EPiC Series in Computing

Volume 100, 2024, Pages 147–164

Proceedings of 25th Conference on Logic for Programming, Artificial Intelligence and Reasoning

VIRAS: Conflict-Driven Quantifier Elimination for Integer-Real Arithmetic

Johannes Schoisswohl[1], Laura Kovács[1], and Konstantin Korovin[2]

[1] TU Wien
{johannes.schoisswohl,laura.kovacs}@tuwien.ac.at
[2] The University of Manchester
konstantin.korovin@manchester.ac.uk

Abstract

We introduce Virtual Integer-Real Arithmetic Substitution (Viras), a quantifier elimination procedure for deciding quantified linear mixed integer-real arithmetic problems. Viras combines the framework of virtual substitutions with conflict-driven proof search and linear integer arithmetic reasoning based on Cooper's method. We demonstrate that Viras gives an exponential speedup over state-of-the-art methods in quantified arithmetic reasoning, proving problems that SMT-based techniques fail to solve.

1 Introduction

Automated reasoning is routinely used in applications of mathematical theory formalisation [11], formal verification [12] and web security [8]. The demand for proving properties with both quantifiers and theories is increasing in these and similar domains, especially in the context of arithmetic reasoning. Common approaches addressing this demand implement incomplete heuristics for quantifier instantiation (QI) [3,14,23] or integrate complete quantifier elimination (QE) [2,4], adjusted for a particular arithmetic domain. In this paper, we improve the state-of-the-art in quantifier elimination by introducing a new calculus for *mixed* integer-real arithmetic, while aiming at reducing computational cost of QE [26].

QE transforms first-order formulas $\exists x.\phi$ or $\forall x.\phi$ into an equivalent formula ϕ' that does not contain the variable x. Seminal works solving QE were introduced within Cylindrical Algebraic Decomposition – CAD [1, 7], lazy model enumaration [22] and virtual substitution [13, 25] for non-linear real arithmetic and Cooper's method for linear integer arithmetic [9]. These techniques have been used and extended with tailored solutions for satisfiability modulo theory (SMT) solving in linear and non-linear real arithmetic (Lra,Nra) [4–6, 15, 16, 19, 20] or linear integer arithmetic (Lia) [2, 17]. Yet, existing solutions [23, 26] fail deciding the mixed theory of linear integer and real arithmetic (Lira) adequately. The work of [26] requires formula normalizations that result in an exponential blow-up in the input formula size, whereas [23] is restricted to $\forall\exists$ problems.

This paper describes the Viras method for solving linear integer-real arithmetic formulas with *arbitrary quantifier alternations* (Sect. 4), using virtual substitutions to implement quantifier elimination in Lira. Within Viras, we combine real and integer arithmetic via a floor

function $\lfloor\cdot\rfloor$ for rounding reals to closest integers. VIRAS uses *virtual substitutions* to eliminate quantified variables x by instantiating with so-called *virtual terms*. We extend the framework of virtual substitutions with so-called $\mathbb{Z}$-terms, allowing us to generalize Cooper's method from LIA to LIRA, and further optimizing it for equality literals (Sect. 5). VIRAS overcomes the burden of arithmetic normalisations performed in [26] and avoids an exponential blow-up in processing LIRA formula (Sect. 5). We further extend VIRAS with conflict-driven proof search (Sect. 6), by generalizing [18] to handle virtual terms involving infinitesimals ε and $\pm\infty$.

Our contributions. In summary, this paper brings the following contributions.

- We present the VIRAS method implementing a quantifier elimination procedure for linear mixed integer-real arithmetic, generalizing both Cooper's method [9] and virtual substitutions [25] by introducing $\mathbb{Z}$-terms (Sect. 4), and prove [1] that VIRAS is indeed a quantifier elimination procedure in Theorem 1.
- We show VIRAS is exponentially faster than related techniques [26]. Moreover, VIRAS can solve problems that SMT-based solutions fail to solve (Sect. 5).
- We enhance VIRAS with conflict-driven proof search, by extending the framework introduced in [18] to support ε and ∞-terms (Sect. 6).

2 Motivating Example

We illustrate LIRA reasoning and the main steps of VIRAS using the formula:

$$\exists x.\phi \quad = \quad \exists x.(\underbrace{\lfloor a\rfloor + \tfrac{1}{3} \leq x}_{L_1} \wedge \underbrace{x \leq \lfloor a\rfloor + \tfrac{2}{3}}_{L_2} \wedge \underbrace{\lceil x\rceil - x \geq c}_{L_3}) \tag{1}$$

where $\lfloor a\rfloor$ denotes the floor of the real number a; that is, the greatest integer such that $\lfloor a\rfloor \leq a$. Eliminating the quantifier $\exists x$ in the LIRA formula (1) comes with the challenge or reasoning floor-expressions within real-integer linear arithmetic.

Note that the literals L_1, L_2 impose respectively lower and upper bounds on the quantified variable x. Intuitively, L_1, L_2 imply that, in order for ϕ to hold for some x, x must be within the non-empty interval $\left[\lfloor a\rfloor + \frac{1}{3}, \lfloor a\rfloor + \frac{2}{3}\right]$. Further, literal L_3 asserts that x is in a periodically repeating set of solutions, for the following reason: as $\lceil x\rceil - x$ can be only within $[0, 1)$, the literal L_3 cannot hold if c belongs to the interval $[1, \infty)$; if $c \in [-\infty, 1)$ then L_3 holds iff $x \in \bigcup_{z\in\mathbb{Z}}(z, z{+}1{-}c]$. As such, the LIRA formula (1) holds iff intersection I of the intervals restricting the values of x, as asserted by L_1, L_2, L_3, is non-empty. Following upon this observation, I is clearly non-empty when $c < 0$. On the other hand, if $c \in [0, 1)$, then I is non-empty iff $(\lfloor a\rfloor, \lfloor a\rfloor + 1 - c] \cap \left[\lfloor a\rfloor + \frac{1}{3}, \lfloor a\rfloor + \frac{2}{3}\right]$ is non-empty, which is the case iff $\lfloor a\rfloor + \frac{1}{3} \leq \lfloor a\rfloor + 1 - c$ In summary, this means a quantifier-free equivalent formula to the LIRA formula (1) is $c \leq \frac{2}{3}$.

Note that by finding a quantifier-free formula $c \leq \frac{2}{3}$ equivalent to formula (1), we applied QE to (1) using arithmetic reasoning with floor-expressions. For automating such a QE process, our VIRAS method implements the following steps. We transform (1) into an equivalent, quantifier-free formula by computing a so-called *elimination set* $\mathsf{elim}(\phi)$ and by *virtually substituting* x with each element of $\mathsf{elim}(\phi)$, allowing us to replace the existentially quantified formula (1) with the following finite disjunction:

$$\exists x.\phi \quad \Longleftrightarrow \quad \phi[\![x \mathbin{/\!/} \lfloor a\rfloor + \tfrac{1}{3}]\!] \vee \phi[\![x \mathbin{/\!/} -\infty]\!] \vee \phi[\![x \mathbin{/\!/} \mathbb{Z}]\!] \vee \phi[\![x \mathbin{/\!/} \mathbb{Z} + \varepsilon]\!] \tag{2}$$

[1] Proofs and additional illustrating examples can be found in the extended version [24] of this paper.

where $\phi[\![x \mathbin{/\!\!/} t]\!]$ denotes the formula obtained from ϕ by virtually substituting x with t. Note that the elements of $\mathsf{elim}(\phi)$ used for substituting x are not just regular terms, but so-called *virtual terms* that include additional symbols: ε for infinitesimal quantities, ∞ for infinity and $\mathbb{Z}$ for periodically repeating solutions. While ε and ∞ are also used [21], the periodic solutions $\mathbb{Z}$-terms are both unique and crucial for Viras. We immediately eliminate the symbols $\varepsilon, \infty, \mathbb{Z}$ in virtual substitutions, so that result of $\phi[\![x \mathbin{/\!\!/} t]\!]$ is a formula using the original signature of ϕ. Concretely, $t + \varepsilon$ is eliminated by replacing $\phi[\![x \mathbin{/\!\!/} t + \varepsilon]\!]$ by its limit value $\lim_{x \to t^+} \phi$; ∞ is eliminated via substituting with a constant greater than any other term, and $t + p\mathbb{Z}$ is eliminated by choosing a sufficient subset $\mathsf{fin}^{\phi}_{t+p\mathbb{Z}}$ of $\{t + pz \mid z \in \mathbb{Z}\}$ for substituting. In the case of this example a sufficient subset $\mathsf{fin}^{\phi}_{\mathbb{Z}}$ of $\mathbb{Z}$ is $\{\lfloor a \rfloor + 1\}$, the integer closest to the lower bound $L_1 = \lfloor a \rfloor + \frac{1}{3} \leq x$. As such, we apply virtual substitution:

$$
\begin{aligned}
\phi[\![x \mathbin{/\!\!/} \lfloor a \rfloor + \tfrac{1}{3}]\!] = \;& \underbrace{\lfloor a \rfloor + \tfrac{1}{3} \leq \lfloor a \rfloor + \tfrac{1}{3}}_{\top} \wedge \underbrace{\lfloor a \rfloor + \tfrac{1}{3} \leq \lfloor a \rfloor + \tfrac{2}{3}}_{\top} \wedge \underbrace{\lceil \lfloor a \rfloor + \tfrac{1}{3} \rceil - \lfloor a \rfloor - \tfrac{1}{3} \geq c}_{\frac{2}{3} \geq c} \\
\phi[\![x \mathbin{/\!\!/} -\infty]\!] = \;& \bot \wedge (L_2 \wedge L_3)[\![x \mathbin{/\!\!/} -\infty]\!] \\
\phi[\![x \mathbin{/\!\!/} \mathbb{Z}]\!] = \;& \textstyle\bigvee_{t \in \mathsf{fin}^{\phi}_{\mathbb{Z}}} \phi[\![x \mathbin{/\!\!/} t]\!] = \phi[\![x \mathbin{/\!\!/} \lfloor a \rfloor + 1]\!] = \underbrace{\lfloor a \rfloor + 1 \leq \lfloor a \rfloor + \tfrac{2}{3}}_{\bot} \wedge (L_1 \wedge L_3)[\![x \mathbin{/\!\!/} \lfloor a \rfloor + 1]\!] \\
\phi[\![x \mathbin{/\!\!/} \mathbb{Z} + \varepsilon]\!] = \;& \textstyle\bigvee_{t \in \mathsf{fin}^{\phi}_{\mathbb{Z}+\varepsilon}} \phi[\![x \mathbin{/\!\!/} t]\!] = \phi[\![x \mathbin{/\!\!/} \lfloor a \rfloor + 1 + \varepsilon]\!] \\
=\;& \lfloor a \rfloor + 1 + \varepsilon \leq \lfloor a \rfloor + \tfrac{2}{3} \wedge (L_1 \wedge L_3)[\![x \mathbin{/\!\!/} \lfloor a \rfloor + 1 + \varepsilon]\!] \\
=\;& \underbrace{\lfloor a \rfloor + 1 < \lfloor a \rfloor + \tfrac{2}{3}}_{\bot} \wedge (L_1 \wedge L_3)[\![x \mathbin{/\!\!/} \lfloor a \rfloor + 1 + \varepsilon]\!]
\end{aligned}
$$

allowing us to reduce (2) to $c \leq \frac{2}{3}$ as the quantifier-free equivalent of the Lira formula (1).

