

# Numerical Invariants through Convex Relaxation and Max-Strategy Iteration<sup>\*</sup>

Thomas Martin Gawlitza<sup>1</sup> and Helmut Seidl<sup>2</sup>

<sup>1</sup> VERIMAG, Grenoble, France [Thomas.Gawlitza@imag.fr](mailto:Thomas.Gawlitza@imag.fr) <sup>\*\*</sup> and The University of Sydney, Australia [gawlitza@it.usyd.edu.au](mailto:gawlitza@it.usyd.edu.au)

<sup>2</sup> Technische Universität München, Institut für Informatik, München, Germany  
[seidl@in.tum.de](mailto:seidl@in.tum.de)

**Abstract.** In this article we develop a max-strategy improvement algorithm for computing least fixpoints of operators on  $\mathbb{R}^n$  (with  $\mathbb{R} := \mathbb{R} \cup \{\pm\infty\}$ ) that are point-wise maxima of finitely many monotone and order-concave operators. Computing the uniquely determined least fixpoint of such operators is a problem that occurs frequently in the context of numerical program/systems verification/analysis. As an example for an application we discuss how our algorithm can be applied to compute numerical invariants of programs by abstract interpretation based on quadratic templates.

## 1 Introduction

### 1.1 Motivation

Finding tight invariants for a given program or system is crucial for many applications related to program respectively system verification. Examples include linear recursive filters and numerical integration schemes. Abstract Interpretation as introduced by Cousot and Cousot [4] reduces the problem of finding tight invariants to the problem of finding the uniquely determined least fixpoint of a monotone operator. In this article, we consider the problem of inferring numerical invariants using abstract domains that are based on *templates*. That is, in addition to the program or system we want to analyze, a set of templates is given. These templates are arithmetic expressions in the program/system variables. The goal then is to compute small safe upper bounds on these templates. We may, for instance, be interested in computing a safe upper bound on the difference  $\mathbf{x}_1 - \mathbf{x}_2$  of two program/system variables  $\mathbf{x}_1, \mathbf{x}_2$  (at some specified control point of the program). Examples for template-based numerical invariants include *intervals* (upper and lower bounds on the values of the numerical program variables) [3], *zones* (intervals and additionally upper and lower bounds on the differences of program variables) [10, 11, 18], *octagons* (zones and additionally upper and lower bounds on the sum of program variables) [12], and,

---

<sup>\*</sup> This work was partially funded by the ANR project ASOPT.

<sup>\*\*</sup> VERIMAG is a joint laboratory of CNRS, Université Joseph Fourier and Grenoble INP.

more generally, *linear templates* (also called *template polyhedra*, upper bounds on arbitrary linear functions in the program variables, where the functions are given a priori) [15]. In this article, we focus on *quadratic templates* as considered by Adjé, Gaubert, and Goubault [1]. That is, a priori, a set of linear and quadratic functions in the program variables (the templates) is given and we are interested in computing small upper bounds on the values of these functions. An example for a quadratic template is represented by the quadratic polynomial  $2x_1^2 + 3x_2^2 + 2x_1x_2$ , where  $x_1$  and  $x_2$  are program variables.

When using such a template-based numerical abstract domain, the problem of finding the minimal inductive invariant, that can be expressed in the abstract domain specified by the templates, can be recast as a purely mathematical optimization problem, where the goal is to minimize a vector  $(\mathbf{x}_1, \dots, \mathbf{x}_n)$  subject to a set of inequalities of the form

$$\mathbf{x}_i \geq f(\mathbf{x}_1, \dots, \mathbf{x}_n). \quad (1)$$

Here,  $f$  is a *monotone* operator. The variables  $\mathbf{x}_1, \dots, \mathbf{x}_n$  take values in  $\mathbb{R} \cup \{\pm\infty\}$ . The variables are representing upper bounds on the values of the templates. Accordingly, the vector  $(\mathbf{x}_1, \dots, \mathbf{x}_n)$  is to be minimized w.r.t. the usual component-wise ordering. Because of the monotonicity of the operators  $f$  occurring in the right-hand sides of the inequalities and the completeness of the linearly ordered set  $\mathbb{R} \cup \{\pm\infty\}$ , the fixpoint theorem of Knaster/Tarski ensures the existence of a uniquely determined least solution.

Computing the least solution of such a constraint system is a difficult task. Even if we restrict our consideration to the special case of intervals as an abstract domain, which is, if the program variables are denoted by  $x_1, \dots, x_n$ , specified by the templates  $-x_1, x_1, \dots, -x_n, x_n$ , the static analysis problem is at least as hard as solving mean payoff games. The latter problem is a long outstanding problem which is in NP and in coNP, but not known to be in P.

A generic way of solving systems of constraints of the form (1) with right-hand sides that are monotone and variables that range over a complete lattice is given through the abstract interpretation framework of Cousot and Cousot [4]. Solving constraint systems in this framework is based on Kleene fixpoint iteration. However, in our case the lattice has infinite ascending chains. In this case, termination of the fixpoint iteration is ensured through an appropriate widening (see Cousot and Cousot [4]). Widening, however, buys termination for precision. Although the lost precision can be partially recovered through a subsequently performed narrowing iteration, there is no guarantee that the computed result is minimal.

## 1.2 Main Contribution

In this article, we study the case where the operators  $f$  in the right-hand side of the systems of equations of the form (1) are not only monotone, but additionally *order-concave* or even *concave* (concavity implies order-concavity, but not vice-versa). In the static program analysis application we consider in this article, the

end up in this comfortable situation by considering a semi-definite relaxation of the abstract semantics. The concavity of the mappings  $f$ , however, does not imply that the problem can be formulated as a convex optimization problem. The feasible space of the resulting mathematical optimization problem is normally neither order-convex nor order-concave and thus neither convex nor concave. In consequence, convex optimization methods cannot be directly applied. For the linear case (obtained when using used linear templates), we solved a long outstanding problem — namely the problem of solving mean payoff games in polynomial time — if we would be able to formalize the problem through a linear programming problem that can be constructed in polynomial time.

In this article, we exploit the fact that the operators  $f$  that occur in the right-hand sides of the system of inequalities of the form (1) we have to solve are not only order-concave, but also *monotone*. In other words: we do not require convexity of the feasible space, but we do require monotonicity in addition to the order-concavity. The main contribution of this article is an algorithm for computing least solutions of such systems of inequalities. The algorithm is based on strategy iteration. That is, we consider the process of solving the system of inequality as a game between a maximizer and a minimizer. The maximizer aims at minimizing the solution, whereas the minimizer aims at maximizing it. The algorithm iteratively constructs a winning strategy for the maximizer — a so-called max-strategy. It uses convex optimization techniques as sub-routines to evaluate parts of the constructed max-strategy. The concrete convex optimization technique that is used for the evaluation depends on the right-hand sides. In some cases linear programming is sufficient (see Gawlitza and Seidl [5, 8]), in other cases more sophisticated convex optimization techniques are required. The application we study in this article will require semi-definite programming.

An important example for monotone and order-concave operators are the operators that are *monotone* and *affine*. The class of monotone and order-concave operators is closed under the point-wise infimum operator. The point-wise infimum of a set of monotone and affine functions, for instance, is monotone and order-concave. Another example is the  $\sqrt{\cdot}$ -operator, which is defined by  $\sqrt{x} = \sup \{y \in \mathbb{R} \mid y^2 \leq x\}$  for all  $x \in \mathbb{R}$ .

An example for a system of inequalities of the class we are considering in this article is the following system of inequalities:

$$\mathbf{x}_1 \geq \frac{1}{2} \quad \mathbf{x}_1 \geq \sqrt{\mathbf{x}_2} \quad \mathbf{x}_2 \geq \mathbf{x}_1 \quad \mathbf{x}_2 \geq 1 + \sqrt{\mathbf{x}_2 - 1} \quad (2)$$

The uniquely determined least solution of the system (2) of inequalities is  $\mathbf{x}_1 = \mathbf{x}_2 = 1$ . We remind the reader again that the important property here is that the right-hand sides of (2) are monotone and order-concave.

The least solution of the system (2) is also the uniquely determined optimal solution of the following convex optimization problem:

$$\max \mathbf{x}_1 + \mathbf{x}_2 \quad \text{subject to} \quad \mathbf{x}_1 \leq \sqrt{\mathbf{x}_2} \quad \mathbf{x}_2 \leq \mathbf{x}_1 \quad (3)$$

Observe that the above convex optimization problem is in some sense a “subsystem” of the system (2). Such a “subsystem”, which we will call a max-strategy

later on, is obtained from the system (2) by selecting exactly one inequality of the form  $\mathbf{x}_i \geq e_i$  from (2) for each variable  $\mathbf{x}_i$  and replacing the relation  $\leq$  by the relation  $\geq$ . Note that there are exponentially many max-strategies. The algorithm we present in this article starts with a max-strategy and assigns a value to it. It then iteratively improves the current max-strategy and assigns a new value to it until the least solution is found. We utilize the monotonicity and the order-concavity of the right-hand sides to prove that our algorithm always terminates with the least solution after at most exponentially many improvement steps. Each improvement step can be executed by solving linearly many convex optimization problems, each of which can be constructed in linear time.

As a second contribution of this article, we show how any algorithm for solving such systems of inequalities, e.g., our max-strategy improvement algorithm, can be applied to infer numerical invariants based on quadratic templates. The method is based on the relaxed abstract semantics introduced by Adjé, Gaubert, and Goubault [1].

### 1.3 Related Work

The most closely related work is the work of Adjé, Gaubert, and Goubault [1]. They apply the *min-strategy improvement approach* of Costan, Gaubert, Goubault, Martel, and Putot [2] to the problem of inferring quadratic invariants of programs. In order to do so, they introduced the relaxed abstract semantics we are going to use in this article.<sup>3</sup> Their method, however, has several drawbacks compared to the method we present in this article. The first drawback is that it does not necessarily terminate after finitely many steps. In addition, even if it terminates, the computed solution is not guaranteed to be minimal. On the other hand, their approach also has substantial advantages that are especially important in practice. Firstly, it can be stopped at any time with a safe over-approximation to the least solution. Secondly, the computational steps that have to be performed are quite cheap compared to the ones we have to perform for the method we propose in this article. This is caused by the fact that the semi-definite programming problems (or in more general cases: convex programming problems) that have to be solved in each iteration are reasonable small. We refer to Gawlitza, Seidl, Adjé, Gaubert, and Goubault [9] for a detailed comparison between the max- and the min-strategy approach.

### 1.4 Previous Publications

Parts of this work were previously published in the proceedings of the Seventeenth International Static Analysis Symposium (SAS 2010). In contrast to the latter version, this article contains the full proofs and the precise treatment of infinities. In order to simplify some argumentations and to deal with infinities, we

---

<sup>3</sup> Adjé, Gaubert, and Goubault [1] in fact use the dual version of the relaxed abstract semantics we use in this article. However, this minor difference does not have any practical consequences.

modified some definitions quite substantially. In addition to these improvements, we provide a much more detailed study of different classes of order-concave functions and the consequences for our max-strategy improvement algorithm. We do not report on experimental results in this article. Such reports can be found in the article in the proceedings of the Seventeenth International Static Analysis Symposium (SAS 2010).

## 1.5 Structure

This article is structured as follows: Section 2 is dedicated to preliminaries. We study the class of *monotone* and *order-concave* operators in Section 3. The results we obtain in Section 3 are important to prove the correctness of our max-strategy improvement algorithm. The method and its correctness proof is presented in Section 4. In Section 5, we discuss the important special cases where the right-hand sides of the system of inequalities are parametrized convex optimization problems. This can be used to evaluate strategies more efficiently. These special cases are important, since they are present especially in the program analysis applications we mainly have in mind. In Section 6, we finally explain how our methods can be applied to a numerical static program analysis based on quadratic templates. We conclude with Section 7.

## 2 Preliminaries

*Vectors and Matrices* We denote the  $i$ -th row (resp.  $j$ -th column) of a matrix  $A$  by  $A_i$  (resp.  $A_{\cdot j}$ ). Accordingly,  $A_{i,j}$  denotes the component in the  $i$ -th row and the  $j$ -th column. We also use these notations for vectors and vector valued functions  $f : X \rightarrow Y^k$ , i.e.,  $f_i(x) = (f(x))_i$  for all  $i \in \{1, \dots, k\}$  and all  $x \in X$ .

*Sets, Functions, and Partial Functions* We write  $A \dot{\cup} B$  for the disjoint union of the two sets  $A$  and  $B$ , i.e.,  $A \dot{\cup} B$  stands for  $A \cup B$ , where we assume that  $A \cap B = \emptyset$ . For sets  $X$  and  $Y$ ,  $X \rightarrow Y$  denotes the set of all functions from  $X$  to  $Y$ , and  $X \rightsquigarrow Y$  denotes the set of all partial functions from  $X$  to  $Y$ . Note that  $X \rightarrow Y \subseteq X \rightsquigarrow Y \subseteq X \times Y$ . Accordingly, we apply the set operators  $\cup$ ,  $\cap$ , and  $\setminus$  also to partial functions. For  $X' \subseteq X$ , the restriction  $f|_{X'} : X' \rightarrow Y$  of a function  $f : X \rightarrow Y$  to  $X'$  is defined by  $f|_{X'} := f \cap X' \times Y$ . The domain and the codomain of a partial function  $f$  are denoted by  $\text{dom}(f)$  and  $\text{codom}(f)$ , respectively. For  $f : X \rightarrow Y$  and  $g : X \rightsquigarrow Y$ , we define  $f \oplus g : X \rightarrow Y$  by  $f \oplus g := f|_{X \setminus \text{dom}(g)} \cup g$ .

*Partially Ordered Sets* Let  $\mathbb{D}$  be a partially ordered set (partially ordered by the binary relation  $\leq$ ). Two elements  $x, y \in \mathbb{D}$  are called *comparable* if and only if  $x \leq y$  or  $y \leq x$ . For all  $x \in \mathbb{D}$ , we set  $\mathbb{D}_{\geq x} := \{y \in \mathbb{D} \mid y \geq x\}$  and  $\mathbb{D}_{\leq x} := \{y \in \mathbb{D} \mid y \leq x\}$ . We denote the *least upper bound* and the *greatest lower bound* of a set  $X \subseteq \mathbb{D}$  by  $\bigvee X$  and  $\bigwedge X$ , respectively, provided that it exists. The least element  $\bigvee \emptyset = \bigwedge \mathbb{D}$  (resp. the greatest element  $\bigwedge \emptyset = \bigvee \mathbb{D}$ ) is

denoted by  $\perp$  (resp.  $\top$ ), provided that it exists. A subset  $C \subseteq \mathbb{D}$  is called a *chain* if and only if  $C$  is linearly ordered by  $\leq$ , i.e., it holds  $x \leq y$  or  $y \leq x$  for all  $x, y \in C$ . For every subset  $X \subseteq \mathbb{D}$  of a set  $\mathbb{D}$  that is partially ordered by  $\leq$ , we set  $X \uparrow^{\mathbb{D}} := \{y \in \mathbb{D} \mid \exists x \in X. x \leq y\}$ . The set  $X \subseteq \mathbb{D}$  is called *upward closed w.r.t.  $\mathbb{D}$*  if and only if  $X \uparrow^{\mathbb{D}} = X$ . We omit the reference to  $\mathbb{D}$ , if  $\mathbb{D}$  is clear from the context.

*Monotonicity* Let  $\mathbb{D}_1, \mathbb{D}_2$  be partially ordered sets (partially ordered by  $\leq$ ). A mapping  $f : \mathbb{D}_1 \rightarrow \mathbb{D}_2$  is called *monotone* if and only if  $f(x) \leq f(y)$  for all  $x, y \in \mathbb{D}_1$  with  $x \leq y$ . A monotone function  $f$  is called *upward-chain-continuous* (resp. *downward-chain-continuous*) if and only if  $f(\bigvee C) = \bigvee f(C)$  (resp.  $f(\bigwedge C) = \bigwedge f(C)$ ) for every non-empty chain  $C$  with  $\bigvee C \in \text{dom}(f)$  (resp.  $\bigwedge C \in \text{dom}(f)$ ). It is called *chain-continuous* if and only if it is *upward-chain-continuous* and *downward-chain-continuous*.

*Complete Lattices* A partially ordered set  $\mathbb{D}$  is called a *complete lattice* if and only if  $\bigvee X$  and  $\bigwedge X$  exist for all  $X \subseteq \mathbb{D}$ . If  $\mathbb{D}$  is a complete lattice and  $x \in \mathbb{D}$ , then the sublattices  $\mathbb{D}_{\geq x}$  and  $\mathbb{D}_{\leq x}$  are also complete lattices. On a complete lattice  $\mathbb{D}$ , we define the binary operators  $\vee$  and  $\wedge$  by

$$x \vee y := \bigvee \{x, y\} \text{ and } x \wedge y := \bigwedge \{x, y\} \quad \text{for all } x, y \in \mathbb{D}, \quad (4)$$

respectively. If the complete lattice  $\mathbb{D}$  is a *complete linearly ordered set* (for instance  $\overline{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$ ), then  $\vee$  is the binary *maximum* operator and  $\wedge$  the binary *minimum* operator. For all binary operators  $\square \in \{\vee, \wedge\}$ , we also consider  $x_1 \square \dots \square x_k$  as the application of a  $k$ -ary operator. This will cause no problems, since the binary operators  $\vee$  and  $\wedge$  are associative and commutative.

*Fixpoints* Assume that the set  $\mathbb{D}$  is partially ordered by  $\leq$  and  $f : \mathbb{D} \rightarrow \mathbb{D}$  is a unary operator on  $\mathbb{D}$ . An element  $x \in \mathbb{D}$  is called *fixpoint* (resp. *pre-fixpoint*, resp. *post-fixpoint*) of  $f$  if and only if  $x = f(x)$  (resp.  $x \leq f(x)$ , resp.  $x \geq f(x)$ ). The set of all fixpoints (resp. pre-fixpoints, resp. post-fixpoints) of  $f$  is denoted by **Fix**( $f$ ) (resp. **PreFix**( $f$ ), resp. **PostFix**( $f$ )). We denote the least (resp. greatest) fixpoint of  $f$  — provided that it exists — by  $\mu f$  (resp.  $\nu f$ ). If the partially ordered set  $\mathbb{D}$  is a complete lattice and  $f$  is monotone, then the fixpoint theorem of Knaster/Tarski [16] ensures the existence of  $\mu f$  and  $\nu f$ . Moreover, we have  $\mu f = \bigwedge \mathbf{PostFix}(f)$  and dually  $\nu f = \bigvee \mathbf{PreFix}(f)$ .

We write  $\mu_{\geq x} f$  (resp.  $\nu_{\leq x} f$ ) for the least element in the set **Fix**( $f$ )  $\cap \mathbb{D}_{\geq x}$  (resp. **Fix**( $f$ )  $\cap \mathbb{D}_{\leq x}$ ). The existence of  $\mu_{\geq x} f$  (resp.  $\nu_{\leq x} f$ ) is ensured if  $\mathbb{D}_{\geq x}$  is a complete lattice and  $f|_{\mathbb{D}_{\geq x}}$  (resp.  $f|_{\mathbb{D}_{\leq x}}$ ) is a monotone operator on  $\mathbb{D}_{\geq x}$  (resp.  $\mathbb{D}_{\leq x}$ ), i.e., if  $\mathbb{D}_{\geq x}$  (resp.  $\mathbb{D}_{\leq x}$ ) is closed under the operator  $f$ . The latter condition is, for instance, fulfilled if  $\mathbb{D}$  is a complete lattice,  $f$  is a monotone operator on  $\mathbb{D}$ , and  $x$  is a pre-fixpoint (resp. post-fixpoint) of  $f$ .

*The Complete Lattice  $\overline{\mathbb{R}}^n$*  The set of real numbers is denoted by  $\mathbb{R}$ , and the complete linearly ordered set  $\mathbb{R} \cup \{\pm\infty\}$  is denoted by  $\overline{\mathbb{R}}$ . Therefore, the set  $\overline{\mathbb{R}}^n$

is a complete lattice that is partially ordered by  $\leq$ , where we write  $x \leq y$  if and only if  $x_{i.} \leq y_{i.}$  for all  $i \in \{1, \dots, n\}$ . As usual, we write  $x < y$  if and only if  $x \leq y$  and  $x \neq y$ . We write  $x \triangleleft y$  if and only if  $x_{i.} < y_{i.}$  for all  $i \in \{1, \dots, n\}$ . For  $f : \mathbb{R}^n \rightsquigarrow \mathbb{R}^m$ , we set  $\text{fdom}(f) := \{x \in \text{dom}(f) \cap \mathbb{R}^n \mid f(x) \in \mathbb{R}^m\}$ .

*The Vector Space  $\mathbb{R}^n$*  The standard base vectors of the Euclidian vector space  $\mathbb{R}^n$  are denoted by  $e_1, \dots, e_n$ . We denote the maximum norm on  $\mathbb{R}^n$  by  $\|\cdot\|$ , i.e.,  $\|x\| = \max \{|x_{i.}| \mid i \in \{1, \dots, n\}\}$  for all  $x \in \mathbb{R}^n$ . A vector  $x \in \mathbb{R}^n$  with  $\|x\| = 1$  is called a *unit vector*.

### 3 Morcave Operators

In this section, we introduce a notion of order-concavity for functions from the set  $\mathbb{R}^n \rightarrow \mathbb{R}^m$ . We then study the properties of functions that are monotone and order-concave. The results obtained in this section are used in Section 4 to prove the correctness of our max-strategy improvement algorithm.

#### 3.1 Monotone Operators on $\mathbb{R}^n$

In this subsection, we collect important properties about monotone operators on  $\mathbb{R}^n$ . We start with the following auxiliary lemma:

**Lemma 1.** *Let  $d, d' \in \mathbb{R}^n$  with  $d \triangleright 0$  and  $d' \geq 0$ . There exist  $j \in \{1, \dots, n\}$  and  $\lambda, \lambda_1, \dots, \lambda_n \geq 0$  such that  $\lambda_j = 0$  and  $\lambda d = d' + \sum_{i=1}^n \lambda_i e_i$ .*

*Proof.* Since  $d \triangleright 0$ , there exist a  $j \in \{1, \dots, n\}$  and a  $\lambda \geq 0$  such that  $\lambda d - d' \geq 0$  and  $(\lambda d - d')_j = 0$ . Thus, there exist  $\lambda_1, \dots, \lambda_n$  with  $\lambda_j = 0$  such that  $\lambda d - d' = \sum_{i=1}^n \lambda_i e_i$ .  $\square$

We now provide a sufficient criterium for a fixpoint  $x$  of a monotone partial operator  $f$  on  $\mathbb{R}^n$  for being the greatest pre-fixpoint of  $f$ .<sup>4</sup> Such sufficient criteria are crucial to prove the correctness of our max-strategy improvement algorithm.

**Lemma 2.** *Let  $f : \mathbb{R}^n \rightsquigarrow \mathbb{R}^n$  be monotone with  $\text{dom}(f)$  upward closed,  $f(x) = x$ , and  $k \in \mathbb{N}$ . Assume that, for every  $\epsilon > 0$ , there exists a unit vector  $d_\epsilon \triangleright 0$  such that  $f^k(x + \lambda d_\epsilon) \triangleleft x + \lambda d_\epsilon$  for all  $\lambda \geq \epsilon$ . Then,  $y \leq x$  for all  $y$  with  $y \leq f(y)$ , i.e.,  $x$  is the greatest pre-fixpoint of  $f$ .*

*Proof.* We show  $y \not\leq x \implies y \not\leq f(y)$ . For that, we first show the following statement:

$$y > x \implies y \not\leq f(y) \tag{5}$$

---

<sup>4</sup> Note that, since  $\mathbb{R}^n$  is not a complete lattice, the greatest pre-fixpoint of  $f$  is not necessarily the greatest fixpoint of  $f$ . The greatest fixpoint of the monotone operators  $f_1, f_2$  defined by  $f_1(x) = \frac{1}{2}x$  and  $f_2(x) = 2x$  for all  $x \in \mathbb{R}$ , for instance, is 0. This is also the greatest pre-fixpoint of  $f_1$ , but not the greatest pre-fixpoint of  $f_2$ , since  $f_2$  has no greatest pre-fixpoint.

