. 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 wrt. 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 [LW93, Stu00]. 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 finite 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, which extend the language of first-order logic. To make this difference clear, we refer to our approach as *weak* quantifier elimination.

In comparison to other work [Wei97a] 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. 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 paper 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 [Wei97c] must 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 [DS97a, DS99]. REDLOG provides an extension of the computer algebra system REDUCE to a computer logic system implementing symbolic algorithms on first-order formulas wrt. 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. After making precise the syntax and the semantics of our framework in Section 2 we introduce in Section 3 the notions of bounded quantifiers and parametric elimination sets. On this basis we give our central elimination theorem. Section 4 then turns to extended quantifier

elimination procedures for the integers. In Section 5 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 Section 6. Some of our examples emphasize the applicability of our methods in practical computer science. In Section 7 we finally summarize and evaluate our work.

## 2 Basic Definitions

We consider the language of ordered rings with congruences. For convenience, we add relation symbols, such that the language becomes closed under logical negation:

$$L = \{0^{(0)}, 1^{(1)}, -^{(1)}, +^{(2)}, \cdot^{(2)}; \neq^{(2)}, \leq^{(2)}, <^{(2)}, >^{(2)}, \geq^{(2)}, \equiv^{(3)}, \neq^{(3)}\}. \quad (1)$$

Our domain are the integers  $\mathbb{Z}$ , which clarifies the semantics of the symbols in  $L$ .

We may assume w.l.o.g. that all terms involving variables  $a_1, \dots, a_r, x_1, \dots, x_s$  are distributive multivariate polynomials in  $\mathbb{Z}[a_1, \dots, a_r, x_1, \dots, x_s]$ . Such a term  $t$  is *linear* in  $x_1, \dots, x_s$  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  $x_1, \dots, x_s$  if so are both  $t$  and  $t'$ , and additionally  $m$  does not contain  $x_1, \dots, x_s$  at all. The same definition applies to incongruences. All other atomic formulas are linear in  $x_1, \dots, x_s$  if so are both contained terms. Finally, a quantifier-free formula is linear in  $x_1, \dots, x_s$  if so are all contained atomic formulas.

A general first-order formula is *linear*, if each contained atomic formula is linear in the set of variables that are bounded by some quantifier at this place. We are soon going to extend our language of first-order logic by further quantifiers in addition to  $\exists x$  and  $\forall x$ . Our notion of linearity will correspondingly be extended to variables bound by these new quantifiers.

Note that linear formulas resemble formulas of Presburger arithmetic with the generalization that all coefficients are polynomials in some parameters  $a_1, \dots, a_r$ . Therefore our framework has also been referred to as *uniform Presburger arithmetic* in the literature [Wei97a].

## 3 Weak Quantifier Elimination

Already in 1990, Weispfenning has given an effective quantifier elimination procedure for conventional Presburger arithmetic [Wei90]. 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 [Wei88, Wei97b, Stu00]. It can, however, be reanalyzed as a procedure of that type. We start out by 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.

### 3.1 Reanalysis of Conventional Presburger Arithmetic

Consider the following formula in conventional Presburger arithmetic:

$$\exists x(5x = b). \quad (2)$$

For this, the following is a quantifier-free equivalent according to [Wei90]:

$$\bigvee_{-5 \leq k \leq 5} (b + k \equiv_5 0 \wedge 5(b + k) = 5b). \quad (3)$$

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 in the right way for our purposes here.

In terms of virtual substitution, we have derived from the original formula an *elimination set*

$$E = \left\{ \left( b + k \equiv_5 0, \frac{b+k}{5} \right) \mid -5 \leq k \leq 5 \right\}. \quad (4)$$

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

$$E = \{ (\gamma_1, t_1), \dots, (\gamma_n, t_n) \} \quad (5)$$

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  $\gamma_i$  are quantifier-free formulas over the original language.

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

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma, t) \in E} (\gamma \wedge \varphi[t//x]), \quad (6)$$

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  $\gamma$  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  $E$  above the guards take care that the pseudo-terms actually describe integers. In fact, the range  $-5 \leq k \leq 5$  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 for Presburger arithmetic. We give the substitution rules for atomic formulas, which then naturally generalize to quantifier-free formulas:

$$\begin{aligned} (ax = b) \left[ \frac{b'}{a'} // x \right] &:= (ab' = a'b), \\ (ax \leq b) \left[ \frac{b'}{a'} // x \right] &:= (aa'b' \leq a'^2b), \\ (ax \equiv_m b) \left[ \frac{b'}{a'} // x \right] &:= (ab' \equiv_{ma'} a'b). \end{aligned} \quad (7)$$

Our slightly more complicated substitution rule for inequalities reflects the fact that  $a'$  is possibly a negative number.

### 3.2 Bounded Quantifiers

Let us now try to generalize the approach sketched in the previous section to our uniform case, where there are also parametric coefficients admitted for quantified variables. The following is a simple corresponding input formula:

$$\exists x(ax = b) \quad (8)$$

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

$$E = \left\{ \left( a \neq 0 \wedge b + k \equiv_a 0, \frac{b+k}{a} \right) \mid -|a| \leq k \leq |a| \right\} \cup \{(\text{true}, 0)\}. \quad (9)$$

We have added the guarded point  $(\text{true}, 0)$  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:

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

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  $L$ -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 [Wei97a]. 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.