3 Preliminaries

We assume familiarity with multi-sorted first-order logic and respectively denote rationals, integers and reals by $\mathbb{Q}, \mathbb{Z}$ and $\mathbb{R}$. We consider the mixed first-order theory of linear integer and real arithmetic (Lira), corresponding to first-order logic with predicate symbols $<, \leq, \geq, >, \approx$; function symbols $+$, $q\cdot$ for $q \in \mathbb{Q}$ and $\lfloor \cdot \rfloor$; and a constant symbol 1, interpreted over $\mathbb{R}$. The function symbols $q\cdot$ are called *numerals*. A term $q \cdot (t)$ is *interpreted* as the term t multiplied by q. For simplicity, we omit parenthesis and $\cdot$ whenever it is clear from context; for example, write $3t$ for $3 \cdot (t)$. We write k for $k \cdot (1)$, $+t$ for $1t$ and $-t$ for $-1t$. By $\approx$ we denote the *equality predicate*. We write $l \not\approx r$ for $\neg(l \approx r)$. The *floor function* $\lfloor \cdot \rfloor$ applied to a term t returns the greatest integer less than or equal to t; hence, $\lfloor t \rfloor \leq t$. The *ceiling function* can be defined as $\lceil x \rceil = -\lfloor -x \rfloor$. While Lira theory does not contain a dedicated sort of integers, it handles integer properties via the $\lfloor \cdot \rfloor$ function. Linear real arithmetic (Lra) is an instance of Lira, without the floor function $\lfloor \cdot \rfloor$, interpreted over the reals $\mathbb{R}$. Linear integer arithmetic (Lia), also known as Presburger arithmetic, restricts Lra to the integer numerals of $\mathbb{Z}$ and is interpreted over $\mathbb{Z}$ instead of $\mathbb{R}$.

Let $\mathbf{V}$ and $\mathbf{T}$ denote respectively the set of Lira variables and terms. We write a, b, c, x, y, z for variables; s, t, u for terms; j, k, q, p for numerals; L for literals; ϕ, ψ for formulas, all possibly with indices. We denote by $\pm$ a symbol in $\{+, -\}$ and write $\mp$ for the respective other symbol; $\diamond$ for a predicate in $\{\approx, \not\approx, >, \geq\}$; and $\gtrsim$ for a predicate in $\{>, \geq\}$. Note that all variables that are not explicitly quantified are considered parameters (i.e., implicitly universally quantified). We write $s \trianglelefteq t$ for s being a subterm of t and $s \triangleleft t$ for s being a strict subterm of t. An expression E is a term, literal or formula. We write $E[x/t]$ for the result of substituting x

by t in E; whenever it is clear from context, we write $E[t]$ for $E[x/t]$. For a formula ϕ we write $\forall\phi$ and $\exists\phi$ for the universal and existential closure of ϕ. For a set E, we write $\bigwedge E$ for $\bigwedge_{e\in E} e$; similarly for $\bigvee, \bigcap, \bigcup$ and Σ. For a formula ϕ and a first-order interpretation $\mathcal{I}$, the set $\mathsf{solSet}^{\phi}_{x,\mathcal{I}} = \{x \in \mathbb{R} \mid \mathcal{I} \models \phi[x]\}$ is the *solution set* of ϕ with respect to $\mathcal{I}$ and x. If ϕ is a conjunction of literals we write $L \in \phi$ to denote that L is a literal of ϕ.

We consider rational numbers $\frac{j}{k} \in \mathbb{Q}$ to be *normalized* such that the greatest common divisor of j, k satisfies $\gcd(j,k) = 1$. Let $\mathsf{den}(\frac{j}{k}) = k$ denote the *denominator* and $\mathsf{num}(\frac{j}{k}) = j$ the *numerator* of $\frac{j}{k}$. By $\mathsf{sgn}(q)$ we denote the *sign* of the number q with $\mathsf{sgn}(q) \in \{0, -, +\}$. We respectively introduce a generalized *quotient* function and *remainder* function as $\mathsf{quot}_p(t) = \lfloor \frac{t}{p} \rfloor$ and $\mathsf{rem}_p(t) = t - p \cdot \mathsf{quot}_p(t)$, both defined over $\mathbb{R}$. *Divisibility constraints* are expressed in Lira as $q \mid t \iff \mathsf{rem}_q(t) \approx 0$, whereas congruence classes are defined as $s \equiv_q t \iff \mathsf{rem}_q(s) \approx \mathsf{rem}_q(t)$. We generalize least common multipliers *lcm* to be used with arbitrary finite sets of rationals $Q \subset \mathbb{Q}$ with $0 \notin Q$, as follows: $\mathsf{lcm}^{\mathbb{Q}}(Q) = \frac{lcm\{\mathsf{num}(q) \mid q \in Q\}}{gcd\{\mathsf{den}(q) \mid q \in Q\}}$. Clearly, for all $q \in Q$, we have $\frac{\mathsf{lcm}^{\mathbb{Q}}(Q)}{q} \in \mathbb{Z}$. We use $\langle\!|$ and $|\!\rangle$ as variables for interval bounds: $\langle\!|$ is either $[$ or $($; and $|\!\rangle$ is $]$ or $)$. For example the interval $\langle\!|l, r]$ could either be $(l, r]$ or $[l, r]$, depending on $\langle\!|$.

4 VIRAS: Virtual Integer Real Arithmetic Substitution

We now introduce the Viras method that performs quantifier elimination (QE) on Lira formulas, by implementing virtual substitutions over integer-real arithmetic. Given a quantified formula $\exists x.\phi$, Viras translates $\exists x.\phi$ into an equivalent quantifier-free formula ϕ', which in the case of a formula where all variables in ϕ are bound means ϕ' is ground and thus can be simply be evaluated (and solved). As we can perform quantifier elimination recursively, universal quantifiers can be expressed in terms of existential ones, and existential quantifiers can be distributed over disjunctions, in the sequel we consider arbitrarily fixed $\exists x.\phi$ formula, where ϕ is a conjunction of literals, that may contain free variables that are considered parameters (i.e., implicitly universally quantified).

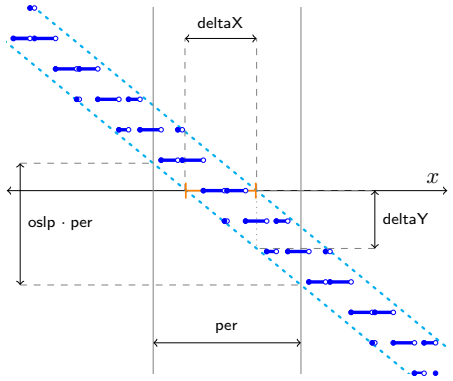
Following the setting of virtual substitutions [13,25], Viras computes a finite but sufficient number of witnesses for $\exists x$ and turns the quantified formula $\exists x.\phi$ into an equaivalent finite disjunction

$$\bigvee_{t \in \mathsf{elim}^x(\phi)} \phi[\![x /\!/ t]\!],$$

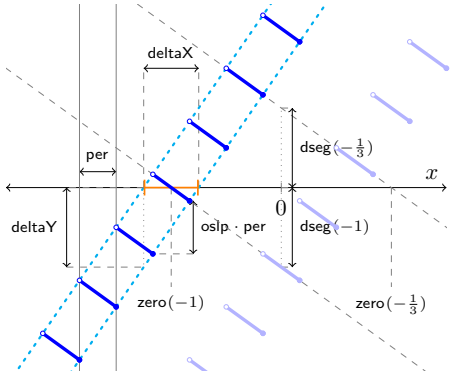
where $\phi[\![x /\!/ t]\!]$ is obtained from ϕ by *virtually substituting* x with the *virtual term* t that does not contain x, and $\mathsf{elim}^x(\phi)$ is the *elimination set* of ϕ.

The core idea for finding finite sets of witnesses $\mathsf{elim}^x(\phi)$ is that every Lira-literal $L \in \phi$ defines a set of solution intervals. Thus, if L holds for some x, then x must be contained in some solution interval S of L; hence, L must also hold for the lower bound of S. As ϕ is a conjunction of such literals, $\exists x.\phi[x]$ holds iff ϕ holds for any of the lower bounds of its literals. Therefore, we can choose the set of all lower bounds of all solution intervals as the elimination set $\mathsf{elim}^x(\phi)$.

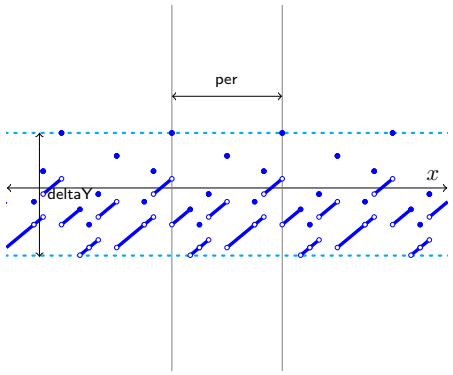
For finding the lower bounds of these solution intervals we introduce key properties of Lira terms and literals in Sect. 4.1. As seen in our motivating example in Sect. 2, solution intervals are not only left-closed (e.g., $[l, r]$) but may also be left-open (e.g., $(l, r]$, $(-\infty, r]$), and may be periodically repeating (e.g., $\cup_{z\in\mathbb{Z}}[l + 2z, r + 2z]$). Thus, we do not only substitute with regular terms, but also with virtual terms in order to include lower bounds such as $l + \varepsilon$, $-\infty$ and $l + 2\mathbb{Z}$. We formally define virtual terms and virtual substitutions in Sect. 4.2. Finally,



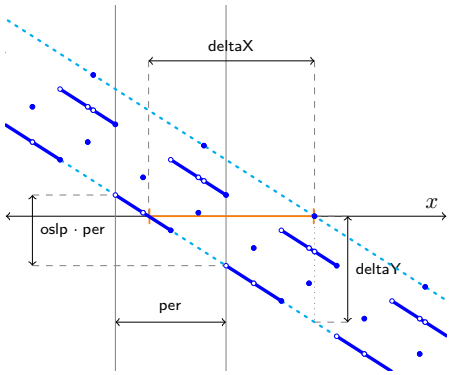
(a) $\lfloor 3x \rfloor - \lfloor 7x + 0.1234 \rfloor$



(b) $t = -\lfloor -3x + z \rfloor - x$ for $z = -1.5$. In pale blue we see t for $z = 1.5$



(c) $x + 2.7 - \lfloor x \rfloor + \frac{1}{2}\lfloor x + \frac{1}{3} \rfloor + \lfloor -\frac{1}{2}x - \frac{1}{3} \rfloor + \frac{1}{2}(\mathsf{dots}_{\frac{1}{2}}(x) + \mathsf{dots}_{\frac{1}{3}}(x)) + 2 \cdot \mathsf{dots}_0(x)$



(d) $-x + 0.8 + \lfloor -x \rfloor - \lfloor -x + \frac{1}{2} \rfloor + \frac{1}{2}(\mathsf{dots}_{\frac{3}{4}}(x) - \mathsf{dots}_{\frac{1}{4}}(x) + \mathsf{dots}_{\frac{1}{5}}(x))$

Figure 1: LIRA terms interpreted as functions in x, with $\mathsf{dots}_t(x) = \lfloor x + t \rfloor + \lfloor -x - t \rfloor$. The function graph is drawn thick and in blue, the function's linear bounds are given by the cyan dashed line, the core interval (Def. 7) is visualized in orange and marked with deltaX.

in Sect. 4.3 we combine the results about LIRA-terms and literals with the virtual substitution operation, allowing us to constructively define $\mathsf{elim}^x(\phi)$ and prove that VIRAS is a QE procedure for LIRA in Theorem 1.

