Precise Relational Invariants Through Strategy Iteration

Thomas Gawlitza and Helmut Seidl

TU München, Institut für Informatik, I2
85748 München, Germany
{gawlitza, seidl}@in.tum.de

Abstract. We present a practical algorithm for computing exact least solutions of systems of equations over the rationals with addition, multiplication with positive constants, minimum and maximum. The algorithm is based on strategy improvement combined with solving linear programming problems for each selected strategy. We apply our technique to compute the abstract least fixpoint semantics of affine programs over the relational template constraint matrix domain [20]. In particular, we thus obtain practical algorithms for computing the abstract least fixpoint semantics over the zone and octagon abstract domain.

1 Introduction

Abstract interpretation aims at inferring run-time invariants of programs [5]. Such an invariant may state, e.g., that a variable $\mathtt{x}$ is always contained in the interval $[2, 99]$ whenever a specific program point is reached. In order to compute such invariants, often an abstract semantics is considered which for each program point over-approximates the collecting semantics at this program point. Technically, the abstract semantics is given as the least fixpoint of an appropriate system of in-equations over a complete lattice. *Any* solution of this system provides safe information while only the *precision* of the information returned by the analysis depends on computing as small solutions as possible. In the example of interval analysis, clearly, the smaller the interval which the analysis returns for a given variable at a program point, the better is the information.

Thus, any ambitious program analyzer aims at computing *least* solutions of the systems of in-equations in question. Since ordinary fixpoint iteration does not provide a terminating algorithm in general, *widening* combined with *narrowing* has been proposed to accelerate fixpoint computation and thus guarantee termination of the analysis algorithm at a moderate loss of precision [7, 8]. Finding useful widening and narrowing operators, however, is a kind of a black art and it is not a priori clear whether the chosen heuristics will be sufficient for a given program. As an alternative to the general technique of widening and narrowing, we are interested in methods which allow to compute least solutions of in-equations *precisely* – at least for certain interesting cases.

Here, we are interested in computing precise abstract least fixpoint semantics of affine programs over a certain *relational* domain which enables us to describe (certain) relations between the values of program variables. Our key techniques refer to the *template constraint matrix (TCMs) abstract domain* introduced by Sankaranarayanan et al.

[20]. Polyhedra of a predefined fixed shape can be represented through elements of this domain. As a particular case, we obtain practical precise algorithms also for intervals, the zone abstract domain and octogons [16, 15].

The key idea for our precise algorithm for equations over rationals is *strategy iteration*. Recently, strategy iteration (called policy iteration in [4, 10]) has been suggested by Costan et al. as an alternative method for the widening and narrowing approach of Cousot and Cousot [7, 8] for computing (hopefully small) solutions of systems of in-equations. Originally, strategy iteration has been introduced by Howard for solving stochastic control problems [13, 19] and is also applied to zero-sum two player games [12, 18, 22] or fixpoints of min-max-plus systems [3]. In general, though, naive strategy iteration will only find some fixpoint — not necessarily the least one [4].

In [4] Costan et al. consider systems of equations over integer intervals. The authors then generalize their idea in [10] to the *zone- and octagon-domain* [16, 15] as well as to the *TCM domain* [20]. Their strategy iteration scheme can be applied to monotone self maps F satisfying a *selection property*. This selection property states that the self map F can be considered as the infimum of a set of simpler self maps. Then the selection property enables to compute a fixpoint of F by successively computing least fixpoints of the simpler maps. In certain cases, e.g., for non-expansive self maps on $\overline{\mathbb{R}}^n$, this approach returns the *least* fixpoint. In many practical cases, however, this cannot be guaranteed. In [11], we provide a practical algorithm for computing least solutions of (in-)equations over integer intervals. This algorithm crucially exploits the fact that the interval bounds are integers. Interestingly, it is not applicable to (in-)equations of intervals with rational bounds or multiplication with fractions such as 0.5.

In contrast to [4, 10] and similar to [11] we do not apply strategy iteration directly to systems of equations over the interval, the zone/octagon or the TCM domain. Instead, we design just one strategy improvement algorithm for computing least solutions of systems of rational equations. Technically, our algorithm in [11] relies on an *instrumentation* of the underlying lattice [11]. This instrumentation is no longer possible for rationals. Our main technical contribution therefore is to construct a precise strategy iteration *without* extra instrumentation. For solving the subsystems selected by a strategy, we use *linear programming* [14, 21]. Using a similar reduction as in [11] for integer intervals, systems of rational equations can be used for interval analysis with rational bounds. Because of lack of space, we do not present this reduction here. Instead, by additionally allowing a (monotone) *linear programming operator* in right-hand sides of equations, we use our techniques for computing abstract least fixpoint semantics of affine programs over the TCM domain. We emphasize that our methods return *precise* answers and do not rely on widening or narrowing. Using the simplex algorithm for solving the occurring linear programs, our algorithm is even uniform, i.e., the number of arithmetic operations does not depend on the sizes of occurring numbers.