A  $k$ -bound is a quantifier-free formula  $\beta(k, a_1, \dots, a_n)$  such that for all interpretations  $z_1, \dots, z_n \in \mathbb{Z}$  of the variables  $a_1, \dots, a_n$  the *solution set*

$$S_\beta^k(z_1, \dots, z_n) = \{ z \in \mathbb{Z} \mid \beta(z, z_1, \dots, z_n) \} \quad (11)$$

wrt.  $k$  and  $z_1, \dots, z_n$  is finite. We call  $k$  the *bound variable* and the other variables  $a_1, \dots, a_n$  the *bound parameters*.

A *bounded existential quantifier* is an existential quantifier  $\exists k$  occurring in a context

$$\exists k(\beta \wedge \varphi), \quad (12)$$

where  $\beta$  is a  $k$ -bound, and  $\varphi$  is an arbitrary first-order formula. We write such subformulas starting with a bounded quantifier as follows:

$$\bigsqcup_{k: \beta} \varphi. \quad (13)$$

Similarly, we define a *bounded universal quantifier* to be a universal quantifier  $\forall k$  occurring in a context  $\forall k(\beta \longrightarrow \varphi)$ , where  $\beta$  is a  $k$ -bound, and  $\varphi$  is an arbitrary first-order formula. We then write  $\bigsqcap_{k: \beta} \varphi$ .

A first-order formula is called *weakly quantifier-free* if all contained quantifiers are in fact bounded quantifiers. In order to avoid confusion, we from now

on refer to first-order formulas that do not contain any quantifiers at all as *strictly quantifier-free* formulas.

A *weak quantifier elimination* procedure is an algorithm assigning to any first-order formula  $\varphi$  a weakly quantifier-free formula  $\varphi'$  such that

$$\varphi' \longleftrightarrow \varphi. \quad (14)$$

Accordingly, a *strict quantifier elimination* procedure assigns to any first-order formula a strictly quantifier-free formula with property (14). 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

$$\bigsqcup_{k: -|a| \leq k \leq |a|} (a \neq 0 \wedge b + k \equiv_a 0 \wedge a(b + k) = ab) \vee (\text{true} \wedge a \cdot 0 = b). \quad (15)$$

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.

### Theorem 1 (Undecidability of Bounded Quantifiers)

*The question, whether a given quantifier is a bounded quantifier, is undecidable.*

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

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

The quantifier  $\exists k$  in  $\exists k(\beta \wedge \varphi)$  is a bounded quantifier if and only if  $\beta$  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 unsolvable [Mat70, Dav73].  $\square$

## 3.3 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 Equation (6) on page 5:

$$\exists x \varphi \longleftrightarrow \bigvee_{(\gamma, t) \in E} (\gamma \wedge \varphi[t//x]). \quad (16)$$

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

$$E = \{ (\gamma_i, t_i, B_i) \mid 1 \leq i \leq n \}, \text{ where } B_i = \{ (k_{ij}, \beta_{ij}) \mid 1 \leq j \leq m_i \}. \quad (17)$$

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

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 Equations (7) on page 6,  $E$  satisfies

$$\exists x\varphi \longleftrightarrow \bigvee_{(\gamma_i, t_i, B_i) \in E} \bigwedge_{k_{i1}: \beta_{i1}} \dots \bigwedge_{k_{im_i}: \beta_{im_i}} (\gamma_i \wedge \varphi[t_i//x]). \quad (18)$$

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  $\gamma_i$  and  $t_i$  are going to be even restricted to the free variables of  $\varphi$  together with  $k_{im_i}$ . Second, for every interpretation of variables satisfying  $\gamma_i$  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 Equation (9):

$$E = \left\{ \left( a \neq 0 \wedge b + k \equiv_a 0, \frac{b+k}{a}, \left( (k, -|a| \leq k \leq |a|) \right), (\text{true}, 0, \emptyset) \right) \right\}. \quad (19)$$

Note that the elimination result discussed as (15) is exactly the result of applying  $E$  in the sense of (18).

We had indicated at the end of the previous section 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 Equation (5). Aside from justifying the term *parametric elimination set*, this is a crucial technical concept for the proof of our elimination theorem in the following section. The remainder of the present section 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. We would like to emphasize, however, that such sets need never be actually computed.

Consider a parametric pre-elimination set  $E$  for  $\exists x\varphi$  as in (17). Let  $a_1, \dots, a_r$  be the parameters of  $\varphi$ , and let  $z_1, \dots, z_r \in \mathbb{Z}$ . Substitution of terms for variables on the guards  $\gamma_i$ , the test points  $t_i$ , and the bounds  $\beta_{ij}$  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

$$E_0 := E[z_1/a_1, \dots, z_r/a_r] = \{(\gamma'_i, t'_i, B'_i) \mid 1 \leq i \leq n\}, \quad (20)$$

where  $B'_i = ((k_{ij}, \beta'_{ij}) \mid 1 \leq j \leq m_i)$ .

