Linear Absolute Value Relation Analysis ⋆

Liqian Chen[1], Antoine Miné[2,3], Ji Wang[1], and Patrick Cousot[2,4]

[1] National Laboratory for Parallel and Distributed Processing, Changsha, P.R.China
{lqchen,wj}@nudt.edu.cn
[2] École Normale Supérieure, Paris, France
{mine,cousot}@di.ens.fr
[3] CNRS, France
[4] CIMS, New York University, New York, NY, USA

Abstract. Linear relation analysis (polyhedral analysis), devoted to discovering linear invariant relations among variables of a program, remains one of the most powerful abstract interpretations but is subject to convexity limitations. *Absolute value* enjoys piecewise linear expressiveness and thus natively fits to encode certain *non-convex* properties. Based on this insight, we propose to use *linear absolute value relation analysis* to discover linear relations among values and absolute values of program variables. Under the framework of abstract interpretation, the analysis yields a new numerical abstract domain, namely the abstract domain of *linear absolute value inequalities* ($\Sigma_k a_k x_k + \Sigma_k b_k |x_k| \leq c$), which can be used to analyze programs involving piecewise linear behaviors (e.g., due to conditional branches or absolute value function calls). Experimental results of our prototype are encouraging; The new abstract domain can find non-convex invariants of interest in practice.

1 Introduction

Abstract interpretation [7] provides a general framework for static analysis. One predominant application is numerical static analysis, i.e., to discover numerical properties of a program statically and automatically. *Linear relation analysis* [9], devoted to discovering linear invariant relations among variables of a program, is one of the earliest but still most powerful abstract interpretations. It yields the known *convex polyhedra* abstract domain ($\Sigma_k a_k x_k \leq c$) [9], since the set of the reachable states at each program point is abstracted as a convex polyhedron. Over the last 30 years, linear relation analysis has a wide range of applications, especially in the field of analysis and verification of programs and hybrid systems [14]. Moreover, a variety of weakly (linear) relational abstract domains have been proposed in recent years, for discovering restricted forms of linear relations, such as the Octagon domain ($\pm x \pm y \leq c$) [21], the Two Variables Per Inequality (TVPI) domain ($ax + by \leq c$) [28], and the Template Polyhedra domain ($\Sigma_k a_k x_k \leq c$ where variable coefficients a_k are fixed beforehand) [25].

The concrete semantics of a program often involves *non-convex* behaviors. E.g., conditional branch statements often introduce disjunctive behaviors, since different computations are performed depending on whether the condition evaluates to true or false.

⋆ This work is supported by the INRIA project "Abstraction" common to CNRS and ENS in France, and by the National Natural Science Foundation of China under Grant No.60725206.

Besides, many program properties that users may be interested in are non-convex, e.g., the division-by-zero error. However, the polyhedra abstract domain together with weakly (linear) relational abstract domains can express only convex sets (without resorting to powerset extensions). The convexity limitations may lead to imprecision in the analysis and thus can cause many false alarms.

Absolute value (AV) is a fundamental concept in mathematics and of high relevance in practice. AV function is essentially a kind of *piecewise linear* functions, and thus fits to express piecewise linear behaviors in a program that account for a large class of non-convex behaviors in practice. Moreover, complex non-linear program behaviors can be abstracted into piecewise linear behaviors, as in the field of hybrid systems. On the other hand, AV functions are provided by many modern programming languages, e.g., the *abs* (absolute value of an integer), *fabs* (absolute value of a floating-point number) functions in the C language. And several commonly used mathematical functions such as *fmin* (minimum value), *fmax* (maximum value), *fdim* (positive difference) in the C99 standard can be also expressed by AV functions, e.g., $\max(x, y) = \frac{1}{2}(|x-y|+x+y)$. Besides, rounding errors in floating-point arithmetic can be also abstracted by AV functions: $|round(x) - x| \leq \varepsilon_{\text{rel}} \cdot |x| + \varepsilon_{\text{abs}}$ where ε_{rel} denotes a relative error and ε_{abs} denotes an absolute error [20]. In addition, in Sect. 2.4, we will show that linear constraints with interval coefficients which may appear in numerical static analysis [4] can be also rewritten via AV functions. However, due to non-linearity, AV functions are rarely considered during program analysis and verification.

In this paper, we propose an analysis to discover *linear absolute value relations* among variables of a program, i.e., linear relations among the values and the absolute values of variables. The analysis yields a new abstract domain, namely the abstract domain of *linear absolute value inequalities* (AVI), to infer relationships of the form $\Sigma_k a_k x_k + b_k|x_k| \leq c$ over program variables x_k ($k = 1, \ldots, n$) where constants $a_k, b_k, c \in \mathbb{R}$ are automatically inferred by the analysis. The new domain is more expressive than the classic convex polyhedra domain and allows expressing certain non-convex (even unconnected) sets due to the utilization of absolute value. Its domain operations are constructed based a double description method. The preliminary experimental results of the prototype implementation are promising on benchmark programs; AVI can find non-convex invariants of interest in practice.

To sum up, this paper aims at exploiting the piecewise-linear expressiveness of absolute value to design non-convex abstract domains which can be used to capture disjunctive information in a program and which for example will apply to programs involving AV(-like) function calls. In other words, this paper is dedicated to coping with disjunctive behaviors of a program at the level of abstract domains, with no need to resort to other techniques to deal with disjunctions [3,8,24].

The rest of the paper is organized as follows. Section 2 shows the equivalence among linear absolute value inequality systems, extended linear complementary problem (XLCP) systems and interval linear inequality systems. Section 3 presents a double description method for XLCP on top of that for polyhedra. Section 4 proposes an abstract domain of linear AV inequalities based on the double description method for XLCP. Section 5 presents our prototype implementation together with preliminary experimental results. Section 6 discusses some related work before Section 7 concludes.

2 Linear absolute value inequality systems and their equivalents

2.1 Linear absolute value inequality systems (AVIs)

Let $|\cdot|$ denote absolute value (AV). We consider the following system of linear absolute value inequalities (AVI)

$$Ax + B|x| \leq c \tag{1}$$

where $A, B \in \mathbb{R}^{m\times n}$ and $c \in \mathbb{R}^m$.

Theorem 1. *Any AV inequality*

$$\textstyle\sum_i a_i x_i + \sum_{i\neq p} b_i|x_i| + b_p|x_p| \leq c$$

where $b_p > 0$, can be reformulated as a conjunction of two AV inequalities

$$\begin{cases} \sum_i a_i x_i + \sum_{i\neq p} b_i|x_i| + b_p x_p \leq c \\ \sum_i a_i x_i + \sum_{i\neq p} b_i|x_i| - b_p x_p \leq c \end{cases}$$

Theorem 1 implies that any AVI system $Ax + B|x| \leq c$ can be reformulated as an AVI system $A'x + B'|x| \leq c$ where $B' \leq 0$.

2.2 Extended linear complementarity problems (XLCPs)

Given a matrix $M \in \mathbb{R}^{n\times n}$ and a vector $q \in \mathbb{R}^n$, the (standard) linear complementarity problem (LCP) is defined as the problem of finding vectors x^+ and x^- such that

$$x^+ = Mx^- + q \tag{2}$$
$$x^+, x^- \geq 0 \tag{3}$$
$$(x^+)^T x^- = 0. \tag{4}$$

Note that if x^+ and x^- are solutions of the above LCP, then it follows from (3-4) that

$$x_i^+ x_i^- = 0 \qquad \text{for } i = 1, \ldots, n$$

i.e., for each i the following holds: If $x_i^+ > 0$ then $x_i^- = 0$ holds, and if $x_i^- > 0$ then $x_i^+ = 0$ holds. In other words, the zero patterns of x_i^+ and x_i^- are complementary. Thus, condition (4) is called the *complementarity condition* of the above LCP. The LCP problem is one of the fundamental problems in mathematical optimization theory, which subsumes many mathematical programming problems such as linear programs, quadratic programs [6]. Here, we present one extension of the LCP that is of interest to us.

Given $M, N \in \mathbb{R}^{m\times n}$ and a vector $q \in \mathbb{R}^m$, find $x^+, x^- \in \mathbb{R}^n$ such that

$$Mx^+ + Nx^- \leq q \tag{5}$$
$$x^+, x^- \geq 0 \tag{6}$$
$$(x^+)^T x^- = 0. \tag{7}$$

We call the above problem eXtended Linear Complementary Problem (XLCP), since it can be proved equivalent to eXtended LCP of Mangasarian and Pang [19].