The paper is organized as follows. Section 2 introduces systems of rational equations and basic notations. Section 3 presents our strategy improvement algorithm for systems of rational equations. Affine programs are discussed in section 4. There we show how to compute the abstract semantics over the TCM domain using systems of rational equations extended with linear programming operators. Solving these systems is discussed in section 5. Finally, we conclude with section 6.

2 Systems of Rational Equations

We are interested in computing least solutions of systems of equations over the rationals. Since the least upper bound of a bounded set of rationals need not be rational any longer, we consider the complete lattice $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$ of real numbers equipped with the natural ordering $\leq$ and extended with $-\infty$ as least and ∞ as greatest element. On $\overline{\mathbb{R}}$ we consider the operations addition, multiplication with positive constants, minimum "$\wedge$" and maximum "$\vee$" which are extended to operands "$-\infty$" and "∞" as follows. We set $x + (-\infty) = y \cdot (-\infty) = -\infty$ for $x \in \overline{\mathbb{R}}, y \geq 0$; we set $x + \infty = y \cdot \infty = \infty$ for $x \in \overline{\mathbb{R}}, y > 0$; and we set $0 \cdot x = 0$ for $x > -\infty$. For $c > 0$, the operations $+$ and $c\cdot$ distribute over $\vee$ and $\wedge$. Moreover $+$ distributes over $c\cdot$. A system of rational equations is a sequence $\mathbf{x}_1 = e_1, \ldots, \mathbf{x}_n = e_n$ of *rational equations* where $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are pairwise distinct variables, and the right-hand sides are expressions e' built up from constants and variables by means of addition, multiplication with positive constants, minimum "$\wedge$" and maximum "$\vee$". Thus, an expression is defined by the grammar

$$e' \ ::= \ a \mid \mathbf{x}_i \mid e'_1 + e'_2 \mid b \cdot e' \mid e'_1 \vee e'_2 \mid e'_1 \wedge e'_2$$

where $a \in \overline{\mathbb{Q}}$, $b \in \mathbb{Q}^{>0}$, $\mathbf{x}_i$ is a variable and e', e'_1, e'_2 are expressions. Note that all occurring constants are rationals. We call a system $\mathcal{E}$ of rational equations *conjunctive* (resp. *disjunctive*) iff no right-hand side of $\mathcal{E}$ contains the maximum-operator "$\vee$" (resp. minimum-operator "$\wedge$"). A system without occurrences of minimum and maximum operators is called *basic*. As usual, every expression e evaluates to a value $[\![e]\!]\mu \in \overline{\mathbb{R}}$ under a *variable assignment* $\mu : \mathbf{X} \to \overline{\mathbb{R}}$. Thus, e.g., $[\![e'_1 + e'_2]\!]\mu = [\![e'_1]\!]\mu + [\![e'_2]\!]\mu$ where e'_1, e'_2 are expressions. Assume that $\mathcal{E}$ denotes the system $\mathbf{x}_1 = e_1, \ldots, \mathbf{x}_n = e_n$ of rational equations. A variable assignment μ which satisfies all equations of $\mathcal{E}$, i.e., $\mu(\mathbf{x}_i) = [\![e_i]\!]\mu$ for $i = 1, \ldots, n$, is called a *solution* of $\mathcal{E}$. Accordingly, we call a variable assignment μ a *pre-solution* of $\mathcal{E}$ iff $\mu(\mathbf{x}_i) \leq [\![e_i]\!]\mu$ for $i = 1, \ldots, n$ and a *post-solution* of $\mathcal{E}$ iff $\mu(\mathbf{x}_i) \geq [\![e_i]\!]\mu$. A solution of $\mathcal{E}$ is a fixpoint of the function given through the right-hand sides of $\mathcal{E}$. Since every right-hand side e_i induces a monotonic function $[\![e_i]\!] : (\mathbf{X} \to \overline{\mathbb{R}}) \to \overline{\mathbb{R}}$, every system $\mathcal{E}$ of rational equations has a least solution. We write $\mu \ll \mu'$ iff $\mu(\mathbf{x}) < \mu'(\mathbf{x})$ for all variables $\mathbf{x}$. Moreover, we write $\underline{-\infty}$ (resp. $\underline{\infty}$) for the variable assignment which maps every variable to $-\infty$ (resp. ∞).

We remark, that least solutions of systems of rational equations cannot effectively be computed by performing ordinary Kleene fixpoint iteration. Even if the least solution is finite, infinitely many iterations may be necessary. A simple example is the equation $\mathbf{x} = 0.5 \cdot \mathbf{x} + 1 \vee 0$, whose least solution maps $\mathbf{x}$ to 2.

As a start, we consider disjunctive systems of rational equations. We recall from [10] that computing the least solution for such a system can be reduced to solving linear programs (LPs). For a set S and a matrix $A \in S^{m \times n}$, we write $A_{i\cdot}$ for the i-th row of A and $A_{\cdot j}$ for the j-th column of A. Accordingly $A_{i \cdot j}$ denotes the element in row i and column j. As usual we identify $S^{m \times 1}$ with S^m. We denote the transposed of A by A^T. For $A \in \mathbb{R}^{m \times n}$ and $c \in \mathbb{R}^n$ we define the operator $LP_{A,c} : \overline{\mathbb{R}}^m \to \overline{\mathbb{R}}$ by

$$LP_{A,c}(b) = \bigvee \{c^T x \mid x \in \mathbb{R}^n, Ax \leq b\}$$

for $b \in \mathbb{R}^m$. This operator is monotone and represents a linear program. If the program is *infeasible*, i.e., $Ax \leq b$ for no x, $LP_{A,c}(b)$ returns $-\infty$. If the program is unbounded, i.e., for all $r \in \mathbb{R}$, $c^T x > r$ for some x satisfying $Ax \leq b$, $LP_{A,c}(b)$ returns ∞.

