Octagons Revisited
Elegant Proofs and Simplified Algorithms

Michael Schwarz[0000−0002−9828−0308] and Helmut Seidl[0000−0002−2135−1593]

Technische Universität München, Garching, Germany
{m.schwarz, helmut.seidl}@tum.de

Abstract. Weakly relational domains have enjoyed tremendous success in the area of program analysis, since they offer a decent compromise between precision and efficiency. *Octagons*, in particular, have widely been studied to obtain efficient algorithms which, however, come with intricate correctness arguments. Here, we provide simplified cubic time algorithms for computing the closure of *Octagon* abstract relations both over the rationals and the integers which avoid introducing auxiliary variables. They are based on a more general formulation by means of 2-*projective* domains which allows for an elegant short correctness proof. The notion of 2-projectivity also lends itself to efficient algorithms for incremental normalization. For the *Octagon* domain, we also provide an improved construction for linear programming based best abstract transformers for affine assignments.

Keywords: weakly relational domains, octagons, 2-decomposable relational domains, Floyd-Warshall algorithm

1 Introduction

While for intricate verification tasks, monolithic relational domains such as the polyhedra abstract domain [8] are indispensable, they are considered prohibitively expensive. Therefore, *weakly relational* domains have been proposed which can only express simple relational properties, but scale better to larger programs. Examples of such domains to capture numerical properties are the *Two Variables Per Inequality* domain [27], or domains given by a finite set of *linear templates* [25]. The most prominent example of a template numerical domain is the *Octagon* domain [20, 21] which allows tracking upper and lower bounds not only of program variables but also of sums and differences of *two* program variables. One such octagon abstract relation could, e.g., be given by the conjunction

$$(-x \leq -5) \wedge (x \leq 10) \wedge (x + y \leq 0) \wedge (x - z \leq 1)$$

Octagons thus can be considered as a mild extension of the non-relational domain of *Intervals* for program variables. An efficient comparison of octagon abstract relations for inclusion, is enabled by canonical representations where all implied

bounds are made explicit. Such representations are called *closed*. In the given example, the upper bounds

$$(y \leq -5) \wedge (-z \leq -4)$$

are implied and therefore are included into the closed representation.

Procedures for computing closures of octagons over rationals or integers have been given by Miné [20] where an improved closure algorithm for integers later has been provided by Bagnara et al. [1, 2]. Further practical improvements are discussed in [4]. All these algorithms have in common that they introduce auxiliary variables for negated program variables $-z$ in order to represent each octagon as a difference bound matrix (DBM), and then apply dedicated techniques for these [19], namely, the *Floyd-Warshall* algorithm [6]. The auxiliary variables, however, must additionally be taken care of by the algorithm which blurs the simplicity of the idea, and also complicates the correctness argument.

Here, we take another approach. To provide efficient procedures for the *Octagon* domain with simple proofs, we identify two generic properties of relational domains which are sufficient for an abstract version of the *Floyd-Warshall* algorithm to provide *normal forms*. Normalization takes calculations on abstract relations between 1, 2, and 3 variables as black boxes and uses these to infer abstract 1 or 2-variable relations mediated by other variables. Our normalization algorithm can be instantiated for rational octagons as well as integer octagons or other instances of the class of weakly relational domains satisfying our criteria.

The first criterion is 2-*decomposability* as introduced in [26] which requires that each abstract relation can be uniquely reconstructed from its projections onto sub-clusters of variables of size at most 2. The second criterion is called *2-projectivity*. This property means that each variable x can be eliminated from an abstract relation by considering projections onto at most 2-variable clusters. If both criteria are satisfied, our algorithm returns the normal form. The key correctness argument can be provided on two pages. Our abstract setting also provides an elegant algorithm for *incremental* normalization, i.e., for reestablishing the normal form after improving the relationship between two variables. In practice, such improvements may occur as the abstract effect of guards in the program which are expressible as abstract relations. For the *Octagon* domain over rationals or integers, we provide improved abstract transformers for affine assignments based on linear programming.

2 Relational Domains

Let us recall basic definitions for relational domains. We mostly follow the notation used in [26] where the notion of 2-decomposability has been introduced. Let $\mathcal{X}$ be some finite set of variables. A *relational domain* $\mathcal{R}$ is a lattice with least element $\bot$ and greatest element $\top$ which provides the monotonic operations

$$\begin{aligned} [\![x \leftarrow e]\!]^\sharp &: \mathcal{R} \to \mathcal{R} \text{ (assignment to variable } x \text{ with right-hand } e) \\ r|_Y &: \mathcal{R} \to \mathcal{R} \text{ (restriction to } Y \subseteq \mathcal{X}) \\ [\![?c]\!]^\sharp &: \mathcal{R} \to \mathcal{R} \text{ (guard for condition } c) \end{aligned}$$

for some languages e of expressions and c of conditions, respectively.

The given operations are meant to provide the abstract transformers for the basic operations of programs. Restricting a relation r to a subset Y of variables amounts to *forgetting* all information about variables in $\mathcal{X} \setminus Y$. Thus, we require that

$$\begin{array}{lll} r|_{\mathcal{X}} & = r & \\ r|_{\emptyset} & = \top & \\ r|_{Y_1} & \sqsupseteq r|_{Y_2} & \text{when } Y_1 \subseteq Y_2 \\ (r|_{Y_1})|_{Y_2} & = r|_{Y_1 \cap Y_2} & \end{array} \tag{1}$$

Restriction therefore is *idempotent*. For guards with condition c, we require that

$$[\![?c]\!]^\sharp r = r \sqcap [\![?c]\!]^\sharp (r|_V) \tag{2}$$

where V is the set of variables occurring inside c.

For a *numerical* relational domain, we additionally require for $Y \subseteq \mathcal{X}$ that

$$([\![x \leftarrow e]\!]^\sharp r)\big|_Y = r|_Y \qquad (x \notin Y) \tag{3}$$

$$([\![x \leftarrow e]\!]^\sharp r)\big|_Y = ([\![x \leftarrow e]\!]^\sharp (r|_{Y \cup V}))\big|_Y \qquad (x \in Y) \tag{4}$$

where V is the set of variables occurring in e. Intuitively, this means that an assignment to the variable x does not affect relational information for any set Y of variables with $x \notin Y$. To determine the effect for a set Y of variables containing x, it suffices to additionally take the variables into account which occur in the right-hand side e. This property may, e.g., be violated if the relational domain also represents points-to information so that updates to x may also affect relational information for sets of variables not containing x.

Example 1. For numerical variables, a variety of such relational domains have been proposed, e.g., (conjunctions of) *affine equalities* [16, 22, 23] or *affine inequalities* [8]. For affine equalities or inequalities, projection onto a subset of Y of variables corresponds to the geometric projection onto the sub-space defined by Y, combined with arbitrary values for variables $z \notin Y$. The abstract effect of a guard c onto a given conjunction r can be realized as $r \wedge c = r \wedge (c \wedge r|_V)$ if c is a linear equality or inequality, respectively, using variables from V. The abstract effect of an assignment $x \leftarrow e$ with affine right-hand side e, finally, can be reduced to the addition of new constraints and projection onto sub-spaces. Relational domains may also be constructed for non-numerical values, e.g., by maintaining *finite* subsets of value maps. □

3 Weakly Relational Domains

One way to tackle the high cost of relational domains is to track relationships not between all variables, but only between *subclusters* of variables. We call such domains *Weakly Relational Domains*.

For a subset $Y \subseteq \mathcal{X}$, let $\mathcal{R}^Y = \{r \mid r \in \mathcal{R}, r|_Y = r\}$ the set of all abstract values from $\mathcal{R}$ that contain only information on those variables in Y. For any

collection $\mathcal{S} \subseteq 2^{\mathcal{X}}$ of *clusters* of variables, a relation $r \in \mathcal{R}$ can be *approximated* by a meet of relations from $\mathcal{R}^Y, Y \in \mathcal{S}$ since for every $r \in \mathcal{R}$,

$$r \sqsubseteq \bigsqcap \{ r|_Y \mid Y \in \mathcal{S} \} \tag{5}$$

holds. Schwarz et al. [26] introduce the notion of 2-*decomposable* relational domains. These are domains where the full value can be recovered from the restriction to all clusters $[\mathcal{X}]_2$ of variables of size at most 2, and all finite least upper bounds can be recovered by computing within these clusters only, i.e., where

$$r = \bigsqcap \left\{ r|_p \mid p \in [\mathcal{X}]_2 \right\} \tag{6}$$

$$(\bigsqcup R)\,|_p = \bigsqcup \left\{ r|_p \mid r \in R \right\} \quad (p \in [\mathcal{X}]_2) \tag{7}$$

holds for each abstract relation $r \in \mathcal{R}$ and each finite set of abstract relations $R \subseteq \mathcal{R}$. The most prominent example of a 2-decomposable domain is the *Octagon* domain [20] – either over rationals or integers, while *affine equalities* or *affine inequalities* are examples of domains that are not 2-decomposable.

Each value r from a 2-decomposable relational domain $\mathcal{R}$ can be represented as the meet of its restrictions to 2-clusters, i.e., by the collection $\left\langle r|_p \right\rangle_{p \in [\mathcal{X}]_2}$. This representation is called 2-*normal*, and an algorithm to compute it, *normalization*. Consider an *arbitrary* collection $\langle s_p \rangle_{p \in [\mathcal{X}]_2}$ with $s_p \in \mathcal{R}^p$ with $r = \bigsqcap \{ s_p \mid p \in [\mathcal{X}]_2 \}$. Then $r|_p \sqsubseteq s_p$ always holds, while equality need not hold. In the *Octagon* domain over the rationals or the integers, the 2-normal representation of an octagon value corresponds to its *strong closure* and *tight closure*, respectively, as described in [1, 20]. Here, we do not distinguish between different types of closure for rational and integer octagons. Instead, we call a non-$\bot$ octagon O over a numerical set of values $\mathbb{I} \in \{\mathbb{Q}, \mathbb{Z}\}$ *closed* if for each octagon combination ℓ, the upper bound b_ℓ equals the minimal value $b \in \mathbb{I}$ such that $\ell \leq b$ is implied by O, or ∞ if no such bound exists.