For that, let  $y > x$ . Let  $\epsilon := \|y - x\|$ . By Lemma 1, there exist  $\lambda, \lambda_1, \dots, \lambda_n \geq 0$  with  $\lambda_j = 0$  for some  $j \in \{1, \dots, n\}$  such that  $\bar{y} := x + \lambda d_\epsilon = y + \sum_{i=1}^n \lambda_i e_i$  holds. We necessarily have  $\lambda \geq \epsilon$ . Using the monotonicity of  $f$  and the fact that  $f^k(\bar{y}) \triangleleft \bar{y}$  holds by assumption, we get  $f_j^k(y) \leq f_j^k(\bar{y}) < \bar{y}_j = y_j$ . Therefore,  $y \not\leq f(y)$ . Thus, we have shown (5). Now, let  $y \not\leq x$ . Thus,  $y' := x \vee y > x$ . Using (5) we get  $y' \not\leq f(y')$ . For the sake of contradiction assume that  $y \leq f(y)$  holds. Then we get  $f(y') = f(x \vee y) \geq f(x) \vee f(y) \geq x \vee y = y'$  — contradiction.  $\square$

In the remainder of this article, we only use the following corollary of Lemma 2:

**Lemma 3.** *Let  $f : \mathbb{R}^n \rightsquigarrow \mathbb{R}^n$  be monotone with  $\text{dom}(f)$  upward closed,  $f(x) = x$ , and  $k \in \mathbb{N}$ . Assume that there exists a unit vector  $d \triangleright 0$  such that  $f^k(x + \lambda d) \triangleleft x + \lambda d$  for all  $\lambda > 0$ . Then,  $y \leq x$  for all  $y$  with  $y \leq f(y)$ , i.e.,  $x$  is the greatest pre-fixpoint of  $f$ .*  $\square$

### 3.2 Monotone and Order-Concave Operators on $\mathbb{R}^n$

A set  $X \subseteq \mathbb{R}^n$  is called *order-convex* if and only if  $\lambda x + (1 - \lambda)y \in X$  for all comparable  $x, y \in X$  and all  $\lambda \in [0, 1]$ . It is called *convex* if and only if this condition holds for all  $x, y \in X$ . Every convex set is order-convex, but not vice-versa. If  $n = 1$ , then every order-convex set is convex. Every upward closed set is order-convex, but not necessarily convex.

A partial function  $f : \mathbb{R}^n \rightsquigarrow \mathbb{R}^m$  is called *order-convex* (resp. *order-concave*) if and only if  $\text{dom}(f)$  is order-convex and

$$f(\lambda x + (1 - \lambda)y) \leq (\text{resp. } \geq) \lambda f(x) + (1 - \lambda)f(y) \quad (6)$$

for all comparable  $x, y \in \text{dom}(f)$  and all  $\lambda \in [0, 1]$  (cf. Ortega and Rheinboldt [14]). A partial function  $f : \mathbb{R}^n \rightsquigarrow \mathbb{R}^m$  is called *convex* (resp. *concave*) if and only if  $\text{dom}(f)$  is convex and

$$f(\lambda x + (1 - \lambda)y) \leq (\text{resp. } \geq) \lambda f(x) + (1 - \lambda)f(y) \quad (7)$$

for all  $x, y \in \text{dom}(f)$  and all  $\lambda \in [0, 1]$  (cf. Ortega and Rheinboldt [14]). Every convex (resp. concave) partial function is order-convex (resp. order-concave), but not vice-versa. Note that  $f$  is (order-)concave if and only if  $-f$  is (order-)convex. Note also that  $f$  is (order-)convex (resp. (order-)concave) if and only if  $f_i$  is (order-)convex (resp. (order-)concave) for all  $i = 1, \dots, m$ . If  $n = 1$ , then every order-convex (resp. order-concave) partial function is convex (resp. concave). Every order-convex/order-concave partial function is chain-continuous. Every convex/concave partial function is continuous.

The set of (order-)convex (resp. (order-)concave) partial functions is not closed under composition. The functions  $f, g$  defined by  $f(x) = (x - 2)^2$  and  $g(x) = \frac{1}{x}$  for all  $x \in \mathbb{R}_{>0}$ , for instance, are both convex and thus also order-convex. However,  $f \circ g$  with  $(f \circ g)(x) = (\frac{1}{x} - 2)^2$  for all  $x \in \mathbb{R}_{>0}$  is neither convex nor order-convex.

In contrast to the set of all order-concave partial functions, the set of all partial functions that are monotone *and* order-concave is closed under composition:



**Lemma 4.** *Let  $f : \mathbb{R}^m \rightsquigarrow \mathbb{R}^n$  and  $g : \mathbb{R}^l \rightsquigarrow \mathbb{R}^m$  be monotone and order-convex (resp. order-concave). Assume that  $\text{codom}(g) \subseteq \text{dom}(f)$ . Then  $f \circ g$  is monotone and order-convex (resp. order-concave).*

*Proof.* We assume that  $f$  and  $g$  are order-convex. The other case can be proven dually. Let  $x, x' \in \text{dom}(g)$  with  $x \leq x'$ ,  $y = g(x)$ ,  $y' = g(x')$ . Since  $g$  is monotone, we get  $y \leq y'$ . Since  $f$  is monotone, we get  $(f \circ g)(x) = f(g(x)) = f(y) \leq f(y') = f(g(x')) = (f \circ g)(x')$ . Hence,  $f \circ g$  is monotone.

Let  $\lambda \in [0, 1]$ . Then  $(f \circ g)(\lambda x + (1 - \lambda)x') = f(g(\lambda x + (1 - \lambda)x')) \leq f(\lambda g(x) + (1 - \lambda)g(x')) = f(\lambda y + (1 - \lambda)y') \leq \lambda f(y) + (1 - \lambda)f(y') = \lambda f(g(x)) + (1 - \lambda)f(g(x')) = \lambda(f \circ g)(x) + (1 - \lambda)(f \circ g)(x')$ , because  $f$  is monotone, and  $f$  and  $g$  are order-convex. Hence,  $f \circ g$  is order-convex.  $\square$

### 3.3 Fixpoints of Monotone and Order-concave Operators on $\mathbb{R}^n$

We now study the fixpoints of monotone and order-concave partial operators on  $\mathbb{R}^n$ . We are in particular interested in developing a simple sufficient criterium for a fixpoint of a monotone and order-concave partial operator on  $\mathbb{R}^n$  for being the greatest pre-fixpoint of this partial operator. To prepare this, we first show the following lemma:

**Lemma 5.** *Let  $f : \mathbb{R}^n \rightsquigarrow \mathbb{R}^n$  be order-convex (resp. order-concave). Let  $x, x^* \in \text{dom}(f)$  with  $x^* = f(x^*)$ ,  $x \triangleright$  (resp.  $\triangleleft$ )  $f(x)$ ,  $d := x^* - x \triangleright 0$ . Then,  $x^* + \lambda d \triangleleft$  (resp.  $\triangleright$ )  $f(x^* + \lambda d)$  for all  $\lambda > 0$  with  $x^* + \lambda d \in \text{dom}(f)$ .*

*Proof.* We only consider the case that  $f$  is order-convex. The proof for the case that  $f$  is order-concave can be carried out dually. Let  $\lambda > 0$ . Assume for the sake of contradiction that there exists some  $i \in \{1, \dots, n\}$  such that  $(x^* + \lambda d)_i \geq f_i(x^* + \lambda d)$ . Since  $f_i$  is order-convex and  $x_i > f_i(x)$  holds, it follows  $x_i^* > f_i(x^*)$  — contradiction.  $\square$

We now use the results obtained so far to prove the following sufficient criterium for a fixpoint of a monotone and order-concave partial operator for being the greatest pre-fixpoint.

**Lemma 6.** *Let  $f : \mathbb{R}^n \rightsquigarrow \mathbb{R}^n$  be monotone and order-concave with  $\text{dom}(f)$  upward closed. Let  $x^*$  be a fixpoint of  $f$ ,  $x$  be a pre-fixpoint of  $f$  with  $x \triangleleft x^*$ , and  $\mu_{\geq x} f = x^*$ . Then,  $x^*$  is the greatest pre-fixpoint of  $f$ .*

*Proof.* Since  $f$  is chain-continuous and  $x \triangleleft x^*$  is a pre-fixpoint of  $f$ , there exists some  $k \in \mathbb{N}$  such that  $x \triangleleft f^k(x)$ . Let  $x'$  be a pre-fixpoint of  $f$ . Let  $d := x^* - x$ . Note that  $d \triangleright 0$ . Since  $f^k|_{\mathbb{R}_{\geq x}^n} = (f|_{\mathbb{R}_{\geq x}^n})^k$  is monotone and order-concave by Lemma 4, and  $x^*$  is a fixpoint of  $f^k$  and thus of  $f^k|_{\mathbb{R}_{\geq x}^n}$ , we get  $f^k(x^* + \lambda d) = f^k|_{\mathbb{R}_{\geq x}^n}(x^* + \lambda d) \triangleleft x^* + \lambda d$  for all  $\lambda \in \mathbb{R}_{>0}$  by Lemma 5. Thus, Lemma 3 gives us  $x' \leq x^*$ .  $\square$

*Example 1.* Let us consider the monotone and concave partial operator  $\sqrt{\cdot} : \mathbb{R} \rightsquigarrow \mathbb{R}$ . The points 0 and 1 are fixpoints of  $\sqrt{\cdot}$ , since  $0 = \sqrt{0}$ , and  $1 = \sqrt{1}$ . Since  $\frac{1}{2}$  is a pre-fixpoint of  $\sqrt{\cdot}$ ,  $\frac{1}{2} < 1$ , and  $\mu_{\geq \frac{1}{2}} \sqrt{\cdot} = 1$ , Lemma 6 implies that 1 is the greatest pre-fixpoint of  $\sqrt{\cdot}$ . Observe that for the fixpoint 0, there is no pre-fixpoint  $x \in \mathbb{R}$  of  $\sqrt{\cdot}$  with  $x < 0$ . Therefore, Lemma 6 cannot be applied.  $\square$

The following example shows that the criterium of Lemma 6 is sufficient, but not necessary:

*Example 2.* Let  $f : \mathbb{R} \rightarrow \mathbb{R}$  be defined by  $f(x) = 0 \wedge x$  for all  $x \in \mathbb{R}$ . Recall that  $\wedge$  denotes the minimum operator. Then, 0 is the greatest pre-fixpoint of  $f$ . However, there does not exist a  $x \in \mathbb{R}$  with  $x < 0$  such that  $\mu_{\geq x} f = 0$ , since  $\mu_{\geq x} f = x$  for all  $x \leq 0$ . Therefore, Lemma 6 cannot be applied to show that 0 is the greatest pre-fixpoint of  $f$ .  $\square$

The set  $\mathbb{R}^n$  can be identified with the set  $\{1, \dots, n\} \rightarrow \mathbb{R}$  which can be identified with the set  $\mathbf{X} \rightarrow \mathbb{R}$ , whenever  $|\mathbf{X}| = n$ . In the remainder of this article, we therefore identify the set  $(\mathbf{X} \rightarrow \mathbb{R}) \rightsquigarrow (\mathbf{X} \rightarrow \mathbb{R})$  with the set  $\mathbb{R}^n \rightsquigarrow \mathbb{R}^n$  — provided that  $|\mathbf{X}| = n$ . Usually, we use  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ . We use one or the other representation depending on which representation is more convenient in the given context.

Our next goal is to weaken the preconditions of Lemma 6, i.e., we aim at providing a weaker sufficient criterium for a fixpoint of a monotone and order-concave partial operator for being the greatest pre-fixpoint than the one provided by Lemma 6. The weaker sufficient criterium we are going to develop can, for instance, be applied to the following example:

*Example 3.* Let us consider the monotone and order-concave partial operator  $f : \mathbb{R}^2 \rightsquigarrow \mathbb{R}^2$  defined by  $f(x_1, x_2) := (x_2 + 1 \wedge 0, \sqrt{x_1})$  for all  $x_1, x_2 \in \mathbb{R}$ . Then,  $x^* = (x_1^*, x_2^*) = (0, 0)$  is the greatest pre-fixpoint of  $f$ . In order to prove this, assume that  $y = (y_1, y_2) > x^*$  is a pre-fixpoint of  $f$ , i.e.,  $y_1 \leq y_2 + 1$ ,  $y_1 \leq 0$ , and  $y_2 \leq \sqrt{y_1}$ . It follows immediately that  $y_1 \leq 0$  and thus  $y_2 \leq \sqrt{y_1} \leq \sqrt{0} = 0$ .

Lemma 6 is not applicable to prove that  $x^*$  is the greatest pre-fixpoint of  $f$ , because there is no pre-fixpoint  $x$  of  $f$  with  $x \triangleleft x^*$ . The situation is even worse: there is no  $x \in \text{dom}(f)$  with  $x \triangleleft x^*$ .

We observe that, locally at  $x^* = (0, 0)$ , the first component  $f_1$  of  $f$  does not depend on the second argument in the following sense: For every  $y = (y_1, y_2) \in \mathbb{R}^2$  with  $y_1 = x_1^* = 0$  and  $y_2 > x_2^* = 0$ , we have  $f_1(y) = 0 = f_1(x^*)$ . The weaker sufficient criterium we develop in the following takes this into account. That is, we will assume that the set of variables can be partitioned according to their dependencies. The sufficient criterium of Lemma 6 should then hold for each partition. In this example this means: there exists some  $x_1 < x_1^*$  with  $x_1 \leq f_1(x_1, x_2^*) = f_1(x_1, 0)$  and  $\mu_{\geq x_1} f_1(\cdot, 0) = x_1^* = 0$ , and there exists some  $x_2 < x_2^*$  with  $x_2 \leq f_2(x_1^*, x_2) = f_2(0, x_2)$  and  $\mu_{\geq x_2} f_2(0, \cdot) = x_2^* = 0$ . We could choose  $x_1 = x_2 = -1$ , for instance.  $\square$

In order to derive a sufficient criterium that is weaker than the sufficient criterium of Lemma 6, we should, as suggested in Example 3, partition the variables

according to their dependencies. In order to define a suitable notion of dependencies, let  $\mathbf{X}$  be a set of variables,  $f : (\mathbf{X} \rightarrow \mathbb{R}) \rightsquigarrow (\mathbf{X} \rightarrow \mathbb{R})$  be a monotone partial operator, and  $\rho : \mathbf{X} \rightarrow \mathbb{R}$ . For  $\mathbf{X}_1 \dot{\cup} \mathbf{X}_2 = \mathbf{X}$ , we write  $\mathbf{X}_1 \xrightarrow{f, \rho} \mathbf{X}_2$  if and only if

1.  $\mathbf{X}_1 = \emptyset$ ,
2.  $\mathbf{X}_2 = \emptyset$ , or
3. there exists an  $\rho' : \mathbf{X}_2 \rightarrow \mathbb{R}$  with  $\rho \oplus \rho' \in \text{dom}(f)$  and  $\rho' \triangleleft \rho|_{\mathbf{X}_2}$  such that  $f(\rho \oplus \rho')|_{\mathbf{X}_1} = f(\rho)|_{\mathbf{X}_1}$ .

Informally spoken,  $\mathbf{X}_1 \xrightarrow{f, \rho} \mathbf{X}_2$  states that — locally at  $\rho$  — the values of the variables from the set  $\mathbf{X}_1$  *do not depend* on the values of the variables from the set  $\mathbf{X}_2$ . Dependencies are only admitted in the opposite direction — from  $\mathbf{X}_1$  to  $\mathbf{X}_2$ .

*Example 4.* Let us again consider the monotone and order-concave partial operator  $f : \mathbb{R}^2 \rightsquigarrow \mathbb{R}^2$  from Example 3 defined by  $f(x_1, x_2) := (x_2 + 1 \wedge 0, \sqrt{x_1})$  for all  $x_1, x_2 \in \mathbb{R}$ . Note that  $f$  is not a total operator, since  $\sqrt{x_1}$  and thus  $f(x_1, x_2)$  is undefined for all  $x_1 < 0$ . Moreover, let  $x := (0, 0)$ . Recall that we identify the set  $\mathbb{R}^2$  with the set  $\{\mathbf{x}_1, \mathbf{x}_2\} \rightarrow \mathbb{R}$ . Especially, we identify  $x$  with the function  $\{\mathbf{x}_1 \mapsto 0, \mathbf{x}_2 \mapsto 0\}$ . Then, we have  $\{\mathbf{x}_1\} \xrightarrow{f, x} \{\mathbf{x}_2\}$ . That is, locally at  $x$ , the first component  $f_1$  of  $f$  does not depend on the second argument. In other words: locally at  $x$ , one can strictly decrease the value of the second argument without changing the value of the first component  $f_1$  of  $f$ . However, the second component  $f_2$  of  $f$  may, locally at  $x$ , depend on the first argument. In this example, this is actually the case: Locally at  $x$ , we cannot decrease the value of the first argument without changing the value of the second component  $f_2$  of  $f$ .  $\square$

If the partial operator  $f$  is monotone and order-concave, then the statement  $\mathbf{X}_1 \xrightarrow{f, \rho} \mathbf{X}_2$  also implies that, locally at  $\rho$ , the values of the  $\mathbf{X}_1$ -components of  $f$  do not *increase* if the values of the variables from  $\mathbf{X}_2$  increase:

**Lemma 7.** *Assume that  $f : (\mathbf{X} \rightarrow \mathbb{R}) \rightsquigarrow (\mathbf{X} \rightarrow \mathbb{R})$  is monotone and order-concave. If  $\mathbf{X}_1 \xrightarrow{f, \rho} \mathbf{X}_2$ , then  $(f(\rho \oplus \rho'))|_{\mathbf{X}_1} = (f(\rho))|_{\mathbf{X}_1}$  for all  $\rho' : \mathbf{X}_2 \rightarrow \mathbb{R}$  with  $\rho' \geq \rho|_{\mathbf{X}_2}$  and  $\rho \oplus \rho' \in \text{dom}(f)$ .  $\square$*

For  $\mathbf{X}_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}_k = \mathbf{X}$ , we write  $\mathbf{X}_1 \xrightarrow{f, \rho} \dots \xrightarrow{f, \rho} \mathbf{X}_k$  if and only if  $k = 1$  or  $\mathbf{X}_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}_j \xrightarrow{f, \rho} \mathbf{X}_{j+1} \dot{\cup} \dots \dot{\cup} \mathbf{X}_k$  for all  $j \in \{1, \dots, k-1\}$ .

Let  $\mathbf{X}$  and  $\mathbb{D}$  be sets,  $f : (\mathbf{X} \rightarrow \mathbb{D}) \rightsquigarrow (\mathbf{X} \rightarrow \mathbb{D})$ , and  $\mathbf{X}_1 \dot{\cup} \mathbf{X}_2 = \mathbf{X}$ . For  $\rho_2 : \mathbf{X}_2 \rightarrow \mathbb{D}$ , we define  $f \leftarrow \rho_2 : (\mathbf{X}_1 \rightarrow \mathbb{D}) \rightsquigarrow (\mathbf{X}_1 \rightarrow \mathbb{D})$  by

$$(f \leftarrow \rho_2)(\rho_1) := (f(\rho_1 \cup \rho_2))|_{\mathbf{X}_1} \quad \text{for all } \rho_1 : \mathbf{X}_1 \rightarrow \mathbb{D}. \quad (8)$$

Informally spoken,  $f \leftarrow \rho_2$  is the function that is obtained from  $f$  by fixing the values of the variables from the set  $\mathbf{X}_2$  according to variable assignment  $\rho_2$  and afterwards removing all variables from the set  $\mathbf{X}_2$ .

*Example 5.* Let us again consider the monotone and order-concave partial operator  $f : \mathbb{R}^2 \rightsquigarrow \mathbb{R}^2$  from Examples 3 and 4 that is defined by  $f(x_1, x_2) := (x_2 + 1 \wedge 0, \sqrt{x_1})$  for all  $x_1, x_2 \in \mathbb{R}$ . Let again  $x := (0, 0)$  be identified with  $x = \{\mathbf{x}_1 \mapsto 0, \mathbf{x}_2 \mapsto 0\}$ . Then  $(f \leftarrow x|_{\{\mathbf{x}_2\}})(\rho_1) = \{\mathbf{x}_1 \mapsto 0\}$  for all  $\rho_1 : \{\mathbf{x}_1\} \rightarrow \mathbb{R}$ , and  $(f \leftarrow x|_{\{\mathbf{x}_1\}})(\rho_2) = \{\mathbf{x}_2 \mapsto 0\}$  for all  $\rho_2 : \{\mathbf{x}_2\} \rightarrow \mathbb{R}$ .  $\square$

The weaker sufficient criterium for a fixpoint of a monotone and order-concave partial operator for being the greatest pre-fixpoint of this partial operator can now be formalized as follows:

**Definition 1 (Feasibility).** *Let  $f : (\mathbf{X} \rightarrow \mathbb{R}) \rightsquigarrow (\mathbf{X} \rightarrow \mathbb{R})$  be monotone and order-concave. A fixpoint  $\rho^*$  of  $f$  is called feasible if and only if there exist  $\mathbf{X}_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}_k = \mathbf{X}$  with  $\mathbf{X}_1 \xrightarrow{f, \rho^*} \dots \xrightarrow{f, \rho^*} \mathbf{X}_k$  such that, for each  $j \in \{1, \dots, k\}$ , there exists some pre-fixpoint  $\rho : \mathbf{X}_j \rightarrow \mathbb{R}$  of  $f \leftarrow \rho^*|_{\mathbf{X} \setminus \mathbf{X}_j}$  with  $\rho \triangleleft \rho^*|_{\mathbf{X}_j}$  such that  $\mu_{\geq \rho}(f \leftarrow \rho^*|_{\mathbf{X} \setminus \mathbf{X}_j}) = \rho^*|_{\mathbf{X}_j}$ .*  $\square$

*Example 6.* Let us again consider the monotone and order-concave partial operator  $f : \mathbb{R}^2 \rightsquigarrow \mathbb{R}^2$  from the Examples 3, 4, and 5 that is defined by  $f(x_1, x_2) := (x_2 + 1 \wedge 0, \sqrt{x_1})$  for all  $x_1, x_2 \in \mathbb{R}$ . We show that  $x := (0, 0)$  is a feasible fixpoint of  $f$ . From Example 3, we know that Lemma 6 is not applicable to prove that  $x$  is the greatest pre-fixpoint. Recall that we can identify the set  $\mathbb{R}^2$  with the set  $\{\mathbf{x}_1, \mathbf{x}_2\} \rightarrow \mathbb{R}$ , and hence  $x$  with  $\{\mathbf{x}_1 \mapsto 0, \mathbf{x}_2 \mapsto 0\}$ . We have  $\{\mathbf{x}_1\} \xrightarrow{f, x} \{\mathbf{x}_2\}$ . Moreover,  $\{\mathbf{x}_1 \mapsto -1\} \triangleleft x|_{\{\mathbf{x}_1\}}$  is a pre-fixpoint of  $f \leftarrow x|_{\{\mathbf{x}_2\}}$  with  $\mu_{\geq \{\mathbf{x}_1 \mapsto -1\}}(f \leftarrow x|_{\{\mathbf{x}_2\}}) = x|_{\{\mathbf{x}_1\}}$ , and  $\{\mathbf{x}_2 \mapsto -1\} \triangleleft x|_{\{\mathbf{x}_2\}}$  is a pre-fixpoint of  $f \leftarrow x|_{\{\mathbf{x}_1\}}$  with  $\mu_{\geq \{\mathbf{x}_2 \mapsto -1\}}(f \leftarrow x|_{\{\mathbf{x}_1\}}) = x|_{\{\mathbf{x}_2\}}$ . Thus,  $x$  is a feasible fixpoint of  $f$ .  $\square$

We now show that feasibility is indeed sufficient for a fixpoint to be the greatest pre-fixpoint. Since any fixpoint that fulfills the criterium given by Lemma 6 is feasible, but, as the Examples 3 and 6 show, not vice-versa, the following lemma is a strict generalization of Lemma 6.