2.3 Interval linear inequality systems (ILIs)

Let $\mathbf{x} = [\underline{x}, \overline{x}]$ be an interval with its bounds (endpoints) $\underline{x} \leq \overline{x}$. Let $\mathbb{IR}$ be the set of all real intervals $[\underline{a}, \overline{a}]$ where $\underline{a}, \overline{a} \in \mathbb{R}$. Let $\underline{A}, \overline{A} \in \mathbb{R}^{m \times n}$ be two matrices with $\underline{A} \leq \overline{A}$ where the order is defined element-wise, then the set of matrices $\mathbf{A} = [\underline{A}, \overline{A}] = \{A \in \mathbb{R}^{m \times n} : \underline{A} \leq A \leq \overline{A}\}$ is called an *interval matrix* and the matrices $\underline{A}, \overline{A}$ are called its bounds. Let us define the *center matrix* of $\mathbf{A}$ as $A_c = \frac{1}{2}(\underline{A} + \overline{A})$ and the *radius matrix* as $\triangle_A = \frac{1}{2}(\overline{A} - \underline{A})$. Then, $\mathbf{A} = [\underline{A}, \overline{A}] = [A_c - \triangle_A, A_c + \triangle_A]$. Note that $\triangle_A \geq 0$ always holds.

Let b be a regular vector in $\mathbb{R}^m$. The following system of interval linear inequalities

$$\mathbf{A}x \leq b$$

denotes an *interval linear inequality system* (ILI), that is, the *family* of all systems of linear inequalities $Ax \leq b$ such that $A \in \mathbf{A}$. A vector $x \in \mathbb{R}^n$ is called a *weak solution* of the interval linear inequality system $\mathbf{A}x \leq b$, if it satisfies $Ax \leq b$ for some $A \in \mathbf{A}$.

2.4 Equivalence among AVIs, XLCPs and ILIs

Equivalence between AVIs and XLCPs. Let $x = (x_i)_{i=1}^n$ be a vector. Let vectors x^+ and x^- be defined by $x^+ = (\max(x_i, 0))_{i=1}^n$ and $x^- = (\max(-x_i, 0))_{i=1}^n$, so that

$$x^+ \geq 0, x^- \geq 0, (x^+)^T x^- = 0$$

and

$$x = x^+ - x^-, \qquad |x| = x^+ + x^- \tag{8}$$

$$x^+ = \frac{1}{2}(x + |x|), \qquad x^- = \frac{1}{2}(|x| - x) \tag{9}$$

wherein $|x| = (|x_i|)_{i=1}^n$.

According to (8), AVI (1) can be reformulated as the following XLCP:

$$\begin{aligned} (A + B)x^+ + (B - A)x^- &\leq c \\ x^+, x^- &\geq 0 \\ (x^+)^T x^- &= 0 \end{aligned}$$

Similarly, according to (9), XLCP (5-7) can be reformulated as the following AVI:

$$\frac{1}{2}(M - N)x + \frac{1}{2}(M + N)|x| \leq q$$

Equivalence between AVIs and ILIs. From Theorem 2.19 in [23] (which states that a vector $x \in \mathbb{R}^n$ is a weak solution of $\mathbf{A}x \leq b$ iff it satisfies $A_c x - \triangle_A|x| \leq b$) together with Theorem 1 in this paper, we can prove that any system of absolute value inequalities $Ax + B|x| \leq b$ can be reformulated as a system of interval linear inequalities $\mathbf{A}'x \leq b'$ where $A, B \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, \mathbf{A}' \in \mathbb{IR}^{k \times n}, b' \in \mathbb{R}^k$. The converse also holds.

Example 1. Consider the following AVI: $\{|x| \leq 1, -|x| \leq -1\}$. Its corresponding XLCP will be $\{x^+ + x^- \leq 1, -x^+ - x^- \leq -1, x^+ \geq 0, x^- \geq 0, (x^+)^T x^- = 0\}$, and its corresponding ILI will be $\{x \leq 1, -x \leq 1, [-1, 1]x \leq -1\}$.

Until now, we have shown the equivalence among AVIs, XLCPs and ILIs, which indicates that we can reuse the method that can solve one of them to solve the others. In this paper, we reduce AVIs (as well as ILIs) to XLCPs and propose a double description method to characterize all solutions of an XLCP. On the other hand, the equivalence implies that the AVI domain proposed in this paper can be reused to infer other kinds of equivalent relations, e.g., to deal with linear constraints with interval coefficients that may appear in numerical static analysis [4]. In Sect. 4.1, we will see that the AVI domain is as expressive as the existing interval polyhedra domain [4] (that employs ILIs for domain representation) but enjoys better (optimal) domain operations.

3 Double description method for XLCP

By Minkowski-Weyl theorem [26], the set $P \subseteq \mathbb{R}^n$ is a polyhedron, iff it is finitely generated, i.e., there exist finite sets $V, R \in \mathbb{R}^n$ such that P can be generated by (V, R):

$$P = \left\{ \sum_{i=1}^{|V|} \lambda_i V_i + \sum_{j=1}^{|R|} \mu_j R_j \;\middle|\; \forall i, \lambda_i \geq 0, \forall j, \mu_j \geq 0, \sum_{i=1}^{|V|} \lambda_i = 1 \right\}$$

where $|V|, |R|$ denote the cardinality of sets V, R respectively. Elements in V are called *extreme points*, while elements in R are called *extreme rays*. Using the double description method, a convex polyhedron can be represented by either its constraint representation $\{Ax \leq b\}$ or its generator representation (V, R). The two representations are duals: Each can be computed from the other by Chernikova's algorithm [18]. And the classic convex polyhedra abstract domain [9] is designed based on the dual representations.

In this section, we will construct a double description method for XLCP, on top of that for convex polyhedra. The main idea is the following. Intuitively, (5-6) of an XLCP describes a convex polyhedron $P = \{x^+ \in \mathbb{R}^n, x^- \in \mathbb{R}^n \mid Mx^+ + Nx^- \leq q, x^+ \geq 0, x^- \geq 0\}$, while the complementary condition (7) specifies that $x_i^+ = 0 \vee x_i^- = 0$ holds for all $i = 1, \ldots, n$, which indicates 2^n complementary patterns. Overall, XLCP (5-7) can be considered as a union of a set of polyhedra, the number of which is in the worst case 2^n (one for each complementary pattern). E.g., when $n = 1$, XLCP (5-7) is equivalent to the union of 2^1 polyhedra: $\{x^+ \in \mathbb{R}^n, x^- \in \mathbb{R}^n \mid Mx^+ + Nx^- \leq q, x^+ = 0, x^- \geq 0\} \cup \{x^+ \in \mathbb{R}^n, x^- \in \mathbb{R}^n \mid Mx^+ + Nx^- \leq q, x^+ \geq 0, x^- = 0\}$. It is worth noting that first not all generators g of P will be the generators of XLCP (5-7), since g may not satisfy the complementary condition. Second, even for those generators of P that satisfy the complementary condition, not all combinations of them will result in solutions of XLCP (5-7). Essentially, we need to group generators according to the complementary patterns such that each group corresponds to a convex polyhedron and any combination of generators in one group will always result in a solution of XLCP (5-7).

Example 2. Consider the following XLCP: $\{-x^+ - x^- \leq -1, x^+ \leq 2, x^- \leq 2, x^+ \geq 0, x^- \geq 0, x^+x^- = 0\}$. As shown in Fig. 1, the polyhedral generators of $\{-x^+ - x^- \leq -1, x^+ \leq 2, x^- \leq 2, x^+ \geq 0, x^- \geq 0\}$ will be

$$(V, R) = \left(\begin{pmatrix} x^+ \\ x^- \end{pmatrix} : \;\; \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix} \right\}, \;\; \emptyset \right)$$

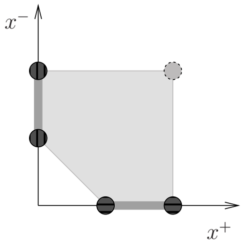


Fig. 1. Generators and their grouping for XLCP

Firstly, the extreme point $(2\ 2)^T$ does not satisfy the complementary condition $x^+x^- = 0$, and thus should be removed, since no combination involving $(2\ 2)^T$ will satisfy $x^+x^- = 0$ either. Secondly, for other extreme points that satisfy the complementary condition, not all convex combinations of them will satisfy $x^+x^- = 0$, e.g., convex combinations of $(1\ 0)^T$ and $(0\ 1)^T$. To precisely characterize the solution set of the original XLCP, two groups need to be constructed such that any convex combination of extreme points from either group (without mixing) forms a XLCP solution:

$$\left\{ \left(\begin{pmatrix} x^+ \\ x^- \end{pmatrix} :\ \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix} \right\},\ \emptyset \right),\quad \left(\begin{pmatrix} x^+ \\ x^- \end{pmatrix} :\ \left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix} \right\},\ \emptyset \right) \right\}$$

Taking the first group as an example, the convex combinations between extreme points $(1\ 0)^T$ and $(2\ 0)^T$ from the first group define points lying on the line segment connecting $(1\ 0)^T$ and $(2\ 0)^T$. Indeed, all those points are solutions of the original XLCP.