While for rational octagons, closure in cubic time was already proposed by Miné [20], it is much more recent that a corresponding algorithm was provided for integer octagons [1, 2]. Here, we re-consider these results. By referring to 2-decomposable domains instead of to octagons, we succeed in providing a conceptually simple normalization algorithm with a simple correctness proof, from which cubic closure algorithms for the *Octagon* domains can be derived.

4 2-Projectivity

Subsequently, we assume that $\mathcal{R}$ is an arbitrary 2-decomposable domain over some set $\mathcal{X}$ of variables. Assume that $r \in \mathcal{R}$ is given by $r = \bigsqcap \{ s_p \mid p \in [\mathcal{X}]_2, s_p \in \mathcal{R}^p \}$. Then, we consider the following constraint system in the unknowns $r_p, p \in [\mathcal{X}]_2$, over $\mathcal{R}$,

$$r_{\{x,y\}} \sqsubseteq s_{\{x,y\}} \sqcap \left(r_{\{x,z\}} \sqcap r_{\{z,y\}} \right)|_{\{x,y\}} \tag{8}$$

for $x, y, z \in \mathcal{X}$. All right-hand sides of the constraint system (8) are monotonic.

Proposition 1. *The collection* $\langle r|_p \rangle_{p \in [\mathcal{X}]_2}$ *is a solution of constraint system* (8).

Proof. Let $x, y, z \in \mathcal{X}$. Then

$$r|_{\{x,y\}} = r|_{\{x,y\}} \sqcap r|_{\{x,y\}} \sqsubseteq s_{\{x,y\}} \sqcap r|_{\{x,y\}} \sqsubseteq s_{\{x,y\}} \sqcap \left(r|_{\{x,z\}} \sqcap r|_{\{z,y\}} \right) \Big|_{\{x,y\}} \quad \square$$

From Proposition 1, we conclude that the *greatest* solution of (8) – if it exists – is an overapproximation of the normal representation of r. In general, the Kleene fixpoint iteration for computing greatest solutions of constraint systems (8) may not terminate. Let us call a 2-decomposable relational domain $\mathcal{R}$ 2-*projective* when from each abstract relation r, each single variable can be eliminated by using projections onto clusters from $[\mathcal{X}]_2$ only, i.e., when for every $Y \subseteq \mathcal{X}$, $z \in \mathcal{X} \setminus Y$, $y_j \in Y \cup \{z\}$, $r' \in \mathcal{R}^Y$, and $r_{\{z,y_j\}} \in \mathcal{R}^{\{z,y_j\}}$,

$$\left(r_{\{z,y_1\}} \sqcap \ldots \sqcap r_{\{z,y_k\}} \sqcap r' \right) \big|_Y = r' \sqcap \prod_{i,j=1}^{k} \left(r_{\{z,y_i\}} \sqcap r_{\{z,y_j\}} \right) \big|_{Y \cap \{y_i, y_j\}} \tag{9}$$

Proposition 2. *The following* 2*-decomposable domains are 2-projective:*

1. *rational octagons;*
2. *integer octagons;*
3. 2*-variable rational affine inequalities;*
4. 2*-variable rational affine equalities.*

Proof. Let us consider the claims (1) and (2) for octagons. Intuitively, their correctness follows from the correctness of *Fourier-Motzkin* elimination of a single variable z from a system of inequalities. In general, this holds only for rational inequalities as considered for claim (1). However, it also holds for systems of integer inequalities – given that all coefficients are integer and all non-zero coefficients of z are either 1 or -1.

Let us call a linear combination $\sum_{x \in \mathcal{X}} a_x \cdot x$ an *octagon* combination if at most two of the coefficients a_x are non-zero and these are then from $\{-1, 1\}$. For a subset Y of variables, let L_Y denote the set of all octagon combinations with variables from Y. An integer octagon constraint is of the form $\ell \leq b$ where ℓ is a linear octagon combination and the bound b is integer or ∞.

Subsequently, we represent an abstract octagon relation over Y by a *closed* conjunction

$$\bigwedge_{\ell \in L_Y} \ell \leq b_\ell \tag{10}$$

of octagon constraints with variables from Y if the octagon is satisfiable, or $\bot$ if it is not. Here, the conjunction (10) is satisfiable and closed iff

$$\begin{array}{ll} 0 \leq b_\ell + b_{-\ell} & \text{if } \ell \in L_Y \\ b_\ell \leq (b_{\ell_1} + b_{\ell_2})/c & \text{if } \ell_1 \neq \ell_2 \text{ and } c \cdot \ell = \ell_1 + \ell_2 \end{array}$$

holds for some $c \in \{1, 2\}$. Here, factor 2 occurs if one variable x occurs both in ℓ_1 and ℓ_2 with the same sign, while another variable y occurs with different signs, i.e.,

$$c \cdot \ell = (x + y) + (x - y) = 2 \cdot x$$

In case of octagons over rationals, the operator "/" denotes division, whereas in case of octagons over integers, it denotes *integer* division, i.e., may include rounding downwards. By definition, the closed representation of an abstract octagon relation is also 2-normal.

For computing the closure for an arbitrary conjunction r of octagon constraints with one or two variables only, we may first determine the least given upper bound b_ℓ for each occurring octagon linear combination ℓ. As a result, we obtain at most 8 octagon constraints for which satifiability (over rationals or integers) can be decided in constant time. Provided the conjunction is satisfiable, all implied tighter upper bounds (over rationals or integers) can be inferred.

Example 2. Consider the integer octagon given by conjunction of the constraints

$$x + y \leq -2 \qquad x - y \leq 5 \qquad -x + y \leq 0$$

By adding up constraints with positive and negative occurrences of the same variable, we derive that

$$y \leq -1 \qquad x \leq 1$$

must also hold, while no further bounds can be inferred. If the conjunction of octagon constraints additionally has the inequality

$$-x - y \leq 0$$

then, by adding this to the first inequality, we derive

$$0 \leq -2$$

– which is false – implying that the octagon equals $\bot$. □

Assume that each non-$\bot$ value $r_{\{y_j,z\}}$, $y_j \in Y \cup \{z\}$, is represented as a *closed* conjunction of octagon constraints with variables from $\{y_j, z\}$. Assume likewise, that $r' \neq \bot$ is represented by a conjunction of octagon constraints with variables from Y only.

For each pair y_i, y_j of variables from $Y \cup \{z\}$, the abstract value

$$\left(r_{\{y_i,z\}} \wedge r_{\{y_j,z\}}\right)\big|_{Y \cap \{y_i,y_j\}} \tag{11}$$

can be obtained by means of Fourier-Motzkin elimination of z, applied to the closed conjunctions of octagon constraints representing $r_{\{y_i,z\}}$, and $r_{\{y_j,z\}}$, respectively. In order to see this, we note that all occurring non-zero coefficients of z in the constraints of $r_{\{y_i,z\}}$ as well as $r_{\{y_j,z\}}$ are from $\{-1, 1\}$. Consider a constraint $\ell \leq b$ of the resulting conjunction. Three cases may occur.

- ℓ may contain occurrences of both variables y_i and y_j – each with coefficients in $\{-1, 1\}$.
- ℓ may contain a single occurrence of one variable, w.l.o.g., y_i, whose coefficient now is in $\{-2, -1, 1, 2\}$. In case the coefficient of y_i is in $\{-2, 2\}$, ℓ is still equivalent to an octagon constraint for y_i only. If the constraint, e.g., is $2 \cdot y_i \leq 7$, then it is equivalent to $y_i \leq 3.5$ over rationals, and to $y_i \leq 3$ over the integers.

– ℓ does not contain any occurrences of variables. In this case, it is either equivalent to true and can be abandoned, or equivalent to false – implying that (11) equals $\bot$.

We conclude that the expression (11), when satisfiable, can be represented by a conjunction of octagon constraints using variables y_i and y_j. Thus, the right-hand side of equation (9) for rational as well as integer octagons is equivalent to the result of Fourier-Motzkin elimination of z. This implies claim (2).

Example 3. Assume an integer octagon $r = r' \wedge r_{\{y_1,z\}} \wedge r_{\{y_2,z\}}$ where

$$\begin{array}{ll} r' & = y_1 + y_2 \leq 7 \\ r_{\{y_1,z\}} & = (y_1 + z \leq -1) \wedge (y_1 \leq 3) \wedge (-z \leq 4) \\ r_{\{y_2,z\}} & = (y_2 - z \leq 5) \wedge (-y_2 \leq 1) \end{array}$$

Fourier-Motzkin elimination of z adds the additional constraint

$$y_1 + y_2 \leq 4$$

Projection onto the subset $Y = \{y_1, y_2\}$ according to (9) therefore results in the conjunction of constraints

$$(y_1 + y_2 \leq 7) \wedge (y_1 \leq 3) \wedge (y_1 + y_2 \leq 4) \wedge (-y_2 \leq 1)$$

which can be further simplified to $(y_1 \leq 3) \wedge (y_1 + y_2 \leq 4) \wedge (-y_2 \leq 1)$. □

Example 4. The following 2-decomposable domains are not 2-projective:

1. Finite sets of 2-variable maps;
2. Implications between interval constraints. □

Proof. For (1), let $\mathcal{X} = \{a, x, y, z\}$ where variables range over values from the set $\{1, 2, 3\}$ and maps from variables to such sets are used as the abstraction. Consider now:

$$r_{\{a,x\}} = \{a \mapsto \{1,2\}\} \qquad r_{\{a,y\}} = \{a \mapsto \{2,3\}\} \qquad r_{\{a,z\}} = \{a \mapsto \{3,1\}\}$$

where all other r_p, $p \in [\mathcal{X}]_2$ have the value $\top$. Then,

$$\left(r_{\{a,x\}} \sqcap r_{\{a,y\}} \sqcap r_{\{a,z\}} \sqcap \top\right)\big|_{\{x,y,z\}} = \bot$$

but, in violation of property (9),

$$\begin{array}{ll} & \top \sqcap (r_{\{a,x\}} \sqcap r_{\{a,x\}})\big|_{\{x\}} \sqcap (r_{\{a,y\}} \sqcap r_{\{a,y\}})\big|_{\{y\}} \sqcap (r_{\{a,z\}} \sqcap r_{\{a,z\}})\big|_{\{z\}} \\ & \sqcap (r_{\{a,x\}} \sqcap r_{\{a,y\}})\big|_{\{x,y\}} \sqcap (r_{\{a,x\}} \sqcap r_{\{a,z\}})\big|_{\{x,z\}} \sqcap (r_{\{a,y\}} \sqcap r_{\{a,z\}})\big|_{\{y,z\}} \\ = & \top \sqcap \top \sqcap \top \sqcap \top \sqcap (\{a \mapsto \{1,2\}\} \sqcap \{a \mapsto \{2,3\}\})|_{\{x,y\}} \sqcap \\ & (\{a \mapsto \{1,2\}\} \sqcap \{a \mapsto \{3,1\}\})|_{\{x,z\}} \sqcap (\{a \mapsto \{2,3\}\} \sqcap \{a \mapsto \{3,1\}\})|_{\{y,z\}} \\ = & (\{a \mapsto \{2\}\})|_{\{x,y\}} \sqcap (\{a \mapsto \{1\}\})|_{\{x,z\}} \sqcap (\{a \mapsto \{3\}\})|_{\{x,z\}} \sqcap \\ = & \top \sqcap \top \sqcap \top = \top \end{array}$$

The domain of implications between interval constraints consists of finite conjunctions of the form

$$x \in I \implies y \in I'$$

for variables x and y and I, I' either intervals or the empty set, ordered by implication. In particular, $x \in \emptyset$ may be written as False, while $x \in [-\infty, \infty]$ is denoted by True.