**Lemma 8.** *Let  $f : (\mathbf{X} \rightarrow \mathbb{R}) \rightsquigarrow (\mathbf{X} \rightarrow \mathbb{R})$  be monotone and order-concave with  $\text{dom}(f)$  upward closed, and  $\rho^*$  be a feasible fixpoint of  $f$ . Then,  $\rho^*$  is the greatest pre-fixpoint of  $f$ .*

*Proof.* Since  $\rho^*$  is a feasible fixpoint of  $f$ , there exists  $\mathbf{X}_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}_k = \mathbf{X}$  with  $\mathbf{X}_1 \xrightarrow{f, \rho^*} \dots \xrightarrow{f, \rho^*} \mathbf{X}_k$  such that, for each  $j \in \{1, \dots, k\}$ , there exists some pre-fixpoint  $\rho_j$  of  $f \leftarrow \rho^*|_{\mathbf{X} \setminus \mathbf{X}_j}$  with  $\rho_j \triangleleft \rho^*|_{\mathbf{X}_j}$  and  $\mu_{\geq \rho_j}(f \leftarrow \rho^*|_{\mathbf{X} \setminus \mathbf{X}_j}) = \rho^*|_{\mathbf{X}_j}$ . Let  $\rho'$  be a pre-fixpoint of  $f$  with  $\rho' \geq \rho^*$  (it is sufficient to consider this case, since the statement that  $\rho''$  is a pre-fixpoint of  $f$  implies that  $\rho' := \rho^* \vee \rho'' \geq \rho^*$  is also a pre-fixpoint of  $f$ ). We show by induction on  $j$  that  $\rho'|_{\mathbf{X}_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}_j} = \rho^*|_{\mathbf{X}_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}_j}$  for all  $j \in \{1, \dots, k\}$ .

Firstly, assume that  $j = 1$ . Since  $\mathbf{X}_1 \xrightarrow{f, \rho^*} \mathbf{X}_2 \dot{\cup} \dots \dot{\cup} \mathbf{X}_k$ , Lemma 7 gives us  $\rho^*|_{\mathbf{X}_1} = (f(\rho^*))|_{\mathbf{X}_1} = (f \leftarrow \rho^*|_{\mathbf{X} \setminus \mathbf{X}_1})(\rho^*|_{\mathbf{X}_1}) = (f \leftarrow \rho'|_{\mathbf{X} \setminus \mathbf{X}_1})(\rho^*|_{\mathbf{X}_1})$ . Using the monotonicity we thus get  $\mu_{\geq \rho_1}(f \leftarrow \rho'|_{\mathbf{X} \setminus \mathbf{X}_1}) = \rho^*|_{\mathbf{X}_1}$ . Hence, Lemma 6 gives us that  $\rho^*|_{\mathbf{X}_1}$  is the greatest pre-fixpoint of  $f \leftarrow \rho'|_{\mathbf{X} \setminus \mathbf{X}_1}$ . Thus,  $\rho'|_{\mathbf{X}_1} = \rho^*|_{\mathbf{X}_1}$ .

Now, assume that  $j \in \{2, \dots, k\}$  and  $\rho'|_{\mathbf{X}_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}_{j-1}} = \rho^*|_{\mathbf{X}_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}_{j-1}}$ . It remains to show that  $\rho'|_{\mathbf{X}_j} = \rho^*|_{\mathbf{X}_j}$ . Since  $\mathbf{X}_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}_j \xrightarrow{f, \rho^*} \mathbf{X}_{j+1} \dot{\cup} \dots \dot{\cup} \mathbf{X}_k$  and  $\rho'|_{\mathbf{X}_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}_{j-1}} = \rho^*|_{\mathbf{X}_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}_{j-1}}$ , Lemma 7 gives us that  $\rho^*|_{\mathbf{X}_j} = (f(\rho^*))|_{\mathbf{X}_j} = (f \leftarrow \rho^*|_{\mathbf{X} \setminus \mathbf{X}_j})(\rho^*|_{\mathbf{X}_j}) = (f \leftarrow \rho'|_{\mathbf{X} \setminus \mathbf{X}_j})(\rho^*|_{\mathbf{X}_j})$ . By monotonicity, we thus get  $\mu_{\geq \rho_j}(f \leftarrow \rho'|_{\mathbf{X} \setminus \mathbf{X}_j}) = \rho^*|_{\mathbf{X}_j}$ . Hence, Lemma 6 gives us that  $\rho^*|_{\mathbf{X}_j}$  is the greatest pre-fixpoint of  $(f \leftarrow \rho'|_{\mathbf{X} \setminus \mathbf{X}_j})$ . Hence  $\rho'|_{\mathbf{X}_j} = \rho^*|_{\mathbf{X}_j}$ . Thus, we get  $\rho'|_{\mathbf{X}_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}_j} = \rho^*|_{\mathbf{X}_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}_j}$ .  $\square$

### 3.4 Morcave Operators on $\overline{\mathbb{R}}^n$

We now study total operators on  $\overline{\mathbb{R}}$  that are monotone and order-concave. For that, we firstly extend the notion of order-concavity that is defined for partial operators on  $\mathbb{R}$  to total operators on  $\overline{\mathbb{R}}$ . Before doing so, we start with the following observation:

**Lemma 9.** *Let  $f : \overline{\mathbb{R}}^n \rightarrow \overline{\mathbb{R}}^m$  be monotone. Then,  $\text{fdom}(f)$  is order-convex.*

*Proof.* Let  $x, y \in \text{fdom}(f)$  with  $x \leq y$  and  $\lambda \in [0, 1]$ . Because of the monotonicity of  $f$ , we get  $-\infty < f(x) \leq f(\lambda x + (1 - \lambda)y) \leq f(y) < \infty$ . Hence,  $\lambda x + (1 - \lambda)y \in \text{fdom}(f)$ . This proves the statement.  $\square$

We extend the notion of (order-)convexity/(order-)concavity from  $\mathbb{R}^n \rightsquigarrow \mathbb{R}$  to  $\overline{\mathbb{R}}^n \rightarrow \overline{\mathbb{R}}$  as follows: let  $f : \overline{\mathbb{R}}^n \rightarrow \overline{\mathbb{R}}$ , and  $I : \{1, \dots, n\} \rightarrow \{-\infty, \text{id}, \infty\}$  be a mapping. Here,  $-\infty$  denotes the function that assigns  $-\infty$  to every argument,  $\text{id}$  denotes the identity function, and  $\infty$  denotes the function that assigns  $\infty$  to every argument. We define the mapping  $f^{(I)} : \overline{\mathbb{R}}^n \rightarrow \overline{\mathbb{R}}$  by

$$f^{(I)}(x) := f(I(1)(x_1), \dots, I(n)(x_n)) \quad \text{for all } x \in \overline{\mathbb{R}}^n. \quad (9)$$

A function  $f : \overline{\mathbb{R}}^n \rightarrow \overline{\mathbb{R}}$  is called *(order-)concave* if and only if the following conditions are fulfilled for all mappings  $I : \{1, \dots, n\} \rightarrow \{-\infty, \text{id}, \infty\}$ :

1.  $\text{fdom}(f^{(I)})$  is (order-)convex.
2.  $f^{(I)}|_{\text{fdom}(f^{(I)})}$  is (order-)concave.
3. If  $\text{fdom}(f^{(I)}) \neq \emptyset$ , then  $f^{(I)}(x) < \infty$  for all  $x \in \mathbb{R}^n$ .

Note that, by Lemma 9, condition 1 is fulfilled for every monotone function  $f : \overline{\mathbb{R}}^n \rightarrow \overline{\mathbb{R}}$  and every mapping  $I : \{1, \dots, n\} \rightarrow \{-\infty, \text{id}, \infty\}$ . A monotone operator is order-concave if and only if the following conditions are fulfilled for all mappings  $I : \{1, \dots, n\} \rightarrow \{-\infty, \text{id}, \infty\}$ :

1.  $\text{fdom}(f^{(I)})$  is upward closed w.r.t.  $\mathbb{R}^n$ .
2.  $f^{(I)}|_{\text{fdom}(f^{(I)})}$  is order-concave.

In order to get more familiar with the above definition, we consider a few examples of order-concave operators on  $\overline{\mathbb{R}}$ :

*Example 7.* We consider the operators  $f : \overline{\mathbb{R}}^2 \rightarrow \overline{\mathbb{R}}$  and  $g : \overline{\mathbb{R}}^2 \rightarrow \overline{\mathbb{R}}$  that are defined by

$$f(x_1, x_2) := \sqrt{x_1}, \quad g(x_1, x_2) := \begin{cases} \sqrt{x_1} & \text{if } x_2 < \infty \\ x_1^2 & \text{if } x_2 = \infty \end{cases} \quad \text{for all } x_1, x_2 \in \overline{\mathbb{R}}. \quad (10)$$

Then,  $f|_{\mathbb{R}^2} = g|_{\mathbb{R}^2} = \{(x_1, x_2) \mapsto \sqrt{x_1} \mid x_1, x_2 \in \mathbb{R}\}$  is a monotone and concave operator on the convex set  $\text{fdom}(f) = \text{fdom}(g) = \mathbb{R}_{\geq 0} \times \mathbb{R}$ . Nevertheless,  $f$  is monotone and order-concave whereas  $g$  is neither monotone nor order-concave. In order to show that  $g$  is not order-concave, let  $I : \{1, 2\} \rightarrow \{-\infty, \text{id}, \infty\}$  be defined by  $I(1) = \text{id}$  and  $I(2) = \infty$ . Then,  $g^{(I)}(x_1, x_2) = x_1^2$  for all  $x_1, x_2 \in \overline{\mathbb{R}}$ . Hence,  $\text{fdom}(g^{(I)}) = \mathbb{R}^2$ . Obviously,  $g^{(I)}|_{\mathbb{R}^2}$  is not order-concave. Therefore,  $g$  is not order-concave.

Another example for a monotone and order-concave operator is the function  $h : \overline{\mathbb{R}}^2 \rightarrow \overline{\mathbb{R}}$  defined by

$$h(x_1, x_2) = \begin{cases} \sqrt{x_1} & \text{if } x_2 < \infty \\ \sqrt{x_1} + 1 & \text{if } x_2 = \infty \end{cases} \quad \text{for all } x_1, x_2 \in \overline{\mathbb{R}}. \quad (11)$$

Although  $h$  is an order-concave operator on  $\overline{\mathbb{R}}$ , it is not upward-chain-continuous, since, for  $C = \{(0, i) \mid i \in \mathbb{R}\}$ , we have  $h(\bigvee C) = h(0, \infty) = 1 > 0 = \bigvee \{0\} = \bigvee h(C)$ . We study different classes of monotone and order-concave functions in the remainder of this article.  $\square$

A mapping  $f : \overline{\mathbb{R}}^n \rightarrow \overline{\mathbb{R}}^m$  is called *(order-)concave* if and only if  $f_i$  is (order-)concave for all  $i \in \{1, \dots, m\}$ . A mapping  $f : \overline{\mathbb{R}}^n \rightarrow \overline{\mathbb{R}}^m$  is called *(order-)convex* if and only if  $-f$  is (order-)concave.

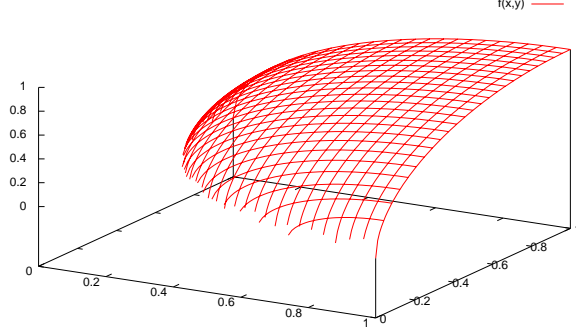
One property we expect from the set of all order-concave functions from  $\overline{\mathbb{R}}^n$  in  $\overline{\mathbb{R}}^m$  is that it is closed under the point-wise infimum operation. This is indeed the case:

**Lemma 10.** *Let  $\mathcal{F}$  be a set of (order-)concave functions from  $\overline{\mathbb{R}}^n$  in  $\overline{\mathbb{R}}^m$ . The function  $g : \overline{\mathbb{R}}^n \rightarrow \overline{\mathbb{R}}^m$  defined by  $g(x) := \bigwedge \{f(x) \mid f \in \mathcal{F}\}$  for all  $x \in \overline{\mathbb{R}}^n$  is (order-)concave.*

*Proof.* The statement can be proven straightforwardly. Note that  $g(x) = (\infty, \dots, \infty)$  for all  $x \in \overline{\mathbb{R}}^n$  if  $\mathcal{F} = \emptyset$ . In this case,  $g$  is concave.  $\square$

Monotone and order-concave functions play a central role in the remainder of this article. For the sake of simplicity, we give names to important classes of monotone and order-concave functions:

**Definition 2 (Morcave, Mcave, Cmorcave, and Cmcave Functions).** *A mapping  $f : \overline{\mathbb{R}}^n \rightarrow \overline{\mathbb{R}}^m$  is called morcave if and only if it is monotone and order-concave. It is called mcave if and only if it is monotone and concave. It is called cmorcave (resp. cmcav) if and only if it is morcave (resp. mcave) and  $f_i^{(I)}$  is upward-chain-continuous on  $\{x \in \overline{\mathbb{R}}^n \mid f_i^{(I)}(x) > -\infty\}$  for all  $I : \{1, \dots, n\} \rightarrow \{-\infty, \text{id}, \infty\}$  and all  $i \in \{1, \dots, m\}$ .*  $\square$



**Fig. 1.** Graph of a morcave operator  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ .

*Example 8.* Figure 1 shows the graph of a morcave function  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ .  $\square$

An important cmcave operator for our applications is the operator  $\wedge$  on  $\overline{\mathbb{R}}^n$ :

**Lemma 11.** *The operator  $\vee$  on  $\overline{\mathbb{R}}^n$  is monotone and convex, but not order-concave. The operator  $\wedge$  on  $\overline{\mathbb{R}}^n$  is cmcave, but not order-convex.*  $\square$

Next, we extend the definition of *affine* functions from  $\mathbb{R}^n \rightarrow \mathbb{R}^m$  to a definition of *affine* functions from  $\overline{\mathbb{R}}^n \rightarrow \overline{\mathbb{R}}^m$ .

**Definition 3 (Affine Functions).** *A function  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is called affine if and only if there exist some  $A \in \mathbb{R}^{m \times n}$  and some  $b \in \mathbb{R}^m$  such that  $f(x) = Ax + b$  for all  $x \in \mathbb{R}^n$ . A function  $f : \overline{\mathbb{R}}^n \rightarrow \overline{\mathbb{R}}^m$  is called affine if and only if there exist some  $A \in \mathbb{R}^{m \times n}$  and some  $b \in \overline{\mathbb{R}}^m$  such that  $f(x) = Ax + b$  for all  $x \in \overline{\mathbb{R}}^n$ .*  $\square$

In the above definition and throughout this article, we use the convention that  $-\infty + \infty = -\infty$ . Observe that an affine function  $f$  with  $f(x) = Ax + b$  is monotone, whenever all entries of the matrix  $A$  are non-negative.

**Lemma 12.** *Every affine function  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is concave and convex. Every monotone and affine function  $f : \overline{\mathbb{R}}^n \rightarrow \overline{\mathbb{R}}^m$  is cmcave.*  $\square$

In contrast to the class of monotone and order-concave operators on  $\mathbb{R}$ , the class of morcave operators on  $\mathbb{R}$  is not closed under functional composition, as the following example shows:

*Example 9.* We consider the functions  $f : \overline{\mathbb{R}} \rightarrow \overline{\mathbb{R}}$  and  $g : \overline{\mathbb{R}} \rightarrow \overline{\mathbb{R}}$  defined by

$$f(x) := \begin{cases} 0 & \text{if } x = -\infty \\ 1 & \text{if } x > -\infty \end{cases} \quad g(x) := \begin{cases} -\infty & \text{if } x < 0 \\ 0 & \text{if } x \geq 0 \end{cases} \quad \text{for all } x \in \overline{\mathbb{R}}. \quad (12)$$

The functions  $f$  and  $g$  are both morcave — even cmcave. However, observe that

$$(f \circ g)(x) = f(g(x)) := \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases} \quad \text{for all } x \in \overline{\mathbb{R}}. \quad (13)$$

Then,  $f \circ g$  is monotone, but not order-concave.  $\square$

As we will see, the composition  $f \circ g$  of two morcave operators  $f$  and  $g$  is again morcave, if  $f$  is additionally strict in the following sense: a function  $f : \overline{\mathbb{R}}^n \rightarrow \overline{\mathbb{R}}$  is called *strict* if and only if  $f(x) = -\infty$  for all  $x \in \overline{\mathbb{R}}^n$  with  $x_{k^*} = -\infty$  for some  $k \in \{1, \dots, n\}$ .

**Lemma 13.** *Let  $f : \overline{\mathbb{R}}^m \rightarrow \overline{\mathbb{R}}$  and  $g : \overline{\mathbb{R}}^n \rightarrow \overline{\mathbb{R}}^m$  be morcave. Assume additionally that  $f$  is strict. Then  $f \circ g$  is morcave.*

*Proof.* Since  $f$  and  $g$  are monotone,  $f \circ g$  is also monotone. In order to show that  $f \circ g$  is order-concave, let  $I : \{1, \dots, n\} \rightarrow \{-\infty, \text{id}, \infty\}$  and  $h := (f \circ g)^{(I)}$ .

1. The set  $\text{fdom}(h)$  is order-convex by Lemma 9, since  $h$  is monotone.
2. Let  $x, y \in \text{fdom}(h)$  with  $x \leq y$ ,  $\lambda \in [0, 1]$ , and  $z := \lambda x + (1 - \lambda)y$ . Moreover, let  $x' := g^{(I)}(x)$ ,  $y' := g^{(I)}(y)$ , and  $z' := g^{(I)}(z)$ . The strictness of  $f$  implies that  $z' \triangleright (-\infty, \dots, -\infty)$ . Since  $g^{(I)}$  is monotone, we get  $x' \leq y'$ . We define  $I' : \{1, \dots, m\} \rightarrow \{-\infty, \text{id}, \infty\}$  by

$$I'(k) = \begin{cases} \text{id} & \text{if } z'_{k^*} \in \mathbb{R} \\ \infty & \text{if } z'_{k^*} = \infty \end{cases} \quad \text{for all } k \in \{1, \dots, m\}.$$

We get:

$$\begin{aligned} h(z) &= f(g^{(I)}(z)) \\ &= f^{(I')}(g^{(I)}(z)) \\ &\geq f^{(I')}(\lambda g^{(I)}(x) + (1 - \lambda)g^{(I)}(y)) \\ &\quad \text{(Monotonicity, Order-Concavity)} \\ &= f^{(I')}(\lambda x' + (1 - \lambda)y') \\ &\geq \lambda f^{(I')}(x') + (1 - \lambda)f^{(I')}(y') && \text{(Order-Concavity)} \\ &= \lambda f^{(I')}(g^{(I)}(x)) + (1 - \lambda)f^{(I')}(g^{(I)}(y)) \\ &\geq \lambda f(g^{(I)}(x)) + (1 - \lambda)f(g^{(I)}(y)) && (f \leq f^{(I')}) \\ &= \lambda h(x) + (1 - \lambda)h(y) \end{aligned}$$

Hence,  $h|_{\text{fdom}(h)}$  is order-concave.

3. Now, assume that  $\text{fdom}(h) \neq \emptyset$ . That is, there exists some  $y \in \mathbb{R}^n$  with  $h(y) = f(g^{(I)}(y)) \in \mathbb{R}$ . Since  $f$  is strict, we get  $y' := g^{(I)}(y) \triangleright (-\infty, \dots, -\infty)$ . Let  $I' : \{1, \dots, m\} \rightarrow \{-\infty, \text{id}, \infty\}$  be defined by

$$I'(k) = \begin{cases} \text{id} & \text{if } y'_{k^*} \in \mathbb{R} \\ \infty & \text{if } y'_{k^*} = \infty \end{cases} \quad \text{for all } k \in \{1, \dots, m\}.$$



Since  $g$  is order-concave, we get  $g_k^{(I)}(x) < \infty$  for all  $x \in \mathbb{R}^n$  and all  $k \in \{1, \dots, m\}$  with  $y'_{k.} \in \mathbb{R}$ . Since  $f$  is order-concave, we get  $f^{(I')}(x) < \infty$  for all  $x \in \mathbb{R}^n$ . Thus, by monotonicity, we get  $f^{(I')} \circ g^{(I)}(x) = f^{(I')}(g^{(I)}(x)) < \infty$  for all  $x \in \mathbb{R}^n$ . Since we have  $h = (f \circ g)^{(I)} \leq f^{(I')} \circ g^{(I)}$  by construction, we get  $h(x) < \infty$  for all  $x \in \mathbb{R}^n$ .  $\square$

## 4 Solving Systems of $\vee$ -morcave Equations

In this section, we present our  $\vee$ -strategy improvement algorithm for computing least solutions of systems of  $\vee$ -morcave equations and prove its correctness.

### 4.1 Systems of $\vee$ -morcave Equations

Assume that a fixed finite set  $\mathbf{X}$  of variables and a complete linearly ordered set  $\mathbb{D}$  is given. Assume that  $\mathbb{D}$  is partially ordered by  $\leq$ . We consider equations of the form  $\mathbf{x} = e$  over  $\mathbb{D}$ , where  $\mathbf{x} \in \mathbf{X}$  is a variable and  $e$  is an expression over  $\mathbb{D}$ . A *system*  $\mathcal{E}$  of (fixpoint-)equations over  $\mathbb{D}$  is a finite set  $\{\mathbf{x}_1 = e_1, \dots, \mathbf{x}_n = e_n\}$  of equations, where  $\mathbf{x}_1, \dots, \mathbf{x}_n$  are pairwise distinct variables. We denote the set  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  of variables occurring in  $\mathcal{E}$  by  $\mathbf{X}_{\mathcal{E}}$ . We drop the subscript, whenever it is clear from the context.

For a variable assignment  $\rho : \mathbf{X} \rightarrow \mathbb{D}$ , an expression  $e$  is mapped to a value  $\llbracket e \rrbracket \rho$  by setting  $\llbracket \mathbf{x} \rrbracket \rho := \rho(\mathbf{x})$ , and  $\llbracket f(e_1, \dots, e_k) \rrbracket \rho := f(\llbracket e_1 \rrbracket \rho, \dots, \llbracket e_k \rrbracket \rho)$ , where  $\mathbf{x} \in \mathbf{X}$ ,  $f$  is a  $k$ -ary operator ( $k = 0$  is possible; then  $f$  is a constant), for instance  $+$ , and  $e_1, \dots, e_k$  are expressions. For every system  $\mathcal{E}$  of equations, we define the unary operator  $\llbracket \mathcal{E} \rrbracket$  on  $\mathbf{X} \rightarrow \mathbb{D}$  by setting  $(\llbracket \mathcal{E} \rrbracket \rho)(\mathbf{x}) := \llbracket e \rrbracket \rho$  for all equations  $\mathbf{x} = e$  from  $\mathcal{E}$  and all  $\rho : \mathbf{X} \rightarrow \mathbb{D}$ . A *solution* is a fixpoint of  $\llbracket \mathcal{E} \rrbracket$ , i.e., it is a variable assignment  $\rho$  such that  $\rho = \llbracket \mathcal{E} \rrbracket \rho$ . We denote the set of all solutions of  $\mathcal{E}$  by  $\mathbf{Sol}(\mathcal{E})$ .