4.1 LIRA Properties

Let us recall our motivating example from Sect. 2. We argued that the literal $L_3 = \lfloor x \rfloor - x \geq c$ of formula (1) has a periodic solution set of solutions $\bigcup_{z \in \mathbb{Z}} (z, z + 1 - c]$. The main idea of building our elimination sets is to cover all lower bounds of the intervals the solution set is composed of (i.e., $z + \varepsilon$ for every $z \in \mathbb{Z}$ in our example). As any LIRA-literal can be normalized to the form $t \diamond 0$ ($\diamond \in \{>, \geq, \approx, \not\approx\}$), we can characterize the lower bounds of solution sets by finding the zero crossings of LIRA-terms t. For finiding these, we introduce relevant properties of LIRA terms and literals.

LIRA Terms. We first illustrate few LIRA terms in Fig. 1, where terms are interpreted as functions in x. Note that each function in Fig. 1 is non-linear and not continuous. Nevertheless, the LIRA terms of Fig. 1 have a linear upper and lower bound with the same slope (Lem. 2), which we call the *outer slope* oslp of terms (Def. 1). The lower and the upper bound distances from the origin are distY^- and distY^+, and their difference is $\mathsf{deltaY} \in \mathbb{Q}^{\geq 0}$ (Def. 2). Even though there is an infinite number of discontinuities in Fig. 1, the function graphs witness periodic repetition, parallely shifted along upper and lower bounds (Lem. 1). The *period* per of a LIRA term refers to the size of the repeating interval of its respective function graph. Fig. 1 also shows that, between each two discontinuities, the function is composed of linear segments (Lem. 4) with the same slope; we refer to these as *segment slopes* sslp (Def. 1). The line segment above any x-value can be described as a linear function (visualized via the thin gray dashed lines in Fig. 1.b) passing through each segment that starts at the term's limit lim_t^x (Def. 3) and is shifted by the distance $\mathsf{dseg}(x)$ (Def. 4) from the origin; thus, the line describing the segment above some value x_0 is given by $\mathsf{sslp} \cdot x + \mathsf{dseg}(x_0)$. It is easy to see that the truth value of a literal $t \diamond 0$ can only change at a discontinuity b or at the zero of some segment $\mathsf{zero}_t(b)$ (Def. 4). Therefore, only these values can be lower bounds of solution intervals of LIRA-literals, defining thus our elimination set (Fig. 2).

Most of the following definitions formalizing these observations will use term and variable subscripts, or superscripts (e.g., per_t^x). We will omit these for the term symbol t and the variable symbol x.

Definition 1 (Slope and Period)**.** *Let t be a* LIRA *term. By recursion on t, we define the* ***period*** per_t^x*,* ***outer slope*** oslp_t^x*, and* ***segment slope*** sslp_t^x *of t as:*

$$\mathsf{oslp}_y^x = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases} \qquad \begin{array}{ll} \mathsf{oslp}_1^x = 0 & \mathsf{oslp}_{s+t}^x = \mathsf{oslp}_s^x + \mathsf{oslp}_t^x \\ \mathsf{oslp}_{kt}^x = k \cdot \mathsf{oslp}_t^x & \mathsf{oslp}_{\lfloor t \rfloor}^x = \mathsf{oslp}_t^x \end{array}$$

$$\mathsf{sslp}_y^x = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases} \qquad \begin{array}{ll} \mathsf{sslp}_1^x = 0 & \mathsf{sslp}_{s+t}^x = \mathsf{sslp}_s^x + \mathsf{sslp}_t^x \\ \mathsf{sslp}_{kt}^x = k \cdot \mathsf{sslp}_t^x & \mathsf{sslp}_{\lfloor t \rfloor}^x = 0 \end{array}$$

$$\mathsf{per}_y^x = \mathsf{per}_1^x = 0 \qquad \mathsf{per}_{kt}^x = \mathsf{per}_t^x$$

$$\mathsf{per}_{s+t}^x = \begin{cases} \mathsf{per}_s^x & \text{if } \mathsf{per}_t^x = 0 \\ \mathsf{per}_t^x & \text{if } \mathsf{per}_s^x = 0 \\ \mathsf{lcm}^{\mathbb{Q}}\{\mathsf{per}_s^x, \mathsf{per}_t^x\} & \text{otherwise} \end{cases} \qquad \mathsf{per}_{\lfloor t \rfloor}^x = \begin{cases} 0 & \text{if } \mathsf{per}_t^x = 0 = \mathsf{oslp}_t^x \\ \frac{1}{|\mathsf{oslp}_t^x|} & \text{if } \mathsf{per}_t^x = 0 \neq \mathsf{oslp}_t^x \\ \mathsf{num}(\mathsf{per}_t^x) \cdot \mathsf{den}(\mathsf{oslp}_t^x) & \text{otherwise} \end{cases}$$

Lemma 1 (Periodic Shift)**.** *If* $\mathsf{per}_t \neq 0$ *then* $\mathbb{R} \models \forall x, y. \big(t[x + \mathsf{per}\, \lfloor y \rfloor\,] \approx t[x] + \mathsf{oslp} \cdot \mathsf{per}\, \lfloor y \rfloor\big)$

Example 1. *Consider the term* $t = -\lfloor -3x + z \rfloor - x$ *of Fig. 1.b. We have* $\mathsf{oslp} = 2$, $\mathsf{sslp} = -1$ *and* $\mathsf{per} = \frac{1}{3}$*. By increasing the value of x by* $\mathsf{per}\, \lfloor y \rfloor$*, the value of* $t[x]$ *increases by* $\mathsf{oslp} \cdot \mathsf{per}\, \lfloor y \rfloor$*, that is:*

$$\begin{aligned} t[x + \underbrace{\mathsf{per}}_{\frac{1}{3}}\, \lfloor y \rfloor] &\approx -\lfloor -3(x + \tfrac{1}{3} \lfloor y \rfloor) + z \rfloor - x - \tfrac{1}{3} \lfloor y \rfloor \\ &\approx -\lfloor -3x + z \rfloor + \lfloor y \rfloor - x - \tfrac{1}{3} \lfloor y \rfloor \approx t[x] + \underbrace{\mathsf{per} \cdot \mathsf{oslp}}_{\frac{2}{3}}\, \lfloor y \rfloor \end{aligned}$$

Definition 2 (Bound Distance)**.** *Let t be a* LIRA*-term. We define* $\mathsf{distY}_{x,t}^{\pm} \in \mathbf{T}$ *and* $\mathsf{deltaY}_{x,t} \in \mathbb{Q}$ *by recursion on t:*

$$\begin{aligned}
\mathsf{deltaY}_{x,y} &= 0 \\
\mathsf{deltaY}_{x,1} &= 0 \\
\mathsf{deltaY}_{x,kt} &= |k|\mathsf{deltaY}_{x,t} \\
\mathsf{deltaY}_{x,s+t} &= \mathsf{deltaY}_{x,s} + \mathsf{deltaY}_{x,t} \\
\mathsf{deltaY}_{x,\lfloor t \rfloor} &= \mathsf{deltaY}_{x,t} + 1 \\
\\
\mathsf{distY}^{+}_{x,t} &= \mathsf{distY}^{-}_{x,t} + \mathsf{deltaY}_{x,t}
\end{aligned}
\qquad
\begin{aligned}
\mathsf{distY}^{-}_{x,y} &= \begin{cases} 0 & \text{if } x = y \\ y & \text{otherwise} \end{cases} \\
\mathsf{distY}^{-}_{x,1} &= 1 \\
\mathsf{distY}^{-}_{x,kt} &= \begin{cases} k \cdot \mathsf{distY}^{-}_{x,t} & \text{if } k \geq 0 \\ k \cdot \mathsf{distY}^{+}_{x,t} & \text{if } k < 0 \end{cases} \\
\mathsf{distY}^{-}_{x,s+t} &= \mathsf{distY}^{-}_{x,s} + \mathsf{distY}^{-}_{x,t} \\
\mathsf{distY}^{-}_{x,\lfloor t \rfloor} &= \mathsf{distY}^{-}_{x,t} - 1
\end{aligned}$$

While the bounds $\mathsf{distY}^{\pm}$ are over-approximations of actual bounds[2], they yield linear bounds with same outer slopes.

Lemma 2 (Linear Bounds). $\mathbb{R} \models \forall x. \Big(\mathsf{oslp} \cdot x + \mathsf{distY}^{-} \leq t \leq \mathsf{oslp} \cdot x + \mathsf{distY}^{+}\Big)$.

Example 2. *Recall term* $t = -\lfloor -3x + z \rfloor - x$ *from Ex. 1, with* $\mathsf{oslp} = 2$. *We have* $\mathsf{distY}^{-} = -z$ *and* $\mathsf{deltaY} = 1$, *which implies that* $2x - z \leq -\lfloor -3x + z \rfloor - x \leq 2x - z + 1$.

We next express that a function defined by a LIRA term is composed from the linear segments between two discontinuities. Therefore, we compute the upper limit lim^x_t of a LIRA term t (Def. 3) and derive each segment's distance to origin (Def. 4).

Definition 3 (Limit). *The* ***limit term*** lim^x_t *of a* LIRA*-term* t *wrt* x *is defined by recursion on* t, *as:*

$$\begin{aligned}
\mathsf{lim}^x_y &= y \\
\mathsf{lim}^x_1 &= 1 \\
\mathsf{lim}^x_{kt} &= k \cdot \mathsf{lim}^x_t
\end{aligned}
\qquad
\begin{aligned}
\mathsf{lim}^x_{s+t} &= \mathsf{lim}^x_s + \mathsf{lim}^x_t \\
\mathsf{lim}^x_{\lfloor t \rfloor} &= \begin{cases} \lfloor \mathsf{lim}^x_t \rfloor & \text{if } \mathsf{sslp}^x_t \geq 0 \\ \lceil \mathsf{lim}^x_t \rceil - 1 & \text{if } \mathsf{sslp}^x_t < 0 \end{cases}
\end{aligned}$$

We write lim_t *for* lim^x_t *if* x *is clear in the context.*

Example 3. *The term* $t = -\lfloor -3x + z \rfloor - x$ *of Fig. 1.b has* $\mathsf{lim}_t = \lfloor 3x - z \rfloor + 1 - x$.