3.1 Conversion from the constraint to the generator representation

3.1.1 Homogeneous case

Finding non-negative solutions of a polyhedral cone satisfying the complementary condition. Let us denote $x = (x^+, x^-)^T$, where $x \in \mathbb{R}^{2n}, x^+, x^- \in \mathbb{R}^n$. Now we consider a set of the form

$$C_\pm = \{x \in \mathbb{R}^{2n} \mid Ax \geq 0, x \geq 0, (x^+)^T x^- = 0\}$$

A vector $y \in \mathbb{R}^{2n}$ is said to be a *complementary generator* of $C_\pm$, iff y is a polyhedral generator of $\{x \mid Ax \geq 0, x \geq 0\}$ and satisfies $(y^+)^T y^- = 0$. One simple way to find all the complementary generators for $C_\pm$ is to first calculate all the polyhedral generators of $\{x \mid Ax \geq 0, x \geq 0\}$ using Chernikova's algorithm [18] and then remove those generators that do not satisfy $(x^+)^T x^- = 0$ at the very end of the whole process. However, this may cause a lot of unnecessary computation, and even combinatorial explosion.

We aim to design a method that can generate directly only complementary generators, by taking into account the complementary condition during the intermediate computation. First, let $C = \{x \mid Ax \geq 0, x \geq 0\}$ be a polyhedral cone and let y be a vector in C. Let D be the matrix associated with the constraint system of C, i.e., $D = \begin{bmatrix} A \\ I \end{bmatrix}$ where $I \in \mathbb{R}^{n \times n}$ denotes the identity matrix. We use $S(y, D) = \{a \mid a \text{ is a row of } D$

such that $ay = 0\}$ to denote the set of rows saturated by y. Let Q be the non-redundant set of polyhedral generators of C. Note that generators of pointed polyhedral cone are all extreme rays except the extreme point 0. We apply the following rule to check adjacency: Two rays y_1 and y_2 are *adjacent* in Q, denoted as $\textit{adjacent}(y_1, y_2) = \textit{true}$, if $|S(y_1, D) \cap S(y_2, D)| \geq 1$ and there is no y' distinct from y_1, y_2 in Q such that $S(y_1, D) \cap S(y_2, D) \subseteq S(y', D)$.

Let $Q_\pm = \{y_1, \ldots, y_r\}$ be the non-redundant set of complementary extreme rays of $C_\pm$. Let $H = \{x \mid cx \geq 0\}$. Three sets can be defined with respect to the product cy_i:

$$Q_\pm^= = \{y \mid y \in Q_\pm, cy = 0\}$$
$$Q_\pm^> = \{y \mid y \in Q_\pm, cy > 0\}$$
$$Q_\pm^< = \{y \mid y \in Q_\pm, cy < 0\}.$$

The non-redundant set of complementary extreme rays of the new cone $C_\pm \cap H$, denoted as $Q'_\pm$, can be then constructed as:

$$Q'_\pm = Q_\pm^= \cup Q_\pm^> \cup \overline{Q}_\pm$$

where $\overline{Q}_\pm$ is defined by:

$$\{y \mid cy = 0, y = \lambda y_1 + \mu y_2, (y_1, y_2) \in Q_\pm^> \times Q_\pm^<, \textit{adjacent}^c(y_1, y_2), \lambda > 0, (y^+)^T y^- = 0\}$$

where $\textit{adjacent}^c(y_1, y_2) = \textit{true}$, if $|S(y_1, D) \cap S(y_2, D)| >= 1$ and there is no y' distinct from y_1, y_2 in $Q_\pm$ such that $S(y_1, D) \cap S(y_2, D) \subseteq S(y', D)$.

Note that to state $\textit{adjacent}^c(y_1, y_2) = \textit{true}$, we check over only the set $Q_\pm$ that is a subset of Q since some of the elements in Q may not satisfy the complementary condition. Thus, it may happen that $\textit{adjacent}^c(y_1, y_2) = \textit{true}$ (defined via $Q_\pm$) but $\textit{adjacent}(y_1, y_2) = \textit{false}$ (defined via Q). However, in fact, in this case it can be proved that no positive combination of such y_1 and y_2 will satisfy the complementary condition. Let R be the set of extreme rays for $C = \{x \mid Ax \geq 0, x \geq 0\}$. Let R^c be the resulting set of complementary extreme rays for $C_\pm = \{x \mid Ax \geq 0, x \geq 0, (x^+)^T x^- = 0\}$ computed by the above method via $Q_\pm$. The following theorem guarantees the correctness of the above incremental process of computing complementary extreme rays for $C_\pm$.

Theorem 2. *R^c contains all and only complementary extreme rays, that is, $R^c = R \cap \{y \mid y^+ y^- = 0\}$.*

Theorem 2 implies that R^c is equivalent to the result given by first computing all extreme rays (i.e., R) for C and then removing at the very end those extreme rays that do not satisfy the complementary condition.

Grouping complementary extreme rays. Note that not all non-negative combinations of rays in R^c satisfy the complementary condition $(x^+)^T x^- = 0$ and thus are necessary in $C_\pm$. To precisely describe $C_\pm$, we need to classify R^c into several groups such that the grouping result $R^{cc} = \langle R^c_{s_1}, \ldots, R^c_{s_i}, \ldots, R^c_{s_m} \rangle$ satisfies

1. $\cup_{i=1}^m R^c_{s_i} = R^c$, and
2. Within each group $R^c_{s_i}$, any nonnegative combination y of rays in $R^c_{s_i}$ satisfies the complementary condition $(y^+)^T y^- = 0$.

Note that R^{cc} is a cover of R^c. To construct such groups, we use the following method. First, we construct an undirected graph Θ, where each $r^c_i \in R^c$ corresponds to one node in Θ. And there is an edge between two nodes r^c_i and r^c_j, if the resulting vector $s = \max\{r^c_i, r^c_j\}$ satisfies $(s^+)^T s^- = 0$. The goal then is to find all the maximal complete subgraphs in Θ, each of which corresponds to one group $R^c_{s_i}$ in R^c. After that, we can characterize $C_\pm$.

Theorem 3. *Let* $C_\pm = \{x \mid Ax \geq 0, x \geq 0, (x^+)^T x^- = 0\}$ *and let* $R^{cc} = \langle R^c_{s_1}, \ldots, R^c_{s_i}, \ldots, R^c_{s_m}\rangle$ *be the grouping result of its complementary extreme rays. Then* $x \in C_\pm$*, iff there exists some* i $(i \in \mathbb{N}, 1 \leq i \leq m)$ *such that*

$$x = \sum_{r^c_k \in R^c_{s_i}} \mu_k r^c_k$$

where $\mu_k \geq 0$.

Intuitively, Theorem 3 states that $C_\pm$ is a union of a set of polyhedral cones and each group $R^c_{s_i}$ corresponds to one polyhedral cone. While the sufficient condition is obvious, the necessary condition can be proved as follows: Suppose $y \in C_\pm = \{x \mid Ax \geq 0, x \geq 0, (x^+)^T x^- = 0\}$. First, we know that y can be and can only be generated through a positive combination of a subset of rays in $R^c = R \cap \{x \mid (x^+)^T x^- = 0\}$, since the result z of any positive combination involving $r \in R \setminus R^c$ will not satisfy $(z^+)^T z^- = 0$. Assume that y can be generated through a positive combination of a set of rays R^c_y where $R^c_y \subseteq R^c$, which implies that any nonnegative combination of rays in R^c_y will satisfy the complementary condition. Hence, any $R^c_{s_i}$ satisfying $R^c_y \subseteq R^c_{s_i}$ can generate y. Therefore, there exists some i $(1 \leq i \leq m)$ such that $y = \sum_{r^c_k \in R^c_{s_i}} \mu_k r^c_k$ where $\mu_k \geq 0$.