The set  $\mathbf{X} \rightarrow \mathbb{D}$  of all *variable assignments* is a complete lattice. For  $\rho, \rho' : \mathbf{X} \rightarrow \mathbb{D}$ , we write  $\rho \triangleleft \rho'$  (resp.  $\rho \triangleright \rho'$ ) if and only if  $\rho(\mathbf{x}) < \rho'(\mathbf{x})$  (resp.  $\rho(\mathbf{x}) > \rho'(\mathbf{x})$ ) for all  $\mathbf{x} \in \mathbf{X}$ . For  $d \in \mathbb{D}$ ,  $\underline{d}$  denotes the variable assignment  $\{\mathbf{x} \mapsto d \mid \mathbf{x} \in \mathbf{X}\}$ . A variable assignment  $\rho$  with  $\underline{\perp} \triangleleft \rho \triangleleft \underline{\top}$  is called *finite*. A pre-solution (resp. post-solution) is a variable assignment  $\rho$  such that  $\rho \leq \llbracket \mathcal{E} \rrbracket \rho$  (resp.  $\rho \geq \llbracket \mathcal{E} \rrbracket \rho$ ) holds. The set of pre-solutions (resp. the set of post-solutions) is denoted by  $\mathbf{PreSol}(\mathcal{E})$  (resp.  $\mathbf{PostSol}(\mathcal{E})$ ). The least solution (resp. the greatest solution) of a system  $\mathcal{E}$  of equations is denoted by  $\mu \llbracket \mathcal{E} \rrbracket$  (resp.  $\nu \llbracket \mathcal{E} \rrbracket$ ), provided that it exists. For a pre-solution  $\rho$  (resp. for a post-solution  $\rho$ ),  $\mu_{\geq \rho} \llbracket \mathcal{E} \rrbracket$  (resp.  $\nu_{\leq \rho} \llbracket \mathcal{E} \rrbracket$ ) denotes the least solution that is greater than or equal to  $\rho$  (resp. the greatest solution that is less than or equal to  $\rho$ ).

An expression  $e$  (resp. an (fixpoint-)equation  $\mathbf{x} = e$ ) is called *monotone* if and only if  $\llbracket e \rrbracket$  is monotone. In our setting, the fixpoint theorem of Knaster/Tarski can be stated as follows: every system  $\mathcal{E}$  of monotone fixpoint equations over a complete lattice has a least solution  $\mu \llbracket \mathcal{E} \rrbracket$  and a greatest solution  $\nu \llbracket \mathcal{E} \rrbracket$ . Furthermore, we have  $\mu \llbracket \mathcal{E} \rrbracket = \bigwedge \mathbf{PostSol}(\mathcal{E})$  and  $\nu \llbracket \mathcal{E} \rrbracket = \bigvee \mathbf{PreSol}(\mathcal{E})$ .

**Definition 4 ( $\vee$ -morcave Equations).** An expression  $e$  (resp. fixpoint equation  $\mathbf{x} = e$ ) over  $\overline{\mathbb{R}}$  is called *morcave* (resp. *cmorcave*, resp. *mcave*, resp. *cmcave*) if and only if  $\llbracket e \rrbracket$  is *morcave* (resp. *cmorcave*, resp. *mcave*, resp. *cmcave*). An expression  $e$  (resp. fixpoint equation  $\mathbf{x} = e$ ) over  $\overline{\mathbb{R}}$  is called  $\vee$ -*morcave* (resp.  $\vee$ -*cmorcave*, resp. *mcave*, resp. *cmcave*) if and only if  $e = e_1 \vee \dots \vee e_k$ , where  $e_1, \dots, e_k$  are *morcave* (resp. *cmorcave*, resp. *mcave*, resp. *cmcave*).  $\square$

*Example 10.* The square root operator  $\sqrt{\cdot} : \overline{\mathbb{R}} \rightarrow \overline{\mathbb{R}}$  (defined by  $\sqrt{x} := \sup \{y \in \mathbb{R} \mid y^2 \leq x\}$  for all  $x \in \overline{\mathbb{R}}$ ) is *cmcave*. The least solution of the system  $\mathcal{E} = \{\mathbf{x} = \frac{1}{2} \vee \sqrt{\mathbf{x}}\}$  of  $\vee$ -*cmcave* equations is  $\mu\llbracket \mathcal{E} \rrbracket = 1$ .  $\square$

**Definition 5 ( $\vee$ -strategies).** A  $\vee$ -strategy  $\sigma$  for a system  $\mathcal{E}$  of equations is a function that maps every expression  $e_1 \vee \dots \vee e_k$  occurring in  $\mathcal{E}$  to one of the immediate sub-expressions  $e_j$ ,  $j \in \{1, \dots, k\}$ . We denote the set of all  $\vee$ -strategies for  $\mathcal{E}$  by  $\Sigma_{\mathcal{E}}$ . We drop the subscript, whenever it is clear from the context. The application  $\mathcal{E}(\sigma)$  of  $\sigma$  to  $\mathcal{E}$  is defined by  $\mathcal{E}(\sigma) := \{\mathbf{x} = \sigma(e) \mid \mathbf{x} = e \in \mathcal{E}\}$ .

*Example 11.* The two  $\vee$ -strategies  $\sigma_1, \sigma_2$  for the system  $\mathcal{E}$  of  $\vee$ -*cmcave* equations defined in Example 10 lead to the systems  $\mathcal{E}(\sigma_1) = \{\mathbf{x} = \frac{1}{2}\}$  and  $\mathcal{E}(\sigma_2) = \{\mathbf{x} = \sqrt{\mathbf{x}}\}$  of *cmcave* equations.  $\square$

## 4.2 The Strategy Improvement Algorithm

We now present the  $\vee$ -strategy improvement algorithm in a general setting. That is, we consider arbitrary systems of monotone equations over arbitrary *complete linearly ordered sets*  $\mathbb{D}$ . The algorithm iterates over  $\vee$ -strategies. It maintains a current  $\vee$ -strategy  $\sigma$  and a current *approximate*  $\rho$  to the least solution. A so-called  *$\vee$ -strategy improvement operator* is used to determine a next, improved  $\vee$ -strategy  $\sigma'$ . Whether or not a  $\vee$ -strategy  $\sigma'$  is an *improvement* of the current  $\vee$ -strategy  $\sigma$  may depend on the current approximate  $\rho$ :

**Definition 6 (Improvements).** Let  $\mathcal{E}$  be a system of monotone equations over a complete linearly ordered set. Let  $\sigma, \sigma' \in \Sigma$  be  $\vee$ -strategies for  $\mathcal{E}$  and  $\rho$  be a pre-solution of  $\mathcal{E}(\sigma)$ . The  $\vee$ -strategy  $\sigma'$  is called an *improvement* of  $\sigma$  w.r.t.  $\rho$  if and only if the following conditions are fulfilled:

1. If  $\rho \notin \mathbf{Sol}(\mathcal{E})$ , then  $\llbracket \mathcal{E}(\sigma') \rrbracket \rho > \rho$ .
2. For all expressions  $e = e_1 \vee \dots \vee e_k$  of  $\mathcal{E}$  the following holds: If  $\sigma'(e) \neq \sigma(e)$ , then  $\llbracket \sigma'(e) \rrbracket \rho > \llbracket \sigma(e) \rrbracket \rho$ .

A function  $P_{\vee}$  that assigns an improvement of  $\sigma$  w.r.t.  $\rho$  to every pair  $(\sigma, \rho)$ , where  $\sigma$  is a  $\vee$ -strategy and  $\rho$  is a pre-solution of  $\mathcal{E}(\sigma)$ , is called a  *$\vee$ -strategy improvement operator*. If it is impossible to improve  $\sigma$  w.r.t.  $\rho$ , then we necessarily have  $P_{\vee}(\sigma, \rho) = \sigma$ .  $\square$

*Example 12.* Consider the system  $\mathcal{E} = \{\mathbf{x}_1 = \mathbf{x}_2 + 1 \wedge 0, \mathbf{x}_2 = -1 \vee \sqrt{\mathbf{x}_1}\}$  of  $\vee$ -cmcave equations. Let  $\sigma_1$  and  $\sigma_2$  be the  $\vee$ -strategies for  $\mathcal{E}$  such that

$$\begin{aligned}\mathcal{E}(\sigma_1) &= \{\mathbf{x}_1 = \mathbf{x}_2 + 1 \wedge 0, \mathbf{x}_2 = -1\}, \text{ and} \\ \mathcal{E}(\sigma_2) &= \{\mathbf{x}_1 = \mathbf{x}_2 + 1 \wedge 0, \mathbf{x}_2 = \sqrt{\mathbf{x}_1}\}.\end{aligned}$$

The variable assignment  $\rho := \{\mathbf{x}_1 \mapsto 0, \mathbf{x}_2 \mapsto -1\}$  is a solution and thus also a pre-solution of  $\mathcal{E}(\sigma_1)$ . The  $\vee$ -strategy  $\sigma_2$  is an improvement of the  $\vee$ -strategy  $\sigma_1$  w.r.t.  $\rho$ .  $\square$

We can now formulate the  $\vee$ -strategy improvement algorithm for computing least solutions of systems of monotone equations over complete linearly ordered sets. This algorithm is parameterized with a  $\vee$ -strategy improvement operator  $P_\vee$ . The input is a system  $\mathcal{E}$  of monotone equations over a complete linearly ordered set, a  $\vee$ -strategy  $\sigma_{\text{init}}$  for  $\mathcal{E}$ , and a pre-solution  $\rho_{\text{init}}$  of  $\mathcal{E}(\sigma_{\text{init}})$ . In order to compute the *least* and not just some solution, we additionally require that  $\rho_{\text{init}} \leq \mu[\mathcal{E}]$  holds:

---

**Algorithm 1** The  $\vee$ -Strategy Improvement Algorithm

---

Parameter: A  $\vee$ -strategy improvement operator  $P_\vee$

Input :  $\begin{cases} \text{A system } \mathcal{E} \text{ of monotone equations over a complete linearly ordered set} \\ \text{A } \vee\text{-strategy } \sigma_{\text{init}} \text{ for } \mathcal{E} \\ \text{A pre-solution } \rho_{\text{init}} \text{ of } \mathcal{E}(\sigma_{\text{init}}) \text{ with } \rho_{\text{init}} \leq \mu[\mathcal{E}] \end{cases}$

Output : The least solution  $\mu[\mathcal{E}]$  of  $\mathcal{E}$

$\sigma \leftarrow \sigma_{\text{init}};$

$\rho \leftarrow \rho_{\text{init}};$

**while** ( $\rho \notin \mathbf{Sol}(\mathcal{E})$ ) {  
 $\sigma \leftarrow P_\vee(\sigma, \rho);$   
 $\rho \leftarrow \mu_{\geq \rho}[\mathcal{E}(\sigma)];$   
}

**return**  $\rho;$

---

*Example 13.* We consider the system

$$\mathcal{E} = \left\{ \mathbf{x} = -\infty \vee \frac{1}{2} \vee \sqrt{\mathbf{x}} \vee \frac{7}{8} + \sqrt{\mathbf{x} - \frac{47}{64}} \right\} \quad (14)$$

of  $\vee$ -cmorcake equations. We start with the  $\vee$ -strategy  $\sigma_0$  that leads to the system

$$\mathcal{E}(\sigma_0) = \{\mathbf{x} = -\infty\} \quad (15)$$

of cmorcake equations. Then  $\rho_0 := \underline{-\infty}$  is a feasible solution of  $\mathcal{E}(\sigma_0)$ . Since  $\rho_0 \notin \mathbf{Sol}(\mathcal{E})$ , we improve  $\sigma_0$  w.r.t.  $\rho_0$  to the  $\vee$ -strategy  $\sigma_1$  that gives us

$$\mathcal{E}(\sigma_1) = \left\{ \mathbf{x} = \frac{1}{2} \right\}. \quad (16)$$

Then,  $\rho_1 := \mu_{\geq \rho_0} \llbracket \sigma_1 \rrbracket = \{\mathbf{x} \mapsto \frac{1}{2}\}$ . Since  $\sqrt{\frac{1}{2}} > \frac{1}{2}$  and  $\frac{7}{8} + \sqrt{\frac{1}{2} - \frac{47}{64}} < \frac{1}{2}$  hold, we improve the strategy  $\sigma_1$  w.r.t.  $\rho_1$  to the  $\vee$ -strategy  $\sigma_2$  with

$$\mathcal{E}(\sigma_2) = \{\mathbf{x} = \sqrt{\mathbf{x}}\}.$$

We get  $\rho_2 := \mu_{\geq \rho_1} \llbracket \sigma_2 \rrbracket = \{\mathbf{x} \mapsto 1\}$ . Since  $\frac{7}{8} + \sqrt{1 - \frac{47}{64}} > \frac{7}{8} + \sqrt{1 - \frac{60}{64}} = \frac{9}{8} > 1$ , we get  $\sigma_3 = \{\mathbf{x} = \frac{7}{8} + \sqrt{\mathbf{x} - \frac{47}{64}}\}$ . Finally we get  $\rho_3 := \mu_{\geq \rho_2} \llbracket \sigma_3 \rrbracket = \{\mathbf{x} \mapsto 2\}$ . The algorithm terminates, because  $\rho_3$  solves  $\mathcal{E}$ . Therefore,  $\rho_3 = \mu \llbracket \mathcal{E} \rrbracket$ .  $\square$

In the following lemma, we collect basic properties that can be proven by induction straightforwardly:

**Lemma 14.** *Let  $\mathcal{E}$  be a system of monotone equations over a complete linearly ordered set. For all  $i \in \mathbb{N}$ , let  $\rho_i$  be the value of the program variable  $\rho$  and  $\sigma_i$  be the value of the program variable  $\sigma$  in the  $\vee$ -strategy improvement algorithm (Algorithm 1) after the  $i$ -th evaluation of the loop-body. The following statements hold for all  $i \in \mathbb{N}$ :*

1.  $\rho_i \leq \mu \llbracket \mathcal{E} \rrbracket$ .
2.  $\rho_i \in \mathbf{PreSol}(\mathcal{E}(\sigma_{i+1}))$ .
3. If  $\rho_i < \mu \llbracket \mathcal{E} \rrbracket$ , then  $\rho_{i+1} > \rho_i$ .
4. If  $\rho_i = \mu \llbracket \mathcal{E} \rrbracket$ , then  $\rho_{i+1} = \rho_i$ .

*If the execution of the  $\vee$ -strategy improvement algorithm terminates, then the least solution  $\mu \llbracket \mathcal{E} \rrbracket$  of  $\mathcal{E}$  is computed.*  $\square$

In the following, we apply our algorithm to solve systems of  $\vee$ -morcave equations. In the next subsection, we show that our algorithm terminates in this case. More precisely, it returns the least solution at the latest after considering every  $\vee$ -strategy at most  $|\mathbf{X}|$  times. We additionally provide an important characterization of  $\mu_{\geq \rho} \llbracket \mathcal{E}(\sigma) \rrbracket$  which allows us to compute it using convex optimization techniques. Here,  $\sigma$  are the  $\vee$ -strategies and  $\rho$  are the pre-solutions  $\rho$  of  $\mathcal{E}(\sigma)$  that can be encountered during the execution of the algorithm.

### 4.3 Feasibility

In this subsection, we extend the notion of feasibility as defined in Definition 1. We then show that feasibility is preserved during the execution of the  $\vee$ -strategy improvement algorithm. In the next subsection, we finally make use of the feasibility.

We denote by  $\mathcal{E}[x_1/\mathbf{X}_1, \dots, x_n/\mathbf{X}_n]$  the equation system that is obtained from the equation system  $\mathcal{E}$  by simultaneously replacing, for all  $i \in \{1, \dots, n\}$ , every occurrence of a variable from the set  $\mathbf{X}_i$  in the right-hand sides of  $\mathcal{E}$  by the value  $x_i$ .

**Definition 7 (Feasibility).** Let  $\mathcal{E}$  be a system of morcave equations. A finite solution  $\rho$  of  $\mathcal{E}$  is called  $(\mathcal{E})$ -feasible if and only if  $\rho$  is a feasible fixpoint of  $\llbracket \mathcal{E} \rrbracket$ . A pre-solution  $\rho$  of  $\mathcal{E}$  with  $\llbracket \mathcal{E} \rrbracket \rho \triangleright \underline{-\infty}$  is called  $(\mathcal{E})$ -feasible if and only if  $\rho'|_{\mathbf{X}'}$  is a feasible finite solution of  $\mathcal{E}' := \{\mathbf{x} = e \in \mathcal{E} \mid \mathbf{x} \in \mathbf{X}'\}[\infty/(\mathbf{X} \setminus \mathbf{X}')]$ , where  $\rho' := \mu_{\geq \rho} \llbracket \mathcal{E} \rrbracket$  and  $\mathbf{X}' := \{\mathbf{x} \in \mathbf{X} \mid \rho'(\mathbf{x}) < \infty\}$ . A pre-solution  $\rho$  of  $\mathcal{E}$  is called feasible if and only if  $e = -\infty$  for all  $\mathbf{x} = e \in \mathcal{E}$  with  $\llbracket e \rrbracket \rho = -\infty$ , and  $\rho|_{\mathbf{X}'}$  is a feasible pre-solution of  $\mathcal{E}' := \{\mathbf{x} = e \in \mathcal{E} \mid \mathbf{x} \in \mathbf{X}'\}[-\infty/(\mathbf{X} \setminus \mathbf{X}')]$ , where  $\mathbf{X}' := \{\mathbf{x} \mid \mathbf{x} = e \in \mathcal{E}, \llbracket e \rrbracket \rho > -\infty\}$ .  $\square$

*Example 14.* We consider the system  $\mathcal{E} = \{\mathbf{x} = \sqrt{\mathbf{x}}\}$  of mcave equations. For all  $x \in \mathbb{R}$ , let  $\underline{x} := \{\mathbf{x} \mapsto x\}$ . From Example 1, we know that the solution  $\underline{0}$  is not feasible, whereas the solution  $\underline{1}$  is feasible. Thus,  $\underline{x}$  is a feasible pre-solution for all  $x \in (0, 1]$ . Note that  $\underline{1}$  is the only feasible finite solution of  $\mathcal{E}$  and thus, by Lemma 8, the greatest finite pre-solution of  $\mathcal{E}$ .  $\square$

*Example 15.* Let us consider the system  $\mathcal{E} = \{\mathbf{x}_1 = \mathbf{x}_2 + 1 \wedge 0, \mathbf{x}_2 = \sqrt{\mathbf{x}_1}\}$  of mcave equations. From Example 6 it follows that  $\rho := \{\mathbf{x}_1 \mapsto 0, \mathbf{x}_2 \mapsto 0\}$  is a feasible finite fixpoint of  $\llbracket \mathcal{E} \rrbracket$ . Thus,  $\{\mathbf{x}_1 \mapsto 0, \mathbf{x}_2 \mapsto x\}$  is a feasible pre-solution for all  $x \in [-1, 0]$ . The solution  $\{\mathbf{x}_1 \mapsto -\infty, \mathbf{x}_2 \mapsto -\infty\}$  is not feasible, since the right-hand sides evaluate to  $-\infty$ , although they are not  $-\infty$ .  $\square$

The following two lemma imply that our  $\vee$ -strategy improvement algorithm stays in the feasible area, whenever it is started in the feasible area.

**Lemma 15.** Let  $\mathcal{E}$  be a system of morcave equations and  $\rho$  be a feasible pre-solution of  $\mathcal{E}$ . Every pre-solution  $\rho'$  of  $\mathcal{E}$  with  $\rho \leq \rho' \leq \mu_{\geq \rho} \llbracket \mathcal{E} \rrbracket$  is feasible.

*Proof.* The statement is an immediate consequence of the definition.  $\square$

**Lemma 16.** Let  $\mathcal{E}$  be a system of  $\vee$ -morcave equations,  $\sigma$  be a  $\vee$ -strategy for  $\mathcal{E}$ ,  $\rho$  be a feasible solution of  $\mathcal{E}(\sigma)$ , and  $\sigma'$  be an improvement of  $\sigma$  w.r.t.  $\rho$ . Then  $\rho$  is a feasible pre-solution of  $\mathcal{E}(\sigma')$ .

*Proof.* Let  $\rho^* := \mu_{\geq \rho} \llbracket \mathcal{E}(\sigma') \rrbracket$ . We w.l.o.g. assume that  $\underline{-\infty} \triangleleft \rho^* \triangleleft \underline{\infty}$ . Hence,  $\rho \triangleleft \underline{\infty}$ . Let

$$\begin{aligned} \mathbf{X}^{\text{old}} &:= \{\mathbf{x} \in \mathbf{X} \mid \rho(\mathbf{x}) > -\infty\}, \text{ and} \\ \mathcal{E}^{\text{old}} &:= \{\mathbf{x} = e \in \mathcal{E}(\sigma) \mid \mathbf{x} \in \mathbf{X}^{\text{old}}\}[-\infty/(\mathbf{X} \setminus \mathbf{X}^{\text{old}})]. \end{aligned}$$

Hence,  $\rho|_{\mathbf{X}^{\text{old}}}$  is a feasible finite solution of  $\mathcal{E}^{\text{old}}$ , i.e., a feasible finite fixpoint of  $\llbracket \mathcal{E}^{\text{old}} \rrbracket$ . Therefore, there exist  $\mathbf{X}_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}_k = \mathbf{X}^{\text{old}}$  with

$$\mathbf{X}_1 \xrightarrow{\llbracket \mathcal{E}^{\text{old}} \rrbracket, \rho|_{\mathbf{X}^{\text{old}}}} \dots \xrightarrow{\llbracket \mathcal{E}^{\text{old}} \rrbracket, \rho|_{\mathbf{X}^{\text{old}}}} \mathbf{X}_k \quad (17)$$

such that, for each  $j \in \{1, \dots, k\}$ , there exists some pre-fixpoint  $\rho'$  of  $\llbracket \mathcal{E}^{\text{old}} \rrbracket \leftarrow \rho|_{\mathbf{X}^{\text{old}} \setminus \mathbf{X}_j}$  with  $\rho' \triangleleft \rho|_{\mathbf{X}_j}$  such that  $\mu_{\geq \rho'}(\llbracket \mathcal{E}^{\text{old}} \rrbracket \leftarrow \rho|_{\mathbf{X}^{\text{old}} \setminus \mathbf{X}_j}) = \rho|_{\mathbf{X}_j}$ .

Let  $\mathbf{X}^{\text{imp}} := \{\mathbf{x} \in \mathbf{X} \mid \rho^*(\mathbf{x}) > \rho(\mathbf{x})\}$ ,  $\mathbf{X}'_j := \mathbf{X}_j \setminus \mathbf{X}^{\text{imp}}$  for all  $j \in \{1, \dots, k\}$ , and  $\mathbf{X}'_{k+1} := \mathbf{X}^{\text{imp}}$ . Obviously, we have  $\mathbf{X}'_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}'_{k+1} = \mathbf{X}$ . It remains to show that the following properties are fulfilled:

1.  $\mathbf{X}'_1 \xrightarrow{[\mathcal{E}(\sigma')], \rho^*} \dots \xrightarrow{[\mathcal{E}(\sigma')], \rho^*} \mathbf{X}'_{k+1}$
2. For each  $j \in \{1, \dots, k+1\}$ , there exists some pre-fixpoint  $\rho'$  with  $\rho' \triangleleft \rho^*|_{\mathbf{X}'_j}$  such that  $\mu_{\geq \rho'}([\mathcal{E}(\sigma')]) \leftarrow \rho^*|_{\mathbf{X} \setminus \mathbf{X}'_j} = \rho^*|_{\mathbf{X}'_j}$ .