The number of bounded quantifiers introduced by the application of  $E_0$  in the sense of (18) 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  $F$ :

$$\mu(F) = \max_{1 \leq i \leq n} m_i \quad \text{and} \quad \nu(F) = \#\{i \in \{1, \dots, n\} \mid m_i = \mu(F)\}. \quad (21)$$

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 w.l.o.g. that  $m_1 = \mu(F)$ . Let us consider separately that first triplet:

$$E_0 = \{(\gamma'_1, t'_1, ((k_{11}, \beta'_{11}), (k_{12}, \beta'_{12}), \dots, (k_{1m_1}, \beta'_{1m_1})))\} \cup E'_0, \quad (22)$$

where  $E'_0 = \{(\gamma'_i, t'_i, B'_i) \mid 2 \leq i \leq n\}$ . Due to our substitution  $[z_1/a_1, \dots, z_r/a_r]$ ,  $\beta'_{11}$  contains only one variable, which is  $k_{11}$ . Since by definition  $\beta_{11}$  is a  $k_{11}$ -bound, the solution set  $S_{\beta'_{11}}^{k_{11}} = S_{\beta_{11}}^{k_{11}}(z_1, \dots, z_r)$  is finite. We define

$$E_1 = \bigcup_{y \in S_{\beta'_{11}}^{k_{11}}} \{(\gamma'_1, t'_1, ((k_{12}, \beta'_{12}), \dots, (k_{1m_1}, \beta'_{1m_1})))\} [y/k_{11}] \cup E'_0. \quad (23)$$

The step from  $E_0$  to  $E_1$  has possibly even increased the total number of bounded quantifiers. It is, however, not hard to see that wrt. the lexicographic order on  $\mathbb{N}^2$  we have achieved that

$$(\mu(E_1), \nu(E_1)) \leq (\mu(E_0), \nu(E_0)). \quad (24)$$

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

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

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

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

such that

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

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

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

*Proof* (i) Inspection of the construction of  $\Pi(E, z_1, \dots, z_r)$  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 an elimination set for  $\exists x\varphi$ , it follows that  $E_0$  is an elimination set for  $\exists x\varphi'$ . That is

$$\exists x\varphi' \longleftrightarrow \bigvee_{(\gamma'_i, t'_i, B'_i) \in E_0} \bigcap_{k_{i1}: \beta'_{i1}} \dots \bigcap_{k_{im_i}: \beta'_{im_i}} (\gamma'_i \wedge \varphi'[t'_i // x]).$$

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

$$\bigsqcup_{k_{11}:\beta'_{11}} \bigsqcup_{k_{12}:\beta'_{12}} \dots \bigsqcup_{k_{1m_1}:\beta'_{1m_1}} \varphi' \longleftrightarrow \bigvee_{y \in S_{\beta'_{11}}^{k_{11}}} \bigsqcup_{k_{12}:\beta'_{12}[y/k_{11}]} \dots \bigsqcup_{k_{1m_1}:\beta'_{1m_1}[y/k_{11}]} \varphi'[y/k_{11}],$$

and consequently

$$\begin{aligned} \exists x \varphi' &\longleftrightarrow \bigvee_{(\gamma'_i, t'_i, B'_i) \in E_0} \bigsqcup_{k_{i1}:\beta'_{i1}} \dots \bigsqcup_{k_{im_i}:\beta'_{im_i}} (\gamma'_i \wedge \varphi'[t'_i//x]) \\ &\longleftrightarrow \bigvee_{(\gamma'_j, t'_j, B'_j) \in E_1} \bigsqcup_{k_{j1}:\beta'_{j1}} \dots \bigsqcup_{k_{jm_j}:\beta'_{jm_j}} (\gamma'_j \wedge \varphi'[t'_j//x]). \end{aligned}$$

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

As next we show the counterpart, that a parametric pre-elimination set for  $\exists x \varphi$  is actually a parametric elimination set for  $\exists x \varphi$ , if every  $(z_1, \dots, z_n)$ -projection  $\Pi(E, z_1, \dots, z_n)$  is a elimination set for  $\varphi' = \varphi[z_1/a_1, \dots, z_n/a_n]$ .

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

$$\exists x \varphi' \longleftrightarrow \bigvee_{(\gamma'_j, t'_j) \in \Pi(E, z_1, \dots, z_n)} (\gamma'_j \wedge \varphi'[t'_j//x]),$$

where  $\varphi' = \varphi[z_1/a_1, \dots, z_r/a_r]$ . 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 holds

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

The direction from the right to the left is trivial. Let  $z_1, \dots, z_r \in \mathbb{Z}$  an interpretation of  $a_1, \dots, a_r$  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  $\Pi(E, z_1, \dots, z_r)$  holds:

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

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

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

It follows that holds

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

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

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

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

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

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

*Proof* Let  $E$  be of the form

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

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

$$\begin{aligned}\exists x\varphi' \longleftrightarrow \exists x\varphi &\longleftrightarrow \bigvee_{(\gamma_i, t_i, B_i) \in E} \bigsqcup_{k_{i1}: \beta_{i1}} \dots \bigsqcup_{k_{im_i}: \beta_{im_i}} (\gamma_i \wedge \varphi[t_i // x]) \\ &\longleftrightarrow \bigvee_{(\gamma_i, t_i, B_i) \in E} \bigsqcup_{k_{i1}: \beta_{i1}} \dots \bigsqcup_{k_{im_i}: \beta_{im_i}} (\gamma_i \wedge \varphi'[t_i // x]). \quad \square\end{aligned}$$

### 3.4 The Elimination Theorem

In this chapter we give a weak quantifier elimination procedure for linear formulas in Presburger arithmetic.