3.1.2 Inhomogeneous case

Finding non-negative solutions of a convex polyhedron. By introducing a fresh variable $h \in \mathbb{R}$, the inhomogeneous linear system $\{Ax \geq b, x \geq 0\}$ (where $x \in \mathbb{R}^{2n}$) can be transformed into an equivalent homogeneous one: $\{[A \ \ -b]y \geq 0, y \geq 0\}$ where $y = (x\ h)^T$ is a column $(2n+1)$-vector with $h \geq 0$.

Each extreme ray r of the above homogeneous system has the form of $r = (x\ h)^T$ with $h \geq 0$. We use r_h to denote the h component of the vector r. For each extreme ray r, there are two possibilities: $r_h = 0$ or $r_h > 0$. The set of extreme rays of the homogeneous system, denoted as R^h, can be divided into two groups: $R^0 = \{r \mid r \in R^h, r_h = 0\}$ and $R^1 = \{r/r_h \mid r \in R^h, r_h > 0\}$. Then, we extract the x part out of the vectors from R^0 and R^1. Assume that we get $R = \{x \mid (x\ 0)^T \in R^0\}$ and $V = \{x \mid (x\ 1)^T \in R^1\}$. The generators in V are called *extreme points* while the generators in R are called *extreme rays* of the inhomogeneous system. In other words, we get the generator representation $G = (V, R)$ for the inhomogeneous system $\{Ax \geq b, x \geq 0\}$.

Finding non-negative solutions of a convex polyhedron satisfying the complementary condition. Similarly as above, from the set of complementary extreme rays of $\{y \mid Ay \geq 0, y \geq 0, (x^+)^T x^- = 0, y = (x^+\ \ x^-\ \ h)^T\}$ which can be obtained via the method in Sect. 3.1.1, we can derive the set of complementary generators $G^c = (V^c, R^c)$ for

$$P_\pm = \{x \mid Ax \geq b, x \geq 0, (x^+)^T x^- = 0\}.$$

Again, to precisely describe $P_{\pm}$, we need to classify G^c into several groups such that the grouping result $G^{cc} = \langle G^c_{s_1}, \ldots, G^c_{s_i}, \ldots, G^c_{s_m} \rangle$ where $G^c_{s_i} = (V^c_{s_i}, R^c_{s_i})$, satisfies

1. $\cup_{i=1}^m V^c_{s_i} = V^c$, $\cup_{i=1}^m R^c_{s_i} = R^c$, and
2. Within each group $G^c_{s_i}$, any sum z of an arbitrary convex combination of extreme points from $V^c_{s_i}$ and an arbitrary nonnegative combination of extreme rays from $R^c_{s_i}$, satisfies the complementary condition $(z^+)^T z^- = 0$.

Similarly, to construct such groups, we can use algorithms that find all the maximal complete subgraphs of an undirected graph. Now we can characterize $P_{\pm}$.

Theorem 4. *Let $P_{\pm} = \{x \in \mathbb{R}^{2n} \mid Ax \geq b, x \geq 0, (x^+)^T x^- = 0\}$, and let $G^{cc} = \langle G^c_{s_1}, \ldots, G^c_{s_i}, \ldots, G^c_{s_m} \rangle$ be the grouping result of its complementary generators where $G^c_{s_i} = (V^c_{s_i}, R^c_{s_i})$. Then $x \in P_{\pm}$, iff there exists some i ($i \in \mathbb{N}, 1 \leq i \leq m$) such that*

$$x = \sum_{v^c_j \in V^c_{s_i}} \lambda_j v^c_j + \sum_{r^c_k \in R^c_{s_i}} \mu_k r^c_k$$

where $\lambda_j, \mu_k \geq 0, \Sigma_j \lambda_j = 1$.

Theorem 4 states that $P_{\pm}$ is a union of a set of convex polyhedra, the number of which is s_m. Each group $G^c_{s_i}$ describes a polyhedron. Note that s_m is not necessarily equal to the number of complementary patterns (i.e., 2^n), since a certain complementary pattern may define an empty polyhedron and the union of some polyhedra stemming from distinct complementary patterns may be exactly representable by a single polyhedron.

It is worth noting that generating all the maximal complete subgraphs of an undirected graph is an NP-complete problem [10]. Fortunately, as we will see in Sect. 4, to design the AVI abstract domain, we do not need to group the complementary generators, since no domain operation requires G^{cc} and all domain operations can be implemented based on only a non-redundant set of complementary generators G^c. In this paper, the notion of G^{cc} is only useful to get Theorem 4 which is interesting as it precisely characterizes the topological properties of $P_{\pm}$ and shows that $P_{\pm}$ is essentially a (possibly) non-convex union of a set of convex polyhedra.

3.2 Conversion from the generator to the constraint representation

Let $G^c = (V^c, R^c)$ be the set of complementary generators of a convex polyhedron $P_{\pm}$ satisfying $\{x \geq 0, (x^+)^T x^- = 0\}$. We now consider the problem of constructing the constraint representation for $P_{\pm}$ from G^c. It can be achieved by the following steps:

1. Consider $G^c = (V^c, R^c)$ as the regular generator representation of some convex polyhedron. Then we use the standard Chernikova's algorithm to compute the corresponding polyhedral constraint representation, i.e., a linear system such as

$$M'x^+ + N'x^- \leq b'$$

2. Add $x^+, x^- \geq 0$ to the above system and remove those constraints from $M'x^+ + N'x^- \leq b'$ that become redundant after adding $x^+, x^- \geq 0$. Suppose we get

$$\begin{aligned} Mx^+ + Nx^- &\leq b \\ x^+, x^- &\geq 0 \end{aligned}$$

3. Add $(x^+)^T x^- = 0$ to the above system, and we get

$$\begin{aligned} Mx^+ + Nx^- &\leq b \\ x^+, x^- &\geq 0 \\ (x^+)^T x^- &= 0 \end{aligned}$$

which will be the XLCP constraint representation for $P_\pm$.

Observe that the resulting XLCP constraint representation for $P_\pm$ is not necessarily non-redundant. However, this does not matter much for designing abstract domains, since a non-redundant generator representation for $P_\pm$ can be ensured.

4 An abstract domain of linear absolute value inequalities

In this section, we propose a new abstract domain, namely the abstract domain of linear absolute value inequalities (AVI). The key point is to use a system of linear absolute value inequalities as the domain representation. AVI can be used to infer relationships of the form $\Sigma_k a_k x_k + \Sigma_k b_k |x_k| \leq c$ over program variables x_k $(k = 1, \ldots, n)$, where constants $a_k, b_k, c \in \mathbb{R}$ are automatically inferred by the analysis.

4.1 Representation

An AVI domain element $\mathbf{P}$ is described as an AVI system $Ax + B|x| \leq c$, where $A, B \in \mathbb{R}^{m\times n}, c \in \mathbb{R}^m$, and m is the number of constraints in the system. It represents the set $\gamma(\mathbf{P}) = \{x \in \mathbb{R}^n \mid Ax + B|x| \leq c\}$, in which each point $x \in \gamma(\mathbf{P})$ represents a possible program environment (or state), i.e., an assignment of numerical/real values to program variables.

From Sect. 2, we know that a linear AV inequality system is equivalent to an interval linear inequality system. Thus the AVI domain is as expressive as the interval polyhedra abstract domain [4]. In other words, each AVI domain element is geometrically an interval polyhedron. Hence, the set of AVI domain elements has the same topological properties as the set of interval polyhedra:

- An AVI domain element is non-convex (even unconnected) in general.
- The intersection of an AVI domain element with each orthant in $\mathbb{R}^n$ gives a (possibly empty) convex polyhedron.

Specifically, from Theorem 6 in Sect. 4.2, we will see that the set union of bounded convex polyhedra with one per each (closed) orthant can be exactly represented by one AVI domain element.

Expressiveness lifting. Note that in the AVI domain representation, absolute value $|\cdot|$ applies to only (single) variables rather than expressions. E.g., consider the relation $y = x - |x+1| + |x-1|$ which encodes the following piecewise linear function

$$y = \begin{cases} x+2 & \text{if } x \leq -1 \\ -x & \text{if } -1 \leq x \leq 1 \\ x-2 & \text{if } x \geq 1 \end{cases}$$

whose plot is shown in Fig. 2. The AVI domain can not express directly this piecewise linear function (in the space of x, y), since $|\cdot|$ applies to two expressions: $x+1$ and $x-1$. Indeed, in Fig. 2 the region in each orthant is not a convex polyhedron.