In order to prove statement 1, let  $j \in \{1, \dots, k\}$ . We have to show that

$$\mathbf{X}'_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}'_j \xrightarrow{[\mathcal{E}(\sigma')], \rho^*} \mathbf{X}'_{j+1} \dot{\cup} \dots \dot{\cup} \mathbf{X}'_{k+1}.$$

Since  $\mathbf{X}_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}_j \xrightarrow{[\mathcal{E}^{\text{old}}], \rho|_{\mathbf{X}^{\text{old}}}} \mathbf{X}_{j+1} \dot{\cup} \dots \dot{\cup} \mathbf{X}_k$ , there exists some variable assignment  $\rho' : \mathbf{X}_{j+1} \dot{\cup} \dots \dot{\cup} \mathbf{X}_k \rightarrow \mathbb{R}$  with  $\rho' \triangleleft \rho|_{\mathbf{X}_{j+1} \dot{\cup} \dots \dot{\cup} \mathbf{X}_k}$  such that

$$([\mathcal{E}^{\text{old}}](\rho|_{\mathbf{X}^{\text{old}} \oplus \rho'}))|_{\mathbf{X}_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}_j} = ([\mathcal{E}^{\text{old}}](\rho|_{\mathbf{X}^{\text{old}}}))|_{\mathbf{X}_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}_j}. \quad (18)$$

We define  $\rho'' : \mathbf{X}'_{j+1} \dot{\cup} \dots \dot{\cup} \mathbf{X}'_{k+1} \rightarrow \mathbb{R}$  by

$$\rho''(\mathbf{x}) = \begin{cases} \rho'(\mathbf{x}) & \text{if } \mathbf{x} \in \mathbf{X}'_{j+1} \dot{\cup} \dots \dot{\cup} \mathbf{X}'_k \\ \rho(\mathbf{x}) & \text{if } \mathbf{x} \in \mathbf{X}'_{k+1} \text{ and } \mathbf{x} \in \mathbf{X}^{\text{old}} \\ \rho^*(\mathbf{x}) - 1 & \text{if } \mathbf{x} \in \mathbf{X}'_{k+1} \text{ and } \mathbf{x} \notin \mathbf{X}^{\text{old}} \end{cases} \quad \text{for all } \mathbf{x} \in \mathbf{X}'_{j+1} \dot{\cup} \dots \dot{\cup} \mathbf{X}'_{k+1}.$$

By construction, we have  $\rho'' \triangleleft \rho^*|_{\mathbf{X}'_{j+1} \dot{\cup} \dots \dot{\cup} \mathbf{X}'_{k+1}}$ . Hence, we get

$$\begin{aligned} ([\mathcal{E}(\sigma')](\rho^*))|_{\mathbf{X}'_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}'_j} &\geq ([\mathcal{E}(\sigma')](\rho^* \oplus \rho''))|_{\mathbf{X}'_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}'_j} & (\rho^* \geq \rho^* \oplus \rho'') \\ &\geq ([\mathcal{E}^{\text{old}}](\rho|_{\mathbf{X}^{\text{old}} \oplus \rho'}))|_{\mathbf{X}'_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}'_j} \\ &= ([\mathcal{E}^{\text{old}}](\rho|_{\mathbf{X}^{\text{old}}}))|_{\mathbf{X}'_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}'_j} & (\text{because of (18)}) \\ &= ([\mathcal{E}^{\text{old}}](\rho^*|_{\mathbf{X}^{\text{old}}}))|_{\mathbf{X}'_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}'_j} & (\text{because of Lemma 7}) \\ &= ([\mathcal{E}(\sigma')](\rho^*))|_{\mathbf{X}'_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}'_j}. \end{aligned}$$

Thus,  $([\mathcal{E}(\sigma')](\rho^* \oplus \rho''))|_{\mathbf{X}'_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}'_j} = ([\mathcal{E}(\sigma')](\rho^*))|_{\mathbf{X}'_1 \dot{\cup} \dots \dot{\cup} \mathbf{X}'_j}$ . This proves statement 1.

In order to prove statement 2, let  $j \in \{1, \dots, k+1\}$ . We distinguish 2 cases. Firstly, assume that  $j \leq k$ . Since  $\rho|_{\mathbf{X}^{\text{old}}}$  is a feasible finite fixpoint of  $[\mathcal{E}^{\text{old}}]$ , there exists some pre-fixpoint  $\rho'$  with  $\rho' \triangleleft \rho|_{\mathbf{X}_j} = \rho^*|_{\mathbf{X}_j}$  such that  $\mu_{\geq \rho'}([\mathcal{E}^{\text{old}}] \leftarrow \rho|_{\mathbf{X}^{\text{old}} \setminus \mathbf{X}_j}) = \rho|_{\mathbf{X}_j} = \rho^*|_{\mathbf{X}_j}$ . Using monotonicity, we get  $\mu_{\geq \rho'}([\mathcal{E}^{\text{old}}] \leftarrow \rho^*|_{\mathbf{X}^{\text{old}} \setminus \mathbf{X}_j}) = \rho|_{\mathbf{X}_j} = \rho^*|_{\mathbf{X}_j}$ . Hence,  $\rho'|_{\mathbf{X}'_j} : \mathbf{X}'_j \rightarrow \mathbb{R}$ ,  $\rho'|_{\mathbf{X}'_j} \triangleleft \rho|_{\mathbf{X}'_j} = \rho^*|_{\mathbf{X}'_j}$ , and  $\mu_{\geq \rho'}([\mathcal{E}^{\text{old}}] \leftarrow \rho^*|_{\mathbf{X}^{\text{old}} \setminus \mathbf{X}'_j}) = \mu_{\geq \rho'}([\mathcal{E}(\sigma')]) \leftarrow \rho^*|_{\mathbf{X} \setminus \mathbf{X}'_j} = \rho^*|_{\mathbf{X}'_j}$ . This proves statement 2 for  $j \leq k$ . Now, assume that  $j = k+1$ . By definition of  $\mathbf{X}'_{k+1}$ ,  $\rho|_{\mathbf{X}'_{k+1}} \triangleleft \rho^*|_{\mathbf{X}'_{k+1}}$ . Moreover, we get immediately that  $\rho|_{\mathbf{X}'_{k+1}}$  is a pre-fixpoint of  $[\mathcal{E}(\sigma')] \leftarrow \rho^*|_{\mathbf{X} \setminus \mathbf{X}'_{k+1}}$  and  $\mu_{\geq \rho|_{\mathbf{X}'_{k+1}}}([\mathcal{E}(\sigma')] \leftarrow \rho^*|_{\mathbf{X} \setminus \mathbf{X}'_{k+1}}) = \rho^*|_{\mathbf{X}'_{k+1}}$ . This proves statement 2.  $\square$

*Example 16.* We continue Example 12. Obviously,  $\rho = \{\mathbf{x}_1 \mapsto 0, \mathbf{x}_2 \mapsto -1\}$  is a feasible solution of  $\mathcal{E}(\sigma_1) = \{\mathbf{x}_1 = \mathbf{x}_2 + 1 \wedge 0, \mathbf{x}_2 = -1\}$ . The  $\vee$ -strategy  $\sigma_2$  is

an improvement of the  $\vee$ -strategy  $\sigma_1$  w.r.t.  $\rho$ . By lemma 16,  $\rho$  is also a feasible pre-solution of  $\mathcal{E}(\sigma_2) = \{\mathbf{x}_1 = \mathbf{x}_2 + 1 \wedge 0, \mathbf{x}_2 = \sqrt{\mathbf{x}_1}\}$ . The fact that  $\rho$  is a feasible pre-solution of  $\mathcal{E}(\sigma_2)$  is also shown in Example 15.  $\square$

The above two lemmas ensure that our  $\vee$ -strategy improvement algorithm stays in the feasible area, whenever it is started in the feasible area. In order to start in the feasible area, we in the following simply assume w.l.o.g. that each equation of  $\mathcal{E}$  is of the form  $\mathbf{x} = -\infty \vee e$ . We say that such a system of fixpoint equations is in *standard form*. Then, we start our  $\vee$ -strategy improvement algorithm with a  $\vee$ -strategy  $\sigma_{\text{init}}$  such that  $\mathcal{E}(\sigma_{\text{init}}) = \{\mathbf{x} = -\infty \mid \mathbf{x} \in \mathbf{X}\}$ . In consequence,  $\underline{-\infty}$  is a feasible solution of  $\mathcal{E}(\sigma_{\text{init}})$ . We get:

**Lemma 17.** *Let  $\mathcal{E}$  be a system of  $\vee$ -morcave equations. For all  $i \in \mathbb{N}$ , let  $\rho_i$  be the value of the program variable  $\rho$  and  $\sigma_i$  be the value of the program variable  $\sigma$  in the  $\vee$ -strategy improvement algorithm (Algorithm 1) after the  $i$ -th evaluation of the loop-body. Then,  $\rho_i$  is a feasible pre-solution of  $\mathcal{E}(\sigma_{i+1})$  for all  $i \in \mathbb{N}$ .  $\square$*

*Example 17.* We again consider the system  $\mathcal{E} = \{\mathbf{x}_1 = -\infty \vee \mathbf{x}_2 + 1 \wedge 0, \mathbf{x}_2 = -\infty \vee -1 \vee \sqrt{\mathbf{x}_1}\}$  of  $\vee$ -morcave equations introduced in Example 12. A run of our  $\vee$ -strategy improvement algorithm gives us

$$\begin{array}{ll} \mathcal{E}(\sigma_0) = \{\mathbf{x}_1 = -\infty, \mathbf{x}_2 = -\infty\} & \rho_0 = \{\mathbf{x}_1 \mapsto -\infty, \mathbf{x}_2 \mapsto -\infty\} \\ \mathcal{E}(\sigma_1) = \{\mathbf{x}_1 = -\infty, \mathbf{x}_2 = -1\} & \rho_1 = \{\mathbf{x}_1 \mapsto -\infty, \mathbf{x}_2 \mapsto -1\} \\ \mathcal{E}(\sigma_2) = \{\mathbf{x}_1 = \mathbf{x}_2 + 1 \wedge 0, \mathbf{x}_2 = -1\} & \rho_2 = \{\mathbf{x}_1 \mapsto 0, \mathbf{x}_2 \mapsto -1\} \\ \mathcal{E}(\sigma_3) = \{\mathbf{x}_1 = \mathbf{x}_2 + 1 \wedge 0, \mathbf{x}_2 = \sqrt{\mathbf{x}_1}\} & \rho_3 = \{\mathbf{x}_1 \mapsto 0, \mathbf{x}_2 \mapsto 0\} \end{array}$$

By Lemma 17,  $\rho_i$  is a feasible pre-solution of  $\mathcal{E}(\sigma_{i+1})$  for all  $i = \{0, 1, 2\}$ .  $\square$

#### 4.4 Evaluating $\vee$ -Strategies / Solving Systems of Morcave Equations

It remains to develop a method for computing  $\mu_{\geq \rho}[\mathcal{E}]$  under the assumption that  $\rho$  is a feasible pre-solution of the system  $\mathcal{E}$  of morcave equations. This is an important step in our  $\vee$ -strategy improvement algorithm (Algorithm 1). Before doing this, we introduce the following notation for the sake of simplicity:

**Definition 8.** *Let  $\mathcal{E}$  be a system of morcave equations and  $\rho$  a pre-solution of  $\mathcal{E}$ . Let*

$$\mathbf{X}_\rho^{-\infty} := \{\mathbf{x} \mid \mathbf{x} = e \in \mathcal{E}, \llbracket e \rrbracket \rho = -\infty\} \quad (19)$$

$$\mathbf{X}_\rho^{\infty} := \{\mathbf{x} \mid \mathbf{x} = e \in \mathcal{E}, \llbracket e \rrbracket \rho = \infty\} \quad (20)$$

$$\mathbf{X}'_\rho := \mathbf{X} \setminus (\mathbf{X}_\rho^{-\infty} \cup \mathbf{X}_\rho^{\infty}) = \{\mathbf{x} \mid \mathbf{x} = e \in \mathcal{E}, \llbracket e \rrbracket \rho \in \mathbb{R}\} \quad (21)$$

$$\mathcal{E}'_\rho = \{\mathbf{x} = e \in \mathcal{E} \mid \mathbf{x} \in \mathbf{X}'_\rho\}[-\infty/\mathbf{X}_\rho^{-\infty}, \infty/\mathbf{X}_\rho^{\infty}] \quad (22)$$

The pre-solution  $\text{suppresol}_\rho[\mathcal{E}]$  of  $\mathcal{E}$  is defined by

$$\text{suppresol}_\rho[\mathcal{E}](\mathbf{x}) := \begin{cases} -\infty & \text{if } \mathbf{x} \in \mathbf{X}_\rho^{-\infty} \\ \sup \{ \hat{\rho}(\mathbf{x}) \mid \hat{\rho} : \mathbf{X}'_\rho \rightarrow \mathbb{R}, \hat{\rho} \leq \llbracket \mathcal{E}' \rrbracket \hat{\rho} \} & \text{if } \mathbf{x} \in \mathbf{X}'_\rho \\ \infty & \text{if } \mathbf{x} \in \mathbf{X}_\rho^\infty \end{cases} \quad (23)$$

for all  $\mathbf{x} \in \mathbf{X}$ .  $\square$

*Remark 1.* The variables assignment  $\text{suppresol}_\rho[\mathcal{E}]$  is by construction a pre-solution of  $\mathcal{E}$ , but, as we will see in Example 18, not necessarily a solution of  $\mathcal{E}$ .  $\square$

Under some constraints, we can compute  $\text{suppresol}_\rho[\mathcal{E}]$  by solving  $|\mathbf{X}|$  convex optimisation problems of linear size. This can be done by general convex optimization methods. For further information on convex optimization, we refer, for instance, to Nemirovski [13].

**Lemma 18.** *Let  $\mathcal{E}$  be a system of mcave equations and  $\rho$  a pre-solution of  $\mathcal{E}$ . Then, the pre-solution  $\text{suppresol}_\rho[\mathcal{E}]$  of  $\mathcal{E}$  can be computed by solving at most  $|\mathbf{X}|$  convex optimization problems.*

*Proof.* Let  $\mathbf{X}_\rho^{-\infty}$ ,  $\mathbf{X}_\rho^\infty$ ,  $\mathbf{X}'_\rho$ , and  $\mathcal{E}'_\rho$  be defined as in Definition 8. We have to compute  $\text{suppresol}_\rho[\mathcal{E}](\mathbf{x}) = \sup \{ \hat{\rho}(\mathbf{x}) \mid \hat{\rho} : \mathbf{X}'_\rho \rightarrow \mathbb{R}, \hat{\rho} \leq \llbracket \mathcal{E}' \rrbracket \hat{\rho} \} = \sup \{ \hat{\rho}(\mathbf{x}) \mid \hat{\rho} : \mathbf{X}'_\rho \rightarrow \mathbb{R}, (\text{id} - \llbracket \mathcal{E}' \rrbracket) \hat{\rho} \leq 0 \}$  for all  $\mathbf{x} \in \mathbf{X}'_\rho$ . Here,  $\text{id}$  denotes the identity function. Therefore, since  $\text{id}$  is affine,  $\llbracket \mathcal{E}' \rrbracket$  is concave (considered as a function that maps values from  $\mathbf{X}'_\rho \rightarrow \mathbb{R}$  to values from  $\mathbf{X}'_\rho \rightarrow (\mathbb{R} \cup \{-\infty\})$ ), and thus  $-\llbracket \mathcal{E}' \rrbracket \hat{\rho}$  is convex (considered as a function that maps values from  $\mathbf{X}'_\rho \rightarrow \mathbb{R}$  to values from  $\mathbf{X}'_\rho \rightarrow (\mathbb{R} \cup \{\infty\})$ ), the mathematical optimization problem  $\sup \{ \hat{\rho}(\mathbf{x}) \mid \hat{\rho} : \mathbf{X}'_\rho \rightarrow \mathbb{R}, (\text{id} - \llbracket \mathcal{E}' \rrbracket) \hat{\rho} \leq 0 \}$  is a convex optimization problem.  $\square$

We will use  $\text{suppresol}_\rho[\mathcal{E}]$  iteratively to compute  $\mu_{\geq \rho}[\mathcal{E}]$  under the assumption that  $\rho$  is a feasible pre-solution of the system  $\mathcal{E}$  of morcave equations. As a first step in this direction, we prove the following lemma, which gives us at least a method for computing  $\mu_{\geq \rho}[\mathcal{E}]$  under the assumption that  $\mathcal{E}$  is a system of *cmorcave* equations.

**Lemma 19.** *Let  $\mathcal{E}$  be a system of morcave equations and  $\rho$  a feasible pre-solution of  $\mathcal{E}$ . Let  $\mathbf{X}_\rho^{-\infty}$ ,  $\mathbf{X}_\rho^\infty$ , and  $\mathbf{X}'_\rho$  be defined as in Definition 8 ((19) - (21)). Then:*

$$\mu_{\geq \rho}[\mathcal{E}](\mathbf{x}) = \text{suppresol}_\rho[\mathcal{E}](\mathbf{x}) = -\infty \quad \text{for all } \mathbf{x} \in \mathbf{X}_\rho^{-\infty} \quad (24)$$

$$\mu_{\geq \rho}[\mathcal{E}](\mathbf{x}) \geq \text{suppresol}_\rho[\mathcal{E}](\mathbf{x}) \quad \text{for all } \mathbf{x} \in \mathbf{X}'_\rho \quad (25)$$

$$\mu_{\geq \rho}[\mathcal{E}](\mathbf{x}) = \text{suppresol}_\rho[\mathcal{E}](\mathbf{x}) = \infty \quad \text{for all } \mathbf{x} \in \mathbf{X}_\rho^\infty \quad (26)$$

If  $\mathcal{E}$  is a system of *cmorcave* equations, then the inequality in (25) is in fact an equality, i.e., we have

$$\mu_{\geq \rho}[\mathcal{E}] = \text{suppresol}_\rho[\mathcal{E}]. \quad (27)$$



*Proof.* Let  $\mathcal{E}'_\rho$  be defined as in Definition 8 (22). We first prove (24) - (26). Let  $x \in \mathbf{X}$ . If  $\mathbf{x} \in \mathbf{X}_\rho^{-\infty} \cup \mathbf{X}_\rho^\infty$ , then the statement is obviously fulfilled, because  $\rho$  is feasible and thus  $e = -\infty$  for all equations  $\mathbf{x} = e$  from  $\mathcal{E}$  with  $\llbracket e \rrbracket \rho = -\infty$ . This gives us (24) and (26). Assume now that  $\mathbf{x} \in \mathbf{X}'_\rho$ . Let  $\rho' := \rho|_{\mathbf{X}'_\rho}$  and  $\rho^* := \mu_{\geq \rho'}[\mathcal{E}'_\rho]$ . We have to show that

$$\rho^*(\mathbf{x}) \geq \sup \{ \hat{\rho}(\mathbf{x}) \mid \hat{\rho} : \mathbf{X}'_\rho \rightarrow \mathbb{R}, \hat{\rho} \leq \llbracket \mathcal{E}'_\rho \rrbracket \hat{\rho} \}. \quad (28)$$

If  $\rho^*(\mathbf{x}) = \infty$ , there is nothing to prove. Therefore, assume that  $\rho^*(\mathbf{x}) < \infty$ . Then  $\rho^*(\mathbf{x}) \in \mathbb{R}$ . Let  $\mathbf{X}''_\rho := \{ \mathbf{x}'' \in \mathbf{X}'_\rho \mid \rho^*(\mathbf{x}'') < \infty \}$ . Then,  $\mathbf{X}''_\rho = \{ \mathbf{x}'' \in \mathbf{X}'_\rho \mid \rho^*(\mathbf{x}'') \in \mathbb{R} \}$ . Let  $\mathcal{E}''_\rho := \{ \mathbf{x}'' = e \in \mathcal{E}'_\rho \mid \mathbf{x}'' \in \mathbf{X}''_\rho \} [\infty / (\mathbf{X}'_\rho \setminus \mathbf{X}''_\rho)]$ , and  $\rho'' := \rho|_{\mathbf{X}''_\rho}$ . The pre-solution  $\rho''$  of  $\mathcal{E}''_\rho$  is feasible. Hence,  $\rho^*|_{\mathbf{X}''_\rho}$  is a feasible finite pre-solution of  $\mathcal{E}''_\rho$ , i.e., a feasible finite fixpoint of  $\llbracket \mathcal{E}''_\rho \rrbracket$ . Therefore, we finally get (25) using Lemma 8.

Before we actually prove (27), we start with an easy observation. The sequence  $(\llbracket \mathcal{E}'_\rho \rrbracket^k \rho')_{k \in \mathbb{N}}$  is increasing, because  $\rho'$  is a pre-solution of  $\mathcal{E}'_\rho$ . Further  $\llbracket \mathcal{E}'_\rho \rrbracket^k \rho' : \mathbf{X}'_\rho \rightarrow \mathbb{R}$  and  $\llbracket \mathcal{E}'_\rho \rrbracket^k \rho' \leq \llbracket \mathcal{E}'_\rho \rrbracket (\llbracket \mathcal{E}'_\rho \rrbracket^k \rho')$  for all  $k \in \mathbb{N}$ . Hence, we get

$$\sup \{ \hat{\rho}(\mathbf{x}) \mid \hat{\rho} : \mathbf{X}'_\rho \rightarrow \mathbb{R}, \hat{\rho} \leq \llbracket \mathcal{E}'_\rho \rrbracket \hat{\rho} \} \geq \sup \{ (\llbracket \mathcal{E}'_\rho \rrbracket^k \rho')(\mathbf{x}) \mid k \in \mathbb{N} \} \quad (29)$$

Now, assume that  $\mathcal{E}$  is a system of cmorcave equations. In order to prove (27), it remains to show that  $\rho^*(\mathbf{x}) \leq \sup \{ \hat{\rho}(\mathbf{x}) \mid \hat{\rho} : \mathbf{X}'_\rho \rightarrow \mathbb{R}, \hat{\rho} \leq \llbracket \mathcal{E}'_\rho \rrbracket \hat{\rho} \}$ . Since  $\llbracket \mathcal{E}'_\rho \rrbracket$  is monotone and upward-chain-continuous on  $\{ \hat{\rho} : \mathbf{X}'_\rho \rightarrow \mathbb{R} \mid \hat{\rho} \geq \rho' \}$ , we have  $\rho^* = \bigvee \{ \llbracket \mathcal{E}'_\rho \rrbracket^k \rho' \mid k \in \mathbb{N} \}$ . Using (29), this gives us  $\rho^*(\mathbf{x}) \leq \sup \{ \hat{\rho}(\mathbf{x}) \mid \hat{\rho} : \mathbf{X}'_\rho \rightarrow \mathbb{R}, \hat{\rho} \leq \llbracket \mathcal{E}'_\rho \rrbracket \hat{\rho} \}$ , as desired.  $\square$

If the equations are morcave but *not* cmorcave, then the inequality in (25) can indeed be strict as the following example shows.

*Example 18.* Let us consider the following system  $\mathcal{E}$  of morcave equations:

$$\mathbf{x}_1 = 1 \quad \mathbf{x}_2 = \mathbf{x}_1 + \mathbf{x}_2 \quad \mathbf{x}_3 = \begin{cases} 0 & \text{if } \mathbf{x}_2 < \infty \\ 1 & \text{if } \mathbf{x}_2 = \infty \end{cases} \quad (30)$$

Observe that the third equation is *not* cmorcave, since, for the ascending chain  $C = \{ \{ \mathbf{x}_2 \mapsto k \} \mid k \in \mathbb{N} \}$ , we have  $\bigvee \{ \llbracket e \rrbracket \rho \mid \rho \in C \} = 0 < 1 = \llbracket e \rrbracket (\bigvee C)$ , where  $e$  denotes the right-hand side of the third equation. The variable assignment

$$\rho := \{ \mathbf{x}_1 \mapsto 0, \mathbf{x}_2 \mapsto 0, \mathbf{x}_3 \mapsto 0 \} \quad (31)$$

is a feasible pre-solution, since

$$\rho^* := \mu_{\geq \rho}[\mathcal{E}] = \{ \mathbf{x}_1 \mapsto 1, \mathbf{x}_2 \mapsto \infty, \mathbf{x}_3 \mapsto 1 \} \quad (32)$$

is a feasible solution of  $\mathcal{E}$ . Now, let the variable assignment  $\rho_1$  be defined by

$$\rho_1 := \text{suppresol}_\rho[\mathcal{E}]. \quad (33)$$

Lemma 19 gives us  $\rho_1 \leq \rho^*$ , but not  $\rho_1 = \rho^*$ . Indeed, we have

$$\rho_1 = \{\mathbf{x}_1 \mapsto 1, \mathbf{x}_2 \mapsto \infty, \mathbf{x}_3 \mapsto 0\} < \rho^*. \quad (34)$$

We emphasize that  $\rho_1(\mathbf{x}_3) = 0$ , because  $\llbracket e \rrbracket \hat{\rho} = 0$  for all  $\hat{\rho} : \mathbf{X} \rightarrow \mathbb{R}$ , where  $e$  denotes the right-hand side of the third equation of 30.