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

$$Q_1, \dots, Q_n \psi, \quad (25)$$

$$k_1:\beta_1 \quad k_n:\beta_n$$

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

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

(ii) *Let  $\varphi$  be a strictly quantifier-free formula. Then there is a strictly quantifier-free positive formula  $\varphi'$  such that  $\varphi' \longleftrightarrow \varphi$ .  $\square$*

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  $\beta_i$  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 bounds  $\beta_i$ .

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** *Let  $\varphi$  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  $\varrho_i \in \{=, \leq, <, \neq, >, \geq\}$  for all  $i \in I_1$ . Then the set  $S_\varphi^x$  is a finite union of intervals. Furthermore, for each interval boundary  $z \in S_\varphi^x$  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. Note that 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  $\varphi$ . 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  $\varrho_i$  equals  $\neq$ , we have

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

By our induction hypothesis, the solution set of each of the two conjunctions above is a finite union of intervals, and  $S_\varphi^x$  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  $\varphi$  and adapting the relation symbols accordingly, we may w.l.o.g. assume that all  $n_i > 0$ . Since  $z \in S$  and  $z - 1 \notin S$ , there is a corresponding atomic formula  $n_i x \varrho_i r_i$  with that very property. It follows from our assumptions made so far that  $\varrho_i \notin \{<, \leq\}$ . If  $n_i = 1$  and in addition  $\varrho_i \in \{=, \geq\}$ , then we choose  $k = 0$ . Otherwise, we choose the unique  $k \in \{1, \dots, |n_i|\}$ , such that  $n_i \mid r_i + k$ . Note that  $r_i/n_i$  is not an integer then.  $\square$

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 a  $m$ -periodic set, where  $m = \text{lcm}\{m_1, m_2\}$ . This transfers to the solution sets of congruences and incongruences both of which are periodic sets:

**Lemma 7** *Let  $\varphi$  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  $\varrho_i$  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  $S_\varphi^x$  is an  $m$ -periodic set with  $m = \text{lcm}\{m_i \mid i \in I_2\}$ .  $\square$*

We now can state the main result of this paper.

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

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

*Let the set of all atomic formulas of  $\psi$  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  $a_1, \dots, a_r$ . The  $s_i$  are polynomials in both the parameters  $a_1, \dots, a_r$  and the bound variables  $k_1, \dots, k_n$ . For  $i \in I_1$ , we have  $\varrho_i \in \{=, \neq, <, \leq, \geq, >\}$ . For  $i \in I_2$ , we have that  $\varrho_i$  is either  $\equiv_{m_i}$  or  $\not\equiv_{m_i}$ , where  $m_i$  is a polynomial in  $a_1, \dots, a_r$ .*

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

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

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

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

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

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

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

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

$$\begin{aligned} n'_i &= n_i[z_1/a_1, \dots, z_r/a_r], & m'_i &= m_i[z_1/a_1, \dots, z_r/a_r], \\ s'_i &= s_i[z_1/a_1, \dots, z_r/a_r], & s_i^{*'} &= s_i^*[z_1/a_1, \dots, z_r/a_r], \\ r'_i &= r_i[z_1/a_1, \dots, z_r/a_r]. \end{aligned}$$

According to Lemma 3 it now suffices to show that

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

where  $\Pi(E, z_1, \dots, z_r)$  is the  $(z_1, \dots, z_r)$ -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  $\varphi'$  into an equivalent formula  $\varphi''$ , 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 a finite disjunction or conjunction. After this, all atomic formulas are of the form  $n'_i x \varphi'_i s''_{i,y} + r'_i$ , where

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

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

$$y_j \in \mathcal{S}_{\beta_i}^{k_i}(\mathbf{z}_1, \dots, \mathbf{z}_r, y_1, \dots, y_{j-1}) = \mathcal{S}_{\beta_i^*}^{k_i^*}(\mathbf{z}_1, \dots, \mathbf{z}_r, y_1, \dots, y_{j-1}).$$

Defining  $s^{*''}_{i,y_1, \dots, y_n} = s_i^{*'}[y_1/k_1^*, \dots, y_n/k_n^*]$  we obviously obtain  $s''_{i,y} = s^{*''}_{i,y}$ .

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  $\varphi''$  equals true, then  $\varphi' \longleftrightarrow \text{true}$ , and  $(\text{true}, 0) \in \Pi(E, z_1, \dots, z_r)$  is a suitable test point. We may from now on assume that  $\varphi''$  does not contain any truth values.

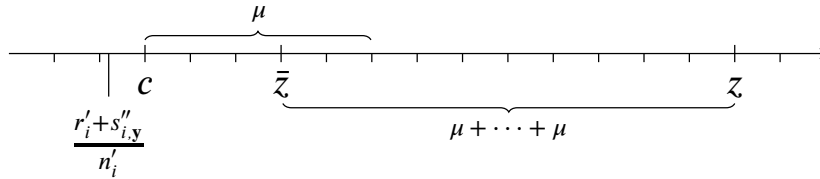
We had assumed that  $\exists x \varphi'$  holds. It follows that the equivalent formula  $\exists x \varphi''$  holds as well. From this it follows in turn that  $\exists x \omega$  holds for at least one conjunction  $\omega$  in the disjunctive normal form  $\varphi''$ . We may assume w.l.o.g. that all atomic formulas in  $\omega$  actually contain  $x$ . We can split  $\omega$  into the conjunction  $\omega_2$  containing all the congruences and incongruences and the conjunction  $\omega_1$  containing all other atomic formulas. Formally,  $\omega = \omega_1 \wedge \omega_2$ , where