Now, consider the same set $\mathcal{X} = \{a, x, y, z\}$ of variables as for claim (1) and let

$$\begin{aligned} r_{\{a,x\}} &= \{\mathsf{True} \implies a \in [1,2]\} \\ r_{\{a,y\}} &= \{\mathsf{True} \implies a \in [2,3]\} \\ r_{\{a,z\}} &= \{a \in [2,2] \implies \mathsf{False}\} \end{aligned}$$

where all other r_p, $p \in [\mathcal{X}]_2$ have the value $\top$. Then,

$$\left(r_{\{a,x\}} \sqcap r_{\{a,y\}} \sqcap r_{\{a,z\}} \sqcap \top\right)\big|_{\{x,y,z\}} = \mathsf{False} = \bot$$

but

$$\begin{aligned} & \top \wedge (r_{\{a,x\}} \wedge r_{\{a,x\}})\big|_{\{x\}} \wedge (r_{\{a,y\}} \wedge r_{\{a,y\}})\big|_{\{y\}} \wedge (r_{\{a,z\}} \wedge r_{\{a,z\}})\big|_{\{z\}} \\ & \wedge (r_{\{a,x\}} \wedge r_{\{a,y\}})\big|_{\{x,y\}} \wedge (r_{\{a,x\}} \wedge r_{\{a,z\}})\big|_{\{x,z\}} \wedge (r_{\{a,y\}} \wedge r_{\{a,z\}})\big|_{\{y,z\}} \\ = {} & \top \wedge \top \wedge \top \wedge \top \wedge (\{\mathsf{True} \implies a \in [1,2]\} \wedge \{\mathsf{True} \implies a \in [2,3]\})\big|_{\{x,y\}} \wedge \\ & (\{\mathsf{True} \implies a \in [1,2]\} \wedge \{a \in [2,2] \implies \mathsf{False}\})\big|_{\{x,z\}} \wedge \\ & (\{\mathsf{True} \implies a \in [2,3]\} \wedge \{a \in [2,2] \implies \mathsf{False}\})\big|_{\{y,z\}} \\ = {} & (\mathsf{True} \implies a \in [2,2])\big|_{\{x,y\}} \wedge (\mathsf{True} \implies a \in [1,1])\big|_{\{x,z\}} \wedge \\ & (\mathsf{True} \implies a \in [3,3])\big|_{\{x,z\}} \\ = {} & \top \wedge \top \wedge \top = \top \end{aligned}$$

which means property (9) is violated. □

Subsequently, assume that the 2-decomposable domain $\mathcal{R}$ is 2-projective. We show that under this assumption, the greatest solution of the constraint system (8) exists and *coincides* with the normal representation. Moreover, we provide an efficient algorithm for performing the normalization.

Assume that $\mathcal{X} = \{x_1 \ldots x_n\}$, and let $X_r = \{x_1, \ldots, x_r\}$, and $\bar{X}_r = \mathcal{X} \setminus X_r$ for $r = 0, \ldots, n$. Assume that we are given $s_p \in \mathcal{R}^p$, $(p \in [\mathcal{X}]_2)$. For $x, y \in \mathcal{X}$, we define the sequence

$$\begin{aligned} s^{(0)}_{\{x,y\}} &= s_{\{x\}} \sqcap s_{\{y\}} \sqcap s_{\{x,y\}} \\ s^{(r)}_{\{x,y\}} &= s^{(r-1)}_{\{x,y\}} \sqcap \left(s^{(r-1)}_{\{x,x_r\}} \sqcap s^{(r-1)}_{\{x_r,y\}}\right)\Big|_{\{x,y\}} \qquad \text{for } r > 0: \end{aligned}$$

Proposition 3. *Let $\bar{s} = \prod\{s_p \mid p \in [\mathcal{X}]_2\}$ be the abstract relation represented by $\langle s_p \rangle_{p \in [\mathcal{X}]_2}$. Let $p \in [\mathcal{X}]_2$. For $r = 0, \ldots, n$,*

1. *$s^{(r)}_p \sqsubseteq s^{(r)}_{\{x\}}$ for each $x \in p$;*
2. $$\bar{s}|_{\bar{X}_r \cup \{x,y\}} = \prod \left\{ s^{(r)}_p \mid p \subseteq \bar{X}_r \cup \{x,y\}, 1 \le |p| \le 2 \right\} \tag{12}$$

Proof. For $r = 0$, the proposition holds by definition. Now assume that $r > 0$ and the assertion already holds for $r-1$. For $p = \{x, y\}$, we calculate

$$\begin{aligned} s^{(r)}_{\{x,y\}} &= s^{(r-1)}_{\{x,y\}} \sqcap \left(s^{(r-1)}_{\{x,x_r\}} \sqcap s^{(r-1)}_{\{x_r,y\}}\right)\Big|_{\{x,y\}} \sqsubseteq s^{(r-1)}_{\{x\}} \sqcap s^{(r-1)}_{\{x,x_r\}}\Big|_{\{x,y\}} \\ &\sqsubseteq s^{(r-1)}_{\{x\}} \sqcap s^{(r-1)}_{\{x,x_r\}}\Big|_{\{x\}} = s^{(r)}_{\{x\}} \end{aligned}$$

and the first claim follows. For the second claim Eq. (12), consider the case $x_r \notin \{x, y\}$. Then

$$\begin{aligned} &\bar{s}|_{\bar{X}_r\cup\{x,y\}} = \left(\bar{s}|_{\bar{X}_{r-1}\cup\{x,y\}}\right)\Big|_{\bar{X}_r\cup\{x,y\}} \\ &= \left(\bigsqcap\left\{s^{(r-1)}_p \mid p \subseteq \bar{X}_{r-1}\cup\{x,y\}, 1{\leq}|p|{\leq}2\right\}\right)\Big|_{\bar{X}_r\cup\{x,y\}} \qquad \text{(by induction hypothesis)} \\ &= \left(\bigsqcap\left\{s^{(r-1)}_p \mid p \subseteq \bar{X}_r\cup\{x,y\}, 1{\leq}|p|{\leq}2\right\} \sqcap \bigsqcap\left\{s^{(r-1)}_{\{z,x_r\}} \mid z \in \bar{X}_{r-1}\cup\{x,y\}\right\}\right)\Big|_{\bar{X}_r\cup\{x,y\}} \\ &= \bigsqcap\left\{s^{(r-1)}_p \mid p \subseteq \bar{X}_r\cup\{x,y\}, 1{\leq}|p|{\leq}2\right\} \sqcap \\ &\quad \bigsqcap\left\{\left(s^{(r-1)}_{\{z_1,x_r\}} \sqcap s^{(r-1)}_{\{x_r,z_2\}}\right)\Big|_{(\bar{X}_r\cup\{x,y\})\cap\{z_1,z_2\}} \mid z_1, z_2 \in \bar{X}_r\cup\{x,y\}\right\} \sqcap \\ &\quad \bigsqcap\left\{\left(s^{(r-1)}_{\{z_1,x_r\}} \sqcap s^{(r-1)}_{\{x_r\}}\right)\Big|_{(\bar{X}_r\cup\{x,y\})\cap\{z_1\}} \mid z_1 \in \bar{X}_r\cup\{x,y\}\right\} \\ &\quad \sqcap s^{(r-1)}_{\{x_r\}}\Big|_{(\bar{X}_r\cup\{x,y\})\cap\{x_r\}} \qquad \text{(by Eq. (9))} \\ &= \bigsqcap\left\{s^{(r-1)}_p \mid p \subseteq \bar{X}_r\cup\{x,y\}, 1{\leq}|p|{\leq}2\right\} \sqcap \\ &\quad \bigsqcap\left\{\left(s^{(r-1)}_{\{z_1,x_r\}} \sqcap s^{(r-1)}_{\{x_r,z_2\}}\right)\Big|_{\{z_1,z_2\}} \mid z_1, z_2 \in \bar{X}_r\cup\{x,y\}\right\} \sqcap \\ &\quad \bigsqcap\left\{\left(s^{(r-1)}_{\{z_1,x_r\}} \sqcap s^{(r-1)}_{\{x_r\}}\right)\Big|_{\{z_1\}} \mid z_1 \in \bar{X}_r\cup\{x,y\}\right\} \sqcap s^{(r-1)}_{\{x_r\}}\Big|_{\emptyset} \\ &= \bigsqcap\left\{s^{(r-1)}_p \mid p \subseteq \bar{X}_r\cup\{x,y\}, 1{\leq}|p|{\leq}2\right\} \sqcap \\ &\quad \bigsqcap\left\{\left(s^{(r-1)}_{\{z_1,x_r\}} \sqcap s^{(r-1)}_{\{x_r,z_2\}}\right)\Big|_{\{z_1,z_2\}} \mid z_1, z_2 \in \bar{X}_r\cup\{x,y\}\right\} \qquad \text{(by claim (1))} \\ &= \bigsqcap\left\{s^{(r)}_p \mid p \subseteq \bar{X}_r\cup\{x,y\}, 1{\leq}|p|{\leq}2\right\} \end{aligned}$$

and the assertion holds. For the second but last equality, we used that the meet in the second but last row is non-empty, since

$$s^{(r-1)}_{\{x_r\}}\Big|_{\emptyset} \sqsupseteq s^{(r-1)}_{\{x_r\}}\Big|_{\{z_1\}} \sqsupseteq s^{(r-1)}_{\{z_1,x_r\}}\Big|_{\{z_1\}} \sqsupseteq s^{(r-1)}_{\{z_1,x_r\}}\Big|_{\{z_1,z_2\}} \sqsupseteq s^{(r-1)}_{\{z_1,x_r\}} \sqcap s^{(r-1)}_{\{z_1,x_r\}}\Big|_{\{z_1,z_2\}}$$

holds for each $z_1, z_2 \in \bar{X}_r \cup \{x, y\}$. Now let $x_r \in \{x, y\}$. Then $\bar{X}_r \cup \{x, y\} = \bar{X}_{r-1} \cup \{x, y\}$. W.l.o.g., let $x = x_r$. Then $s^{(r-1)}_{\{x,x_r\}} = s^{(r-1)}_{\{x\}}$ and $s^{(r-1)}_{\{x_r,y\}} = s^{(r-1)}_{\{x,y\}}$.