How we can actually compute  $\rho^*$ , remains an open question. The discontinuity at  $\mathbf{x}_2 = \infty$  is the reason for the strict inequality in (34). However, since upward discontinuities can only be present at  $\infty$ , there are at most  $n$  upward discontinuities, where  $n$  is the number of variables of the equation system. Hence, we could think of using (33) to get over at least one discontinuity.

Let us perform a second iteration for the example. We know that  $\rho_1 \leq \rho^*$ . Moreover, by definition,  $\rho_1$  is also a feasible pre-solution of  $\mathcal{E}$ . For the variable assignment  $\rho_2$  that is defined by  $\rho_2 := \text{suppresol}_{\rho_1}[\mathcal{E}]$  we obviously have  $\rho^* = \rho_2$ . We will see that this method can always be applied. More precisely, we can always compute  $\rho^*$  after performing at most  $n$  such iterations.  $\square$

In order to deal not only with systems of cmorcake equations, but also with systems of morcake equations, we use Lemma 19 iteratively until we reach a solution. That is, we generalize the statement of Lemma 19 as follows:

**Lemma 20.** *Let  $\mathcal{E}$  be a system of morcake equations and  $\rho$  a feasible pre-solution of  $\mathcal{E}$ . For all  $i \in \mathbb{N}$ , let  $\text{suppresol}_{\rho}^i[\mathcal{E}]$  be defined by*

$$\text{suppresol}_{\rho}^0[\mathcal{E}] := \rho \quad (35)$$

$$\text{suppresol}_{\rho}^{i+1}[\mathcal{E}] := \text{suppresol}_{\text{suppresol}_{\rho}^i[\mathcal{E}]}[\mathcal{E}] \quad \text{for all } i \in \mathbb{N}. \quad (36)$$

Then, the following statements hold:

1.  $(\text{suppresol}_{\rho}^i[\mathcal{E}])_{i \in \mathbb{N}}$  is an increasing sequence of feasible pre-solutions of  $\mathcal{E}$ .
2.  $\text{suppresol}_{\rho}^i[\mathcal{E}] \leq \mu_{\geq \rho}[\mathcal{E}]$  for all  $i \in \mathbb{N}$ .
3.  $\text{suppresol}_{\rho}^{|\mathbf{X}|}[\mathcal{E}] = \mu_{\geq \rho}[\mathcal{E}]$ .
4.  $\text{suppresol}_{\rho}[\mathcal{E}] = \mu_{\geq \rho}[\mathcal{E}]$ , whenever  $\mathcal{E}$  is a system of cmorcake equations.

*Proof.* The first two statements can be proven by induction on  $i$  using Lemma 19. The third statement follows from the fact that, for any feasible pre-solution  $\rho$  of a system  $\mathcal{E}$  of morcake equations,  $\text{suppresol}_{\rho}[\mathcal{E}] < \mu_{\geq \rho}[\mathcal{E}]$  implies that there exists some variable  $\mathbf{x} \in \mathbf{X}$  such that  $\rho(\mathbf{x}) < \infty$  and  $\text{suppresol}_{\rho}[\mathcal{E}](\mathbf{x}) = \infty$ . The fourth statement is the second statement of Lemma 19.  $\square$

*Example 19.* For the situation in Example 18, we have  $\mu_{\geq \rho}[\mathcal{E}] = \text{suppresol}_{\rho}^3[\mathcal{E}] = \text{suppresol}_{\rho}^2[\mathcal{E}] > \text{suppresol}_{\rho}^1[\mathcal{E}] > \rho$ .  $\square$

Because of the definition of  $\text{suppresol}_{\rho}$  (see Definition 8), Lemma 20 implies the following corollary:

**Corollary 1.** *Let  $\mathcal{E}$  be a system of morcake equations and  $\rho$  a feasible pre-solution of  $\mathcal{E}$ . Then, the value  $\mu_{\geq \rho}[\mathcal{E}]$  only depends on  $\mathcal{E}$  and  $\mathbf{X}_{\rho}^{\infty} := \{\mathbf{x} \mid \mathbf{x} = e \in \mathcal{E}, \llbracket e \rrbracket \rho = \infty\}$ .*  $\square$

## 4.5 Termination

It remains to show that our  $\vee$ -strategy improvement algorithm (Algorithm 1) terminates. That is, we have to come up with an upper bound on the number of iterations of the loop. In each iteration, we have to compute  $\mu_{\geq \rho}[\mathcal{E}(\sigma)]$ , where  $\rho$  is a feasible pre-solution of  $\mathcal{E}(\sigma)$ . This has to be done until we have found a solution. By Corollary 1,  $\mu_{\geq \rho}[\mathcal{E}(\sigma)]$  only depends on the  $\vee$ -strategy  $\sigma$  and the set  $\mathbf{X}_\rho^\infty := \{\mathbf{x} \mid \mathbf{x} = e \in \mathcal{E}(\sigma), \llbracket e \rrbracket \rho = \infty\}$ . During the run of our  $\vee$ -strategy improvement algorithm, the set  $\mathbf{X}_\rho^\infty$  monotonically increases. This implies that we have to consider each  $\vee$ -strategy  $\sigma$  at most  $|\mathbf{X}|$  times. That is, the number of iterations of the loop is bounded from above by  $|\mathbf{X}| \cdot |\Sigma|$ . Summarizing, we have shown our main theorem:

**Theorem 1.** *Let  $\mathcal{E}$  be a system of  $\vee$ -morcave equations in standard form. Our  $\vee$ -strategy improvement algorithm computes  $\mu[\mathcal{E}]$  and performs at most  $|\mathbf{X}| \cdot |\Sigma|$   $\vee$ -strategy improvement steps.  $\square$*

In our experiments, we did not observe the exponential worst-case behavior. All examples we know of require linearly many  $\vee$ -strategy improvement steps. We are also not aware of a class of examples, where we would be able to observe the exponential worst-case behavior. Therefore, our conjecture is that for practical examples our algorithm terminates after linearly many iterations.

## 5 Parametrized Optimization Problems as Right-hand sides

In the static program analysis application that we discuss in Section 6, the right-hand sides of the fixpoint equation systems we have to solve are maxima of finitely many *parametrized optimization problems*. In this special situation, we can evaluate  $\vee$ -strategies more efficiently than by solving general convex optimization problems as described in Section 4 (see Lemma 18, 19, and 20). We provide a in-depth study of this special situation in this section.

### 5.1 Parametrized Optimization Problems

We now consider the case that a system  $\mathcal{E}$  of fixpoint equations is given, where the right-hand sides are *parametrized optimization problems*. In this article, we call an operator  $g : \overline{\mathbb{R}}^n \rightarrow \overline{\mathbb{R}}$  a *parametrized optimization problem* if and only if

$$g(x) = \sup \{f(y) \mid y \in Y(\mathbf{x}_1, \dots, \mathbf{x}_n)\} \quad \text{for all } x \in \overline{\mathbb{R}}^n, \quad (37)$$

where  $f : \mathbb{R}^k \rightarrow \mathbb{R}$  is an objective function, and  $Y : \overline{\mathbb{R}}^n \rightarrow 2^{\mathbb{R}^k}$  is a mapping that assigns a set  $Y(x) \subseteq \mathbb{R}^k$  of states to any vector of bounds  $x \in \overline{\mathbb{R}}^n$ . The parametrized optimization problem  $g$  is monotone on  $\overline{\mathbb{R}}^n$ , whenever  $Y$  is monotone on  $\overline{\mathbb{R}}^n$ . It is monotone on  $\overline{\mathbb{R}}^n$  and upward chain continuous on

$g^{-1}(\overline{\mathbb{R}} \setminus \{-\infty\})$ <sup>5</sup> whenever  $f$  is continuous on  $\mathbb{R}^k$  and  $Y$  is monotone on  $\overline{\mathbb{R}}^n$  and upward chain continuous on  $Y^{-1}(2^{\mathbb{R}^k} \setminus \{\emptyset\})$ <sup>6</sup>. In the following, we are concerned with the latter situation. A parametrized optimization problem  $g$  that is monotone on  $\overline{\mathbb{R}}^n$  and upward chain continuous on  $g^{-1}(\overline{\mathbb{R}} \setminus \{-\infty\})$  is called *upward chain continuous parametrized optimization problem*.

*Example 20.* Assume that  $Y$  and  $f$  are given by

$$Y(x) := \{y \in \mathbb{R}^k \mid Ay \leq x\} \quad \text{for all } x \in \overline{\mathbb{R}}^n, \text{ and} \quad (38)$$

$$f(y) := b + c^\top y \quad \text{for all } y \in \mathbb{R}^k, \quad (39)$$

where  $A \in \mathbb{R}^{n \times k}$ ,  $b \in \mathbb{R}$ , and  $c \in \mathbb{R}^k$ . Then,  $g$  is defined through Equation (37) is an upward chain continuous parametrized optimization problems. To be more precise, it is a *parametrized linear programming problem* (to be defined). Although this is also an interesting case (cf. Gawlitza and Seidl [8]), in the following, we mainly focus on the more general case where the right-hand sides are *parametrized semi-definite programming problems* (to be defined). In this example, the right-hand side is not only upward chain continuous, it is even concave. To be more precise, on the set of points where it returns a value greater than  $-\infty$  it is a point-wise minimum of finitely many monotone and affine operators.  $\square$

## 5.2 Fixpoint Equations with Parametrized Optimization Problems

Assume now that we have a system of fixpoint equations, where the right-hand sides are point-wise maxima of finitely many upward chain continuous parametrized optimization problems. If we use our  $\vee$ -strategy improvement algorithm to compute the least solution, then, for each  $\vee$ -strategy improvement step, we have to compute  $\mu_{\geq \rho_0} \llbracket \mathcal{E} \rrbracket$  for a system  $\mathcal{E}$  of fixpoint equations whose right-hand sides are upward chain continuous parametrized optimization problems, and  $\rho_0$  is a pre-solution of  $\mathcal{E}$ . We study this case in the following:

Assume that  $\mathcal{E}$  is a system of fixpoint equations, where the right-hand sides are *upward chain continuous parametrized optimization problems*. For simplicity and without loss of generality, we additionally assume that a variable assignment  $\rho_0 : \mathbf{X} \rightarrow \mathbb{R}$  is given such that

$$-\infty \triangleleft \rho_0 \leq \llbracket \mathcal{E} \rrbracket \rho_0 \triangleleft \infty. \quad (40)$$

We are interested in computing the pre-solution  $\text{suppresol}_{\rho_0} \llbracket \mathcal{E} \rrbracket$  of  $\mathcal{E}$ . In the case at hand, this means that we need to compute  $\rho^* : \mathbf{X} \rightarrow \overline{\mathbb{R}}$  that is defined by

$$\rho^*(\mathbf{x}) := \sup \{\rho(\mathbf{x}) \mid \rho : \mathbf{X} \rightarrow \mathbb{R} \text{ and } \rho \leq \llbracket \mathcal{E} \rrbracket \rho\} \quad \text{for all } \mathbf{x} \in \mathbf{X}. \quad (41)$$

<sup>5</sup> A monotone function  $g : \overline{\mathbb{R}}^n \rightarrow \overline{\mathbb{R}}$  is upward chain continuous on an upward closed set  $X \subseteq \overline{\mathbb{R}}^n$  if and only if  $g(\bigvee C) = \bigvee g(C)$  for all non-empty chains  $C \subseteq X$ .

<sup>6</sup> A monotone function  $Y : \overline{\mathbb{R}}^n \rightarrow 2^{\mathbb{R}^k}$  is upward chain continuous on an upward closed set  $X \subseteq \overline{\mathbb{R}}^n$  if and only if  $Y(\bigvee C) = \bigcup Y(C)$  for all chains  $C \subseteq X$ .

**Algorithm EvalForMaxAtt** As a start, we firstly consider the case where all right-hand sides are upward chain continuous parametrized optimisation problems of the form  $\sup \{f(y) \mid y \in Y(\mathbf{x}_1, \dots, \mathbf{x}_n)\}$ , where

$$\sup \{f(y) \mid y \in Y(x_1, \dots, x_n)\} = \max \{f(y) \mid y \in Y(x_1, \dots, x_n)\} \quad (42)$$

for all  $x_1, \dots, x_n \in \overline{\mathbb{R}}$  with  $-\infty < \sup \{f(y) \mid y \in Y(x_1, \dots, x_n)\} < \infty$ . We say that such a parametrized optimization problem *attains its optimal value for all parameter values*. Parametrized linear programming problems, for instance, are parametrized optimization problems that attain their optimal values for all parameter values. In the case at hand, the variable assignment  $\rho^*$  can be characterized as follows:

$$\rho^*(\mathbf{x}) := \sup \{\rho(\mathbf{x}) \mid \rho : \mathbf{X}_{\mathcal{C}(\mathcal{E})} \rightarrow \mathbb{R} \text{ and } \rho \leq \llbracket \mathcal{C}(\mathcal{E}) \rrbracket \rho\} \quad \text{for all } \mathbf{x} \in \mathbf{X}, \quad (43)$$

where the constraint system  $\mathcal{C}(\mathcal{E})$  is obtained from  $\mathcal{E}$  by replacing every equation

$$\mathbf{x} = \sup \{f(y) \mid y \in Y(\mathbf{x}_1, \dots, \mathbf{x}_n)\} \quad (44)$$

with the constraints

$$\mathbf{x} \leq f(\mathbf{y}_1, \dots, \mathbf{y}_k) \quad (\mathbf{y}_1, \dots, \mathbf{y}_k) \in Y(\mathbf{x}_1, \dots, \mathbf{x}_n), \quad (45)$$

where  $\mathbf{y}_1, \dots, \mathbf{y}_k$  are fresh variables.

As we will see in the remainder of this section, the above characterization enable us to compute  $\rho^*$  using specialized convex optimization techniques. If, for instance, the right-hand sides are *parametrized linear programming problems* (to be defined), then we can compute  $\rho^*$  through linear programming. Likewise, if the right-hand sides are *parametrized semi-definite programming problems* (to be defined), then we can compute  $\rho^*$  through semi-definite programming.

*Example 21.* Let us consider the system  $\mathcal{E}$  of equations that consist of the following equations:

$$\mathbf{x}_1 = \sup \{x'_1 \in \mathbb{R} \mid x'_1 \in \mathbb{R}, x'_1 \leq 0\} \quad (46)$$

$$\mathbf{x}_2 = \sup \{x''_2 \in \mathbb{R} \mid x'_2, x''_2 \in \mathbb{R}, 0 \leq x'_2 \leq \mathbf{x}_1, x''_2 \leq 1\} \quad (47)$$

We aim at computing the variable assignment  $\rho^* : \mathbf{X} \rightarrow \overline{\mathbb{R}}$  defined by

$$\rho^*(\mathbf{x}) := \sup \{\rho(\mathbf{x}) \mid \rho : \mathbf{X} \rightarrow \mathbb{R} \text{ and } \rho \leq \llbracket \mathcal{E} \rrbracket \rho\} \quad \text{for all } \mathbf{x} \in \mathbf{X}. \quad (48)$$

All right-hand sides of the equations are upward continuous parametrized optimization problems that attain their optimal value for all parameter values. Hence, we can apply the above described method to compute  $\rho^*$ . If we do so, the system  $\mathcal{C}(\mathcal{E})$  of inequalities consist of the following inequalities:

$$\mathbf{x}_1 \leq \mathbf{x}'_1 \quad \mathbf{x}'_1 \leq 0 \quad \mathbf{x}_2 \leq \mathbf{x}''_2 \quad 0 \leq \mathbf{x}'_2 \leq \mathbf{x}_1 \quad \mathbf{x}''_2 \leq 1. \quad (49)$$

According to Equation (43), for all  $i \in \{1, 2\}$ , we thus have

$$\rho^*(\mathbf{x}_i) = \sup \{ \mathbf{x}_i \mid \mathbf{x}_1, \mathbf{x}'_1, \mathbf{x}_2, \mathbf{x}'_2, \mathbf{x}''_2 \in \mathbb{R}, \quad (50)$$

$$\mathbf{x}_1 \leq \mathbf{x}'_1, \mathbf{x}'_1 \leq 0, \mathbf{x}_2 \leq \mathbf{x}''_2, 0 \leq \mathbf{x}'_2 \leq \mathbf{x}_1, \mathbf{x}''_2 \leq 1 \} \quad (51)$$

Observe that these optimization problems are actually linear programming problems. Solving these linear programming problems gives us, as desired,  $\rho^* = \{\mathbf{x}_1 \mapsto 0, \mathbf{x}_2 \mapsto 1\}$ .  $\square$

**Algorithm EvalForGen** If we are not in the nice situation that all parametrized optimization problems attain their optimal values for all parameter values, then we have to apply a more sophisticated method to compute  $\rho^*$ . The following example, that is obtained from Example 21, illustrates the need for more sophisticated methods.

*Example 22.* We now slightly modify the fixpoint equation system  $\mathcal{E}$  from Example 21 by replacing Equation (46) by the equation  $\mathbf{x}_1 = \sup \{x'_1 \in \mathbb{R} \mid x'_1 \in \mathbb{R}, x'_1 < 0\}$ . That is, we are now concerned with strict inequality instead of non-strict inequality. In consequence, the parametrized optimization problem does not attain its optimal value for any parameter value. The fixpoint equation system  $\mathcal{E}$  now consists of the following equations:

$$\mathbf{x}_1 = \sup \{x'_1 \in \mathbb{R} \mid x'_1 \in \mathbb{R}, x'_1 < 0\} \quad (52)$$

$$\mathbf{x}_2 = \sup \{x''_2 \in \mathbb{R} \mid x'_2, x''_2 \in \mathbb{R}, 0 \leq x'_2 \leq \mathbf{x}_1, x''_2 \leq 1\} \quad (53)$$

This modification does not change the value of  $\rho^*$  (defined by Equation (48)), since the right-hand side of the first equation still evaluates to 0. However, the system  $\mathcal{C}(\mathcal{E})$  of inequalities is now given by

$$\mathbf{x}_1 < \mathbf{x}'_1 \quad \mathbf{x}'_1 \leq 0 \quad \mathbf{x}_2 \leq \mathbf{x}''_2 \quad 0 \leq \mathbf{x}'_2 \leq \mathbf{x}_1 \quad \mathbf{x}''_2 \leq 1. \quad (54)$$

Since the above inequalities imply  $0 \leq \mathbf{x}'_2 \leq \mathbf{x}_1 < \mathbf{x}'_1 \leq 0$  and thus  $0 < 0$ , there is no solution to the above inequalities. Therefore, we cannot apply the methods we applied in Example 21 to compute  $\rho^*$ .  $\square$

We now describe a more sophisticated method to compute  $\rho^*$ . For all variable assignments  $\rho_0$  and  $\rho$ , we define the system  $\mathcal{E}_{\rho_0, \rho}$  of equations as follows:

$$\begin{aligned} \mathcal{E}_{\rho_0, \rho} := & \{ \mathbf{x} = \rho_0(\mathbf{x}) \mid \mathbf{x} = e \in \mathcal{E} \text{ and } \rho_0(\mathbf{x}) \geq \llbracket e \rrbracket \rho \} \\ & \cup \{ \mathbf{x} = e \mid \mathbf{x} = e \in \mathcal{E} \text{ and } \rho_0(\mathbf{x}) < \llbracket e \rrbracket \rho \} \end{aligned} \quad (55)$$

That is,  $\mathcal{E}_{\rho_0, \rho}$  contains all equations  $\mathbf{x} = e$  of  $\mathcal{E}$  whose right-hand sides  $e$  evaluate under  $\rho$  to a value greater than  $\rho_0(\mathbf{x})$ . The other equations of  $\mathcal{E}$  are replaced by  $\mathbf{x} = \rho_0(\mathbf{x})$ . We again assume that  $\rho_0$  is a variable assignment with  $-\infty \triangleleft \rho_0 \leq \llbracket \mathcal{E} \rrbracket \rho_0 \triangleleft \infty$ . For all  $k \in \mathbb{N}_{>0}$ , we then define the variable assignment  $\rho_k$  inductively by

$$\rho_k(\mathbf{x}) := \sup \{ \rho(\mathbf{x}) \mid \rho : \mathbf{X} \rightarrow \mathbb{R} \text{ and } \rho \leq \llbracket \mathcal{E}_{\rho_0, \rho_{k-1}} \rrbracket \rho \} \quad \text{for all } \mathbf{x} \in \mathbf{X}. \quad (56)$$

Now,  $\rho^*$  is the limit of the sequence  $(\rho_k)_{k \in \mathbb{N}}$  and the sequence reaches its limits after at most  $|\mathbf{X}|$  steps:

**Lemma 21.** *The sequence  $(\rho_k)_{k \in \mathbb{N}}$  of variables assignments is increasing,  $\rho_k \leq \rho^*$  for all  $k \in \mathbb{N}$ ,  $\rho_{k+1} > \rho_k$  if  $\rho_k < \rho^*$ , and  $\rho_{|\mathbf{X}|} = \rho^*$ . Moreover,  $\rho_k(\mathbf{x}) = \sup \{\rho(\mathbf{x}) \mid \rho : \mathbf{X}_{\mathcal{C}(\mathcal{E}_{\rho_0, \rho_{k-1}})} \rightarrow \mathbb{R} \text{ and } \rho \leq \llbracket \mathcal{C}(\mathcal{E}_{\rho_0, \rho_{k-1}}) \rrbracket \rho\}$  for all  $k$  and all  $\mathbf{x} \in \mathbf{X}$ .*  $\square$

*Example 23.* Let us again consider the fixpoint equation system  $\mathcal{E}$  from Example 22. We again aim at computing the variable assignment  $\rho^* : \mathbf{X} \rightarrow \overline{\mathbb{R}}$  that is defined by  $\rho^*(\mathbf{x}) := \sup \{\rho(\mathbf{x}) \mid \rho : \mathbf{X} \rightarrow \mathbb{R} \text{ and } \rho \leq \llbracket \mathcal{E} \rrbracket \rho\}$  for all  $\mathbf{x} \in \mathbf{X}$ . Since  $-\infty < \rho_0 \leq \llbracket \mathcal{E} \rrbracket \rho_0 < \infty$  for  $\rho_0 := \{\mathbf{x}_1 \mapsto 0, \mathbf{x}_2 \mapsto 0\}$ , we can apply the method we just developed. The system  $\mathcal{E}_{\rho_0, \rho_0}$  is given by

$$\mathbf{x}_1 = 0 \quad \mathbf{x}_2 = \sup \{x_2'' \in \mathbb{R} \mid x_2', x_2'' \in \mathbb{R}, 0 \leq x_2' \leq \mathbf{x}_1, x_2'' \leq 1\} \quad (57)$$

Therefore, the constraint system  $\mathcal{C}(\mathcal{E}_{\rho_0, \rho_0})$  is given by

$$\mathbf{x}_1 \leq 0 \quad \mathbf{x}_2 \leq \mathbf{x}_2'' \quad 0 \leq \mathbf{x}_2' \leq \mathbf{x}_1 \quad \mathbf{x}_2'' \leq 1 \quad (58)$$

Solving the optimization problems that aims at maximizing  $\mathbf{x}_1$  and  $\mathbf{x}_2$ , respectively, we get  $\rho_1 = \{\mathbf{x}_1 \mapsto 0, \mathbf{x}_2 \mapsto 1\}$ . We then construct the fixpoint equation system  $\mathcal{E}_{\rho_0, \rho_1}$ . The system  $\mathcal{E}_{\rho_0, \rho_1}$  is equal to the system  $\mathcal{E}_{\rho_0, \rho_0}$ , and thus  $\mathcal{C}(\mathcal{E}_{\rho_0, \rho_1})$  is equal to the system  $\mathcal{C}(\mathcal{E}_{\rho_0, \rho_0})$ . Therefore, we get  $\rho^* = \rho_1$  by Lemma 21.  $\square$

**Algorithm EvalForCmorcave** In our static program analysis application we discuss in the next section, we have the comfortable situation that our right-hand sides are not only upward continuous parametrized optimization problems, but they are additionally cmcave. We can utilize this in order to simplify the above developed procedure EvalForGen. The following lemma is the key ingredient for this optimization:

**Lemma 22.** *Let  $\rho$  be a feasible pre-solution of a system  $\mathcal{E}$  of cmorcave equations. For all  $\mathbf{x} \in \mathbf{X}$ , we have  $\mu_{\geq \rho} \llbracket \mathcal{E} \rrbracket(\mathbf{x}) > \rho(\mathbf{x})$  if and only if  $(\llbracket \mathcal{E} \rrbracket^{|\mathbf{X}|} \rho)(\mathbf{x}) > \rho(\mathbf{x})$ .*

*Proof (Sketch).* Since  $\rho$  is a feasible pre-solution of  $\mathcal{E}$ , we can w.l.o.g. assume that  $\llbracket e \rrbracket \rho > -\infty$  for all equations  $\mathbf{x} = e$  of  $\mathcal{E}$ . Therefore,  $\llbracket \mathcal{E} \rrbracket$  is upward chain continuous on  $(\mathbf{X} \rightarrow \overline{\mathbb{R}})_{\geq \rho}$ . The statement finally follows from the fact that  $\llbracket \mathcal{E} \rrbracket$  is additionally monotone and order-concave.  $\square$

Assume now that we want to use our  $\vee$ -strategy improvement algorithm to compute the least solution of a system of  $\vee$ -cmorcave equations. In each  $\vee$ -strategy improvement step, we are then in the situation that we have to compute  $\rho^* := \mu_{\geq \rho} \llbracket \mathcal{E} \rrbracket$ , where  $\rho$  is a feasible pre-solution of a system  $\mathcal{E}$  of morcave equations (cf. Lemma 17). By Lemma 22, we can compute the set

$$\mathbf{X}' := \{\mathbf{x} \in \mathbf{X} \mid \rho^*(\mathbf{x}) > \rho(\mathbf{x})\} \quad (59)$$

by performing  $|\mathbf{X}|$  Kleene iteration steps. We then construct the equation system

$$\mathcal{E}' := \{\mathbf{x} = e \in \mathcal{E} \mid \mathbf{x} \in \mathbf{X}'\} \cup \{\mathbf{x} = \rho(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X} \setminus \mathbf{X}'\} \quad (60)$$

By construction, we get:

**Lemma 23.**  $\rho^*(\mathbf{x}) = \sup \{\rho(\mathbf{x}) \mid \rho : \mathbf{X}_{\mathcal{C}(\mathcal{E}')} \rightarrow \mathbb{R} \text{ and } \rho \leq \llbracket \mathcal{C}(\mathcal{E}') \rrbracket \rho\}$  for all  $\mathbf{x} \in \mathbf{X}$ .  $\square$

In consequence, we can compute  $\rho^*$  by performing  $|\mathbf{X}|$  Kleene iteration steps followed by solving  $|\mathbf{X}|$  optimization problems.