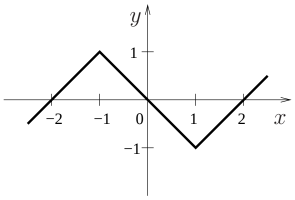


Fig. 2. A piecewise linear function

In order to express such piecewise linear relations, we lift the expressiveness of the AVI domain by introducing new auxiliary variables to denote those expressions that appear inside the AV function. E.g., we could introduce two auxiliary variables v_1, v_2 to denote the values of the expressions $x + 1$ and $x - 1$ respectively. Then using AVI domain elements in the space with higher dimension (involving 4 variables: x, y, v_1, v_2), such as $\{y = x - |v_1| + |v_2|, v_1 = x + 1, v_2 = x - 1\}$, we could express complex piecewise linear relations in the space over lower dimension (involving 2 variables: x, y), such as $y = x - |x + 1| + |x - 1|$. Note that $\{y = x - |v_1| + |v_2|, v_1 = x + 1, v_2 = x - 1\}$ is indeed an AVI domain element. Following the same strategy, we can also express piecewise linear relations with nestings of absolute value functions. E.g., to express $y = ||x| - 1| + ||z| - 2|$, by introducing auxiliary variables v_1, v_2, we could use $\{y = |v_1| + |v_2|, v_1 = |x| - 1, v_2 = |z| - 2\}$.

In fact, a large subclass of piecewise linear functions of practical interest can be represented via AV functions through a so-called canonical (piecewise linear) representation [5], as known in the field of circuits and systems. Thus, most piecewise linear relations of interest in the program could be also expressed by the AVI domain, provided that necessary auxiliary variables are introduced.

4.2 Domain operations

In the convex polyhedra domain, domain operations can be implemented based on the double description method for convex polyhedra [9]. Similarly, we will construct domain operations for the AVI domain based on the double description method for AVI systems (which are equivalent to XLCPs). During the implementation, we maintain the map between abstract environments over x and abstract environments over x^+, x^- as:

$$x = x^+ - x^-, \qquad |x| = x^+ + x^-$$
$$x^+ = \frac{1}{2}(x + |x|), \qquad x^- = \frac{1}{2}(|x| - x)$$

where x^+, x^- satisfy

$$x^+ \geq 0, x^- \geq 0, (x^+)^T x^- = 0$$

More precisely, we will construct domain operations for the AVI domain over x, based on the double description method for XLCP over x^+, x^-. Note that for the implementation of the AVI domain, we need only the set of complementary generators $G^c = (V^c, R^c)$, without resorting to the grouping information G^{cc} of the complementary generators. And the cost of the AVI domain is dominated by the dual conversions between XLCP constraints and complementary generators.

For the sake of simplicity, from now on, we assume that the AVI element $\mathbf{P}$ corresponds to the following XLCP system:

$$\begin{array}{c} Mx^+ + Nx^- \leq b \\ x^+ \geq 0, x^- \geq 0, (x^+)^T x^- = 0 \end{array}$$

and we denote its set of complementary generators as

$$G^c = (V^c, R^c).$$

Now, we describe the implementation of most common domain operations required for static analysis over the AVI domain, some of which require only constraints or generators while some of which require both.

(1) Lattice operations

- Emptiness test: $\mathbf{P}$ is empty, iff $V^c = \emptyset$.

  From now on, let $\mathbf{P}, \mathbf{P}'$ be two non-empty AVI domain elements.
- Inclusion test: $\mathbf{P} \sqsubseteq \mathbf{P}'$ that is $\gamma(\mathbf{P}) \subseteq \gamma(\mathbf{P}')$, iff

$$\forall v \in V^c, M'\, v^+ + N'\, v^- \leq b' \quad \wedge \quad \forall r \in R^c, M'\, r^+ + N'\, r^- \leq 0$$

- Meet: $\mathbf{P} \sqcap \mathbf{P}'$ is an AVI domain element whose XLCP system is

$$\begin{array}{c} Mx^+ + Nx^- \leq b \\ M'x^+ + N'x^- \leq b' \\ x^+ \geq 0, x^- \geq 0, (x^+)^T x^- = 0 \end{array}$$

- Join: $\mathbf{P} \sqcup \mathbf{P}'$ is the least AVI domain element containing $\mathbf{P}$ and $\mathbf{P}'$, whose set of complementary generators is the union of those of $\mathbf{P}$ and $\mathbf{P}'$: $(V^c \cup V'^c, R^c \cup R'^c)$. We show by the following theorem that this join operation is optimal, i.e., its output gives the smallest AVI domain element containing the two input elements.

Theorem 5. *Given two* AVI *domain elements* $\mathbf{P}$ *and* $\mathbf{P}'$*, for any* AVI *domain element* $\mathbf{Q}$ *satisfying* $\gamma(\mathbf{P}) \subseteq \gamma(\mathbf{Q})$ *and* $\gamma(\mathbf{P}') \subseteq \gamma(\mathbf{Q})$*, we have* $\gamma(\mathbf{P} \sqcup \mathbf{P}') \subseteq \gamma(\mathbf{Q})$.

From this theorem together with Theorem 4, we have the following theorem that explores further the expressiveness of the AVI domain.

Theorem 6. *Given a set of bounded convex polyhedra with one per each closed orthant, their set union can be exactly represented by one* AVI *domain element (through the same set of variables).*

Note that, however, Theorem 6 may not hold when one of the input AVI domain elements is not bounded. Theorem 6 implies that given two AVI systems that are bounded, the result of the AVI join is equivalent to the result given by the set union of convex polyhedral hulls in each orthant.

(2) Transfer functions

- Test transfer function: $\tau[\![cx + d|x| \leq e]\!]^{\#}(\mathbf{P})$, whose XLCP system is defined as

$$\begin{aligned} Mx^{+} + Nx^{-} &\leq b \\ (c+d)x^{+} + (d-c)x^{-} &\leq e \\ x^{+} \geq 0, x^{-} \geq 0, (x^{+})^{T}x^{-} &= 0 \end{aligned}$$

- Projection: $\tau[\![x_j := \mathit{random}()]\!]^{\#}(\mathbf{P})$, whose set of complementary generators is defined as $(V^c, R^c \cup \{e_j^+, e_j^-, -e_j^+, -e_j^-\})$, where e_j^+ denotes a canonical basis vector wherein all the components are 0 except $x_j^+ = 1$, and e_j^- denotes a canonical basis vector wherein all the components are 0 except $x_j^- = 1$. Observe that $\tau[\![x_j := \mathit{random}()]\!]^{\#}(\mathbf{P})$ is optimal in the AVI domain, although its result may be less precise than $\exists x_j.\mathbf{P} \overset{\text{def}}{=} \{x[x_j/y] \mid x \in \gamma(\mathbf{P}), y \in \mathbb{R}\}$ which may be not an AVI domain element, where $x[x_j/y]$ denotes the vector x in which the j-th element is replaced with y.
- Assignment transfer function: $\tau[\![x_j := \Sigma_i a_i x_i + \Sigma_i b_i |x_i| + c]\!]^{\#}(\mathbf{P})$, can be modeled using test transfer function, projection and variable renaming as follows:

$$\left(\tau[\![x_j : \mathit{random}()]\!]^{\#} \circ \tau[\![\Sigma_i a_i x_i + \Sigma_i b_i |x_i| + c - x'_j = 0]\!]^{\#}(\mathbf{P})\right)[x'_j/x_j]$$

  Note that the assignment transfer function is optimal but not exact. E.g., assignments may cause a polyhedron in one orthant to cross orthant boundaries. In such case, the result in each orthant is then updated to a possible overapproximation of the polyhedral convex hull of the regions which belong to that orthant after the transfer operation.