Definition 4 (Segment Line). *The* ***segment distance*** $\mathsf{dseg}_t(x_0)$ *of a* LIRA*-term* t *at* x_0 *is:*

$$\mathsf{dseg}^x_t(x_0) = -\mathsf{sslp}^x_t \cdot x_0 + \mathsf{lim}^x_t[x_0] \qquad \mathsf{zero}^x_t(x_0) = x_0 - \frac{\mathsf{lim}^x_t[x_0]}{\mathsf{sslp}_t}$$

The ***segment line*** *of* t *at* x_0 *is* $\mathsf{sslp}_t \cdot x + \mathsf{dseg}_t(x_0)$, *whereas* $\mathsf{zero}_t(x_0)$ *is the* ***zero of the segment*** *of* t *at* x_0.

Example 4. *In* t *from Fig. 1.b, we have* $\mathsf{dseg}(b) = \lfloor 3b - z \rfloor + 1$. *Hence,* $\mathsf{dseg}(-1/3) = \lfloor -z \rfloor$, $\mathsf{dseg}(-1) = \lfloor -z \rfloor - 2$, *and* $\mathsf{zero}(b) = \mathsf{dseg}(b)$. *The dotted lines of Fig. 1).b show* $\mathsf{sslp} \cdot x + \mathsf{dseg}(-1/3)$ *and* $\mathsf{sslp} \cdot x + \mathsf{dseg}(-1)$, *with the corresponding zeros* $\mathsf{zero}(-1/3)$ *and* $\mathsf{zero}(-1)$. *The lines' distances to the origin are* $\mathsf{dseg}(-1/3)$ *and* $\mathsf{dseg}(-1)$.

We next introduce the set breaks^{∞} of discontinuities, which is infinite but periodically repeating. We therefore specify finite sets breaks of terms with a formal parameter $\mathbb{Z}$, capturing that if $t + p\mathbb{Z} \in \mathsf{breaks}$ then $\{t + pz \mid z \in \mathbb{Z}\} \subseteq \mathsf{breaks}^{\infty}$.

Example 5. *For* $-\lfloor -3x + z \rfloor - x$ *in Fig. 1.b, we have* $\mathsf{breaks}^{\infty} = \{\frac{z}{3} + \frac{i}{3} \mid i \in \mathbb{Z}\}$ *and* $\mathsf{breaks}_t = \{\frac{z}{3} + \frac{1}{3}\mathbb{Z}\}$.

[2] Ex. 14 in Appendix A.1 in [24] shows that finding tight(er) bounds is very expensive

To define breaks^∞, we compute the intersection of infinite sets defined by $t + p\mathbb{Z}$ with constant-sized intervals $(\!|l, l+q|\!)$ $(l, t \in \mathbf{T}, q, p \in \mathbb{Q}^{\geq 0})$, using *grid instersections*.

Definition 5 (Grid Intersection)**.** *For $s, t \in \mathbf{T}$ and $p, k \in \mathbb{Q}^{>0}$, the **grid intersection** is*

$$(s + p\mathbb{Z}) \sqcap (\!|t, t+k|\!) = \{start_{(\!|} + np \mid n \in \mathbb{N}, np \lessdot_{|\!)} k\}$$

where

$$\lceil t \rceil^{s+p\mathbb{Z}} = t + \mathsf{rem}_p(s - t) \qquad \lceil t + \varepsilon \rceil^{s+p\mathbb{Z}} = \lfloor t + p \rfloor^{s+p\mathbb{Z}} \qquad start_{[\!|} = \lceil t \rceil^{s+p\mathbb{Z}} \qquad \lessdot_{]} = \leq$$
$$\lfloor t \rfloor^{s+p\mathbb{Z}} = t - \mathsf{rem}_p(t - s) \qquad \lfloor t - \varepsilon \rfloor^{s+p\mathbb{Z}} = \lceil t - p \rceil^{s+p\mathbb{Z}} \qquad start_{(\!|} = \lceil t + \varepsilon \rceil^{s+p\mathbb{Z}} \qquad \lessdot_{)} = <$$

Intuitively, a term $t + p\mathbb{Z}$ is a grid that starts at value t and repeats with period p. The operation $\sqcap$ intersects this grid with an interval, whereas the operations $\lfloor s \rfloor^{t+p\mathbb{Z}}$ and $\lceil s \rceil^{t+p\mathbb{Z}}$ are rounding the grid value next to s. We thus have the following result.

Lemma 3 (Grid Intersection)**.** $(s + p\mathbb{Z}) \sqcap (\!|t, t+k|\!) \supseteq (\{s + pz \mid z \in \mathbb{Z}\} \cap (\!|t, t+k|\!))$

Example 6. *Consider the interval $[a, a+4)$ and the grid $1 + 2\mathbb{Z}$. As*

$I = (1 + 2\mathbb{Z}) \sqcap [a, a+4) = \{\lceil a \rceil^{1+2\mathbb{Z}} + i \mid i \in \{0, 2\}\} = \{a + \mathsf{rem}_2(1 - a) + i \mid i \in \{0, 2\}\}$,
we obtain $I = \{1 - 2\left\lfloor \frac{1-a}{2} \right\rfloor, 3 - 2\left\lfloor \frac{1-a}{2} \right\rfloor\}$. Hence, the values in I are in $G = \{1 + 2z \mid z \in \mathbb{Z}\}$. Further, since $\mathsf{rem}_2(1 - a) \in [0, 2)$ yields that $\lceil a \rceil^{1+2\mathbb{Z}} = a + \mathsf{rem}_2(1 - a)$ is the smallest value in $G \cap [a, a+4)$, which means $I \subseteq G \cap [a, a+4)$.

We have now all ingredients to define the set breaks of discontinuities, using over-approximation as for linear bounds (Lem. 2).

Definition 6. *The set of **discontinuities** breaks^x_t of a* Lira*-term t wrt variable x is defined by recursion on t, as:*

$$\mathsf{breaks}^x_y = \mathsf{breaks}^x_1 = \emptyset \qquad \mathsf{breaks}^x_{kt} = \mathsf{breaks}^x_t \qquad \mathsf{breaks}^x_{s+t} = \mathsf{breaks}^x_s \cup \mathsf{breaks}^x_t$$

$$\mathsf{breaks}^x_{\lfloor t \rfloor} = \begin{cases} \mathsf{breaks}^x_t & \text{if } \mathsf{sslp}_t = 0 \\ \{\mathsf{zero}_t(0) + \mathsf{per}_{\lfloor t \rfloor}\mathbb{Z}\} & \text{if } \mathsf{breaks}^x_t = \emptyset \;\&\; \mathsf{sslp}_t \neq 0 \\ \mathsf{breaks}^x_t \cup \mathsf{breaksInSeg}^x_t & \text{if } \mathsf{breaks}^x_t \neq \emptyset \;\&\; \mathsf{sslp}_t \neq 0 \end{cases}$$

$$\mathsf{breaksInSeg}^x_t = \left\{ b + \mathsf{per}_{\lfloor t \rfloor}\mathbb{Z} \;\middle|\; \begin{array}{ll} b \in (\mathsf{zero}(b_0) + \frac{1}{\mathsf{sslp}_t}\mathbb{Z}) \sqcap [b_0, b_0 + p^{\min}_t) & \text{where} \\ b_0 \in (b'_0 + p\mathbb{Z}) \sqcap [b'_0, b'_0 + \mathsf{per}_{\lfloor t \rfloor}) & \text{where} \\ b'_0 + p\mathbb{Z} \in \mathsf{breaks}_t & \end{array} \right\}$$

$$p^{\min}_t = \min\{p \mid b + p\mathbb{Z} \in \mathsf{breaks}^x_t\}$$
$$\mathsf{breaks}^{x,\infty}_t = \{t + pz \mid z \in \mathbb{Z}, t + p\mathbb{Z} \in \mathsf{breaks}^x_t\}$$

The piecewise linearity of functions defined by Lira terms is then expressed as: between any two neighbouring breaks b^+ and b^-, the term t is described by a linear function $\mathsf{sslp} \cdot x + \mathsf{dseg}(b^-)$.

Lemma 4 (Piecewise Linearity)**.** *Let $\mathcal{I}$ be an $\mathbb{R}$-interpretation, $x \in \mathbf{V}$ and t a* Lira*-term such that $\mathsf{breaks} \neq \emptyset$ and $b^- \in \mathsf{breaks}^\infty$. Let $b^+ = \min\{b \mid b \in \mathsf{breaks}^\infty, \mathcal{I} \models b > b^-\}$, and $\pm \in \{+, -\}$. Then*

$$\mathcal{I} \models \forall x \in (b^-, b^+), y \in [b^-, b^+).\, (t[x] \approx \mathsf{lim}_t[x] \approx \mathsf{sslp} \cdot x + \mathsf{dseg}(y))\,.$$

LIRA Literals. Let us now introduce some key properties of LIRA literals that will allow us to specify finite elimination sets. We assume LIRA literals to be normalized to $t \diamond 0$, and distinguish two kinds of LIRA-literals: a LIRA literal is *periodic* if $\mathsf{oslp}_t = 0$, and *aperiodic* otherwise. Solution sets of periodic literals repeat periodically, which allows us to finitely specify the lower bounds of their solution sets using $\mathbb{Z}$-terms (formally defined in Sect. 4.2). On the other hand, aperiodic literals only have a finite number of solution intervals that can be found using the bounds of their so-called core intervals (Def. 7).

Fig. 1.b shows that the solutions of periodic literals repeat in a periodic manner:

Lemma 5 (Periodic Literals). *If $L = t \diamond 0$ is a periodic* LIRA*-literal ($\mathsf{oslp}_t = 0$), then*
$$\mathbb{R} \models \forall y.\big(L[x] \leftrightarrow L[x + \mathsf{per}_t \lfloor y \rfloor]\big)$$

Example 7. *Consider the literal $L_3 = t \geq 0$, with $t = \lceil x \rceil - x - c$, $\mathsf{oslp}_t = 0$ and $\mathsf{per}_t = 1$ from the motivating example of Sect. 2. We have $t[x + \lfloor s \rfloor] \approx t[x]$ for any s. Hence, $L_3[x + \lfloor s \rfloor] \leftrightarrow L_3[x]$.*

The truth values of aperiodic literals do not repeat. Instead they have a constant limit value $\mathsf{lim}^{\pm\infty}$ and a so-called *core interval*.

Definition 7 (Core Interval). *Let t be a* LIRA*-term with $\mathsf{oslp}_t \neq 0$. The **core interval** of t is $[\mathsf{distX}_t^-, \mathsf{distX}_t^+]$, where*
$$\mathsf{distX}_t^- = -\frac{\mathsf{distY}_t^{\mathsf{sgn}(\mathsf{oslp}_t)}}{\mathsf{oslp}_t} \qquad \mathsf{deltaX}_t = \frac{\mathsf{deltaY}_t}{|\mathsf{oslp}_t|} \qquad \mathsf{distX}_t^+ = \mathsf{distX}_t^- + \mathsf{deltaX}_t$$

Bounds of the core intervals are given by the zeros of the linear bounds from Lem. 2. Within a core interval, a LIRA literal may be evaluated to both true and false, while outside of the interval the literal's value is equal to the constant value $\mathsf{lim}_L^{\pm\infty} \in \{\top, \bot\}$, as next given.