*Example 24.* Let us again consider the fixpoint equation system  $\mathcal{E}$  from Example 22 and 23. That is,  $\mathcal{E}$  consists of the following equations:

$$\mathbf{x}_1 = \sup \{x'_1 \in \mathbb{R} \mid x'_1 \in \mathbb{R}, x'_1 < 0\} \quad (61)$$

$$\mathbf{x}_2 = \sup \{x''_2 \in \mathbb{R} \mid x'_2, x''_2 \in \mathbb{R}, 0 \leq x'_2 \leq \mathbf{x}_1, x''_2 \leq 1\} \quad (62)$$

The fixpoint equation system  $\mathcal{E}$  is a system of cmorcave equations. The pre-solution  $\rho := \{\mathbf{x}_1 \mapsto 0, \mathbf{x}_2 \mapsto 0\}$  of  $\mathcal{E}$  is feasible. Moreover, we have  $-\infty \triangleleft \rho \leq \llbracket \mathcal{E} \rrbracket \rho \triangleleft \infty$ . We aim at computing  $\rho^* := \mu_{\geq \rho} \llbracket \mathcal{E} \rrbracket$ .

We have  $\llbracket \mathcal{E} \rrbracket^{|\mathbf{X}|} \rho = \llbracket \mathcal{E} \rrbracket^2 \rho = \{\mathbf{x}_1 \mapsto 0, \mathbf{x}_2 \mapsto 1\}$ . By Lemma 22, we thus get  $\mathbf{X}' := \{\mathbf{x} \in \mathbf{X} \mid \rho^*(\mathbf{x}) > \rho(\mathbf{x})\} = \{\mathbf{x}_2\}$  (cf. (59)). Lemma 23 finally gives us

$$\rho^*(\mathbf{x}_i) = \sup \{\mathbf{x}_i \mid \mathbf{x}_1, \mathbf{x}'_1, \mathbf{x}_2, \mathbf{x}'_2, \mathbf{x}''_2 \in \mathbb{R}, \mathbf{x}_1 \leq 0, \mathbf{x}_2 \leq \mathbf{x}''_2, 0 \leq \mathbf{x}'_2 \leq \mathbf{x}_1, \mathbf{x}''_2 \leq 1\}. \quad (63)$$

for all  $i \in \{1, 2\}$  (cf. Example 23). This is the desired result. We performed two Kleene iteration steps and solved two mathematical optimization problems.  $\square$

### 5.3 Parameterized Linear Programming Problems

We now introduce *parameterized linear programming problems*. We do this as follows. For all  $A \in \mathbb{R}^{k \times m}$  and all  $c \in \mathbb{R}^m$ , we define the operator  $\mathbf{LP}_{A,c} : \overline{\mathbb{R}}^k \rightarrow \overline{\mathbb{R}}$  which solves a *parametrized linear programming problem* by

$$\mathbf{LP}_{A,c}(b) := \sup \{c^\top x \mid x \in \mathbb{R}^m \text{ and } Ax \leq b\} \quad \text{for all } b \in \overline{\mathbb{R}}^k. \quad (64)$$

We use the LP-operators in the right-hand sides of fixpoint equation systems:

**Definition 9.** (*LP-equations,  $\vee$ -LP-equations*) A fixpoint equation  $\mathbf{x} = e$  is called LP-equation if and only if  $e$  is a parametrized linear programming problem. It is called  $\vee$ -LP-equation if and only if  $e$  is a point-wise maximum of finitely many semi-definite programming problems.  $\square$

LP-operators have the following important properties:

**Lemma 24.** The following statements hold for all  $A \in \mathbb{R}^{k \times m}$  and all  $c \in \mathbb{R}^m$ :

1. The operator  $\mathbf{LP}_{A,c}$  is cmcave.
2.  $\mathbf{LP}_{A,c}(b) = \max \{c^\top x \mid x \in \mathbb{R}^m \text{ and } Ax \leq b\}$  for all  $b \in \overline{\mathbb{R}}^k$  with  $-\infty < \mathbf{LP}_{A,c}(b) < \infty$ . That is, the parametrized optimization problem  $\mathbf{LP}_{A,c}$  attains its optimal value for all parameter values.



*Proof.* We do not prove the first statement, since, as we will see, it is just a special case of Lemma 25 (see below). This second statement is a direct consequence of the fact that the optimal value of a feasible and bounded linear programming problem is attained at the edges of the feasible space.  $\square$

If we apply our  $\vee$ -strategy improvement algorithm for solving a system of  $\vee$ -LP-equations, then, because of Lemma 24, we have the convenient situation that we can apply Algorithm **EvalForMaxAtt** instead of its more general variant **EvalForGen** for evaluating a single  $\vee$ -strategy that is encountered during the  $\vee$ -strategy iteration (see Section 5.2). We thus obtain the following result:

**Theorem 2.** *If  $\mathcal{E}$  is a system of  $\vee$ -LP-equations, then the evaluation of a  $\vee$ -strategy that is encountered during the  $\vee$ -strategy iteration can be performed by solving  $|\mathbf{X}|$  linear programming problems, each of which can be constructed in polynomial time. In consequence, a  $\vee$ -strategy improvement step can be performed in polynomial time.*  $\square$

Theorem 1 implies that our  $\vee$ -strategy improvement algorithm terminates after at most  $|\mathbf{X}| \cdot |\Sigma|$   $\vee$ -strategy improvement steps, whenever it runs on a system  $\mathcal{E}$  of  $\vee$ -LP-equations.

A consequence of the fact that we can evaluate  $\vee$ -strategies in polynomial time is the following decision problem is in **NP**: Decide whether or not, for a given system  $\mathcal{E}$  of  $\vee$ -LP-equations, a given variable  $\mathbf{x} \in \mathbf{X}$ , and a given value  $b \in \mathbb{R}$ , the statement  $\mu[\mathcal{E}](\mathbf{x}) \leq b$  holds. This decision problem is at least as hard as the problem of computing the winning regions in mean payoff games. However, whether or not it is **NP**-hard is an open question.

#### 5.4 Parameterized Semi-Definite Programming Problems

As a strict generalization of *parameterized linear programming problems*, we now introduce *parameterized semi-definite programming problems*. Before we can do so, we have to briefly introduce semi-definite programming.

**Semi-definite Programming**  $S\mathbb{R}^{n \times n}$  (resp.  $S\mathbb{R}_+^{n \times n}$ ) denotes the set of symmetric matrices (resp. the set of positive semidefinite matrices).  $\preceq$  denotes the Löwner ordering of symmetric matrices, i.e.,  $A \preceq B$  if and only if  $B - A \in S\mathbb{R}_+^{n \times n}$ .  $\text{Tr}(A)$  denotes the trace of a square matrix  $A \in \mathbb{R}^{n \times n}$ , i.e.,  $\text{Tr}(A) = \sum_{i=1}^n A_{i,i}$ . The inner product of two matrices  $A$  and  $B$  is denoted by  $A \bullet B$ , i.e.,  $A \bullet B = \text{Tr}(A^\top B)$ . For  $\mathcal{A} = (A_1, \dots, A_m)$  with  $A_i \in \mathbb{R}^{n \times n}$  for all  $i = 1, \dots, m$ , we denote the vector  $(A_1 \bullet X, \dots, A_m \bullet X)^\top$  by  $\mathcal{A}(X)$ . For all  $x \in \mathbb{R}^n$ , the dyadic matrix  $X(x)$  is defined by

$$X(x) := \begin{pmatrix} 1 \\ x \end{pmatrix} (1, x^\top). \quad (65)$$

We consider semidefinite programming problems (SDP problems for short) of the form

$$z^* = \sup \{C \bullet X \mid X \in S\mathbb{R}_+^{n \times n}, \mathcal{A}(X) = a, \mathcal{B}(X) \leq b\}, \quad (66)$$

where  $\mathcal{A} = (A_1, \dots, A_m)$ ,  $a \in \mathbb{R}^m$ ,  $A_1, \dots, A_m \in S\mathbb{R}^{n \times n}$ ,  $\mathcal{B} = (B_1, \dots, B_k)$ ,  $B_1, \dots, B_k \in S\mathbb{R}^{n \times n}$ ,  $b \in \mathbb{R}^k$ , and  $C \in S\mathbb{R}^{n \times n}$ . The set  $\{X \in S\mathbb{R}_+^{n \times n} \mid \mathcal{A}(X) = a, \mathcal{B}(X) \leq b\}$  is called the *feasible space*. The problem is called *feasible* if and only if the feasible space is non-empty. It is called *infeasible* otherwise. An element of the feasible space is called *feasible solution*. The value  $z^*$  is called *optimal value*. The problem is called *bounded* iff  $z^* < \infty$ . It is called *unbounded*, otherwise. A feasible solution  $X^*$  is called an *optimal solution* if and only if  $z^* = C \bullet X^*$ . In contrast to the situation for linear programming, there exist feasible and bounded semi-definite programming problem that have no optimal solution.

For semi-definite programming problems, fast algorithms exist. Semi-definite programming is polynomial time solvable if an a priori bound on the size of the solutions is known and provided as an input.

For more detailed information on semi-definite programming, or, more generally, on convex optimization, we refer, for instance, to Nemirovski [13], Todd [17].

**Parametrized SDP Problems** For  $\mathcal{A} = (A_1, \dots, A_m)$ ,  $A_1, \dots, A_m \in S\mathbb{R}^{n \times n}$ ,  $a \in \mathbb{R}^m$ ,  $\mathcal{B} = (B_1, \dots, B_k)$ ,  $B_1, \dots, B_k \in S\mathbb{R}^{n \times n}$ , and  $C \in S\mathbb{R}^{n \times n}$ , we define the operator  $\mathbf{SDP}_{\mathcal{A}, a, \mathcal{B}, C} : \mathbb{R}^k \rightarrow \overline{\mathbb{R}}$  which solves a *parametrized SDP problem* by

$$\mathbf{SDP}_{\mathcal{A}, a, \mathcal{B}, C}(b) := \sup \{C \bullet X \mid X \in S\mathbb{R}_+^{n \times n}, \mathcal{A}(X) = a, \mathcal{B}(X) \leq b\} \text{ for all } b \in \overline{\mathbb{R}}^k.$$

The SDP-operators generalizes the LP-operators in the same way as semi-definite programming generalizes linear programming. That is, for every LP-operator we can construct an equivalent SDP-operator.

**Definition 10.** (*SDP-equations,  $\vee$ -SDP-equations*) A fixpoint equation  $\mathbf{x} = e$  is called SDP-equation if and only if  $e$  is a parametrized semi-definite programming problem. It is called  $\vee$ -SDP-equation if and only if  $e$  is a point-wise maximum of finitely many semi-definite programming problems.  $\square$

For this article, the following properties of SDP-operators are important:

**Lemma 25.** *The operator  $\mathbf{SDP}_{\mathcal{A}, a, \mathcal{B}, C}$  is cmcave.*

*Proof.* Let  $f := \mathbf{SDP}_{\mathcal{A}, a, \mathcal{B}, C}$ . For all  $b \in \mathbb{R}^k$ , let  $M(b) := \{X \in S\mathbb{R}_+^{n \times n} \mid \mathcal{A}(X) = a, \mathcal{B}(X) \leq b\}$ . Therefore,  $f(b) = \sup \{C \bullet X \mid X \in M(b)\}$  for all  $b \in \mathbb{R}^k$ . We do not need to consider all  $I : \{1, \dots, k\} \rightarrow \{-\infty, \text{id}, \infty\}$ , because, for all  $I : \{1, \dots, k\} \rightarrow \{-\infty, \text{id}, \infty\}$ ,  $f^{(I)}$  can be obtained by choosing appropriate  $\mathcal{A}, a, \mathcal{B}, C$ . The fact that  $f$  is monotone is obvious. Firstly, we show that  $f(b) < \infty$  holds for all  $b \in \mathbb{R}^k$ , whenever  $\text{fdom}(f) \neq \emptyset$ . For the sake of contradiction assume that there exist  $b_1, b_2 \in \mathbb{R}^k$  such that  $f(b_1) \in \mathbb{R}$  and  $f(b_2) = \infty$  hold. Note that  $M(b_i)$  are convex sets for all  $i \in \{1, 2\}$ . Thus, there exists some  $D \in S\mathbb{R}_+^{n \times n}$  such that  $C \bullet D > 0$  and  $M(b_2) + \{\lambda D \mid \lambda \in \mathbb{R}_{\geq 0}\} \subseteq M(b_2)$  hold. Therefore,  $\mathcal{A}(D) = 0$  and  $\mathcal{B}(D) \leq 0$ . Let  $X_1 \in S\mathbb{R}_+^{n \times n}$  with  $\mathcal{A}(X_1) = a$  and  $\mathcal{B}(X_1) \leq b_1$ . Then  $\mathcal{A}(X_1 + \lambda D) = \mathcal{A}(X_1) + \lambda \mathcal{A}(D) = a$  and  $\mathcal{B}(X_1 + \lambda D) = \mathcal{B}(X_1) + \lambda \mathcal{B}(D) \leq b_1$

hold for all  $\lambda > 0$ . Thus,  $f(b_1) = \infty$  — contradiction. Thus,  $f(b) < \infty$  holds for all  $b \in \mathbb{R}^k$ , whenever  $\text{fdom}(f) \neq \emptyset$ .

Next, we show that  $\text{fdom}(f)$  is convex and  $f|_{\text{fdom}(f)}$  is concave. Assume that  $\text{fdom}(f) \neq \emptyset$ . Thus,  $f(b) < \infty$  for all  $b \in \mathbb{R}^k$ . Let  $b_1, b_2 \in \text{fdom}(f)$ ,  $\lambda \in [0, 1]$ , and  $b := \lambda b_1 + (1 - \lambda)b_2$ . In order to show that

$$\lambda M(b_1) + (1 - \lambda)M(b_2) \subseteq M(b) \quad (67)$$

holds, let  $X_i \in M(b_i)$ ,  $i = 1, 2$ , and  $X = \lambda X_1 + (1 - \lambda)X_2$ . Since  $X_i \in S\mathbb{R}_+^{n \times n}$ ,  $\mathcal{A}(X_i) = a$ , and  $\mathcal{B}(X_i) \leq b_i$  for all  $i = 1, 2$ , we have  $X \in S\mathbb{R}_+^{n \times n}$ ,  $\mathcal{A}(X) = \lambda \mathcal{A}(X_1) + (1 - \lambda)\mathcal{A}(X_2) = a$ ,  $\mathcal{B}(X) = \lambda \mathcal{B}(X_1) + (1 - \lambda)\mathcal{B}(X_2) \leq \lambda b_1 + (1 - \lambda)b_2 = b$ . Therefore,  $X \in M(b)$ . Using (67), we finally get:

$$f(b) = \sup\{C \bullet X \mid X \in M(b)\} \quad (68)$$

$$\geq \lambda \sup\{C \bullet X_1 \mid X_1 \in M(b_1)\} + (1 - \lambda) \sup\{C \bullet X_2 \mid X_2 \in M(b_2)\} \quad (69)$$

$$= \lambda f(b_1) + (1 - \lambda)f(b_2) > -\infty \quad (70)$$

Therefore,  $\text{fdom}(f)$  is convex and  $f|_{\text{fdom}(f)}$  is concave.

It remains to show that  $f$  is upward chain continuous on  $f^{-1}(\overline{\mathbb{R}} \setminus \{-\infty\})$ . For that, let  $B \subseteq f^{-1}(\overline{\mathbb{R}} \setminus \{-\infty\})$  be a chain. We have

$$f(\bigvee B) = \sup\{C \bullet X \mid X \in M(\bigvee B)\} \quad (71)$$

$$= \sup\{C \bullet X \mid X \in \bigcup\{M(b) \mid b \in B\}\} \quad (M \text{ is continuous}) \quad (72)$$

$$= \sup\{\sup\{C \bullet X \mid X \in M(b)\} \mid b \in B\} \quad (73)$$

$$= \sup\{f(b) \mid b \in B\} \quad (74)$$

This proves that  $f$  is upward chain continuous on  $f^{-1}(\overline{\mathbb{R}} \setminus \{-\infty\})$ .  $\square$

The next example shows that the square root operator can be expressed through a SDP-operator:

*Example 25.* The square root operator  $\sqrt{\cdot} : \overline{\mathbb{R}} \rightarrow \overline{\mathbb{R}}$  is defined by  $\sqrt{b} := \sup\{x \in \mathbb{R} \mid x^2 \leq b\}$  for all  $b \in \overline{\mathbb{R}}$ . Note that  $\sqrt{b} = -\infty$  for all  $b < 0$ , and  $\sqrt{\infty} = \infty$ . Let

$$\mathcal{A} := \left( \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \right), \quad a := 1, \quad \mathcal{B} := \left( \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \right), \quad C := \left( \begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix} \right). \quad (75)$$

For  $x, b \in \mathbb{R}_{\geq 0}$ , the statement  $x^2 \leq b$  is equivalent to the statement  $\exists b'. x^2 \leq b' \leq b$ . By the Schur complement theorem (c.f. Section 3, Example 5 of Todd [17], for instance), this is equivalent to

$$\exists b'. \begin{pmatrix} 1 & x \\ x & b' \end{pmatrix} \succeq 0 \wedge b' \leq b. \quad (76)$$

This is equivalent to  $\exists X \in S\mathbb{R}_+^{2 \times 2}. x = X_{1,2} = X_{2,1} \wedge \mathcal{A}(X) = a \wedge \mathcal{B}(X) \leq b$ . Thus,  $\sqrt{b} = \mathbf{SDP}_{\mathcal{A}, a, \mathcal{B}, C}(b)$  for all  $b \in \overline{\mathbb{R}}$ .  $\square$

If  $\mathcal{E}$  is a system of  $\vee$ -SDP-equations, then, because of Lemma 25, we have the convenient situation that we can apply Algorithm `EvalForCmorcave` instead of its more general variant `EvalForGen` (see Section 5.2) to evaluate the  $\vee$ -strategies that are encountered during the  $\vee$ -strategy iteration. This case is in particular interesting for the static program analysis application we will describe in Section 6.

**Theorem 3.** *If  $\mathcal{E}$  is a system of  $\vee$ -SDP-equations, then the evaluation of a  $\vee$ -strategy that is encountered during the  $\vee$ -strategy iteration can be performed by performing  $|\mathbf{X}|$  Kleene iteration steps and subsequently solving  $|\mathbf{X}|$  semi-definite programming problems, each of which can be constructed in polynomial time.  $\square$*

Theorem 1 implies that our  $\vee$ -strategy improvement algorithm terminates after at most  $|\mathbf{X}| \cdot |\Sigma|$   $\vee$ -strategy improvement steps, whenever it runs on a system  $\mathcal{E}$  of  $\vee$ -LP-equations.

## 6 Quadratic Zones and Relaxed Abstract Semantics

In this section, we apply our  $\vee$ -strategy improvement algorithm to a static program analysis problem. For that, we first introduce our programming model as well as its collecting and its abstract semantics. We then relax the abstract semantics along the same lines as Adjé, Gaubert, and Goubault [1] using Shor’s semidefinite relaxation schema. Finally, we show how we can use our finding to compute the relaxation of the abstract semantics.

### 6.1 Collecting Semantics

In our programming model, we consider statements of the following two forms:

1.  $x := Ax + b$ , where  $A \in \mathbb{R}^{n \times n}$ , and  $b \in \mathbb{R}^n$  (*affine assignments*)
2.  $x^\top Ax + 2b^\top x \leq c$ , where  $A \in S\mathbb{R}^{n \times n}$ ,  $b \in \mathbb{R}^n$ , and  $c \in \mathbb{R}$  (*quadratic guards*)

Here,  $x \in \mathbb{R}^n$  denotes the vector of program variables. We denote the set of statements by **Stmt**. The *collecting semantics*  $\llbracket s \rrbracket : 2^{\mathbb{R}^n} \rightarrow 2^{\mathbb{R}^n}$  of a statement  $s \in \mathbf{Stmt}$  is defined by:

$$\llbracket x := Ax + b \rrbracket X := \{Ax + b \mid x \in X\} \quad \text{for all } X \subseteq \mathbb{R}^n \quad (77)$$

$$\llbracket x^\top Ax + 2b^\top x \leq c \rrbracket X := \{x \in X \mid x^\top Ax + 2b^\top x \leq c\} \quad \text{for all } X \subseteq \mathbb{R}^n \quad (78)$$

A program  $G$  is a triple  $(N, E, \mathbf{st}, I)$ , where  $N$  is a finite set of *control-points*,  $E \subseteq N \times \mathbf{Stmt} \times N$  is a finite set of control-flow edges,  $\mathbf{st} \in N$  is the start control-point, and  $I \subseteq \mathbb{R}^n$  is a set of initial values. The *collecting semantics*  $V$  of a program  $G = (N, E, \mathbf{st}, I)$  is then the least solution of the following constraint system:

$$\mathbf{V}[\mathbf{st}] \supseteq I \quad \mathbf{V}[v] \supseteq \llbracket s \rrbracket(\mathbf{V}[u]) \quad \text{for all } (u, s, v) \in E \quad (79)$$

Here, the variables  $\mathbf{V}[v]$ ,  $v \in N$  take values in  $2^{\mathbb{R}^n}$ . The components of the collecting semantics  $V$  are denoted by  $V[v]$  for all  $v \in N$ .