(3) Widening

- Widening: Given two AVI domain elements $\mathbf{P} \sqsubseteq \mathbf{P}'$, we define

$$\mathbf{P} \triangledown \mathbf{P}' \overset{\text{def}}{=} \mathcal{S}_1 \cup \mathcal{S}_2 \cup \{x^{+}, x^{-} \geq 0, (x^{+})^{T}x^{-} = 0\}$$

  where

$$\begin{aligned} \mathcal{S}_1 &= \{\, \varphi_1 \in (Mx^{+} + Nx^{-} \leq b) \mid \mathbf{P}' \models \varphi_1 \,\}, \\ \mathcal{S}_2 &= \left\{ \varphi_2 \in (M'x^{+} + N'x^{-} \leq b') \left| \begin{array}{l} \exists\, \varphi_1 \in (Mx^{+} + Nx^{-} \leq b), \\ \gamma(\mathbf{P}) = \gamma((\mathbf{P} \setminus \{\, \varphi_1 \,\}) \cup \{\, \varphi_2 \,\}) \end{array} \right. \right\} \end{aligned}$$

  The above widening for the AVI domain is designed following the same principle as the standard widening of the convex polyhedra domain. The first set $\mathcal{S}_1$ contains all inequalities from the $Mx^{+} + Nx^{-} \leq b$ part of $\mathbf{P}$ that are not violated by the larger $\mathbf{P}'$, while $\mathcal{S}_2$ consists of inequalities from the $M'x^{+} + N'x^{-} \leq b'$ part of $\mathbf{P}'$ that can be exchanged with an inequality from the $Mx^{+} + Nx^{-} \leq b$ part of $\mathbf{P}$ without changing the represented state. $\mathcal{S}_2$ ensures that the result is independent of the (syntactic) representation of $\mathbf{P}$ and $\mathbf{P}'$. Here, we use $\varphi_1 \in (Mx^{+} + Nx^{-} \leq b)$ to denote that φ_1 is one constraint from the system $Mx^{+} + Nx^{-} \leq b$. Let φ_1 be $(cx^{+} + dx^{-} \leq e)$. The entailment $\mathbf{P}' \models \varphi_1$ can be implemented by checking

$$\forall v' \in V'^c, c\,v'^+ + d\,v'^- \le e \quad \wedge \quad \forall r' \in R'^c, c\,r'^+ + d\,r'^- \le 0$$

Next, we use the following example to show in detail how AVI domain operations can be constructed based on the double description method. We choose to show the join operation, since the join is rather interesting (especially when comparing it with polyhedral convex hull of the convex polyhedra domain [9] as well as weak join of the interval polyhedra domain [4]).

Example 3. Consider two AVI domain elements $\mathbf{P}' = \{(x\ y)^T \mid 1 \le x \le 2, -1 \le y \le 1\} = \{(x^+\ x^-\ y^+\ y^-)^T \mid 1 \le x^+ - x^- \le 2, -1 \le y^+ - y^- \le 1, x^+ \ge 0, x^- \ge 0, y^+ \ge 0, y^- \ge 0, x^+x^- = 0, y^+y^- = 0\}$ and $\mathbf{P}'' = \{(x\ y)^T \mid -2 \le x \le -1\} = \{(x^+\ x^-\ y^+\ y^-)^T \mid -2 \le x^+ - x^- \le -1, x^+ \ge 0, x^- \ge 0, y^+ \ge 0, y^- \ge 0, x^+x^- = 0, y^+y^- = 0\}$, shown in Figure 3(1). Note that $\mathbf{P}'$ is a bounded convex polyhedron while $\mathbf{P}''$ is an unbounded convex polyhedron. And the polyhedral convex hull of $\mathbf{P}'$ and $\mathbf{P}''$ results in $\{(x\ y)^T \mid -2 \le x \le 2\}$ that is a convex polyhedron. Since $\mathbf{P}''$ is unbounded, we can not apply Theorem 6 to the set union of $\mathbf{P}'$ and $\mathbf{P}''$ which indeed cannot be exactly described by any AVI domain element (through the same set of variables).

First, if we omit the condition $x^+x^- = 0 \wedge y^+y^- = 0$, the set of regular (polyhedral) generators for $\mathbf{P}'$ and $\mathbf{P}''$ over $(x^+, x^-, y^+, y^-)^T$ will be respectively

$$(V_{\mathbf{P}'}, R_{\mathbf{P}'}) = \left(\begin{pmatrix} x^+ \\ x^- \\ y^+ \\ y^- \end{pmatrix} : \left\{\begin{pmatrix}1\\0\\0\\0\end{pmatrix},\begin{pmatrix}2\\0\\0\\0\end{pmatrix},\begin{pmatrix}1\\0\\0\\1\end{pmatrix},\begin{pmatrix}2\\0\\0\\1\end{pmatrix},\begin{pmatrix}1\\0\\1\\0\end{pmatrix},\begin{pmatrix}2\\0\\1\\0\end{pmatrix}\right\}, \left\{\begin{pmatrix}1\\1\\0\\0\end{pmatrix},\begin{pmatrix}0\\0\\1\\1\end{pmatrix}\right\}\right)$$

$$(V_{\mathbf{P}''}, R_{\mathbf{P}''}) = \left(\begin{pmatrix} x^+ \\ x^- \\ y^+ \\ y^- \end{pmatrix} : \left\{\begin{pmatrix}0\\1\\0\\0\end{pmatrix},\begin{pmatrix}0\\2\\0\\0\end{pmatrix}\right\}, \left\{\begin{pmatrix}0\\0\\0\\1\end{pmatrix},\begin{pmatrix}0\\0\\1\\0\end{pmatrix},\begin{pmatrix}1\\1\\0\\0\end{pmatrix}\right\}\right)$$

If we take into account $x^+x^- = 0 \wedge y^+y^- = 0$, we get the sets of complementary generators for $\mathbf{P}'$and $\mathbf{P}''$ over $(x^+, x^-, y^+, y^-)^T$:

$$(V^c_{\mathbf{P}'}, R^c_{\mathbf{P}'}) = \left(\begin{pmatrix} x^+ \\ x^- \\ y^+ \\ y^- \end{pmatrix} : \left\{\begin{pmatrix}1\\0\\0\\0\end{pmatrix},\begin{pmatrix}2\\0\\0\\0\end{pmatrix},\begin{pmatrix}1\\0\\0\\1\end{pmatrix},\begin{pmatrix}2\\0\\0\\1\end{pmatrix},\begin{pmatrix}1\\0\\1\\0\end{pmatrix},\begin{pmatrix}2\\0\\1\\0\end{pmatrix}\right\}, \emptyset\right)$$

$$(V^c_{\mathbf{P}''}, R^c_{\mathbf{P}''}) = \left(\begin{pmatrix} x^+ \\ x^- \\ y^+ \\ y^- \end{pmatrix} : \left\{\begin{pmatrix}0\\1\\0\\0\end{pmatrix},\begin{pmatrix}0\\2\\0\\0\end{pmatrix}\right\}, \left\{\begin{pmatrix}0\\0\\0\\1\end{pmatrix},\begin{pmatrix}0\\0\\1\\0\end{pmatrix}\right\}\right)$$

And $(V^c_{\mathbf{P}'} \cup V^c_{\mathbf{P}''}, R^c_{\mathbf{P}'} \cup R^c_{\mathbf{P}''})$ will be the set of complementary generators for $\mathbf{P}' \sqcup \mathbf{P}''$, i.e.,

$$\left(\begin{pmatrix} x^+ \\ x^- \\ y^+ \\ y^- \end{pmatrix} : \left\{\begin{pmatrix}1\\0\\0\\0\end{pmatrix},\begin{pmatrix}2\\0\\0\\0\end{pmatrix},\begin{pmatrix}1\\0\\0\\1\end{pmatrix},\begin{pmatrix}2\\0\\0\\1\end{pmatrix},\begin{pmatrix}1\\0\\1\\0\end{pmatrix},\begin{pmatrix}2\\0\\1\\0\end{pmatrix},\begin{pmatrix}0\\1\\0\\0\end{pmatrix},\begin{pmatrix}0\\2\\0\\0\end{pmatrix}\right\}, \left\{\begin{pmatrix}0\\0\\0\\1\end{pmatrix},\begin{pmatrix}0\\0\\1\\0\end{pmatrix}\right\}\right)$$