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

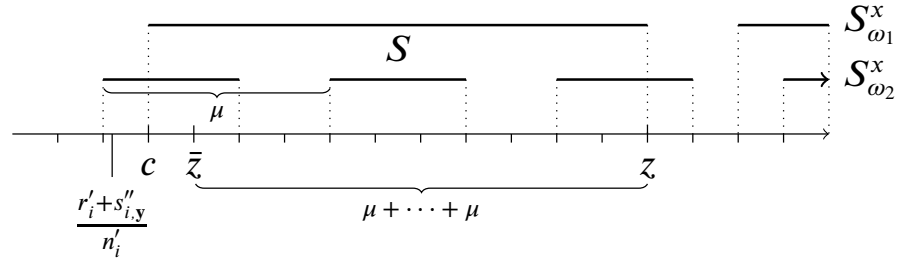
It is not hard to see that  $J_1 \subseteq I_1 \cup J$  and  $J_2 \subseteq I_2 \setminus J$ . We assume w.l.o.g. that  $n'_i > 0$  and  $\gcd\{n'_i, s''_{i,y} + r'_i\} = 1$  for all  $i \in J_1$ .

Consider the solution sets  $S_{\omega_1}^x$  and  $S_{\omega_2}^x$ . If  $S_{\omega_1}^x = S_{\omega_2}^x = \mathbb{Z}$ , then  $\omega \longleftrightarrow \text{true}$ , and again  $(\text{true}, 0) \in \Pi(E, z_1, \dots, z_r)$  is a suitable test point. Otherwise it is helpful to distinguish cases on the form of  $S_{\omega_1}^x$  for a moment. Recall that  $S_{\omega_2}^x$  is periodic by Lemma 7. Denoting by  $\mu$  its minimal positive period, we have  $1 \leq \mu \leq m' = m[z_1/a_1, \dots, z_r/a_r]$ :

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



- (b) If  $S_{\omega_1}^x \neq \mathbb{Z}$ , then still  $z \in S_{\omega_1}^x$ . According to Lemma 6,  $z$  lies in some interval  $S \subseteq S_{\omega_1}^x$  that is w.l.o.g. 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  $i \in J_1$ ,  $y \in Y_i$ , and  $\Delta \in \mathbb{Z}$  with  $0 \leq \Delta \leq n'_i$  such that  $c = (r'_i + s''_{i,y} + \Delta)/n'_i$ . Due to periodicity there is some  $\bar{z} \in S_{\omega_2}^x$  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  $\bar{z} \in S \subseteq S_{\omega_1}^x$ , and furthermore  $\bar{z} \in S_{\omega_1}^x \cap S_{\omega_2}^x = S_{\omega}^x$ .



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

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

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

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

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

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

Consequently  $\omega[t' // x]$  is equivalent to  $\omega[\bar{z} // x]$ , which 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

$$\bigvee_{(\gamma', t') \in \Pi(E, z_1, \dots, z_n)} (\gamma' \wedge \varphi'[t' // x])$$

holds, which is what had to be shown.  $\square$

### Theorem 9 (Uniform Weak Elimination Theorem)

*The full linear theory of 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  $\varphi$  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  $\varphi$  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

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

We obtain  $\hat{\varphi}'$  from  $\hat{\varphi}$  by equivalently replacing  $\exists x\varphi$  with  $\varphi'$ . 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.  $\square$

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.

## 4 Extended Weak Quantifier Elimination

Let us return to our introductory example for conventional elimination sets from Section 3.1 on page 5: For the conventional Presburger formula  $\exists x(5x = b)$ , there is a regular elimination set

$$E = \left\{ \left( b + k \equiv_5 0, \frac{b+k}{5} \right) \mid -5 \leq k \leq 5 \right\}. \quad (26)$$

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

$$\bigvee_{-5 \leq k \leq 5} (b + k \equiv_5 0 \wedge 5(b + k) = 5b), \quad (27)$$

which is in turn obviously equivalent to  $b \equiv_5 0$ . From the original quantified formula it is easy to see 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 conventional substitution methods such information can be derived systematically. A corresponding procedure has been originally introduced for the reals by Weispfenning, who referred to the problem as *extended linear elimination problem* [Wei97c]. Instead of forming a disjunction, we write down the following data:

$$\left[ \begin{array}{l|l} b + (-5) \equiv_5 0 \wedge 5(b + (-5)) = 5b & \left\{ x = \frac{b+(-5)}{5} \right\} \\ \vdots & \vdots \\ b + 0 \equiv_5 0 \wedge 5(b + 0) = 5b & \left\{ x = \frac{b+0}{5} \right\} \\ \vdots & \vdots \\ b + 5 \equiv_5 0 \wedge 5(b + 5) = 5b & \left\{ x = \frac{b+5}{5} \right\} \end{array} \right]. \quad (28)$$

For any choice of  $b$ , the left hand side formula of exactly the row displayed in the middle, which originates 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 (5) the corresponding substitution scheme (6) can be replaced by the following *extended quantifier elimination* scheme:

$$\exists x \varphi \rightsquigarrow \left[ \begin{array}{c|c} \gamma_1 \wedge \varphi[t_1//x] & \{x = t_1\} \\ \vdots & \vdots \\ \gamma_n \wedge \varphi[t_n//x] & \{x = t_n\} \end{array} \right]. \quad (29)$$

We have to clarify the relation between the left hand side and the right hand side in (29). For obvious syntactic reasons there cannot hold equivalence. Nevertheless, there is an 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.  $\square$*

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 a 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.  $\square$*

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 for the outermost block.