Hence by claim (1), $s^{(r)}_{\{x,y\}} = s^{(r-1)}_{\{x,y\}}$. Accordingly,

$$\begin{aligned}
&\bar{s}|_{\bar{X}_r \cup \{x,y\}} = \bar{s}|_{\bar{X}_{r-1} \cup \{x,y\}} \\
&= \bigsqcap \left\{ s^{(r-1)}_p \mid p \subseteq \bar{X}_{r-1} \cup \{x,y\}, 1 \leq |p| \leq 2 \right\} \qquad \text{(by induction hypothesis)} \\
&= \bigsqcap \left\{ s^{(r-1)}_{\{z_1,z_2\}} \sqcap s^{(r-1)}_{\{z_1,x\}} \sqcap s^{(r-1)}_{\{x,z_2\}} \mid z_1, z_2 \in \bar{X}_{r-1} \cup \{x,y\} \right\} \\
&= \bigsqcap \left\{ s^{(r)}_p \mid p \subseteq \bar{X}_r \cup \{x,y\}, 1 \leq |p| \leq 2 \right\}
\end{aligned}$$

□

Thus, provided $\mathcal{R}$ fulfills Eq. (9), we obtain for $k = n$:

$$\bar{s}|_{\{x,y\}} = s^{(n)}_{\{x,y\}} \sqcap s^{(n)}_{\{x\}} \sqcap s^{(n)}_{\{y\}} = s^{(n)}_{\{x,y\}}$$

Subsequently, we consider Algorithm 1. It consists of one application of the *Floyd-Warshall* algorithm, as is. For that to be sufficient, an initialization round is performed upfront to ensure that each value $t_{\{x,y\}}$ not only subsumes $s_{\{x,y\}}$, but also $s_{\{x\}}$ and $s_{\{y\}}$. The complexity of the proposed algorithm is $\mathcal{O}(n^3)$ if calculations with abstract relations over at most three variables, i.e., from $\mathcal{R}^Y$ for every $Y \subseteq \mathcal{X}$ with $|Y| \leq 3$, can be performed in constant time. For Algorithm 1, we find:

Theorem 1. *Assume that $\langle t_p \rangle_{p \in [\mathcal{X}]_2}$ is the collection of values returned by Algorithm 1 for the collection $\langle s_p \rangle_{p \in [\mathcal{X}]_2}$. Let $\bar{s} = \bigsqcap \{ s_p \mid p \in [\mathcal{X}]_2 \}$ the abstract relation represented by $\langle s_p \rangle_{p \in [\mathcal{X}]_2}$. Then for each $p \in [\mathcal{X}]_2$,*

1. *$\bar{s}|_p \sqsubseteq t_p$;*
2. *If the 2-decomposable domain $\mathcal{R}$ is 2-projective, then $\bar{s}|_p = t_p$ holds. In that case, $\langle t_p \rangle_{p \in [\mathcal{X}]_2}$ is the greatest solution of the constraint system* (8).

Thus, Algorithm 1 provides a cubic time normalization procedure – whenever $\mathcal{R}$ is 2-decomposable and 2-projective. We remark that the initializing first loop cannot be abandoned. When $\mathcal{R}$ is not 2-projective, but 2-decomposable, the algorithm still computes *overapproximations* of normal representations.

Algorithm 1: The variant of the *Floyd-Warshall* algorithm to compute (an overapproximation of) normalization.

```
for x, y ∈ 𝒳 do
    t_{x,y} := s_{x,y} ⊓ s_{x} ⊓ s_{y}          // initialization
for z ∈ 𝒳 do
    for x, y ∈ 𝒳 do
        t_{x,y} := t_{x,y} ⊓ (t_{x,z} ⊓ t_{z,y}) |_{x,y}
return ⟨t_p⟩_{p∈[𝒳]_2}
```

Proof. Let $p \in [\mathcal{X}]_2$. By Proposition 1, $\bar{s}|_p \sqsubseteq t_p$ holds, since the right-hand sides of the constraint system (8) are all monotonic, and starting from the initial values provided in the first loop, each update to some $t_{\{x,y\}}$ in the second loop, corresponds to one update performed by the evaluation of some right-hand side of (8). Therefore, the first assertion follows.

Now assume that the 2-decomposable relational domain $\mathcal{R}$ additionally is 2-projective. Let $t_p^{(r)}$ denote the value of t_p attained after the iteration of the second loop for the variable x_r. By induction on r, we verify by means of Proposition 3 that for all $p \in [\mathcal{X}]_2$, $t_p^{(r)} \sqsubseteq s_p^{(r)}$ holds for all $r = 0, \ldots, n$. In particular, $t_p = t_p^{(n)} \sqsubseteq \bar{s}|_p$, and the second assertion of the theorem follows. □

Example 5. Given a (finite) set of constants, the *Pairs* domain consists of false or conjunctions $\bigwedge\{\phi_p \mid p \in [\mathcal{X}]_2\}$ where for $p \in [\mathcal{X}]_2$, ϕ_p is true or a disjunction of conjunctions of atomic propositions $x = c$, $x \in p$. It is ordered by logical implication. Consider, e.g., $r = \phi_{\{x,y\}} \wedge \phi_{\{y,z\}}$ with $\phi_{\{x,y\}} \equiv (x = a) \vee (x = b \wedge y = c)$ and $\phi_{\{y,z\}} \equiv (y = d \wedge z = b)$. Then $r|_{\{x,y\}} = (x = a \wedge y = d)$. Likewise, $r|_{\{y,z\}} = (y = d \wedge z = b)$ and $r|_{\{x,z\}} = (x = a \wedge z = b)$.

Assume each $r \in R$ is represented by $r = \bigwedge\{r|_p \mid p \in [\mathcal{X}]_2\}$, and define for $p \in [\mathcal{X}]_2$, ϕ_p as the least upper bound of formulas $r|_p, r \in R$. Then $\bar{r} = \bigwedge\{\phi_p \mid p \in [\mathcal{X}]_2\}$ is an upper bound of R and, in fact, the least upper bound. For some $p \in [\mathcal{X}]_2$, then by definition, $\bar{r}|_p \Rightarrow \phi_p$. By monotonicity of the restriction, on the other hand, $r|_p \Rightarrow \bar{r}|_p$ for all $r \in R$. Therefore, $\phi_p \Rightarrow \bar{r}|_p$ as well, and the claim follows. While being 2-decomposable, the *Pairs* domain is not 2-projective. Let, e.g.,

$$\begin{aligned} s_{\{w,x\}} &= (w = \texttt{"fun1"} \wedge x = \texttt{\&f1}) \vee (w = \texttt{"fun3"} \wedge x = \texttt{\&f2}) \\ s_{\{w,y\}} &= (w = \texttt{"fun2"}) \vee (w = \texttt{"fun3"}) \\ s_{\{w,z\}} &= (w = \texttt{"fun1"} \wedge z = \texttt{\&f1}) \vee (w = \texttt{"fun2"} \wedge z = \texttt{\&f1}) \end{aligned}$$

and all other $s_p = \mathsf{true}$. Then, Algorithm 1 computes

$$\begin{aligned} t_{\{w\}} &= t_{\{w,x\}} = t_{\{w,y\}} = t_{\{w,z\}} = \mathsf{false} & t_{\{y\}} &= \mathsf{true} \\ t_{\{x\}} &= t_{\{x,y\}} = (x = \texttt{\&f1}) \vee (x = \texttt{\&f2}) & t_{\{y,z\}} &= t_{\{z\}} = (z = \texttt{\&f1}) \\ t_{\{x,z\}} &= (x = \texttt{\&f1} \wedge z = \texttt{\&f1}) \vee (x = \texttt{\&f2} \wedge z = \texttt{\&f1}) \end{aligned}$$

which is an overapproximation of the normalization given by $\bar{s}|_p = \mathsf{false}$ for $p \in [\mathcal{X}]_2$. Here, the normalization happens to coincide with the greatest solution of constraint system (8). □

Example 6. According to Proposition 2, the domains of rational as well as integer octagons are 2-decomposable and 2-projective. Therefore, Algorithm 1 computes the exact 2-normal form, and thus provides us with cubic time closure algorithms for these. □

5 Incremental Normalization

If the condition c of a guard can be abstracted by some abstract relation $r_c \in \mathcal{R}$, then the transfer function $[\![?c]\!]^\sharp$ can be chosen as $[\![?c]\!]^\sharp r = r \sqcap r_c$. Assume

Algorithm 2: Incremental version of the FLOYD-WARSHALL algorithm to incrementally compute (an overapproximation of) 2-normal forms when clusters t_p, $p \subseteq V$, with $|p| = 2$ have potentially received new values.

```
for z ∈ V do
    for x, y ∈ X do
        t_{x,y} := t_{x,y} ⊓ (t_{x,z} ⊓ t_{z,y}) |_{x,y}
return ⟨t_p⟩_{p∈[X]_2}
```

that the relational domain $\mathcal{R}$ is 2-decomposable as well as 2-projective, and that r_c is represented as the meet $r_{p_1} \sqcap \ldots \sqcap r_{p_k}$ for $p_j \in [\mathcal{X}]_2$. Then, the normalization of $r \sqcap r_c$ can be computed *incrementally*. For the octagon domain over integers, Chawdhary et al. [4] give quadratic incremental closure algorithms. Just like theirs, our algorithm for incremental normalization is based on the Floyd-Warshall algorithm, i.e., Algorithm 1.