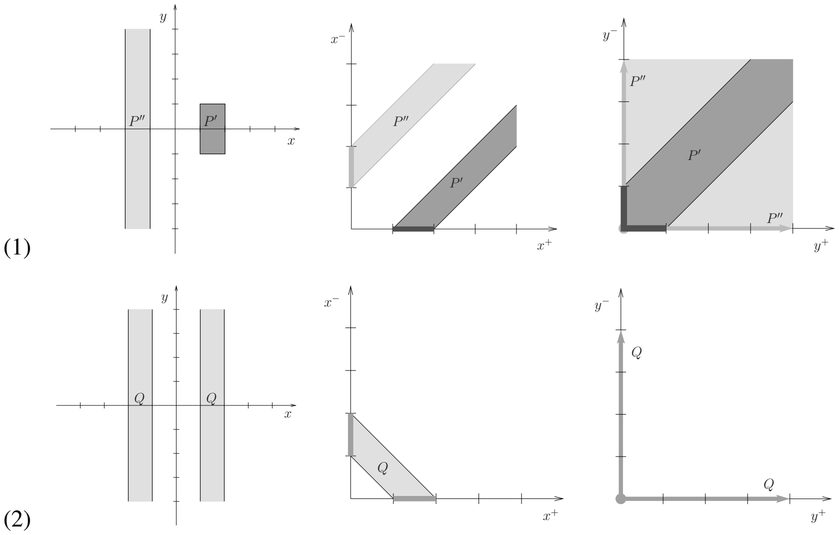
Then, by converting it into the constraint representation, we will get the following XLCP system for $\mathbf{P}' \sqcup \mathbf{P}''$:

$$\{1 \le x^+ + x^- \le 2, x^+ \ge 0, x^- \ge 0, x^+x^- = 0, y^+ \ge 0, y^- \ge 0, y^+y^- = 0\}$$

Finally, we can get the following AVI representation for $\mathbf{P}' \sqcup \mathbf{P}''$:

$$\mathbf{P}' \sqcup \mathbf{P}'' = \{(x\ y)^T \mid 1 \le |x| \le 2\}$$

And the regions of the inputs $\mathbf{P}'$ and $\mathbf{P}''$ together with the output $\mathbf{Q}$ of the AVI join operation are shown in Figure 3.



(a) xy (b) x^+x^- (c) y^+y^-

Fig. 3. Subfigure (1) shows the two input AVI domain elements of the join: $\mathbf{P}' = \{1 \le x \le 2, -1 \le y \le 1\}$ and $\mathbf{P}'' = \{-2 \le x \le -1\}$, while subfigure (2) shows the join over the AVI domain: $\mathbf{Q} = \mathbf{P}' \sqcup \mathbf{P}'' = \{1 \le |x| \le 2\}$. The columns (a), (b), (c) depict the regions over the xy, x^+x^-, y^+y^- planes respectively.

5 Implementation and experimental results

Our prototype domain, rAVI, is developed based on Sect. 4 using multi-precision rational numbers. It makes use of GMP (the GNU Multiple Precision arithmetic library) [1] and NewPolka [15] that is a rational implementation of the convex polyhedra domain. rAVI is interfaced to the Apron numerical abstract domain library [15]. Our experiments were conducted using the Interproc [16] static analyzer. In order to assess the precision and efficiency of rAVI, we compare the obtained invariants and the performance of rAVI with NewPolka as well as our previous work *itvPol* which is a sound floating-point implementation of the interval polyhedra domain [4].

To demonstrate the expressiveness of rAVI, two simple programs are shown in Figs. 4-5, together with the invariants generated by the analyzer. In Fig. 4, for *AVtest1*, the initial state consists of four points that are respectively from 4 different orthants over the x-y plane: $(1, 1), (-1, 1), (-1, -1), (1, -1)$. The loop increases the values of x and y in each orthant simultaneously, along the direction $y = x$ and $y = -x$ respectively. At program point ①, rAVI can prove that $|y| = |x| \wedge |x| \geq 1$ while NewPolka obtains no information. *itvPol* can only prove that $[-1, 1]x \leq -1 \wedge [-1, 1]y \leq -1$ (i.e., $|x| \geq 1 \wedge |y| \geq 1$) and thus can not find any relations among x and y due to the weak join used in *itvPol* [4].

```
real x, y;
assume x = 1 or x = −1;
assume y = 1 or y = −1;
while (true) {
① if (x ≥ 0) { x := x + 1; }
   else        { x := x − 1; }
   if (y ≥ 0) { y := y + 1; }
   else        { y := y − 1; }
}
```

Loc	NewPolka	*itvPol*	rAVI
①	$\top$ (no information)	$[-1, 1]x \leq -1$ $\wedge[-1, 1]y \leq -1$	$\|x\| = \|y\| \wedge \|x\| \geq 1$

Fig. 4. Program AVtest1 (left) and the generated invariants (right)

The program CmplxTest1 shown in Fig. 5 comes from [13] where it is used as an example for analyzing time complexity of the program. Here, we modify a bit the program by introducing a fresh variable t to denote the value of $n - x0$. The main goal is to find an upper bound for the loop counter i. However, NewPolka and *itvPol* can not find any upper bound for i, while rAVI can prove that $i \leq \frac{1}{2}(|t|+t)$, i.e., $i \leq \max(0, n-x0)$, which shows that the time complexity of CmplxTest1 is $\max(0, n - x0)$ in terms of the input parameters $x0, n$.

```
CmplxTest1(int x0, n)
    x := x0; i := 0;
    t := n − x0;
    while (x < n) {
      i := i + 1;
      x := x + 1;
    } ①
```

Loc	NewPolka	*itvPol*	rAVI
①	$i \geq 0$ $\wedge\, i = x - x0$	$i \geq 0$	$i \geq 0$ $\wedge\, i = x - x0$ $\wedge\, i \leq \frac{1}{2}(\|t\| + t)$ $\wedge \ldots$

Fig. 5. Program CmplxTest1 (left) and the generated invariants related to i (right)

Table 1 shows the comparison of performance and result invariants for a selection of benchmark examples. Programs AVtest1, CmplxTest1 correspond to those programs shown in Figs. 4-5. CmplxTest1-3 come from [13] wherein they are used for analyzing time complexity of programs. program4 and program5 come from our previous work [4]. “#vars” indicates the total number of program variables in each program. And for each program, the value of the widening delay parameter for Interproc is set to 1. “#iter.” gives the number of increasing iterations during the analysis.

Table 1. Experimental results for benchmark examples

Program		NewPolka		itvPol		rAVI		Res.	
name	#vars	#iter.	$t(ms)$	#iter.	$t(ms)$	#iter.	$t(ms)$	Inv.	
AVtest1	2	4	11	4	45	4	48	⊐	⊐
AVtest2	2	4	8	3	14	4	31	⊐	⊐
AVtest3	2	4	9	4	16	5	73	⊐	⊐
CmplxTest1	5	4	7	4	26	4	57	⊐	⊐
CmplxTest2	5	6	10	6	34	6	150	⊐	⊐
CmplxTest3	8	4	17	4	242	4	310	⊐	⊐
program4	1	5	2	4	4	4	10	⊐	=
program5	2	6	9	5	20	8	45	⊐	⊐

Invariants. The column “Res. Inv.” compares the invariants obtained. The left sub-column compares rAVI with NewPolka while the right sub-column compares rAVI with itvPol. A “⊐” indicates that rAVI outputs stronger invariants than NewPolka or *itvPol*, while a “=” indicates that rAVI outputs equivalent invariants as NewPolka or *itvPol*. The results in Table 1 show that rAVI outputs stronger invariants than NewPolka for all these examples. Note that traditional convex domains (such as the convex polyhedra domain) are not fit for the benchmark examples shown in Table 1, since these programs involve non-convex properties that are out of the expressiveness of convex domains.

And in most cases, rAVI outputs stronger invariants than *itvPol*, although the two domains have the same expressiveness. This is because domain operations in rAVI are optimal while most domain operations in *itvPol* are weak (e.g., the join operation). For *program4*, the two domains generate equivalent invariants, because this program involves only one variable and most domain operations in *itvPol* become optimal in this case. During the experiments, we observed that most linear absolute value invariants generated by rAVI are essentially due to piecewise linear behaviors in the program, e.g., branches inside loops. In the examples CmplxTest1-3 that are used to show time complexity, the piecewise linear behaviors mainly come from case by case discussions over the difference between the loop counter and the input parameter (or the initial value), e.g., whether the difference is greater than 0 or not.

Performance. The column “t(ms)” presents the analysis times in milliseconds when the analyzer runs on a 2.4GHz PC with 2GB of RAM running Fedora Linux. From Table 1, we can see that rAVI is much less efficient than NewPolka, because for these examples the polyhedra generated by NewPolka during the analysis are rather simple (with very few or even no non-trivial constraints). Similarly, we can see that rAVI is less efficient than *itvPol*, because *itvPol* is implemented based on floating-point arithmetic and also because domain operations in *itvPol* are weak operations with low computational cost.