Lemma 6 (Limit Value). *If $L = t \diamond 0$ is an aperiodic* LIRA*-literal ($\mathsf{oslp}_t \neq 0$), then the values outside of the core interval of t satisfy the following:*
$$\mathbb{R} \models \forall x < \mathsf{distX}_t^-.(L[x] \leftrightarrow \mathsf{lim}_L^{-\infty}) \qquad \mathbb{R} \models \forall x > \mathsf{distX}_t^+.(L[x] \leftrightarrow \mathsf{lim}_L^{+\infty})$$
where
$$\mathsf{lim}_{t \approx 0}^{\pm\infty} = \bot \qquad \mathsf{lim}_{t \not\approx 0}^{\pm\infty} = \top \qquad \mathsf{lim}_{t \gtrsim 0}^{\pm\infty} = \pm\mathsf{oslp} > 0$$

Example 8. *Consider again our term $t = -\lceil -3x + z \rceil - x$ and the literal $L = t > 0$. We have $\mathsf{distX}^- = \frac{z-1}{2}$, $\mathsf{deltaX} = \frac{1}{2}$, $\mathsf{lim}_L^{+\infty} = \top$ and $\mathsf{lim}_L^{-\infty} = \bot$. Therefore, L is $\bot$ for all values less than $\frac{z-1}{2}$ and $\top$ for all values greater than $\frac{z}{2}$.*

4.2 Virtual Substitutions in VIRAS

Recall that virtual substitutions do not replace variables by regular terms, but by *virtual terms* from an extended language. Formally, we have the following.

Definition 8 (Virtual Term). *A **virtual term** v is a sum $t + e\varepsilon + z\mathbb{Z} + i\infty$ with $t \in \mathbf{T}, e \in \{0, 1\}, z \in \mathbb{Q}^{\geq 0}, i \in \{0, +, -\}$, where $z = 0$ or $i = 0$. We may omit summands with zero coefficients. We write $\mathbb{Z}(v) = z$, $\varepsilon(v) = e$ and $\infty(v) = i$. A virtual term is **plain** if $e = z = i = 0$ and **proper** otherwise.*

The new symbols $\varepsilon, \mathbb{Z}, \infty$ do not occur in the result of applying virtual substitution. Instead, the *virtual substitution function* (Def. 9) eliminates these auxiliary symbols, as follows. As ε represents an infinitesimal quantity, we compute $L[\![x \mathbin{/\!\!/} s+\varepsilon]\!]$ by replacing it by $\lim_{x\to s^+} L$ (cases 4–5 of Def. 9). The summand ∞ represents an infinitely large constant that is divisible by every rational number. Thus we compute $\phi[\![x \mathbin{/\!\!/} t \pm \infty]\!]$ by replacing all aperiodic literals $A \in \phi$ by $\mathsf{lim}_A^{\pm\infty}$ and replacing periodic literals $P \in \phi$ by $P[\![x \mathbin{/\!\!/} t]\!]$ (case 3 of Def. 9).

Virtual terms $t + p\mathbb{Z}$ represent infinite sets of substitutions: $\phi[\![x \mathbin{/\!\!/} t + p\mathbb{Z}]\!]$ is true iff $\exists z \in \mathbb{Z}.\phi[\![x \mathbin{/\!\!/} t + pz]\!]$; hence, we compute a finite subset $\mathsf{fin}^{\phi}_{t+p\mathbb{Z}} \subset \{t + pz \mid z \in \mathbb{Z}\}$ such that $\exists z \in \mathbb{Z}.\phi[\![x \mathbin{/\!\!/} t + pz]\!] \leftrightarrow \bigvee_{t' \in \mathsf{fin}^{\phi}_{t+p\mathbb{Z}}} \phi[\![x \mathbin{/\!\!/} t']\!]$ (case 1 of Def. 9). Such a finite subset was given in Sect. 2, where $\mathsf{fin}^{\phi}_{\mathbb{Z}} = \{\lfloor a \rfloor + 1\}$, the smallest integer satisfying the lower bound $\lfloor a \rfloor + \frac{1}{3}$.

Definition 9 (Virtual Substitution)**.** *A* ***virtual substitution function*** $\circ[\![\circ \mathbin{/\!\!/} \circ]\!]$ *maps a conjunction of* LIRA*-literals, a variable, and a virtual term to a formula. We write* $\phi[\![t]\!]$ *for* $\phi[\![x \mathbin{/\!\!/} t]\!]$*. Let* ϕ *be a conjunction of* LIRA*-literals,* t *a term,* v *a virtual term with* $\mathbb{Z}(v) = 0$*,* $P = \{L \in \phi \mid L \text{ is periodic}\}$*, and* $A = \{L \in \phi \mid L \text{ is aperiodic}\}$*. Then,*

1. $\phi[\![x \mathbin{/\!\!/} t + e\varepsilon + p\mathbb{Z}]\!] = \bigvee_{t' \in \mathsf{fin}^{\phi}_{t+p\mathbb{Z}}} \phi[\![x \mathbin{/\!\!/} t' + e\varepsilon]\!]$ *where*

   V1. if $\forall L \in A.\mathsf{lim}_L^{\pm\infty} = \top$*:* $\quad \mathsf{fin}^{\phi}_{t+p\mathbb{Z}} = \{s \pm \infty \mid s \in \big(t + p\mathbb{Z} \sqcap [t, t+\lambda)\big)\}$

   V2. if $\exists L \in A.L = u \approx 0$*:* $\quad \mathsf{fin}^{\phi}_{t+p\mathbb{Z}} = \big(t + p\mathbb{Z} \sqcap [\mathsf{distX}^-_{u\approx 0}, \mathsf{distX}^+_{u\approx 0}]\big)$

   V3. otherwise: $\quad \mathsf{fin}^{\phi}_{t+p\mathbb{Z}} = \bigcup_{L\in A, \mathsf{lim}_L^{-\infty} = \bot} \big(t + p\mathbb{Z} \sqcap [\mathsf{distX}^-_L, \mathsf{distX}^+_L + \lambda]\big)$

   $\lambda = \mathsf{lcm}^{\mathbb{Q}}(\{p\} \cup \{\mathsf{per}_L \mid L \in P\})$

2. $(\bigwedge_{L\in\phi} L)[\![x \mathbin{/\!\!/} v]\!] = \bigwedge_{L\in\phi} (L[\![x \mathbin{/\!\!/} v]\!])$

3. $(s \diamond 0)[\![x \mathbin{/\!\!/} v \pm \infty]\!] = \begin{cases} \mathsf{lim}^{\pm\infty}_{s\diamond 0} & \text{if } s \diamond 0 \text{ is aperiodic } (\mathsf{oslp}_s = 0) \\ (s \diamond 0)[\![x \mathbin{/\!\!/} v]\!] & \text{if } s \diamond 0 \text{ is periodic } (\mathsf{oslp}_s \neq 0) \end{cases}$

4. $((\neg)s \approx 0)[\![x \mathbin{/\!\!/} t + \varepsilon]\!] = \begin{cases} (\neg)\bot & \text{if } \mathsf{sslp}_s \neq 0 \\ (\neg)\mathsf{lim}_s[t] \approx 0 & \text{if } \mathsf{sslp}_s = 0 \end{cases}$

5. $(s \gtrsim 0)[\![x \mathbin{/\!\!/} t + \varepsilon]\!] = \begin{cases} \mathsf{lim}_s[t] \geq 0 & \text{if } \mathsf{sslp}_s > 0 \\ \mathsf{lim}_s[t] \gtrsim 0 & \text{if } \mathsf{sslp}_s = 0 \\ \mathsf{lim}_s[t] > 0 & \text{if } \mathsf{sslp}_s < 0 \end{cases}$

6. $(s \diamond 0)[\![x \mathbin{/\!\!/} t]\!] = s[x/t] \diamond 0$

Note that for finding $\mathsf{fin}^{\phi}_{t+p\mathbb{Z}}$ in general (case 1 of Def. 9), we use periodic literals (Lem. 5) and core intervals (Lem. 6), as follows. Literals L with $\mathsf{lim}_L^{+\infty} = \top$ and $\mathsf{lim}_L^{-\infty} = \bot$ can only be true from the beginning of the core interval $[\mathsf{distX}^-_L, \infty)$, thus we only need to instantiate with values in this interval for every such literal. Further in the interval $(\mathsf{distX}^+, \infty)$ the literal L will always be $\mathsf{lim}_L^{+\infty} = \top$, while the truth value of periodic literals will repeat with a period of λ. Thus if there is a solution in $(\mathsf{distX}^+, \infty)$, then there must be one in $(\mathsf{distX}^+, \mathsf{distX}^+ + \lambda]$. This means it is sufficient for $\mathsf{fin}^{\phi}_{t+p\mathbb{Z}}$ to contain all values in $[\mathsf{distX}^-_L, \mathsf{distX}^+_L + \lambda] \cap \{t + pz \mid z \in \mathbb{Z}\}$

for such L. This reasoning correponds to case (V3) of Def. 9 and illustrated in Ex. 15 in Appendix A.1 in [24]. The cases (V1), (V2) of Def. 9 handle formulas where there is no such literal L. The cases (V1), (V3) of Def. 9 generalize Cooper's method for LIA [9], as discussed in Sect. 5.

4.3 Quantifier Elimination via Elimination Sets

To find sufficient finite elimination sets, we proceed as follows. If there is some x such that a formula ϕ is true, then x is an element of some solution interval I of ϕ. Therefore, ϕ is true for the lower bound of I. Hence, if we take all terms that might be lower bounds of a solution interval of any literal of ϕ, we obtain an elimination set for ϕ. It is easy to see that $\exists x.\phi$ holds if there is a t such that $\phi[\![t]\!]$ holds; thus we may compute an over-approximation of the exact set of lower bounds.

Example 9. *In Sect. 2, the solution set of L_3 is $\bigcup_{z\in\mathbb{Z}}(z, z+1-c]$. Hence, we may derive an elimination set as $\{z+\varepsilon \mid z \in \mathbb{Z}\}$, which is finitely represented as $\{\mathbb{Z}+\varepsilon\}$.*

Definition 10 (Elimination Set)**.** *The **elimination set** $\mathsf{elim}^x(\phi)$ of a conjunction of literals ϕ with respect to the variable x is defined in Fig. 2.*

Let us make the following remarks upon Def. 10. If $\mathsf{breaks} = \emptyset$, we have a simple linear function; in this case, the lower bounds of the solution intervals can be computed as in LRA. For literals $t \diamond 0$ where $\mathsf{breaks}_t \neq \emptyset$, firstly notice that every discontinuity b of t can be the lower bound of a solution interval $[b, b]$. Therefore, we add ebreak to the elimination set. For periodic literals, ebreak is breaks_t a finite representation of the full infinite set of discontinuities, while for aperiodic literals we only add discontinuities within the core interval $(\mathsf{distX}^-, \mathsf{distX}^+)$. Between any two discontinuities, t can be described as segment of a linear function (Lem. 4). Therefore, we find the lower bounds eseg of the solution intervals of these segments using the zeros of the segments $\mathsf{zero}(b)$, as well as the discontinuities b bounding the segments. For periodic literals, we only consider all periodically repeating values; whereas for aperiodic literals consider those in the core interval. Both eseg and ebreak are limited to the core interval $(\mathsf{distX}^-, \mathsf{distX}^+)$ for aperiodic literals, thus we also need to cover the lower bounds $\mathsf{ebound}^{\pm}$ of solution sets outside of the core interval.