## 6.2 Quadratic Zones and Abstract Semantics

Along the lines of Adjé, Gaubert, and Goubault [1], we define *quadratic zones* as follows: A set  $P$  of templates  $p : \mathbb{R}^n \rightarrow \mathbb{R}$  is a *quadratic zone* if and only if every template  $p \in P$  can be written as

$$p(x) = x^\top A_p x + 2b_p^\top x \quad \text{for all } x \in \mathbb{R}^n, \quad (80)$$

where  $A_p \in S\mathbb{R}^{n \times n}$  and  $b_p \in \mathbb{R}^n$  for all  $p \in P$ . In the remainder of this article, we assume that  $P = \{p_1, \dots, p_m\}$  is a finite quadratic zone. Moreover, we assume w.l.o.g. that  $p_i \neq 0$  for all  $i = 1, \dots, m$ . The *abstraction*  $\alpha : 2^{\mathbb{R}^n} \rightarrow P \rightarrow \overline{\mathbb{R}}$  and the *concretization*  $\gamma : (P \rightarrow \overline{\mathbb{R}}) \rightarrow 2^{\mathbb{R}^n}$  are defined as follows:

$$\gamma(v) := \{x \in \mathbb{R}^n \mid \forall p \in P. p(x) \leq v(p)\} \quad \text{for all } v : P \rightarrow \overline{\mathbb{R}} \quad (81)$$

$$\alpha(X) := \bigwedge \{v : P \rightarrow \overline{\mathbb{R}} \mid \gamma(v) \supseteq X\} \quad \text{for all } X \subseteq \mathbb{R}^n \quad (82)$$

As shown by Adjé, Gaubert, and Goubault [1],  $\alpha$  and  $\gamma$  form a Galois-connection. The elements from  $\gamma(P \rightarrow \overline{\mathbb{R}})$  and the elements from  $\alpha(2^{\mathbb{R}^n})$  are called *closed*.  $\alpha(\gamma(v))$  is called the *closure of  $v : P \rightarrow \mathbb{R}$* . Accordingly,  $\gamma(\alpha(X))$  is called the *closure of  $X \subseteq \mathbb{R}^n$* .

As usual, the *abstract semantics*  $\llbracket s \rrbracket^\# : (P \rightarrow \overline{\mathbb{R}}) \rightarrow P \rightarrow \overline{\mathbb{R}}$  of a statement  $s$  is defined by  $\llbracket s \rrbracket^\# := \alpha \circ \llbracket s \rrbracket \circ \gamma$ . The *abstract semantics*  $V^\#$  of a program  $G = (N, E, \text{st}, I)$  is then the least solution of the following constraint system:

$$\mathbf{V}^\#[\text{st}] \geq \alpha(I) \quad \mathbf{V}^\#[v] \geq \llbracket s \rrbracket^\#(\mathbf{V}^\#[u]) \quad \text{for all } (u, s, v) \in E \quad (83)$$

Here, the variables  $\mathbf{V}^\#[v]$ ,  $v \in N$  take values in  $P \rightarrow \overline{\mathbb{R}}$ . The components of the abstract semantics  $V^\#$  are denoted by  $V^\#[v]$  for all  $v \in N$ .

## 6.3 Relaxed Abstract Semantics

The problem of deciding, whether or not, for a given quadratic zone  $P$ , a given  $v : P \rightarrow \mathbb{Q}$ , a given  $p \in P$ , and a given  $q \in \mathbb{Q}$ ,  $\alpha(\gamma(v))(p) \leq q$  holds, is NP-hard (cf. Adjé et al. [1]) and thus intractable. Therefore, we use the *relaxed abstract semantics*  $V^\mathcal{R}$  introduced by Adjé, Gaubert, and Goubault [1]. It is based on Shor's semidefinite relaxation schema. In order to fit it into our framework, we have to switch to the semi-definite dual. This is not a disadvantage. It is actually an advantage, since we gain additional precision through this step.

**Definition 11** ( $\llbracket x := Ax + b \rrbracket^\mathcal{R}$ ). *We define the relaxed abstract semantics  $\llbracket x := Ax + b \rrbracket^\mathcal{R} : (P \rightarrow \overline{\mathbb{R}}) \rightarrow P \rightarrow \overline{\mathbb{R}}$  of an affine assignment  $x := Ax + b$  by*

$$\llbracket x := Ax + b \rrbracket^\mathcal{R} v(p) \quad (84)$$

$$:= \sup \{ \overline{A}(p) \bullet X \mid \forall p' \in P. \overline{A}_{p'} \bullet X \leq v(p'), X \succeq 0, X_{1,1} = 1 \} \quad (85)$$

for all  $v : P \rightarrow \mathbb{R}$  and all  $p \in P$ , where, for all  $p' \in P$ ,

$$A(p) := A^\top A_p A, \quad b(p) := A^\top A_p b + A^\top b_p, \quad c(p) := b^\top A_p b + 2b_p^\top b \quad (86)$$

$$\overline{A}(p) := \begin{pmatrix} c(p) & b^\top(p) \\ b(p) & A(p) \end{pmatrix}, \quad \overline{A}_{p'} := \begin{pmatrix} 0 & b_{p'}^\top \\ b_{p'} & A_{p'} \end{pmatrix}. \quad \square \quad (87)$$

**Definition 12** ( $\llbracket x^\top Ax + 2b^\top x \leq c \rrbracket^\mathcal{R}$ ). We define the relaxed abstract semantics  $\llbracket x^\top Ax + 2b^\top x \leq c \rrbracket^\mathcal{R} : (P \rightarrow \mathbb{R}) \rightarrow P \rightarrow \mathbb{R}$  of a quadratic guard  $x^\top Ax + 2b^\top x \leq c$  by

$$\llbracket x^\top Ax + 2b^\top x \leq c \rrbracket^\mathcal{R} v(p) \quad (88)$$

$$:= \sup \{ \bar{A}_p \bullet X \mid \forall p' \in P. \bar{A}_{p'} \bullet X \leq v(p'), \tilde{A} \bullet X \leq 0, X \succeq 0, X_{1,1} = 1 \} \quad (89)$$

for all  $v : P \rightarrow \mathbb{R}$  and all  $p \in P$ , where, for all  $p' \in P$ ,

$$\tilde{A} := \begin{pmatrix} -c & b^\top \\ b & A \end{pmatrix}, \quad \bar{A}_{p'} := \begin{pmatrix} 0 & b_{p'}^\top \\ b_{p'} & A_{p'} \end{pmatrix}. \quad \square \quad (90)$$

The relaxed abstract semantics  $\llbracket \cdot \rrbracket^\mathcal{R}$  is the semidefinite dual of the one used by Adjé, Gaubert, and Goubault [1]. By weak-duality, it is at least as precise as the one used by Adjé, Gaubert, and Goubault [1].

Next, we show that the relaxed abstract semantics is indeed a relaxation of the abstract semantics, and that the relaxed abstract semantics of a statement is expressible through a SDP-operator.

**Lemma 26.** *The following statements hold for every statement  $s \in \mathbf{Stmt}$ :*

1.  $\llbracket s \rrbracket^\# \leq \llbracket s \rrbracket^\mathcal{R}$
2. For every  $i \in \{1, \dots, m\}$ , there exist  $\mathcal{A}, a, \mathcal{B}, C$  such that

$$\llbracket s \rrbracket^\mathcal{R} v(p_i) = \mathbf{SDP}_{\mathcal{A}, a, \mathcal{B}, C}(v(p_1), \dots, v(p_m)) \quad (91)$$

for all  $v : P \rightarrow \mathbb{R}$ . From  $s$ , the values  $\mathcal{A}$ ,  $a$ ,  $\mathcal{B}$ , and  $C$  can be computed in polynomial time.  $\square$

*Proof.* Since the second statement is obvious, we only prove the first one. We only consider the case that  $s$  is an affine assignment  $x := Ax + b$ . The case that  $s$  is a quadratic guard can be treated along the same lines. Let  $v : P \rightarrow \mathbb{R}$ ,  $p \in P$ , and  $v' := \llbracket x := Ax + b \rrbracket^\# v$ . Then,

$$v'(p) = \sup \{ p(Ax + b) \mid x \in \mathbb{R}^n, \forall p' \in P. p'(x) \leq v(p') \} \quad (92)$$

$$= \sup \{ x^\top A(p)x + 2b^\top(p)x + c(p) \mid \quad (93)$$

$$x \in \mathbb{R}^n, \forall p' \in P. x^\top A_{p'}x + 2b_{p'}^\top x \leq v(p') \} \quad (94)$$

$$= \sup \{ (1, x^\top) \bar{A}(p) (1, x^\top)^\top \mid \forall p' \in P. (1, x^\top) \bar{A}_{p'} (1, x^\top)^\top \leq v(p') \} \quad (95)$$

$$= \sup \{ \bar{A}(p) \bullet X(x) \mid \forall p' \in P. \bar{A}_{p'} \bullet X(x) \leq v(p') \} \quad (96)$$

$$\leq \sup \{ \bar{A}(p) \bullet X \mid \forall p' \in P. \bar{A}_{p'} \bullet X \leq v(p'), X \succeq 0, X_{1,1} = 1 \}. \quad (97)$$

The last inequality holds, because  $X(x) \succeq 0$  and  $X(x)_{1,1} = 1$  for all  $x \in \mathbb{R}^n$ . This completes the proof of statement 1.  $\square$

A relaxation of the closure operator  $\alpha \circ \gamma$  is given by  $\llbracket x := x \rrbracket^\mathcal{R}$ . That is,  $\alpha \circ \gamma \leq \llbracket x := x \rrbracket^\mathcal{R}$ .

The *relaxed abstract semantics*  $V^{\mathcal{R}}$  of a program  $G = (N, E, \mathbf{st}, I)$  is finally defined as the least solution of the following constraint system:

$$\mathbf{V}^{\mathcal{R}}[\mathbf{st}] \geq \alpha(I) \quad \mathbf{V}^{\mathcal{R}}[v] \geq \llbracket s \rrbracket^{\mathcal{R}}(\mathbf{V}^{\mathcal{R}}[u]) \quad \text{for all } (u, s, v) \in E$$

Here, the variables  $\mathbf{V}^{\mathcal{R}}[v]$ ,  $v \in N$  take values in  $P \rightarrow \overline{\mathbb{R}}$ . The components of the relaxed abstract semantics  $V^{\mathcal{R}}$  are denoted by  $V^{\mathcal{R}}[v]$  for all  $v \in N$ .

Because of Lemma 26, the relaxed abstract semantics of a program is a safe over-approximation of its abstract semantics. If all templates and all guards are linear, then the relaxed abstract semantics is precise (cf. Adjé et al. [1]):

**Lemma 27.** *We have  $V^{\sharp} \leq V^{\mathcal{R}}$ . Moreover, if all templates and all guards are linear, then  $V^{\sharp} = V^{\mathcal{R}}$ .*  $\square$

#### 6.4 Computing Relaxed Abstract Semantics

We now use our  $\vee$ -strategy improvement algorithm to compute the relaxed abstract semantics  $V^{\mathcal{R}}$  of a program  $G = (N, E, \mathbf{st}, I)$  w.r.t. a given finite quadratic zone  $P = \{p_1, \dots, p_m\}$ . For that, we define  $\mathcal{C}$  to be the constraint system

$$\mathbf{x}_{\mathbf{st}, p} \geq \alpha(I)(p) \quad \text{for all } p \in P \quad (98)$$

$$\mathbf{x}_{v, p} \geq (\llbracket s \rrbracket^{\mathcal{R}}(\mathbf{x}_{u, p_1}, \dots, \mathbf{x}_{u, p_m})^{\top})(p) \quad \text{for all } (u, s, v) \in E, \text{ and all } p \in P \quad (99)$$

which uses the variables  $\mathbf{X} = \{\mathbf{x}_{v, p} \mid v \in N, p \in P\}$ . The value of the variable  $\mathbf{x}_{v, p}$  is the bound on the template  $p$  at control-point  $v$ .

Because of Lemma 26, from  $\mathcal{C}$  we can construct a system  $\mathcal{E}$  of SDP-equations with  $\mu[\mathcal{E}] = \mu[\mathcal{C}]$  in polynomial time. Finally, we have:

**Lemma 28.**  *$V^{\mathcal{R}}[v](p) = \mu[\mathcal{E}](\mathbf{x}_{v, p})$  for all  $v \in N$  and all  $p \in P$ .*  $\square$

Since  $\mathcal{E}$  is a system of  $\vee$ -SDP-equations, by Theorem 1 and Theorem 3, we can compute the least solution  $\mu[\mathcal{E}]$  of  $\mathcal{E}$  using our  $\vee$ -strategy improvement algorithm. Thus, we have finally shown the following main result for the static program analysis application:

**Theorem 4.** *We can compute the relaxed abstract semantics  $V^{\mathcal{R}}$  of a program  $G = (N, E, \mathbf{st}, I)$  using our  $\vee$ -strategy improvement algorithm. Each  $\vee$ -strategy improvement step can be performed by performing  $|N| \cdot |P|$  Kleene iteration steps and solving  $|N| \cdot |P|$  SDP problems, each of which can be constructed in polynomial time. The number of strategy improvement steps is exponentially bounded by the product of the number of merge points in the program and the number of program variables.*  $\square$

*Example 26.* In order to give a complete picture of our method, we now discuss the harmonic oscillator example of Adjé et al. [1] in detail. The program consists only of the simple loop

$$\text{while ( true ) } x := Ax, \quad (100)$$

where  $x = (x_1, x_2)^\top \in \mathbb{R}^2$  is the vector of program variables and

$$A = \begin{pmatrix} 1 & 0.01 \\ -0.01 & 0.99 \end{pmatrix}. \quad (101)$$

We assume that the two-dimensional interval  $I = [0, 1] \times [0, 1]$  is the set of initial states. The set of control-points just consists of  $\mathbf{st}$ , i.e.  $N = \{\mathbf{st}\}$ . The set  $P = \{p_1, \dots, p_5\}$  of templates is given by

$$p_1(x_1, x_2) = -x_1 \quad p_2(x_1, x_2) = x_1 \quad p_3(x_1, x_2) = -x_2 \quad (102)$$

$$p_4(x_1, x_2) = x_2 \quad p_5(x_1, x_2) = 2x_1^2 + 3x_2^2 + 2x_1x_2 \quad (103)$$

The abstract semantics is thus given by the least solution of the following system of  $\vee$ -SDP-equations:

$$\mathbf{x}_{\mathbf{st}, p_1} = -\infty \vee 0 \vee \mathbf{SDP}_{A, a, \mathcal{B}, C_1}(\mathbf{x}_{\mathbf{st}, p_1}, \mathbf{x}_{\mathbf{st}, p_2}, \mathbf{x}_{\mathbf{st}, p_3}, \mathbf{x}_{\mathbf{st}, p_4}, \mathbf{x}_{\mathbf{st}, p_5}) \quad (104)$$

$$\mathbf{x}_{\mathbf{st}, p_2} = -\infty \vee 1 \vee \mathbf{SDP}_{A, a, \mathcal{B}, C_2}(\mathbf{x}_{\mathbf{st}, p_1}, \mathbf{x}_{\mathbf{st}, p_2}, \mathbf{x}_{\mathbf{st}, p_3}, \mathbf{x}_{\mathbf{st}, p_4}, \mathbf{x}_{\mathbf{st}, p_5}) \quad (105)$$

$$\mathbf{x}_{\mathbf{st}, p_3} = -\infty \vee 0 \vee \mathbf{SDP}_{A, a, \mathcal{B}, C_3}(\mathbf{x}_{\mathbf{st}, p_1}, \mathbf{x}_{\mathbf{st}, p_2}, \mathbf{x}_{\mathbf{st}, p_3}, \mathbf{x}_{\mathbf{st}, p_4}, \mathbf{x}_{\mathbf{st}, p_5}) \quad (106)$$

$$\mathbf{x}_{\mathbf{st}, p_4} = -\infty \vee 1 \vee \mathbf{SDP}_{A, a, \mathcal{B}, C_4}(\mathbf{x}_{\mathbf{st}, p_1}, \mathbf{x}_{\mathbf{st}, p_2}, \mathbf{x}_{\mathbf{st}, p_3}, \mathbf{x}_{\mathbf{st}, p_4}, \mathbf{x}_{\mathbf{st}, p_5}) \quad (107)$$

$$\mathbf{x}_{\mathbf{st}, p_5} = -\infty \vee 7 \vee \mathbf{SDP}_{A, a, \mathcal{B}, C_5}(\mathbf{x}_{\mathbf{st}, p_1}, \mathbf{x}_{\mathbf{st}, p_2}, \mathbf{x}_{\mathbf{st}, p_3}, \mathbf{x}_{\mathbf{st}, p_4}, \mathbf{x}_{\mathbf{st}, p_5}) \quad (108)$$

Here

$$\begin{aligned} \mathcal{A} &= \left( \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \right) \quad a = (1) \\ \mathcal{B} &= \left( \begin{pmatrix} 0 & -0.5 & 0 \\ -0.5 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0.5 & 0 \\ 0.5 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & -0.5 \\ 0 & 0 & 0 \\ -0.5 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0.5 \\ 0 & 0 & 0 \\ 0.5 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 3 \end{pmatrix} \right) \\ C_1 &= \begin{pmatrix} 0 & -0.5 & -0.005 \\ -0.5 & 0 & 0 \\ -0.005 & 0 & 0 \end{pmatrix} \quad C_2 = \begin{pmatrix} 0 & 0.5 & 0.005 \\ 0.5 & 0 & 0 \\ 0.005 & 0 & 0 \end{pmatrix} \\ C_3 &= \begin{pmatrix} 0 & 0.005 & -0.495 \\ 0.005 & 0 & 0 \\ -0.495 & 0 & 0 \end{pmatrix} \quad C_4 = \begin{pmatrix} 0 & -0.005 & 0.495 \\ -0.005 & 0 & 0 \\ 0.495 & 0 & 0 \end{pmatrix} \\ C_5 &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1.9803 & 0.9802 \\ 0 & 0.9802 & 2.9603 \end{pmatrix} \end{aligned}$$

In this example we have  $3^5 = 243$  different  $\vee$ -strategies. Assuming that the algorithm always chooses the best local improvement, in the first step it switch to the  $\vee$ -strategy that is given by the finite constants. At each equation, it then can switch to the **SDP**-expression, but then, because it constructs a strictly increasing sequence, it can never return to the constant. Summarizing, because of the simple structure, it is clear that our  $\vee$ -strategy improvement algorithm will perform at most 6  $\vee$ -strategy improvement steps. In fact our prototypical implementation performs 4  $\vee$ -strategy improvement steps on this example.  $\square$



## 7 Conclusion

We introduced and studied systems of  $\vee$ -morcave equations — a natural and strict generalization of systems of rational equations that were previously studied by Gawlitza and Seidl [5, 8]. We showed how the  $\vee$ -strategy improvement approach from Gawlitza and Seidl [5, 6] can be generalized to solve these fix-point equation systems. We provided full proves and a in-depth discussion on the different cases.

On the practical side, we showed that our algorithm can be applied to perform static program analysis w.r.t. quadratic templates using the relaxed abstract semantics of Adjé et al. [1] (based on Shor’s semi-definite relaxation schema). This analysis can, for instance, be used to verify linear recursive filters and numerical integration schemes. In the conference article that appears in the proceedings of the Seventeenth International Static Analysis Symposium (SAS 2010) we report on experimental results that were obtained through our proof-of-concept implementation [7].

For future work, we are interested in studying the use of other convex relaxation schemes to deal with more sophisticated cases, a problem already posed by Adjé et al. [1]. This would partially abolish the restriction to affine assignments and quadratic guards. Currently, we apply our  $\vee$ -strategy improvement algorithm only to numerical static analysis of programs. It remains to investigate in how far the  $\vee$ -strategy improvement algorithm we developed can be applied to other applications — maybe in other fields of computer science. Since our methods are solving quite general fixpoint problems, we have some hope that this is the case. Natural candidates could perhaps be found in the context of two-players zero-sum games.

## Bibliography

- [1] A. Adjé, S. Gaubert, and E. Goubault. Coupling policy iteration with semi-definite relaxation to compute accurate numerical invariants in static analysis. In A. D. Gordon, editor, *ESOP*, volume 6012 of *LNCS*, pages 23–42. Springer, 2010. ISBN 978-3-642-11956-9.
- [2] A. Costan, S. Gaubert, E. Goubault, M. Martel, and S. Putot. A Policy Iteration Algorithm for Computing Fixed Points in Static Analysis of Programs. In *Computer Aided Verification, 17th Int. Conf. (CAV)*, pages 462–475. LNCS 3576, Springer Verlag, 2005.
- [3] P. Cousot and R. Cousot. Static Determination of Dynamic Properties of Programs. In *Second Int. Symp. on Programming*, pages 106–130. Dunod, Paris, France, 1976.
- [4] P. Cousot and R. Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *POPL*, pages 238–252, 1977.
- [5] T. Gawlitza and H. Seidl. Precise relational invariants through strategy iteration. In J. Duparc and T. A. Henzinger, editors, *CSL*, volume 4646 of *LNCS*, pages 23–40. Springer, 2007. ISBN 978-3-540-74914-1.

- [6] T. Gawlitza and H. Seidl. Precise fixpoint computation through strategy iteration. In R. D. Nicola, editor, *ESOP*, volume 4421 of *LNCS*, pages 300–315. Springer, 2007. ISBN 978-3-540-71314-2.
- [7] T. M. Gawlitza and H. Seidl. Computing relaxed abstract semantics w.r.t. quadratic zones precisely. In R. Cousot and M. Martel, editors, *SAS*, volume 6337 of *Lecture Notes in Computer Science*, pages 271–286. Springer, 2010. ISBN 978-3-642-15768-4.
- [8] T. M. Gawlitza and H. Seidl. Solving systems of rational equations through strategy iteration. *ACM Trans. Program. Lang. Syst.*, 33(3):11, 2011.
- [9] T. M. Gawlitza, H. Seidl, A. Adjé, S. Gaubert, and É. Goubault. Abstract interpretation meets convex optimization. *WING-JSC*, 2011.
- [10] K. G. Larsen, F. Larsson, P. Pettersson, and W. Yi. Efficient verification of real-time systems: compact data structure and state-space reduction. In *IEEE Real-Time Systems Symposium*, pages 14–24. IEEE Computer Society, 1997.
- [11] A. Miné. A new numerical abstract domain based on difference-bound matrices. In O. Danvy and A. Filinski, editors, *PADO*, volume 2053 of *LNCS*, pages 155–172. Springer, 2001. ISBN 3-540-42068-1.
- [12] A. Miné. The octagon abstract domain. In *WCRE*, pages 310–, 2001.
- [13] A. Nemirovski. *Modern Convex Optimization*. Department ISYE, Georgia Institute of Technology, 2005.
- [14] J. Ortega and W. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. Academic Press, 1970.
- [15] S. Sankaranarayanan, H. B. Sipma, and Z. Manna. Scalable analysis of linear systems using mathematical programming. In R. Cousot, editor, *VMCAI*, volume 3385 of *LNCS*, pages 25–41. Springer, 2005. ISBN 3-540-24297-X.
- [16] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pac. J. Math.*, 5:285–309, 1955.
- [17] M. J. Todd. Semidefinite optimization. *Acta Numerica*, 10:515–560, 2001.
- [18] S. Yovine. Model checking timed automata. In G. Rozenberg and F. W. Vaandrager, editors, *European Educational Forum: School on Embedded Systems*, volume 1494 of *LNCS*, pages 114–152. Springer, 1996. ISBN 3-540-65193-4.