## 4.1 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

$$E = \left\{ \left( a \neq 0 \wedge b + k \equiv_a 0, \frac{b+k}{a}, ((k, -|a| \leq k \leq |a|)) \right), (\text{true}, 0, \emptyset) \right\}. \quad (30)$$

To start with, we copy the idea of our extended quantifier elimination scheme (29). 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:

$$\left[ \begin{array}{c|c} \bigsqcup_{k: -|a| \leq k \leq |a|} (a \neq 0 \wedge b + k \equiv_a 0 \wedge a(b+k) = ab) & \left\{ x = \frac{b+k}{a} \right\} \\ \text{true} \wedge a \cdot 0 = b & \{x = 0\} \end{array} \right]. \quad (31)$$

For any choice 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 at most  $2a$  trial evaluations, we can find out that the formula holds due to the possible choice  $k = 0$ . 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 suitable assignment. 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 we have described for regular elimination sets in the previous section. For all other choices of  $a$  and  $b$  the original quantified formula does not hold.

The procedure described above by means of an example is generally applicable and correct. There is, however, one subtle point about the syntactical form of our extended quantifier elimination result (31). 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 occur freely. This observation is in fact evidence for a much deeper problem, which becomes apparent when simplification gets involved.

## 4.2 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 [DS97b, Las05].

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 (31):

$$\left[ \begin{array}{c|c} \bigsqcup_{k: -5 \leq k \leq 5} (5 \neq 0 \wedge b + k \equiv_5 0 \wedge 5(b + k) = 5b) & \left\{ x = \frac{b+k}{5} \right\} \\ \text{true} \wedge 5 \cdot 0 = b & \{x = 0\} \end{array} \right]. \quad (32)$$

By first replacing both left hand side formulas with simpler but equivalent ones and then dropping the second row where the left hand side formula has been replaced by false we derive the following simpler result:

$$\left[ b \equiv_5 0 \mid \left\{ x = \frac{b+k}{5} \right\} \right]. \quad (33)$$

At that point the bound variable  $k$  occurs in the sample point but not 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 (32) that we have to consider for  $k$  the finite range  $\{-5, \dots, 5\}$ . So we have to learn that in our situation the classical scheme for extended quantifier elimination does not work in combination with simplification, which is in turn inevitable.

Note that our counter-example is even a regular Presburger formula. For such formulas, the problem can obviously be resolved by immediately expanding all bounded quantifiers into finite disjunctions. This, however, increases both time and space complexity by one exponential step from doubly exponential to triply exponential.

Our general solution is to modify the extended quantifier elimination scheme (29) such that we additionally keep the bound conditions contained in the parametric elimination sets. With respect to a parametric elimination set

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

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

$$\exists x \varphi \rightsquigarrow \left[ \begin{array}{c|c|c} \bigsqcup_{k_{11}: \beta_{11}} \dots \bigsqcup_{k_{1m_1}: \beta_{1m_1}} (\gamma_1 \wedge \varphi[t_1//x]) & \{\beta_{11}, \dots, \beta_{1m_1}\} & \{x = t_1\} \\ \vdots & \vdots & \vdots \\ \bigsqcup_{k_{n1}: \beta_{n1}} \dots \bigsqcup_{k_{nm_n}: \beta_{nm_n}} (\gamma_n \wedge \varphi[t_n//x]) & \{\beta_{n1}, \dots, \beta_{nm_n}\} & \{x = t_n\} \end{array} \right]. \quad (34)$$

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  $k_{i1}, \dots, k_{im_i}$ . 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.  $\square$

Given a quantified formula  $\exists x\varphi$ , good choices for  $k_{i1}, \dots, k_{im_i}$  can always be determined by successively plugging into  $\varphi$  the values of the terms  $t_i$  for the finitely many possible choices within the computed range.

In general, simplification will not remove all the bounded quantifiers from the left hand side formulas. Whenever there are 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, good ones for the corresponding right hand side assignment as well. 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

$$\left[ b \equiv_5 0 \mid \{-5 \leq k \leq 5\} \mid \left\{x = \frac{b+k}{5}\right\} \right]. \quad (35)$$

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

## 5 Implementation

The procedures described in this paper have been implemented in REDLOG, which stands for REDUCE *logic system* [DS97a, DS99]. It provides an extension of the computer algebra system REDUCE to a computer logic system implementing symbolic algorithms on first-order formulas wrt. temporarily fixed first-order languages and theories. Such a choice of language and theory is called a *context*. So far, the following REDLOG contexts had been available:

ACFSF (Algebraically closed fields, standard form representation of terms). The class of algebraically closed fields such as the complex numbers over the language of rings.

DCFSF (Differentially closed fields, standard form representation of terms). A context for dealing with 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 [DS04].