Based on our definition of elimination sets and virtual substitution, we obtain the following result, asserting that elim^x can be used to eliminate existential quantifiers.

Theorem 1 (Quantifier Elimination)**.** *Let ϕ be a non-empty conjunction of* LIRA*-literals.*

$$\mathbb{R} \models \exists x.\phi \leftrightarrow \bigvee_{t\in\mathsf{elim}(\phi)} \phi[\![t]\!]$$

Example 10. *Consider formula ϕ from Sect. 2, with*

$$\begin{aligned}\mathsf{elim}^x(\phi) &= \mathsf{elim}^x(\lfloor a\rfloor + \tfrac{1}{3} \le x) \wedge \mathsf{elim}^x(x \le \lfloor a\rfloor + \tfrac{2}{3}) \wedge \mathsf{elim}^x(\lfloor x\rfloor - x \ge c)\\ &= \mathsf{elim}^x(\underbrace{x - \lfloor a\rfloor - \tfrac{1}{3}}_{t_1} \ge 0) \wedge \mathsf{elim}^x(\underbrace{\lfloor a\rfloor + \tfrac{2}{3} - x}_{t_2} \ge 0) \wedge \mathsf{elim}^x(\underbrace{\lfloor x\rfloor - x - c}_{t_3} \ge 0)\end{aligned}$$

As $\mathsf{breaks}_{t_1} = \mathsf{breaks}_{t_2} = \emptyset$, we compute the elimination sets $\mathsf{elim}^x(t_1 \ge 0)$ and $\mathsf{elim}^x(t_2 \ge 0)$, resulting in $\mathsf{elim}^x(t_1 \ge 0) = \{-\lfloor a\rfloor - \frac{1}{3}\}$ and $\mathsf{elim}^x(t_2 \ge 0) = \{-\infty\}$.

For $\mathsf{elim}^x(t_3 \ge 0)$, we have $\mathsf{breaks}^x_{t_3} = \{\mathbb{Z}\}$. As $t_3 \ge 0$ is periodic ($\mathsf{oslp}_{t_3} = 0$), the elimination set $\mathsf{elim}^x(t_3 \ge 0)$ consists of all discontinuities $\mathsf{ebreak} = \mathsf{breaks} = \{\mathbb{Z}\}$ and eseg. The intuition of eseg is the least value t within two breaks $t \in (b^-, b+)$ for which $t_3 \ge 0$ can hold. As the slope of the segment is negative $\mathsf{sslp}_{t_3} = -1$, this value must be $b^- + \varepsilon$. Therefore, $\mathsf{eseg} = \{b+\varepsilon \mid b \in \mathsf{breaks}\} = \{\mathbb{Z}+\varepsilon\}$. Thus, we derive $\mathsf{elim}^x(\phi) = \{-\lfloor a\rfloor - \frac{1}{3}, -\infty, \mathbb{Z}, \mathbb{Z}+\varepsilon\}$.

for conjunctions of literals ϕ and ψ: $\mathsf{elim}^x(\phi \wedge \psi) = \mathsf{elim}^x(\phi) \cup \mathsf{elim}^x(\psi)$

if $\mathsf{breaks} = \emptyset$

$$\mathsf{elim}(t \diamond 0) = \{-\infty\} \qquad \text{if } \mathsf{sslp} = 0$$
$$\mathsf{elim}(t \not\approx 0) = \{-\infty, \mathsf{zero}_t(0) + \varepsilon\}$$
$$\mathsf{elim}(t \approx 0) = \{\mathsf{zero}_t(0)\}$$
$$\mathsf{elim}(t \gtrsim 0) = \begin{cases} \{\mathsf{zero}_t(0)\} & \text{if } \mathsf{sslp} > 0 \;\&\; \gtrsim = \geq \\ \{\mathsf{zero}_t(0) + \varepsilon\} & \text{if } \mathsf{sslp} > 0 \;\&\; \gtrsim = > \\ \{-\infty\} & \text{if } \mathsf{sslp} < 0 \end{cases}$$

if $\mathsf{breaks} \neq \emptyset$

$$\mathsf{elim}(t \diamond 0) = \begin{cases} \mathsf{ebreak} \cup \mathsf{eseg} & \text{if } t \diamond 0 \text{ is periodic} \\ \mathsf{ebreak} \cup \mathsf{eseg} \cup \mathsf{ebound}^+ \cup \mathsf{ebound}^- & \text{if } t \diamond 0 \text{ is aperiodic} \end{cases}$$

$$\mathsf{ebound}^+ = \begin{cases} \{\mathsf{distX}^+, \mathsf{distX}^+ + \varepsilon\} & \text{if } \mathsf{lim}^{+\infty} = \top \\ \{\mathsf{distX}^+\} & \text{if } \mathsf{lim}^{+\infty} = \bot \end{cases}$$

$$\mathsf{ebound}^- = \begin{cases} \{\mathsf{distX}^-, -\infty\} & \text{if } \mathsf{lim}^{-\infty} = \top \\ \{\mathsf{distX}^-\} & \text{if } \mathsf{lim}^{-\infty} = \bot \end{cases}$$

$$\mathsf{ebreak} = \begin{cases} \{\, b + p\mathbb{Z} \mid b + p\mathbb{Z} \in \mathsf{breaks}\} & \text{if } t \diamond 0 \text{ is periodic} \\ \bigcup\{(b + p\mathbb{Z}) \sqcap (\mathsf{distX}^-, \mathsf{distX}^+) \mid b + p\mathbb{Z} \in \mathsf{breaks}\} & \text{if } t \diamond 0 \text{ is aperiodic} \end{cases}$$

$$\mathsf{eseg} = \begin{cases} \{t + \varepsilon \mid t \in \mathsf{ebreak}\} & \text{if } \mathsf{sslp} = 0 \text{ or } \mathsf{sslp} < 0 \;\&\; \diamond \in \{>, \geq\} \\ \{t + \varepsilon \mid t \in \mathsf{ebreak}\} \cup \{t \mid t \in \mathsf{ezero}\} & \text{if } \mathsf{sslp} > 0 \;\&\; \diamond \in \{\geq\} \\ \{t + \varepsilon \mid t \in \mathsf{ebreak}\} \cup \{t + \varepsilon \mid t \in \mathsf{ezero}\} & \text{if } \mathsf{sslp} > 0 \;\&\; \diamond \in \{>\} \\ \{t + \varepsilon \mid t \in \mathsf{ebreak} \cup \mathsf{ezero}\} & \text{if } \mathsf{sslp} \neq 0 \;\&\; \diamond \in \{\not\approx\} \\ \mathsf{ezero} & \text{if } \mathsf{sslp} \neq 0 \;\&\; \diamond \in \{\approx\} \end{cases}$$

$$\mathsf{ezero} = \begin{cases} \{\mathsf{zero}(b) + p\mathbb{Z} \mid b + p\mathbb{Z} \in \mathsf{breaks}\} & \text{if } t \diamond 0 \text{ is periodic} \\ \{\mathsf{zero}(b) \mid b + p\mathbb{Z} \in \mathsf{breaks}\} & \text{if } t \diamond 0 \text{ is aperiodic } \&\; \mathsf{oslp} = \mathsf{sslp} \\ \bigcup \left\{ \begin{pmatrix} \mathsf{zero}(b) + (1 - \frac{\mathsf{oslp}}{\mathsf{sslp}})p\mathbb{Z} \\ \sqcap(\mathsf{distX}^-, \mathsf{distX}^+) \end{pmatrix} \middle| b + p\mathbb{Z} \in \mathsf{breaks} \right\} & \text{if } t \diamond 0 \text{ is aperiodic } \&\; \mathsf{oslp} \neq \mathsf{sslp} \end{cases}$$

Figure 2: Definition of the elimination set elim^x computed by Viras.

5 VIRAS and Related Methods

We discuss and highlight the main differences of Viras compared to the state-of-the-art algorithms in solving quantified linear arithmetic problems. In a nutshell, we generalize Cooper's method [9] for Lia to be used with Lira while allowing for additional optimization for equality literals. Further, virtual substitutions in Viras yield an exponential speed-up compared to the method for solving Lira described by [26]. As a result and thanks to its Lira reasoning, Viras solves problems that state-of-the-art SMT techniques [3, 10] fail to solve.

VIRAS Generalizations upon Cooper's Method. While Cooper's method [9] implements a QE procedure only for LIA, our VIRAS calculus solves full LIRA formulas[3]. Similarly to VIRAS splitting literals into periodic P and aperiodic literal A (Def. 9.1), Cooper's method splits a formula $\phi = \mathcal{L} \wedge \mathcal{U} \wedge \mathcal{D}$ into literals capturing lower bounds $\mathcal{L}$, upper bounds $\mathcal{U}$ and divisibility constraints $\mathcal{D}$. The solution to $\mathcal{D}$ are found by (i) computing λ, the *lcm* of all divisibility constraints, and (ii) instantiating $\mathcal{D}$ with one number for every congruence class modulo λ. For also solving $\mathcal{L} \wedge \mathcal{U}$, the formula ϕ is instantiated with $\{l, \ldots, l + \lambda - 1\}$ for every lower bound $x \geq l \in \mathcal{L}$. Generalizing Cooper's method to LIRA is however not straightforward, as bounds and equivalence classes over $\mathbb{R}$ differ from the ones over $\mathbb{Z}$. While Cooper's method requires that a solution to $\mathcal{D}$ is one of the congruence classes $\{0 \ldots \lambda - 1\}$, as proper real numbers ($\mathbb{R} \setminus \mathbb{Z}$) cannot be captured by these equivalence classes. In VIRAS, we therefore compute equivalence classes of solutions using elim over $\mathbb{Z}$-terms (e.g. using $\frac{1}{2} + 2\mathbb{Z}$). We compute values of these equivalence classes to be the closest values the lower bound literals L ($L \in A, \mathsf{lim}^{-\infty} L = \bot$ in item (V3) of case 1 of Def. 9), using core intervals of these L and λ. Cooper's method contains an optimization for formulas where $\mathcal{L}$ or $\mathcal{U}$ is empty, we implement this optimization in (V1). An additional optimization offered by VIRAS that is not present in Cooper's method is (V2).

Exponential Speed-Up of VIRAS. The work of [26] provides a quantifier elimination procedure for LIRA based on the following idea. A variable x is split into its integer $\lfloor x \rfloor$ and fractional $x - \lfloor x \rfloor$ parts, allowing for the separate uses of external QE procedures for LIA and LRA, respectively. Doing so, formula preprocessing comes with a heavy normalization burden: formulas are normalized such that their literals are of the form $jx + k \lfloor x \rfloor + t \diamond 0$, where $x \notin t$. With such normalizations, Ex. 11 shows an exponential blow-up in the formula size. Unlike this, VIRAS does not use external QE procedures but operates directly on LIRA terms, implementing virtual substitutions.

Example 11. *We illustrate the* VIRAS *benefits in avoiding the expensive normalizations of [26]. Let $n \in \mathbb{N}^{>0}$ and $t_i \in \mathbf{T}$ such that $x \notin t_i$. Consider the formula $\phi = \sum_{i=1}^{n} \lfloor 2x + t_i \rfloor \approx 0$.*

The work of [26] normalizes ϕ to $\phi' = \bigvee_{j_1=0}^{2} \ldots \bigvee_{j_n=0}^{2} \sum_{i=1}^{n} (2 \lfloor x \rfloor + j_i + \lfloor t_i \rfloor) \approx 0$ and eliminates quantifiers of ϕ'. Note that the size of ϕ' is $O(3^n)$ in the size of ϕ.