6 Related work

In numerical static analysis, linear relations are considered as the most important kind of numerical relations among variables of a program. The convex polyhedra abstract domain [9], devoted to linear relation analysis, is one of the earliest but still remains one of the most powerful and commonly used numerical abstract domains. For the sake of efficiency, a variety of weakly relational abstract domains are designed as subdomains of the convex polyhedra domain, such as the Octagon domain [21], the Two Variables Per Inequality (TVPI) domain [28], the Template Polyhedra domain [25], and the SubPolyhedra domain [17]. However, this paper goes the other direction. Rather than aiming at discovering restricted forms of linear relations, we generalize the linear relation analysis to linear absolute value relation analysis that allows discovering a kind of piecewise linear relations.

Numerical abstract domains often use conjunctions of convex constraints as the domain representation, and thus most domains can only represent convex sets. The convexity limitations may lead to imprecision during analysis. To deal with disjunctions, a known solution in abstract interpretation is to use disjunctive completion [8,11], such as powerset extension. However, it can be very costly and widening operators for such domains are difficult to design [3].

There also exists much work on elaborating the control flow information of the program to improve the precision. Rival and Mauborgne [22] proposed the trace partitioning abstract domain, which is based on the partitioning of program traces. Sankaranarayanan et al. [24] showed that a fixed point computed over a powerset extension corresponds to a fixed point over the base domain computed on an elaboration of the control flow graph of the program. Simon [27] used a boolean flag to encode the union of two polyhedra and to perform control flow splitting when necessary.

This paper aims at designing abstract domains that can natively encode non-convex information. Until now, few existing abstract domains natively allow representing non-convex sets, e.g., congruences [12], max-plus polyhedra [2], domain lifting by max expressions [13], interval polyhedra [4].

The AVI domain that we introduce in this paper is closest to our previous work on the interval polyhedra domain [4]. The AVI domain is as expressive as the interval polyhedra domain, but differs from it in the following respects: First, the AVI domain enjoys optimal domain operations while operations in the interval polyhedra domain are not optimal; Second, for representation, the AVI domain uses the double description method while the interval polyhedra domain uses solely constraints; Third, to implement domain operations, the AVI domain employs Chernikova's algorithm while the interval polyhedra domain employs linear programming and Fourier-Motzkin elimination algorithms; Finally, prototype rAVI for the AVI domain is implemented via rational numbers while prototype *itvPol* for the interval polyhedra domain in [4] is implemented via floating point numbers.

7 Conclusion

In this paper, we present an analysis to discover linear absolute value relations among variables of a program ($\Sigma_k a_k x_k + \Sigma_k b_k |x_k| \leq c$), which generalizes the classic linear relation analysis ($\Sigma_k a_k x_k \leq c$) [9]. The analysis explores absolute value (AV) to describe piecewise linear relations in the program, as a mean to deal with non-convex or non-linear behaviors in the program. First, we show the equivalence among linear AV inequality systems, extended linear complementarity problem (XLCP) systems and interval linear inequality systems. The equivalence implies that linear AV relation analysis can be reused to infer other kinds of equivalent relations in a program, such as interval linear relations which is of high relevance in numeric static analysis [4]. Then, we construct a double description method for XLCP on top of that for convex polyhedra. On this basis, we propose an abstract domain of linear AV inequalities that natively allows expressing non-convex properties and enjoys optimal transfer functions. The AVI domain is implemented using rational numbers based on the double description method for XLCP. Experimental results are encouraging: The AVI domain can discover interesting non-convex properties, especially for programs involving piecewise linear behaviors.

It remains for future work to consider automatic methods to introduce auxiliary variables on the fly that can be used inside the AV function to improve the precision of AVI analysis. Another direction of work is to consider weakly relational abstract domains over absolute value, with less expressiveness but higher efficiency.

References

1. Gnu multiple precision arithmetic library. `http://gmplib.org/`.
2. X. Allamigeon, S. Gaubert, and E. Goubault. Inferring min and max invariants using max-plus polyhedra. In *SAS'08*, volume 5079 of *LNCS*, pages 189–204. Springer Verlag, 2008.
3. R. Bagnara, P. M. Hill, and E. Zaffanella. Widening operators for powerset domains. In *VMCAI'04*, volume 2937 of *LNCS*, pages 135–148. Springer Verlag, 2004.
4. L. Chen, A. Miné, J. Wang, and P. Cousot. Interval polyhedra: An abstract domain to infer interval linear relationships. In *SAS'09*, volume 5673 of *LNCS*, pages 309–325. Springer Verlag, 2009.
5. L.O. Chua and A.-C. Deng. Canonical piecewise-linear representation. *IEEE Trans. on Circuits and Systems*, 35(1):101–111, 1988.
6. R.W. Cottle, J.-S. Pang, and R.E. Stone. *The Linear Complementarity Problem*. Academic Press, New York, 1992.
7. P. Cousot and R. Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *ACM POPL'77*, pages 238–252. ACM Press, New York, 1977.
8. P. Cousot and R. Cousot. Systematic design of program analysis frameworks. In *ACM POPL'79*, pages 269–282. ACM Press, New York, 1979.
9. P. Cousot and N. Halbwachs. Automatic discovery of linear restraints among variables of a program. In *ACM POPL'78*, pages 84–96. ACM Press, New York, 1978.
10. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.
11. R. Giacobazzi and F. Ranzato. Optimal domains for disjunctive abstract interpretation. *Sci. Comput. Program*, 32(1-3):177–210, 1998.

12. P. Granger. Static analysis of arithmetical congruences. *International Journal of Computer Mathematics*, pages 165–199, 1989.
13. B. S. Gulavani and S. Gulwani. A numerical abstract domain based on expression abstraction and max operator with application in timing analysis. In *CAV'08*, volume 5123 of *LNCS*, pages 370–384. Springer-Verlag, 2008.
14. N. Halbwachs, Y.E. Proy, and P. Roumanoff. Verification of real-time systems using linear relation analysis. *Formal Methods in System Design*, 11(2):157–185, 1997.
15. B. Jeannet and A. Miné. Apron: A library of numerical abstract domains for static analysis. In *CAV'09*, volume 5643 of *LNCS*, pages 661–667. Springer, 2009.
16. G. Lalire, M. Argoud, and B. Jeannet. Interproc. `http://pop-art.inrialpes.fr/people/bjeannet/bjeannet-forge/interproc/`.
17. V. Laviron and F. Logozzo. Subpolyhedra: A (more) scalable approach to infer linear inequalities. In *VMCAI'09*, volume 5403 of *LNCS*, pages 229–244. Springer Verlag, 2009.
18. H. LeVerge. A note on Chernikova's algorithm. Technical Report 635, IRISA, France, 1992.
19. O. L. Mangasarian and J. S. Pang. The extended linear complementarity problem. *SIAM J. Matrix Anal. Appl.*, 16(2):359–368, 1995.
20. A. Miné. Relational abstract domains for the detection of floating-point run-time errors. In *ESOP'04*, volume 2986 of *LNCS*, pages 3–17. Springer, 2004.
21. A. Miné. The octagon abstract domain. *Higher-Order and Symbolic Computation*, 19(1):31–100, 2006.
22. X. Rival and L. Mauborgne. The trace partitioning abstract domain. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 29(5), 2007.
23. J. Rohn. Solvability of systems of interval linear equations and inequalities. In *Linear Optimization Problems with Inexact Data*, pages 35–77. Springer, 2006.
24. S. Sankaranarayanan, F. Ivancic, I. Shlyakhter, and A. Gupta. Static analysis in disjunctive numerical domains. In *SAS'06*, volume 4134 of *LNCS*, pages 3–17. Springer, 2006.
25. S. Sankaranarayanan, H. Sipma, and Z. Manna. Scalable analysis of linear systems using mathematical programming. In *VMCAI'05*, volume 3385 of *LNCS*, pages 25–41. Springer Verlag, 2005.
26. A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, Inc., 1986.
27. A. Simon. Splitting the Control Flow with Boolean Flags. In *SAS'08*, volume 5079 of *LNCS*, pages 315–331. Springer, 2008.
28. A. Simon, A. King, and J. M. Howe. Two Variables per Linear Inequality as an Abstract Domain. In *LOPSTR'03*, volume 2664 of *LNCS*, pages 71–89. Springer, 2003.