DVFSF (Discretely valued fields, standard form representation of terms). The most prominent example for discretely valued fields are  $p$ -adic numbers for some prime  $p$  with abstract divisibility relations encoding order between values. All DVFSF algorithms are optionally uniform in  $p$  [Stu00].

IBALP (Initial Boolean algebras, Lisp prefix representation of terms). The class of Boolean algebras with two elements. These algebras are uniquely determined up to isomorphisms. IBALP comprises quantified propositional calculus [SS03].

OFSF (Ordered fields, standard form representation of terms). 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 [DSW98].

The work discussed here establishes another such context:

PASF (Presburger arithmetic, standard form representation of terms). 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. To our knowledge our implementation is the only one available for the full linear theory of the integers.

Besides the core elimination methods described in the present paper there has considerable further theoretical research entered the implementation. This comprises is 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 [DS97b]. The resulting simplification theory for the integers has so far been described only in the diploma thesis of the first author [Las05].

The remainder of this section is devoted to a very short *primer* on REDLOG and PASF. 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 invoking REDUCE, the following commands load the REDLOG package and switch to the context PASF:

```
▷ load_package redlog;
▷ rlset pasf;
```

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

```
▷ on time;
```

The timings given are CPU times. They 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. Weak quantifier elimination can be applied as follows:

```
▷ sol := rlwqe phi;
```

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,phi);
```

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 phi;
```