In contrast, VIRAS *computes the elimination set of ϕ as $\mathsf{elim}(\phi) = (\mathsf{ebreak}) \cup \mathsf{eseg} \cup \mathsf{ebound}^{+} \cup \mathsf{ebound}^{-}$ where $|\mathsf{ebound}^{+} \cup \mathsf{ebound}^{-}|$ is $O(1)$ and $|(\mathsf{ebreak})|$ is $O(|\mathsf{breaks}|2\mathsf{deltaX})$ in the size of ϕ. As $\mathsf{breaks} = \{-\frac{t_i}{2} + \mathbb{Z} \mid i \in \{1 \ldots n\}\}$ and $\mathsf{deltaX} = 2n$, we derive $|(\mathsf{ebreak})|$ being $O(n^2)$ in the size of ϕ. Further, the size of eseg is $O(|\mathsf{ebreak}|) = O(n^2)$. Using Theorem 1,* VIRAS *thus solves ϕ exponentially faster compared to [26].*

Solving Quantified SMT Problems in LIRA. Thanks to its sound and complete LIRA reasoning, VIRAS solves LIRA problems that existing SMT techniques fail to solve, like the example below.

Example 12. *Consider the formula:*

$$\forall x, z. \Big(\underbrace{\lceil x + z \rceil > \lfloor x + z \rfloor}_{L_1} \wedge \underbrace{\lceil z \rceil \approx \lfloor z \rfloor}_{L_2} \rightarrow \underbrace{\lfloor x \rfloor \not\approx x}_{L_3} \Big) \tag{3}$$

Literal L_1 is true iff $x + z$ is not an integer, L_2 is true iff z is an integer and L_3 is true iff x is not an integer. Formula (3) thus captures that, if the sum $x + z$ is not an integer and z is

[3]Note that LIA (aka. Presubrger Arithmetic) is sometimes defined with an auxiliary divisibility predicate $q \mid t$ (read "q divides t). This predicate can be expressed in LIRA as $\exists x. q \lfloor x \rfloor \approx t$.

an integer, then x cannot be an integer, which is clearly valid. Existing SMT techniques [3, 10] fail to solve (3)*, whereas* Viras *can easily prove*[4] (3)*.*

Conflict-Driven Reasoning. A complementary approach to Viras comes with conflict-driven proof search for arithmetic reasoning [18]. Within [18], validity of $\exists x_1 \dots x_n.\phi$ is (dis)proved, where ϕ is a conjunction of literals. Therefore the algorithm attempts at building a satisfying assignment $x_1 \leftarrow t_1 \dots x_n \leftarrow t_n$ using terms from the elimination set for t_i. If the assignment makes ϕ true, $\exists\phi$ must be valid. Whenever some partial assignment makes ϕ false, we speak of a *conflict*. Then a lemma is learned to block the generation of such a conflicting assignment, and proof search backtracks. When no more backtracking is possible, ϕ is unsatisfiable. While learning lemmas is central in [18], the approach is limited to elimination sets with plain virtual terms, that is to virtual terms not containing ε or $\pm\infty$, which is essential for Viras. In Sect. 6 we generalize lemma learning from [18], allowing us to handle proper virtual terms and improve Viras with conflict-driven proof search.

6 Conflict-Driven VIRAS

For extending Viras with conflict-driven lemma learning during proof search as introduced in [18], we need to resolve the following limitation. In case [18] identifies an assigment $x \leftarrow t$ as a conflict, the approach will introduce a lemma $x \not\approx t$ [5] to exclude this assignment. Simply using this approach in Viras is not sufficient since [18] can not handle the assigments of $x \leftarrow t + \varepsilon$, as $x \not\approx t + \varepsilon$ is not a formula in our Lira signature. To address this limitation, we introduce a function lemma_ϕ (Def. 12) to generate lemmas that exclude assignments for arbitrary virtual terms t from Def. 8; in particular, we generate ε-lemmas and ∞-lemmas. In order for the calculus using the lemma function lemma_ϕ to be sound, we impose that, if $\neg\phi[\![x \mathbin{/\!\!/} t]\!]$ and $\phi[x]$, then $\mathsf{lemma}_\phi(x \not\approx t)$ (Lem. 7.1). Further, to ensure completeness of lemma_ϕ, we exclude the current assignment $\neg\mathsf{lemma}_\phi(x \not\approx t)[\![x \mathbin{/\!\!/} t]\!]$ (Lem. 7.2). In what follows, we formalize this setting, allowing us to integrate Viras with conflict-driven proof, resulting in our improved CD-Viras calculus for QE over Lira formulas $\exists x.\phi$.

ε-Lemmas We first focus on finding a lemma that is false when we virtually substitute x by $t + \varepsilon$; we denote such lemmas as ε-lemmas. For these lemmas, we can use any formula $x \leq t \vee u < x$ for any $u > t$. To find such a u reason as follows. If ϕ does not hold at some point $t+\varepsilon$, there must be some non-empty interval (t, u) where ϕ does not hold. ϕ can only change its truth value at a value v when one of its literals $s \diamond 0$ changes its truth value at v. For deriving an ε-lemma, we use Fig. 3 to compute $\mathsf{nxt}^{\top}_{s\diamond 0}(t + \varepsilon)$ as an overapproximation of the set of all such v. In particular, if $\mathsf{breaks}_s = \emptyset$, the truth value of $s \diamond 0$ can only change from false to true if the linear function defined by $t[x]$ intersects with zero. Note that literals $-x > 0$ can only change their truth values from true to false, but not from false to true; hence $\mathsf{nxt}^{\top}_{-x>0}(t) = \emptyset$. If $\mathsf{breaks}_s \neq \emptyset$, then $s \diamond 0$ can only change its truth value if either the line segment of s at point $t + \varepsilon$ intersects with zero ($\mathsf{curZero}(s + \varepsilon)$ in Fig. 3) or at the next discontinuity b where that segment ends ($\mathsf{nextBreak}(s + \varepsilon)$ in Fig. 3). Based on this reasoning, we introduce formula $\mathsf{inFalseInterval}^{\phi}_{t+\varepsilon}(x)$ to define an interval I with lower bound $t + \varepsilon$, that includes only values for which ϕ is false (given $\phi[\![t + \varepsilon]\!]$ is false). [6]

[4] see Appendix A.2 in [24]

[5] To see how derivations and lemma derival works in detail see Ex. 16 and 17 in Appendix B.2 in [24]

[6] Example 18 in Appendix B.2 in [24] illustrates $\mathsf{inFalseInterval}^{\phi}_{t+\varepsilon}(x)$.

$$\mathsf{inFalseInterval}^{\phi}_{t+\varepsilon}(x) = (t < x \wedge \bigwedge_{L\in\phi} \bigwedge_{e\in\mathsf{nxt}^{\top}_{L}(t+\varepsilon)} (t+\varepsilon \lessdot e \rightarrow x \lessdot e))$$

$$s+\varepsilon \lessdot t \quad = s < t \qquad s \lessdot t \quad = s < t$$
$$s+\varepsilon \lessdot t+\varepsilon = s < t \qquad s \lessdot t+\varepsilon = s \leq t$$

if $\mathsf{breaks} = \emptyset$

$$\mathsf{nxt}^{\top}_{t\diamond 0}(s+\varepsilon) = \emptyset \quad \text{if } \mathsf{sslp} = 0$$

$$\mathsf{nxt}^{\top}_{t\approx 0}(s+\varepsilon) = \{\mathsf{zero}_t(0)\} \qquad \mathsf{nxt}^{\top}_{t\gtrsim 0}(s+\varepsilon) = \begin{cases} \{\mathsf{zero}_t(0)\} & \text{if } \mathsf{sslp} > 0 \;\&\; \gtrsim \;=\; \geq \\ \{\mathsf{zero}_t(0)+\varepsilon\} & \text{if } \mathsf{sslp} > 0 \;\&\; \gtrsim \;=\; > \\ \emptyset & \text{if } \mathsf{sslp} < 0 \end{cases}$$
$$\mathsf{nxt}^{\top}_{t\not\approx 0}(s+\varepsilon) = \emptyset$$

if $\mathsf{breaks} \neq \emptyset$

$$\mathsf{nxt}^{\top}_{t\diamond 0}(s+\varepsilon) = \mathsf{nextBreak}(s) \qquad \text{if } \mathsf{sslp} = 0$$
$$\mathsf{nxt}^{\top}_{t\diamond 0}(s+\varepsilon) = \mathsf{nextBreak}(s) \cup \mathsf{curZero}(s+\varepsilon) \qquad \text{if } \mathsf{sslp} \neq 0$$

$$\mathsf{nextBreak}(s) = \{\lceil s+\varepsilon \rceil^{b+p\mathbb{Z}} \mid b+p\mathbb{Z} \in \mathsf{breaks}\}$$

$$\mathsf{curZero}(s) = \begin{cases} \{\mathsf{zero}_t(s)+\varepsilon\} & \text{if } \diamond = > \;\&\; \mathsf{sslp} > 0 \\ \{\mathsf{zero}_t(s)\} & \text{if } \diamond = \geq \;\&\; \mathsf{sslp} > 0 \text{ or } \diamond \;= \approx \\ \emptyset & \text{if } \diamond \in \{>, \geq\} \;\&\; \mathsf{sslp} < 0 \text{ or } \diamond \;= \not\approx \end{cases}$$

Figure 3: Definition of $\mathsf{inFalseInterval}^{\phi}_{t+\varepsilon}(x)$.

Definition 11 (False Interval)**.** *Let ϕ be a conjunction of literals. The **false interval** of ϕ at $t+\varepsilon$ is denoted as $\mathsf{inFalseInterval}^{\phi}_{t+\varepsilon}(x)$ and defined in Fig. 3.*

∞-Lemmas We next derive lemmas to exclude assignments using virtual terms containing $\pm\infty$; we refer to these lemmas as ∞-lemmas. For $\phi[\![x /\!/ t+\infty]\!]$ to be false, there are two options: (i) either one of its aperiodic literals L has a limit $\mathsf{lim}^{\pm\infty}_{L} = \bot$, or (ii) one of its periodic literal L is false at t. For (i), we simply derive the ∞-lemma of $x \leq \mathsf{distX}^{+}_{L}$ or $\mathsf{distX}^{-}_{L} \leq x$. For (ii), our ∞-lemma has to exclude the solution t. A naïve approach would derive the ∞-lemma of $x \not\approx t$; this lemma however not suffice as $\mathsf{lim}^{\pm\infty}_{x\not\approx t} = \top$, hence $(x \not\approx t)[\![t \pm \infty]\!] = \top$. Therefore, we need to find some periodic literal that excludes the solution t. As L is periodic, we have $L[\![t]\!] \leftrightarrow L[\![t + \lambda \lfloor z \rfloor]\!]$, thus we obtain the ∞-lemma $\mathsf{rem}_{\lambda}(x) \approx \mathsf{rem}_{\lambda}(t)$, which is equaivalent to $x \approx t + \lambda(\mathsf{quot}_{\lambda}(x) - \mathsf{quot}_{\lambda}(t))$; this ∞-lemma is to be used for any t that does not contain ε. With a similar reasoning for $t+\varepsilon+\infty$ and by using ε-lemmas, we derive the ∞-lemma $\neg\mathsf{inFalseInterval}^{\phi}_{t+\lambda(\mathsf{quot}_{\lambda}(x)-\mathsf{quot}_{\lambda}(t))}(x)$.