In our setting, adding new constraints amounts to improving some clusters $r_{\{a,b\}}$ where a and b are from some set $V \subseteq \mathcal{X}$. For simplicity, we require that only clusters $r_{\{a,b\}}$ with $a \neq b$ are improved. This allows us in the adaption of Algorithm 1 to avoid the initialization loop. Whenever $\mathcal{X}$ contains more than one variable, this extra requirement is no limitation, though, as a constraint involving only the variable z may just be added to any 2-variable cluster p with $z \in p$. (When $\mathcal{X}$ contains only one variable, no normalization is required.) Normalization then is computed by the modified version of Algorithm 1 given in Algorithm 2.

Theorem 2. *Assume a 2-normal collection of values of some 2-decomposable relational domain $S = \langle s_p \rangle_{p \in [\mathcal{X}]_2}$, and a collection $S_1 = \langle s'_{p'} \rangle_{p' \subseteq V, |p'|=2}$ with $s'_{p'} \sqsubseteq s_{p'}$ for all p'. Assume that $\langle t_p \rangle_{p \in [\mathcal{X}]_2}$ is the collection of values returned by Algorithm 2 for the collection $S' = \langle s_p \rangle_{p \in [\mathcal{X}]_2, (p \not\subseteq V \vee |p| \neq 2)} \cup S_1$ Let $\bar{s} = \bigsqcap S'$ the abstract relation represented by S'. Then for each $p \in [\mathcal{X}]_2$,*

1. *$\bar{s}|_p \sqsubseteq t_p$;*
2. *If the 2-decomposable domain $\mathcal{R}$ is 2-projective, then $\bar{s}|_p = t_p$ holds. In that case, $\langle t_p \rangle_{p \in [\mathcal{X}]_2}$ is the greatest solution of constraint system (8).*

Proof. Let $p \in [\mathcal{X}]_2$. $\bar{s}|_p \sqsubseteq t_p$ holds since, as observed before, all right-hand sides of the constraint system (8) are monotonic and the individual update steps of Algorithm 2 each correspond to updates performed by the evaluations of the right-hand sides of (8). Thus, the first statement follows.

Now consider the case where the relational domain is additionally 2-projective. The invariant which the non-incremental Algorithm 1 attains after the initialization holds by construction here. Let $t_p^{(r)}$ denote the value of t_p attained after the iteration of the second loop for the r-th variable in the non-incremental Algorithm 1. We choose the order of the iteration of variables in the second loop such

that the variables in V are considered last. Then, for the first $|\mathcal{X} \setminus V|$ iterations $t_p^{(r-1)} = t_p^{(r)}$, as the original collection $\langle s_p \rangle_{p \in [\mathcal{X}]_2}$ was normalized. Therefore, it suffices to execute the last $|V|$ iterations of the second loop of Algorithm 1 which is identical to Algorithm 2. Thus, by Theorem 1, the claim follows. □

We have thus shown that re-establishing normalization (and thus closure) after adding octagon constraints for m variables is in $\mathcal{O}(m \cdot n^2)$.

6 Abstract Transformers for Linear Assignments

Assume we are given a normalized value r over the set $\mathcal{X}$ of program variables from some 2-decomposable relational domain. Assume further that we are given an assignment a of the form $x \leftarrow e$ where e is an expression over some subset $V \subseteq \mathcal{X}$, and assume that the relational domain satisfies properties (3) and (4). Let $r \in \mathcal{R}$ denote the relational value before the assignment and assume r is already normalized where $r_p = r|_p$ has already been computed for all $p \in [\mathcal{X}]_2$. Let $r' = [\![\mathsf{a}]\!]^\sharp\, r$ denote the relational value after the assignment. Then, for every $p \in [\mathcal{X}]_2$ with $x \notin p$, $r'|_p = r|_p = r_p$. In order to compute the normalization of r', it therefore suffices to compute the values $r'_p = r'|_p$ for $x \in p$, i.e., a *linear* number of clusters p. Now consider some variable $y \in \mathcal{X}$. Because of property (4), we have that

$$\begin{aligned} r'_p &= [\![\mathsf{a}]\!]^\sharp r\big|_{\{x,y\}} \\ &= ([\![\mathsf{a}]\!]^\sharp r|_{V \cup \{x,y\}})\Big|_{\{x,y\}} \\ &= ([\![\mathsf{a}]\!]^\sharp(\bigsqcap\{r_p \mid p \subseteq V \cup \{x,y\}\}))\big|_{\{x,y\}} \end{aligned}$$

i.e., the abstract value $r'_{\{x,y\}}$ requires taking into account only clusters $p \in [\mathcal{X}]_2$ with variables from $V \cup \{x, y\}$. We conclude:

Proposition 4. *Assume that computations on abstract relations from $\mathcal{R}$ over a bounded set of variables is constant time, and assume that the assignment a refers only to a bounded number of variables. Assume further that the abstract relation $r \in \mathcal{R}$ is normalized. Then a normalization of the relation $[\![\mathsf{a}]\!]^\sharp r$ can be computed in linear time.* □

7 Linear Programming with Octagon Constraints

Let us turn to the implementation of best abstract transformers for assignments for the octagon domain (over rationals as well as over integers). For the octagon domain, an abstract transformer for assignments can be constructed by adding octagon constraints. This works well for right-hand sides of the form $y + c$ or $-y + c$ for variables y and constants c. For more general right-hand sides such as, e.g., $3 \cdot y - 2 \cdot z$, the best transformer can instead be expressed by means of *optimization* problems [25].

Assume that the octagon is provided by bounds $b_\ell, \ell \in L_V$ for some subset $V \subseteq \mathcal{X}$ of variables. Depending on the sign of a variable occurring in a linear combination ℓ, we say it occurs *positively* or *negatively*. Consider the optimization problem of maximizing a linear objective function taking variables from V subject to the given set of octagon constraints

$$\begin{array}{l} \textbf{maximize } \sum_{z \in V} a_z \cdot z \\ \textbf{subject to } \ell \leq b_\ell \qquad (\ell \in L_V) \end{array} \tag{13}$$

When interpreted over the rationals, optimal solutions can be computed in time *polynomial* in the *size* of the linear program (i.e., the number of bits to spell it out) [15] or exponential time in the number of variables if simplex type algorithms are used [17]. To this general approach, we here add one more observation, namely, that over the rationals, the set of octagon constraints to be satisfied in optimization problems can be restricted to constraints where each occurring variable $z \in V$ occurs with the same sign as the coefficient a_z of z in the objective function: this considerably reduces the number of constraints to be considered.

Proposition 5. *Assume that we are given the rational octagon linear program* (13) *where $a_z > 0$ for all $z \in V$. If the octagon corresponding to the constraints is closed, then the same result is obtained when the constraints are restricted to octagon linear combinations z and $z + y$ for $z, y \in V$ and $z \neq y$.*

Proof. The proof of the proposition is obtained by means of the *dual* linear program:

$$\begin{array}{l} \textbf{minimize } \sum_{\ell \in L_V} y_\ell \cdot b_\ell \\ \textbf{subject to } (\sum_{z \text{ in } \ell} y_\ell) - (\sum_{-z \text{ in } \ell} y_\ell) = a_z \quad (z \in V) \\ \qquad\qquad\qquad\qquad\qquad\quad y_\ell \geq 0 \quad (\ell \in L_V) \end{array} \tag{14}$$

If the original program is unbounded, then so is the program with the restricted set of constraints. Therefore, assume that the original linear program is bounded. Then the dual optimization problem has a feasible solution $y_\ell, \ell \in L_V$, where the minimal gain b is attained, i.e., $\sum_{\ell \in L_V} y_\ell \cdot b_\ell = b$. It remains to prove that b can be attained by a feasible solution $y_\ell, \ell \in L$, where $y_\ell = 0$ for all octagon combinations ℓ which contain negations. We proceed by induction on the number of octagon combinations ℓ with negative occurrences of variables from V. Assume that there are octagon combinations ℓ with negated occurrences of z and $y_\ell > 0$. Consider the linear constraint in (13) for z

$$\left(\sum_{j=1}^{r} y_\ell\right) - \left(\sum_{j'=1}^{r'} y_{\ell'_{j'}}\right) = a_z$$

where ℓ_j enumerates all octagon combinations with positive and $\ell'_{j'}$ enumerates all octagon combinations with negative occurrences of z. Since $r' > 0$ and $a_z > 0$, also $r > 0$. If $y_{\ell_r} \geq y_{\ell'_{r'}}$, we proceed to eliminate the octagon combination $\ell'_{r'}$ with a negative occurrence of z and proceed to eliminate also all other negative occurrences of z by constructing a solution y'_ℓ with the same gain b where $y'_{\ell'_{r'}} =$

0. If $\ell_r + \ell'_{r'} = 0$, then either no further variable is contained in $\ell_r, \ell'_{r'}$ or the same variable z' occurs with opposite signs. Then we set $y'_{\ell_r} = y'_{\ell'_{r'}} = 0$ and $y'_p = y_p$ otherwise.