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;
```

## 6 Benchmarks and Application Examples

In this section we collect some practical computations with our implementation in REDLOG in combination with the current REDUCE 3.8. 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.

Our Implementation is part of the current development system of REDLOG. The code is going to be distributed with REDUCE 3.9. Until then it is available from the authors on request. All computations have been performed on a 2 GHz Pentium 4 using 128 MB of RAM. The times given are CPU times.

### 6.1 Planar Transportation Problems

Multidimensional planar transportation problems have been a challenging benchmark for real quantifier elimination for many years [LW93, Vla86]. 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 [LW93]. All formulas are going to be regular Presburger formulas. In dimension 1, the input formulas are of the form

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

For clarifying the entries in the tables below, let us discuss  $T_{1,1}$  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.

$m$	time ( <code>rlwqe</code> )	size	time ( <code>rlexpand</code> )	result	time (total)
5	0.06 s	192	0.02 s	$a \geq 0$	0.08 s
6	0.14 s	480	0.06 s	$a \geq 0$	0.20 s
7	0.34 s	1152	0.24 s	$a \geq 0$	0.58 s
8	0.80 s	2688	0.74 s	$a \geq 0$	1.54 s
9	1.89 s	6144	2.39 s	$a \geq 0$	4.28 s
10	4.43 s	13824	10.05 s	$a \geq 0$	14.48 s
11	11.10 s	30720	44.43 s	$a \geq 0$	55.53 s

For comparison, the current standard real quantifier 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 wrt. redundancy.

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

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

In contrast to dimension 1, our final results here are so large in general that it is not reasonable to display them. We instead give the sizes. For  $m = 1$ , there do not any bounded quantifiers come into existence.

$m$	time (rlwqe)	size	time (rlexpand)	size	time (total)
1	< 0.01 s	2	–	2	< 0.01 s
2	0.01 s	24	< 0.01 s	32	< 0.02 s
3	0.94 s	5032	39.62 s	32766	40.56 s

For comparison, real quantifier elimination in REDLOG applied to  $T_{2,3}$ ,  $T_{2,7}$ , and  $T_{2,8}$  yields 36 (in 0.01 s), 12936 (in 8.85 s), and 54912 (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  $a_1 = b_1 \wedge a_1 \geq 0$ .

## 6.2 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 [Gri04]. In the following 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:

$$\exists i \exists j \exists i' \exists j' (0 \leq i \leq m \wedge 0 \leq j \leq m \wedge 0 \leq i' \leq m \wedge 0 \leq j' \leq m \wedge (i \neq i' \vee j \neq j') \wedge i + j \neq i' + j' \wedge ni + j = ni' + j').$$

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  $n \neq 1$  on the reasonable assumption that both  $m$  and  $n$  are greater than 0.

Now let us 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 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 replace the definition of  $m$  by  $m = \text{lcm}\{m_i \mid i \in I_2\}$  because  $m_i > 0$  for all  $i$ . That problem can be overcome in future version 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  $I_1 = \emptyset$ .

### 6.3 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 [Sne05, SRK06].

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.



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  $-5 \leq k_1 = 4 \leq 5$  is the only choice satisfying both the congruence and the inequality  $a^2 k_1 \geq 0$ . Accordingly,  $x = 1$  is one solution of the given constraint system for our fixed choice of parameters.

## 7 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 as well as to weak quantifier elimination for the full linear theory. We have given corresponding extended elimination procedures. 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

- [Ber77] Leonard Berman. Precise bounds for presburger arithmetic and the reals with addition. In *18th Annual Symposium on Foundations of Computer Science, 31 October-2 November, Providence, Rhode Island, USA*, pages 95–99. IEEE, 1977.
- [Ber80] Leonard Berman. The complexity of logical theories. *Theoretical Computer Science*, 11:71–77, 1980.
- [Coo72] D. C. Cooper. Theorem proving in arithmetic without multiplication. In *Machine Intelligence*, volume 7, pages 91–99, New York, 1972. American Elsevier.
- [Dav73] Martin Davis. Hilbert’s tenth problem is unsolvable. *American Mathematical Monthly*, 80:233–269, 1973.
- [DS97a] Andreas Dolzmann and Thomas Sturm. Redlog: Computer algebra meets computer logic. *ACM SIGSAM Bulletin*, 31(2):2–9, June 1997.

- [DS97b] Andreas Dolzmann and Thomas Sturm. Simplification of quantifier-free formulae over ordered fields. *Journal of Symbolic Computation*, 24(2):209–231, August 1997.
- [DS99] Andreas Dolzmann and Thomas Sturm. Redlog user manual. Technical Report MIP-9905, FMI, Universität Passau, D-94030 Passau, Germany, April 1999. Edition 2.0 for Version 2.0.
- [DS04] Andreas Dolzmann and Thomas Sturm. Generalized constraint solving over differential algebras. In V. G. Ganzha, E. W. Mayr, and E. V. Vorozhtsov, editors, *Computer Algebra in Scientific Computing. Proceedings of the CASC 2004*, pages 111–125. Institut für Informatik, Technische Universität München, 2004.
- [DSW98] Andreas Dolzmann, Thomas Sturm, and Volker Weispfenning. Real quantifier elimination in practice. In B. H. Matzat, G.-M. Greuel, and G. Hiss, editors, *Algorithmic Algebra and Number Theory*, pages 221–247. Springer, Berlin, 1998.
- [FM74] M.J. Fischer and Rabin M. Super-exponential complexity of Presburger arithmetic. *SIAM-AMS Proceedings*, 7:27–41, 1974.
- [FR75] Jeanne Ferrante and Charles W. Rackoff. A decision procedure for the first-order theory of real addition with order. *SIAM Journal on Computing*, 4:69–77, 1975.
- [FR79] Jeanne Ferrante and Charles W. Rackoff. *The Computational Complexity of Logical Theories*. Number 718 in Lecture Notes in Mathematics. Springer-Verlag, Berlin, 1979.
- [Gri04] Martin Griebel. *Automatic Parallelization of Loop Programs for Distributed Memory Architectures*. Habilitationsschrift, University of Passau, 94030 Passau, Germany, 2004.
- [Las05] Aless Lasaruk. Parametrisches Integer-Solving. Diploma thesis, Universität Passau, D-94030 Passau, Germany, April 2005.
- [LW93] Rüdiger Loos and Volker Weispfenning. Applying linear quantifier elimination. *THE Computer Journal*, 36(5):450–462, 1993. Special issue on computational quantifier elimination.
- [Mat70] Yuri V. Matiyasevich. Enumerable sets are Diophantine. *Soviet Mathematics. Doklady*, 11(2):354–358, 1970. Translation of Doklady Akademii Nauk SSSR 191, 279–282 (1970).

- [Pre29] Mojzesz Presburger. Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. In *Comptes Rendus du premier congrès de Mathématiciens des Pays Slaves*, pages 92–101, Warsaw, Poland, 1929.
- [Sne05] Gregor Snelting. Quantifier elimination and information flow control for software security. In Andreas Dolzmann, Andreas Seidl, and Thomas Sturm, editors, *Algorithmic Algebra and Logic. Proceedings of the A3L 2005*. BoD, Norderstedt, 2005.
- [SRK06] Gregor Snelting, Torsten Robschink, and Jens Krinke. Efficient path conditions in dependence graphs for software safety analysis. *ACM Transactions on Software Engineering and Methodology*, 2006. To appear.
- [SS03] Andreas M. Seidl and Thomas Sturm. Boolean quantification in a first-order context. In V. G. Ganzha, E. W. Mayr, and E. V. Vorozhtsov, editors, *Computer Algebra in Scientific Computing. Proceedings of the CASC 2003*, pages 329–345. Institut für Informatik, Technische Universität München, Passau, 2003.
- [Stu00] Thomas Sturm. Linear problems in valued fields. *Journal of Symbolic Computation*, 30(2):207–219, August 2000.
- [Vla86] Milan Vlach. Conditions for the existence of solutions of the three-dimensional planar transportation problem. *Discrete Applied Mathematics*, 13:61–78, 1986.
- [vzGS78] J. von zur Gathen and M. Sieveking. A bound on solutions of linear integer equalities and inequalities. *Proceedings of the AMS*, 72:155–158, 1978.
- [Wei88] Volker Weispfenning. The complexity of linear problems in fields. *Journal of Symbolic Computation*, 5(1&2):3–27, February–April 1988.
- [Wei90] Volker Weispfenning. The complexity of almost linear diophantine problems. *Journal of Symbolic Computation*, 10(5):395–403, November 1990.
- [Wei97a] Volker Weispfenning. Complexity and uniformity of elimination in presburger arithmetic. In Wolfgang W. Kuchlin, editor, *Proceedings of*

*the 1997 International Symposium on Symbolic and Algebraic Computation (ISSAC 97)*, pages 48–53. ACM Press, New York, 1997, Maui, HI, July 1997.

- [Wei97b] Volker Weispfenning. Quantifier elimination for real algebra—the quadratic case and beyond. *Applicable Algebra in Engineering Communication and Computing*, 8(2):85–101, February 1997.
- [Wei97c] Volker Weispfenning. Simulation and optimization by quantifier elimination. *Journal of Symbolic Computation*, 24(2):189–208, August 1997. Special issue on applications of quantifier elimination.