$\mathbb{Z}$-flattening Lemmas for virtual terms $t+p\mathbb{Z}$ could be computed similarly to ∞ lemmas using $\mathsf{rem}_p(t) \not\approx \mathsf{rem}_p(x)$. Nevertheless, virtual substitutions with $\mathbb{Z}$-terms pose another challenge: as [18] transforms literals into disjunctions, the assumption of ϕ being a conjunction of literals required by the conflict-driven framework is violated. We resolve this difficulty by transforming the elimination set elim^{x} into the flattened version $\mathsf{elim}^{x}_{\mathsf{flat}}(\phi) = \{t \mid t+0\mathbb{Z} \in \mathsf{elim}^{x}(\phi)\} \cup \bigcup\{\mathsf{fin}^{\phi}_{t+p\mathbb{Z}} \mid t+p\mathbb{Z} \in \mathsf{elim}^{x}(\phi), p \not\approx 0\}$. Clearly, $\mathsf{elim}^{x}_{\mathsf{flat}}(\phi)$ fulfils Theorem 1 but does not

contain $\mathbb{Z}$-terms. Therefore we use $\mathsf{elim}^x_{\mathsf{flat}}$ instead of elim^x, allowing us to only deal with conjuctions of literals and replacing the need to generate lemmas for $\mathbb{Z}$-terms.

CD-VIRAS By using ε-lemmas, ∞-lemmas and $\mathbb{Z}$-flattening, we combine VIRAS with conflict-driven proof search, resulting in our CD-VIRAS calculus. Doing so, we ajdust only two rules from [18], namely INNER CONFLICT and LEAF CONFLICT[7] as named in [18]. Instead of lemmas $\bigvee_{i\in I} x_i \not\approx t_i$ introduced by these rules in [18], in CD-VIRAS we use the lemmas $\bigvee_{i\in I} \mathsf{lemma}_\phi(x_i \not\approx t_i)$ using the lemma function lemma_ϕ defined below.

Definition 12 (CD-VIRAS Lemmas)**.** *Let ϕ be a conjunction of literals, t a term, and $A = \{L \mid L \in \phi, \mathsf{oslp}_L \neq 0\}$. The* ***lemma function*** *of* CD-VIRAS *is defined as: conflicts are:*

$$
\begin{array}{lll}
\mathsf{lemma}_\phi\ (x \not\approx t) & = x \not\approx t & \\
\mathsf{lemma}_\phi\ (x \not\approx t + \varepsilon) & = \neg\mathsf{inFalseInterval}^\phi_{t+\varepsilon}(x) & \\
\mathsf{lemma}_\phi\ (x \not\approx t + e\varepsilon + \infty) & = x \leq \mathsf{distX}^+_L & \text{if } \mathsf{lim}^{+\infty}_L = \bot \text{ for some } L \in A \\
\mathsf{lemma}_\phi\ (x \not\approx t + e\varepsilon - \infty) & = \mathsf{distX}^-_L \leq x & \text{if } \mathsf{lim}^{-\infty}_L = \bot \text{ for some } L \in A \\
\mathsf{lemma}_\phi\ (x \not\approx t \pm \infty) & = \mathsf{rem}_\lambda(x) \not\approx \mathsf{rem}_\lambda(t) & \\
\mathsf{lemma}_\phi\ (x \not\approx t + \varepsilon \pm \infty) & = \neg\mathsf{inFalseInterval}^\phi_{t+\lambda(\mathsf{quot}_\lambda(x)-\mathsf{quot}_\lambda(t))+\varepsilon}(x) &
\end{array}
$$

Soundness and completeness of the lemma function is established next, yielding that the calculus CD-VIRAS itself is sound and complete.

Lemma 7. *Let ϕ be a conjunction of literals and v be a virtual term with $\mathbb{Z}(v) = 0$. Our function lemma_ϕ satisfies the following properties:*

1. $\neg\phi[\![x \mathbin{/\!\!/} v]\!] \to \forall x(\phi \to \mathsf{lemma}_\phi(x \not\approx v))$. *(soundness)*
2. $\neg\mathsf{lemma}_\phi[\![x \mathbin{/\!\!/} v]\!]$. *(completeness)*

Using Lem. 7, soundness and completeness of the calculus CD-VIRAS is proved in the same way as in [18]. Lem. 7.1 is needed for soundness, while Lem. 7.2 is needed for completeness. [8]

7 Conclusion

We introduce the VIRAS calculus as a new quantifier elimination procedure for solving quantifier formulas with mixed linear integer-real arithmetic. VIRAS uses virtual substitutions and can be integrated with conflict-driven proof search. Computing more accurate bounds $\mathsf{distY}^\pm$, as well as more accurate discontinuity sets breaks, is an interesting line for future research, with the purpose of more efficient proof search. Implementing VIRAS is another challenge for further work. We pointed out that our method gives an exponential speed-up over [26] for some classes of formulas. Nevertheless finding actual complexity bounds for our method remains for future research.

Acknowledgements. We acknowledge funding from the ERC Consolidator Grant ARTIST 101002685, the TU Wien SecInt Doctoral College, the FWF SFB project SpyCoDe F8504, the WWTF ICT22-007 grant ForSmart, the Amazon Research Award 2023 QuAT and the EPSRC SCorCH project grant EP/V000497/1.

[7] Modified rules are given in Fig. 4 in Sect. 6 in [24].
[8] For a detailed explanation of the proof we refer to Appendix B.3 in [24].

References

[1] Dennis S. Arnon, George E. Collins, and Scott McCallum. Cylindrical algebraic decomposition I: the basic algorithm. *SIAM J. Comput.*, 13(4):865–877, 1984.

[2] Philipp Bär, Jasper Nalbach, Erika Ábrahám, and Christopher W. Brown. Exploiting Strict Constraints in the Cylindrical Algebraic Covering. In *SMT*, volume 3429 of *CEUR Workshop Proceedings*, pages 33–45. CEUR-WS.org, 2023.

[3] Haniel Barbosa, Clark W. Barrett, Martin Brain, Gereon Kremer, Hanna Lachnitt, Makai Mann, Abdalrhman Mohamed, Mudathir Mohamed, Aina Niemetz, Andres Nötzli, Alex Ozdemir, Mathias Preiner, Andrew Reynolds, Ying Sheng, Cesare Tinelli, and Yoni Zohar. cvc5: A versatile and industrial-strength SMT solver. In *TACAS*, volume 13243 of *LNCS*, pages 415–442. Springer, 2022.

[4] Nikolaj S. Bjørner and Mikolás Janota. Playing with quantified satisfaction. In *LPAR*, volume 35 of *EPiC Series in Computing*, pages 15–27. EasyChair, 2015.

[5] Franz Brauße, Konstantin Korovin, Margarita V. Korovina, and Norbert Th. Müller. A CDCL-style calculus for solving non-linear constraints. In *FroCoS*, pages 131–148, 2019.

[6] Alessandro Cimatti, Alberto Griggio, Ahmed Irfan, Marco Roveri, and Roberto Sebastiani. Incremental linearization for satisfiability and verification modulo nonlinear arithmetic and transcendental functions. *ACM Trans. Comput. Log.*, 19(3):19:1–19:52, 2018.

[7] George E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition-preliminary report. *SIGSAM Bull.*, 8(3):80–90, 1974.

[8] Byron Cook. Formal Reasoning About the Security of Amazon Web Services. In *CAV*, volume 10981 of *LNCS*, pages 38–47. Springer, 2018.

[9] David C Cooper. Theorem proving in arithmetic without multiplication. *Machine intelligence*, 7(91-99):300, 1972.

[10] Leonardo Mendonça de Moura and Nikolaj S. Bjørner. Z3: an efficient SMT solver. In *TACAS*, volume 4963 of *LNCS*, pages 337–340. Springer, 2008.

[11] Martin Desharnais, Petar Vukmirovic, Jasmin Blanchette, and Makarius Wenzel. Seventeen Provers Under the Hammer. In *ITP*, volume 237 of *LIPIcs*, pages 8:1–8:18, 2022.

[12] Dino Distefano, Manuel Fähndrich, Francesco Logozzo, and Peter W. O'Hearn. Scaling Static Analyses at Facebook. *Commun. ACM*, 62(8):62–70, 2019.

[13] Andreas Dolzmann, Thomas Sturm, and Volker Weispfenning. Real quantifier elimination in practice. In *Algorithmic Algebra and Number Theory*, pages 221–247, 1997.

[14] Bruno Dutertre. Solving exists/forall problems with yices. 2015.

[15] Bruno Dutertre and Leonardo Mendonça de Moura. A fast linear-arithmetic solver for DPLL(T). In *CAV*, volume 4144 of *LNCS*, pages 81–94. Springer, 2006.

[16] Dejan Jovanovic and Leonardo de Moura. Solving non-linear arithmetic. *ACM Commun. Comput. Algebra*, 46(3/4):104–105, 2012.

[17] Dejan Jovanovic and Leonardo Mendonça de Moura. Cutting to the Chase - Solving Linear Integer Arithmetic. *J. Autom. Reason.*, 51(1):79–108, 2013.

[18] Konstantin Korovin, Marek Kosta, and Thomas Sturm. Towards Conflict-Driven Learning for Virtual Substitution. In *CASC*, pages 256–270. Springer, 2014.

[19] Konstantin Korovin, Nestan Tsiskaridze, and Andrei Voronkov. Conflict resolution. In *CP*, volume 5732 of *LNCS*, pages 509–523. Springer, 2009.

[20] Konstantin Korovin and Andrei Voronkov. Solving systems of linear inequalities by bound propagation. In *CADE*, volume 6803 of *LNCS*, pages 369–383. Springer, 2011.

[21] Rüdiger Loos and Volker Weispfenning. Applying linear quantifier elimination. *Comput. J.*, 36(5):450–462, 1993.

[22] David Monniaux. Quantifier elimination by lazy model enumeration. In *CAV*, volume 6174 of

LNCS, pages 585–599. Springer, 2010.

[23] Andrew Reynolds, Tim King, and Viktor Kuncak. Solving quantified linear arithmetic by counterexample-guided instantiation. *Formal Methods Syst. Des.*, 51(3):500–532, 2017.

[24] Johannes Schoisswohl, Laura Kovács, and Konstantin Korovin. VIRAS: Conflict-Driven Quantifier Elimination for Integer-Real Arithmetic (Extended Version). EasyChair Preprint no. 13150, 2024.

[25] Thomas Sturm. Thirty years of virtual substitution: Foundations, techniques, applications. In *ISSAC*, pages 11–16. ACM, 2018.

[26] Volker Weispfenning. Mixed real-integer linear quantifier elimination. In *ISSAC*, pages 129–136. ACM, 1999.