Now assume that $\ell_r + \ell'_{r'}$ is a linear combination different from 0. Then it either is equivalent to an octagon combination not involving variable z, or $2z'$ or $2 \cdot (-z')$ for some variable z' different from z. In order to deal with all these cases consistently, we introduce a correction factor c as 1 if the sum is an octagon linear combination, and 2 otherwise. Let q denote the octagon combination with $c \cdot q = \ell_r + \ell'_{r'}$. Since the octagon r is closed, $c \cdot b_q \leq b_{\ell_r} + b_{\ell'_{r'}}$ holds. Let $y'_\ell, \ell \in L_V$, be defined by

$$y'_\ell = \begin{cases} y_{\ell_r} - y_{\ell'_{r'}} \text{ if } \ell = \ell_r \\ 0 \text{ if } \ell = \ell'_{r'} \\ y_\ell + c \cdot y_{\ell'_{r'}} \text{ if } c \cdot \ell = q \\ y_\ell \text{ otherwise} \end{cases}$$

We claim that $y'_\ell, \ell \in L$, is again a feasible solution, i.e., satisfies all constraints, where the same gain b is attained. Concerning the gain, we have

$$\begin{aligned} y_{\ell_r} \cdot b_{\ell_r} + y_{\ell'_{r'}} \cdot b_{\ell'_{r'}} + y_q \cdot b_q &= (y_{\ell_r} - y_{\ell'_{r'}}) \cdot b_{\ell_r} + y_{\ell'_{r'}} \cdot (b_{\ell_r} + b_{\ell'_{r'}}) + y_q \cdot b_q \\ &\geq y'_{\ell_r} \cdot b_{\ell_r} + y'_q \cdot b_q \end{aligned}$$

As the gain b was already minimal, we conclude that the gain for the y'_ℓ has not changed. It remains to show that the y'_ℓ form a feasible solution of the constraints in (13). By construction, the equation for z is satisfied (we reduce y_{ℓ_r} with a positive occurrence of z by the same amount as $y_{\ell'_{r'}}$ with a negative occurrence). If q contains a variable z' which is then different from z, then this variable must occur in $\ell_r, \ell'_{r'}$ or both and if so, with the same sign. If it is contained only in $\ell'_{r'}$, then $y_{\ell'_{r'}}$ in the left-hand side of the constraint for z' is replaced with 0, while at the same time y_q is increased with y_{ℓ_r}. If it is contained only in ℓ_r, then y_{ℓ_r} in the left-hand side of the constraint for z' is decreased with $y_{\ell'_{r'}}$, while at the same time y_q is increased with y_{ℓ_r}. If it is contained both in ℓ_r and $\ell'_{r'}$, then y_{ℓ_r} in the left-hand side of the constraint for z' is decreased with $y_{\ell'_{r'}}$, $y_{\ell'_{r'}}$ is set to 0, y_q is increased with $2 \cdot y_{\ell'_{r'}}$.

Thus, in all cases, the equation is satisfied for the y'_p.

We conclude that the combination ℓ_r can equivalently be removed by means of the octagon combination q not involving the variable z.

Therefore, now assume that $y_{\ell'_{r'}} > y_{\ell_r}$ where, w.l.o.g., the maximal value of the non-zero y_{ℓ_j} equals y_{ℓ_r}. If $\ell_r + \ell'_{r'} = 0$, then $b_{\ell_r} + b_{\ell'_{r'}} = 0$ (otherwise the gain were not minimal). Therefore, we set $y'_{\ell_r} = 0$, $y'_{\ell'_{r'}} = y_{\ell'_{r'}} - y_{\ell_r}$, and $y'_\ell = \ell_p$ otherwise to obtain a feasible solution where the minimal gain is attained. At the same time, the number of octagon combinations ℓ with $y'_\ell > 0$ where z occurs positively has decreased. Therefore, assume that $\ell_r + \ell'_{r'}$ is different from 0. Then there is a coefficient $c \in \{1, 2\}$ and an octagon constraint q such that

$c \cdot q = \ell_r + \ell'_{r'}$ and $c \cdot b_q \leq b_{\ell_r} + b_{\ell'_{r'}}$. Then we set

$$y'_\ell = \begin{cases} 0 \text{ if } \ell = \ell_r \\ y_{\ell'_{r'}} - y_{\ell_r} \text{ if } \ell = \ell'_{r'} \\ y_q + c \cdot y_{\ell_r} \text{ if } \ell = q \\ y_\ell \text{ otherwise} \end{cases}$$

Again, we obtain a feasible solution where the gain has not increased, but the number of octagon combinations ℓ with $y'_\ell > 0$ where z occurs positively has decreased. Altogether, we conclude that, without increasing the gain, the feasible solution y_ℓ can be adjusted such that $y_\ell = 0$ for ℓ whenever ℓ contains negative occurrences of variables in V.

As a result, we obtain as the dual of the simplified LP problem

$$\begin{array}{lll} \textbf{minimize} & \sum_{z_1 \in V} y_{z_1} \cdot b_{z_1} + \sum_{z_2 \in V \setminus \{z_1\}} y_{z_1+z_2} \cdot b_{z_1+z_2} & \\ \textbf{subject to} & y_{z_1} + \sum_{z_2 \in V \setminus \{z_1\}} y_{z_1+z_2} = a_{z_1} & (z_1 \in V) \\ & y_{z_1} \geq 0 & (z_1 \in V) \\ & y_{z_1+z_2} \geq 0 & (z_1, z_2 \in V, z_1 \neq z_2) \end{array} \tag{15}$$

Example 7. Assume that the set of program variables consists of x, z_1, z_2, z_3, that our goal is to maximize the linear objective function $2z_1 + 3z_2 + z_3$ subject to the octagon constraints

$$z_1 + z_2 \leq 10 \qquad z_1 + z_3 \leq 1 \qquad z_2 + z_3 \leq 1$$

The dual linear program then is given by

$$\begin{array}{ll} \textbf{minimize} & y_1 \cdot 10 + y_2 + y_3 \\ \textbf{subject to} & y_1 + y_2 = 2 \quad y_1 + y_3 = 3 \quad y_2 + y_3 = 1 \\ & y_1, y_2, y_3 \geq 0 \end{array}$$

In this case, there is just one possible solution for the y_i, namely,

$$y_1 = 2.5 \quad y_2 = 0.5 \quad y_3 = 0.5$$

— implying that the optimal value is given by $25 + 0.5 + 0.5 = 26$. □

For an optimization problem with integer octagon constraints, we may, in principle, proceed as for rationals. Solving integer linear programs with octagon constraints precisely, however, is NP-hard. This can be seen, e.g., by reduction from the NP-complete *maximum clique problem*, i.e., the problem of deciding whether the maximal size of a clique in an undirected graph exceeds some bound. Let $G = (V, E)$ denote a finite undirected graph, and choose V as the set of variables. Then we construct the integer optimization problem

$$\begin{array}{lll} \textbf{maximize} & \sum_{x \in V} x & \\ \textbf{subject to} & x + y \leq 1 & (\{x, y\} \notin E) \\ & -x \leq 0 & (x \in V) \\ & x \leq 1 & (x \in V) \end{array}$$

The constraints are all integer octagon constraints, while the solution to the optimization problem equals the maximal size of a clique. Since the construction of the integer optimization problem from the instance of the clique problem can be done in polynomial time, it follows that to decide whether the optimal value for an integer linear program with octagon constraints exceeds some value, is NP-hard.

8 Abstract Assignments for Octagons

Assume that we are given an affine assignment of the form

$$x \leftarrow b + \textstyle\sum_{z \in V} a_z \cdot z$$

and that the octagon before the assignment is a closed octagon r with coefficients $b_\ell, \ell \in L_{\mathcal{X}}$. W.l.o.g., assume that x does not occur in the right-hand side, i.e., $x \notin V$. Over the rationals, the best upper bound b'_ℓ for the octagon combination ℓ with x occurring in ℓ is obtained by a linear program of the form (13). Depending on ℓ, the objective functions are

ℓ	objective function
x	$\sum_{z \in V} a_z \cdot z$
$-x$	$\sum_{z \in V} -a_z \cdot z$
$x + y$	$y + \sum_{z \in V} a_z \cdot z$
$x - y$	$-y + \sum_{z \in V} a_z \cdot z$
$-x + y$	$y + \sum_{z \in V} -a_z \cdot z$
$-x - y$	$-y + \sum_{z \in V} -a_z \cdot z$

The best abstract transformer $[\![\mathsf{a}]\!]^\sharp$ then is given by

$$[\![\mathsf{a}]\!]^\sharp(r) = r|_{\mathcal{X} \setminus \{x\}} \wedge r_x \tag{16}$$

where r_x denotes the conjunction

$$\begin{aligned}(x \le b + b'_x) \ \wedge \ \textstyle\bigwedge_{z \neq x} & (x + z \le b + b'_{x+z}) \wedge (x - z \le b + b'_{x-z}) \wedge \\ & (-x + z \le b'_{-x+z} - b) \wedge (-x - z \le b'_{-x-z} - b)\end{aligned}$$

Over the integers, we can proceed analogously to the rational case by solving the corresponding integer optimization problems. Since these, in general, are NP-hard, we prefer for integer octagons, to rely on rational *relaxations* of the corresponding ILP problems. This means that for each octagon combination ℓ, we determine the best rational upper bound b_ℓ after the assignment (as determined by the corresponding LP problem) which is tightened to $\lfloor b_\ell \rfloor$ to obtain a sound upper bound for ℓ over the integers. We remark that for integer octagons, an alternative formulation of abstract transformers for affine assignments has been provided in [21]. The transformer there is based on the optimal abstract transformer for rational polyhedra in [9] whose bounds are tightened and subsequently over-approximated by octagon constraints. The latter step also requires

solving appropriate (relaxed) LP problems, which are essentially the same as we solve – only that we benefit from a reduced number of octagon constraints to be taken into account by each LP problem. We obtain:

Theorem 3. *For the octagon domain over the rationals, the best transformer* (16) *for a linear assignment can be computed in polynomial time. For n program variables and a constant number of variables occurring in the assignment, the best transformer can be computed in time $\mathcal{O}(n)$.*

□

Proof. Assume that the octagon before the assignment is closed. Due to Proposition 5, the octagon transformer for linear assignments satisfies properties (4) and (3). Therefore by Proposition 4, only a linear number of optimization problems must be solved. Over the rationals, the optimal upper bound to an octagon combination can be determined by solving an LP problem – which is known to be possible in polynomial time. Note that due to Proposition 5, the set of octagon constraints to be taken into account can be reduced to constraints with octagon combinations where the signs of variables match the corresponding signs occurring in the objective function.

If the right-hand side contains only a bounded number of variables, each of the LP problems will refer to a bounded number of variables only, and thus can be solved in constant time (e.g., by using the Simplex algorithm). Since only $\mathcal{O}(n)$ many of these problems must be solved, the overall runtime is linear. □

Over the integers, on the other hand, the solution of the relaxed integer LP problem for a sound bound to an octagon combination can be obtained as the solution to the corresponding relaxed rational LP problem, and the argument proceeds as in the rational case. As a corollary, we therefore obtain:

Corollary 1. *For the octagon domain over the integers, the integer relaxation of* (16) *for a linear assignment can be computed in polynomial time. For n program variables and a constant number of variables occurring in the assignment, the relaxed best transformer can be computed in time $\mathcal{O}(n)$.* □

9 Related Work

Since being introduced by Miné [20, 21], the weakly relational numerical domain of *Octagons* has found widespread application in the analysis and verification of programs and is part, e.g., of the highly successful static analyzer Astrée [3, 7]. While normalization has been known to be cubic time for rational octagons right from the beginning [20], it was open whether this also holds true for integer octagons. This question has been settled affirmatively by Bagnara et al. [1]. Sankaranarayanan et al. [25] proposed using techniques from linear programming to compute best transformers for linear assignments. Chawdhary et al. [4] investigated the problem of improved quadratic algorithms for incremental closure, i.e., adding one further octagon constraint. Implementations of

Domain	2-decomposable	2-projective	Normalization
Integer Octagons [20]	✓	✓	$\mathcal{O}(n^3)$
Rational Octagons [20]	✓	✓	$\mathcal{O}(n^3)$
TVPI [27][1]	✓	✓	$\mathcal{O}(n^3 \log^2 n)$
Pentagons [18]	✓	✓	$\mathcal{O}(n^3)$
Weighted Hexagons [11]	✓	✓	$\mathcal{O}(n^3)$
Logahedra [13]	✓	✓	$\mathcal{O}(n^3)$
dDBM [24][2]	✓	✗ (Appendix A)	$O(n^3)$ rat.; $O(n^5)$ ints
AVO [5]	✓	✗ (Appendix A)	$O(2^n \cdot n^3)$ rat.; ? ints
Pairs (Example 5)	✓	✗	?

Fig. 1: Various weakly relational domains, whether they are 2-projective and 2-decomposable, and the complexity of their normalization operation.

Octagons are provided, e.g., by the Apron library [14] and Elina [10]. Various *Octagon* algorithms are practically evaluated by Gange et al. [12].

Extensions of octagons have been considered by Péron and Halbwachs [24] and Chen et al. [5]. For these extensions, however, known normalization algorithms turn out to be rather expensive so that more practical *approximate* normalizations have been proposed. Fig. 1 gives an overview over some weakly relational domains, whether they are 2-decomposable and whether they are also 2-projective as well as the best time complexities for (approximate) normalization in the number of variables.

10 Conclusion and Future Work

We have provided an algorithm for normalizing octagon abstract relations over rationals as well as over integers. For that, we introduced the notion of 2-decomposability for relational domains and provided a cubic-time algorithm based on *Floyd-Warshall* which overapproximates normalization. For the subclass of 2-projective domains comprising, e.g., integer or rational *Octagons*, it computes the *exact* 2-normal form. The major benefit of the resulting algorithm is its simplicity. For the instance of the *Octagon* domain, e.g., the closure is obtained without duplication of variables. The general setup also provides us with a quadratic algorithm for *incremental* normalization. For octagons, we also reconsidered the construction of best abstract transformers for affine assignments by means of linear programming. Over the rationals, we observe that only those octagon constraints need to be taken into account where the sign of each occurring variable z agrees with the sign of the occurrence of z in the respective

[1] For TVPI: As operations on values for 3 variables are in $\mathcal{O}(\log^2 n)$.

[2] For int dDBM: Approximate normalization up-to emptiness. Checking emptiness is exponential.

objective functions. This, again, may result in a significant speedup when it comes to practical implementations.

In future work, we would like to provide a new implementation of *Octagon* domains based on our algorithms and evaluate its practical performance on realistic examples. Combining our algorithms with orthogonal techniques such as online decomposition [28] in particular seems like a promising line of inquiry. We also would like to explore in greater detail the potential of further, perhaps non-numerical 2-decomposable domains.

Acknowledgements. This work was supported in part by Deutsche Forschungsgemeinschaft (DFG) – 378803395/2428 ConVeY.

A 2-Projectivity for Extensions of Octagons

Here, we investigate extensions to the *Octagon* domain and the domain of difference bounds, respectively, that have been proposed in the literature, and investigate whether they are 2-decomposable and 2-projective.

Example 8. Consider the domain of difference-bound matrices enhanced with disequalities [24] where $\mathcal{X} = \{a, b, c\}$. This domain is 2-decomposable. Now, for 2-projectivity, let, e.g.,

$$\begin{aligned} r_{\{a,b\}} &= (a - b \leq -1 \wedge b \neq 98 \wedge b \neq 97) \\ r_{\{a,c\}} &= (c \leq 99) \\ r_{\{b,c\}} &= (b - c \leq -1) \end{aligned}$$

and all other $r_p = \top$. We remark that, by abuse of notation, we write $b \neq 98$ instead of introducing a dedicated variable c_{98} and constraints $b \neq c_{98} \wedge c_{98} \leq 98 \wedge 0 - c_{98} \leq -98$, and analogously for $b \neq 97$. Now, consider (9) with $Y = \{a, c\}$, $z = b$, $r' = \{c \leq 99\}$ Then,

$$\begin{aligned} & (r_{\{b\}} \wedge r_{\{b,a\}} \wedge r_{\{b,c\}} \wedge r')\big|_Y \\ = \; & (r_{\{b\}} \wedge r_{\{b,a\}} \wedge r_{\{b,c\}} \wedge r')\big|_{\{a,c\}} \\ = \; & (c \leq 99 \wedge a - c \leq -2 \wedge c \leq 95) \\ \neq \; & (c \leq 99 \wedge a - c \leq -2 \wedge c \leq 97) \\ = \; & (c \leq 99) \wedge \top \wedge \top \wedge \top \wedge (a - c \leq -2) \wedge \top \wedge \top \\ = \; & (c \leq 99) \wedge (r_{\{b\}} \wedge r_{\{b\}})\big|_{\emptyset} \wedge (r_{\{a,b\}} \wedge r_{\{a,b\}})\big|_{\{a\}} \wedge (r_{\{b,c\}} \wedge r_{\{b,c\}})\big|_{\{c\}} \\ & \wedge (r_{\{a,b\}} \wedge r_{\{b,c\}})\big|_{\{a,c\}} \wedge (r_{\{b\}} \wedge r_{\{b,a\}})\big|_{\{a\}} \wedge (r_{\{b\}} \wedge r_{\{b,c\}})\big|_{\{c\}} \\ = \; & r' \wedge \textstyle\bigwedge_{i,j=1}^{k} (r_{\{b,y_i\}} \wedge r_{\{b,y_j\}})\big|_{Y \cap \{y_i, y_j\}} \end{aligned}$$

and the domain thus is not 2-projective. □

Example 9. Consider the domain of octagons enhanced with additional constraints for the absolute values of variables [5], i.e., with additional constraints of the form $\pm|x| \pm |y| \leq c$ and $\pm|x| \pm y \leq c$. This domain is 2-decomposable.

Now, for 2-projectivity, let, e.g.,

$$\begin{aligned} r_{\{a,d\}} &= a - |d| \leq 2 \\ r_{\{b,c\}} &= b + c \leq 5 \\ r_{\{b,d\}} &= b - d \leq 5 \\ r_{\{c,d\}} &= -c + d \leq 2 \wedge -|d| \leq 0 \end{aligned}$$

with all other $r_p = \top$ for $p \in [\mathcal{X}]_2$. Now, consider (9) with $Y = \{a, b, c\}$, $z = d$, $r' = (b + c \leq 5)$.

$$\begin{aligned} & (r_{\{d\}} \wedge r_{\{d,a\}} \wedge r_{\{d,b\}} \wedge r_{\{d,c\}} \wedge r')\big|_Y \\ &= (r_{\{d\}} \wedge r_{\{d,a\}} \wedge r_{\{d,b\}} \wedge r_{\{d,c\}} \wedge r')\big|_{\{a,b,c\}} \\ &= b + c \leq 5 \wedge b - c \leq 7 \wedge b \leq 6 \wedge a + b \leq 9 \\ &\neq b + c \leq 5 \wedge b - c \leq 7 \wedge b \leq 6 \\ &= b + c \leq 5 \wedge \top \wedge \top \wedge \{b - c \leq 7\} \\ &= b + c \leq 5 \wedge (a - |d| \leq 2 \wedge b - d \leq 5)|_{\{a,b\}} \\ &\quad \wedge (a - |d| \leq 2 \wedge (-c + d \leq 2 \wedge -|d| \leq 0))|_{\{a,c\}} \wedge \\ &\quad (b - d \leq 5 \wedge (-c + d \leq 2 \wedge -|d| \leq 0))|_{\{b,c\}} \\ &= b + c \leq 5 \wedge (r_{\{d,a\}} \wedge r_{\{d,b\}})\big|_{\{a,b\}} \wedge (r_{\{d,a\}} \wedge r_{\{d,c\}})\big|_{\{a,c\}} \wedge (r_{\{d,b\}} \wedge r_{\{d,c\}})\big|_{\{b,c\}} \\ &= \{b + c \leq 5\} \wedge \\ &\quad (r_{\{d,a\}})\big|_{\{a\}} \wedge (r_{\{d,a\}} \wedge r_{\{d,b\}})\big|_{\{a,b\}} \wedge (r_{\{d,a\}} \wedge r_{\{d,c\}})\big|_{\{a,c\}} \wedge \\ &\quad (r_{\{d,b\}})\big|_{\{b\}} \wedge (r_{\{d,b\}} \wedge r_{\{d,c\}})\big|_{\{b,c\}} \wedge \\ &\quad (r_{\{d,c\}})\big|_{\{c\}} \\ &= \{b + c \leq 5\} \wedge \\ &\quad (r_{\{d,a\}} \wedge r_{\{d,a\}})\big|_{\{a\}} \wedge (r_{\{d,a\}} \wedge r_{\{d,b\}})\big|_{\{a,b\}} \wedge (r_{\{d,a\}} \wedge r_{\{d,c\}})\big|_{\{a,c\}} \\ &\quad \wedge (r_{\{d,a\}} \wedge r_{\{d,d\}})\big|_{\{a\}} \wedge \\ &\quad (r_{\{d,b\}} \wedge r_{\{d,b\}})\big|_{\{b\}} \wedge (r_{\{d,b\}} \wedge r_{\{d,c\}})\big|_{\{b,c\}} \wedge (r_{\{d,b\}} \wedge r_{\{d,d\}})\big|_{\{b\}} \wedge \\ &\quad (r_{\{d,c\}} \wedge r_{\{d,c\}})\big|_{\{c\}} \wedge (r_{\{d,c\}} \wedge r_{\{d,d\}})\big|_{\{c\}} \wedge \\ &\quad (r_{\{d,d\}} \wedge r_{\{d,d\}})\big|_{\{\emptyset\}} \\ &= r' \wedge \textstyle\bigwedge_{i,j=1}^{k} (r_{\{d,y_i\}} \wedge r_{\{d,y_j\}})\big|_{Y \cap \{y_i, y_j\}} \end{aligned}$$

and the domain thus is not 2-projective. □

Bibliography

[1] Bagnara, R., Hill, P.M., Zaffanella, E.: An improved tight closure algorithm for integer octagonal constraints. In: Logozzo, F., Peled, D.A., Zuck, L.D. (eds.) Verification, Model Checking, and Abstract Interpretation, pp. 8–21, Springer Berlin Heidelberg, Berlin, Heidelberg (2008), ISBN 978-3-540-78163-9

[2] Bagnara, R., Hill, P.M., Zaffanella, E.: Weakly-relational shapes for numeric abstractions: improved algorithms and proofs of correctness. Formal Methods Syst. Des. **35**(3), 279–323 (2009), DOI: 10.1007/s10703-009-0073-1, URL https://doi.org/10.1007/s10703-009-0073-1

[3] Blanchet, B., Cousot, P., Cousot, R., Feret, J., Mauborgne, L., Miné, A., Monniaux, D., Rival, X.: A static analyzer for large safety-critical software. In: Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation, p. 196–207, PLDI '03, Association for Computing Machinery, New York, NY, USA (2003), ISBN 1581136625, DOI: 10.1145/781131.781153, URL https://doi.org/10.1145/781131.781153

[4] Chawdhary, A., Robbins, E., King, A.: Incrementally closing octagons. Formal Methods Syst. Des. **54**(2), 232–277 (2019), DOI: 10.1007/s10703-017-0314-7, URL https://doi.org/10.1007/s10703-017-0314-7

[5] Chen, L., Liu, J., Miné, A., Kapur, D., Wang, J.: An abstract domain to infer octagonal constraints with absolute value. In: Müller-Olm, M., Seidl, H. (eds.) Static Analysis - 21st International Symposium, SAS 2014, Munich, Germany, September 11-13, 2014. Proceedings, Lecture Notes in Computer Science, vol. 8723, pp. 101–117, Springer (2014), DOI: 10.1007/978-3-319-10936-7_7, URL https://doi.org/10.1007/978-3-319-10936-7_7

[6] Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. MIT Press, Cambridge (2009)

[7] Cousot, P., Cousot, R., Feret, J., Mauborgne, L., Miné, A., Rival, X.: Why does astrée scale up? Form. Methods Syst. Des. **35**(3), 229–264 (dec 2009), ISSN 0925-9856, DOI: 10.1007/s10703-009-0089-6, URL https://doi.org/10.1007/s10703-009-0089-6

[8] Cousot, P., Halbwachs, N.: Automatic discovery of linear restraints among variables of a program. In: Aho, A.V., Zilles, S.N., Szymanski, T.G. (eds.) Conference Record of the Fifth Annual ACM Symposium on Principles of Programming Languages, Tucson, Arizona, USA, January 1978, pp. 84–96, ACM Press (1978), DOI: 10.1145/512760.512770, URL https://doi.org/10.1145/512760.512770

[9] Cousot, P., Halbwachs, N.: Automatic discovery of linear restraints among variables of a program. In: Proceedings of the 5th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages, p. 84–96, POPL

'78, Association for Computing Machinery, New York, NY, USA (1978), ISBN 9781450373487, DOI: 10.1145/512760.512770, URL https://doi.org/10.1145/512760.512770

[10] ELINA: Elina: Eth library for numerical analysis. http://elina.ethz.ch/ (2018)

[11] Fulara, J., Durnoga, K., Jakubczyk, K., Schubert, A.: Relational abstract domain of weighted hexagons. Electron. Notes Theor. Comput. Sci. **267**(1), 59–72 (2010), DOI: 10.1016/j.entcs.2010.09.006, URL https://doi.org/10.1016/j.entcs.2010.09.006

[12] Gange, G., Ma, Z., Navas, J.A., Schachte, P., Søndergaard, H., Stuckey, P.J.: A fresh look at zones and octagons. ACM Trans. Program. Lang. Syst. **43**(3) (sep 2021), ISSN 0164-0925, DOI: 10.1145/3457885, URL https://doi.org/10.1145/3457885

[13] Howe, J.M., King, A.: Logahedra: A new weakly relational domain. In: Liu, Z., Ravn, A.P. (eds.) Automated Technology for Verification and Analysis, 7th International Symposium, ATVA 2009, Macao, China, October 14-16, 2009. Proceedings, Lecture Notes in Computer Science, vol. 5799, pp. 306–320, Springer (2009), DOI: 10.1007/978-3-642-04761-9_23, URL https://doi.org/10.1007/978-3-642-04761-9_23

[14] Jeannet, B., Miné, A.: Apron: A library of numerical abstract domains for static analysis. In: Bouajjani, A., Maler, O. (eds.) Computer Aided Verification, 21st International Conference, CAV 2009, Grenoble, France, June 26 - July 2, 2009. Proceedings, LNCS, vol. 5643, pp. 661–667, Springer (2009), DOI: 10.1007/978-3-642-02658-4_52, URL https://doi.org/10.1007/978-3-642-02658-4_52

[15] Karmarkar, N.: A new polynomial-time algorithm for linear programming. In: Proceedings of the sixteenth annual ACM symposium on Theory of computing, pp. 302–311 (1984)

[16] Karr, M.: Affine relationships among variables of a program. Acta Informatica **6**, 133–151 (1976), DOI: 10.1007/BF00268497, URL https://doi.org/10.1007/BF00268497

[17] Klee, V., Minty, G.J.: How good is the simplex algorithm. Inequalities **3**(3), 159–175 (1972)

[18] Logozzo, F., Fähndrich, M.: Pentagons: A weakly relational abstract domain for the efficient validation of array accesses. In: Proceedings of the 2008 ACM Symposium on Applied Computing, p. 184–188, SAC '08, Association for Computing Machinery, New York, NY, USA (2008), ISBN 9781595937537, DOI: 10.1145/1363686.1363736, URL https://doi.org/10.1145/1363686.1363736

[19] Miné, A.: A new numerical abstract domain based on difference-bound matrices. In: Danvy, O., Filinski, A. (eds.) Programs as Data Objects, Second Symposium, PADO 2001, Aarhus, Denmark, May 21-23, 2001, Proceedings, LNCS, vol. 2053, pp. 155–172, Springer (2001), DOI: 10.1007/3-540-44978-7_10, URL https://doi.org/10.1007/3-540-44978-7_10

[20] Miné, A.: The octagon abstract domain. In: WCRE' 01, p. 310, IEEE Computer Society (2001), DOI: 10.1109/WCRE.2001.957836

[21] Miné, A.: The octagon abstract domain. Higher Order Symbol. Comput. **19**(1), 31–100 (mar 2006), ISSN 1388-3690, DOI: 10.1007/s10990-006-8609-1, URL https://doi.org/10.1007/s10990-006-8609-1

[22] Müller-Olm, M., Seidl, H.: Precise interprocedural analysis through linear algebra. In: Jones, N.D., Leroy, X. (eds.) Proceedings of the 31st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2004, Venice, Italy, January 14-16, 2004, pp. 330–341, ACM (2004), DOI: 10.1145/964001.964029, URL https://doi.org/10.1145/964001.964029

[23] Müller-Olm, M., Seidl, H.: Analysis of modular arithmetic. ACM Trans. Program. Lang. Syst. **29**(5), 29 (2007), DOI: 10.1145/1275497.1275504, URL https://doi.org/10.1145/1275497.1275504

[24] Péron, M., Halbwachs, N.: An abstract domain extending difference-bound matrices with disequality constraints. In: Cook, B., Podelski, A. (eds.) Verification, Model Checking, and Abstract Interpretation, 8th International Conference, VMCAI 2007, Nice, France, January 14-16, 2007, Proceedings, Lecture Notes in Computer Science, vol. 4349, pp. 268–282, Springer (2007), DOI: 10.1007/978-3-540-69738-1_20, URL https://doi.org/10.1007/978-3-540-69738-1_20

[25] Sankaranarayanan, S., Sipma, H.B., Manna, Z.: Scalable analysis of linear systems using mathematical programming. In: Cousot, R. (ed.) Verification, Model Checking, and Abstract Interpretation, LNCS, vol. 3385, pp. 25–41, Springer, Berlin, Heidelberg (2005), ISBN 978-3-540-30579-8, DOI: 10.1007/978-3-540-30579-8_2

[26] Schwarz, M., Saan, S., Seidl, H., Erhard, J., Vojdani, V.: Clustered relational thread-modular abstract interpretation with local traces. In: Wies, T. (ed.) Programming Languages and Systems - 32nd European Symposium on Programming, ESOP 2023, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2023, Paris, France, April 22-27, 2023, Proceedings, Lecture Notes in Computer Science, vol. 13990, pp. 28–58, Springer (2023), DOI: 10.1007/978-3-031-30044-8_2, URL https://doi.org/10.1007/978-3-031-30044-8_2

[27] Simon, A., King, A., Howe, J.M.: Two variables per linear inequality as an abstract domain. In: Leuschel, M. (ed.) Logic Based Program Synthesis and Transformation, 12th International Workshop, LOPSTR 2002, Madrid, Spain, September 17-20,2002, Revised Selected Papers, LNCS, vol. 2664, pp. 71–89, Springer (2002), DOI: 10.1007/3-540-45013-0_7, URL https://doi.org/10.1007/3-540-45013-0_7

[28] Singh, G., Püschel, M., Vechev, M.: A practical construction for decomposing numerical abstract domains. Proc. ACM Program. Lang. **2**(POPL) (dec 2018), DOI: 10.1145/3158143, URL https://doi.org/10.1145/3158143