Our goal is to compute the least solution of a system $\mathcal{E}$ of disjunctive rational equations. For simplicity, we assume that all maximum operators in right-hand sides of $\mathcal{E}$ occur on top-level such as in:

$$\mathbf{x}_1 = \tfrac{1}{3}\mathbf{x}_2 + 3 \vee 1 \qquad \mathbf{x}_2 = 2\mathbf{x}_1 - 6 \vee 5\mathbf{x}_2 - 1$$

Assume that $\mathcal{E}$ has n variables and the least solution is given by μ^*. In the first step, we compute the set of variables $\mathbf{x}_i$ with $\mu^*(\mathbf{x}_i) = -\infty$. This can be done in time $\mathcal{O}(n \cdot |\mathcal{E}|)$ by performing n rounds of fixpoint iteration which results in a variable assignment μ with $\mu(\mathbf{x}) = -\infty$ iff $\mu^*(\mathbf{x}) = -\infty$ for all variables $\mathbf{x}$. Accordingly, the least solution of the example system returns values exceeding $-\infty$ for both $\mathbf{x}_1$ and $\mathbf{x}_2$.

Having determined the set of variables $\mathbf{x}_i$ with $\mu^*(\mathbf{x}_i) = -\infty$, we can remove these variables from our system of equations. Therefore, we now w.l.o.g. may assume that $\mu^* \gg \underline{-\infty}$. Also, we may assume that the constant $-\infty$ does not occur in $\mathcal{E}$. For a moment assume furthermore that $\mu^* \ll \underline{\infty}$. From the set of equations we can extract a set of constraints (here in-equations) which are satisfied exactly by all post-solutions of $\mathcal{E}$. In our example these are given by:

$$\mathbf{x}_1 \geq \tfrac{1}{3}\mathbf{x}_2 + 3 \qquad \mathbf{x}_1 \geq 1 \qquad \mathbf{x}_2 \geq 2\mathbf{x}_1 - 6 \qquad \mathbf{x}_2 \geq 5\mathbf{x}_2 - 1$$

Since $\underline{-\infty} \ll \mu^* \ll \underline{\infty}$, the least solution μ^* can be characterized as the (unique) vector $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ that represents a solution of the above constraints and for which $-(x_1 + \cdots + x_n)$ is maximal. Thus, x can be determined by solving the appropriate LP. In the example, this results in the vector $x = (3, 0)$.

In general, it might not be the case that $\mu^* \ll \underline{\infty}$. If this is not the case, the LP corresponding to the system $\mathcal{E}$ is not feasible. In order to deal with this case as well, we consider the variable dependency graph $G = (V, \to)$ of $\mathcal{E}$ where the set of vertices V is the set of variables and the set of edges $\to \subseteq V^2$ is the smallest set s.t. $\mathbf{x}_j \to \mathbf{x}_i$ iff $\mathbf{x}_i = e_i \in \mathcal{E}$ and $\mathbf{x}_j$ occurs in e_i. Since $\mu^* \gg \underline{-\infty}$ and $-\infty$ does not occur as a constant in $\mathcal{E}$, $[\![e]\!]\mu^* > -\infty$ for every subexpression occurring in $\mathcal{E}$. Thus, $\mu^*(\mathbf{x}_j) = \infty$ and $\mathbf{x}_j \to^* \mathbf{x}_i$ implies $\mu^*(\mathbf{x}_i) = \infty$. In particular if $\mu^*(\mathbf{x}_i) = \infty$ for some variable $\mathbf{x}_i$ of a strongly connected component (SCC), then $\mu^*(\mathbf{x}_j) = \infty$ for every variable $\mathbf{x}_j$ of the same SCC. Therefore, we proceed by processing one maximal SCC after the other. Thereby we start with a maximal SCC $G' = (V', \to')$ without in-going edges. The least solution of the subsystem of $\mathcal{E}$ described by G' can be computed using linear programming as sketched above. If the corresponding LP is infeasible, then $\mu^*(\mathbf{x}_i) = \infty$ for all variables $\mathbf{x}_i$ of the SCC and in fact for all variables $\mathbf{x}_i$ reachable from this SCC. The corresponding LP cannot be unbounded, since this would be a contradiction to $\mu^* \gg \underline{-\infty}$.

Having computed the values of all variables in the first maximal SCC, we replace all occurrences of these variables in the remaining equations by their values and proceed with another maximal SCC without in-going edges. In essence, this is the algorithm of [10] simplified for systems of rational constraints. Summarizing, we have:

Theorem 1 (Costan et al. 2007). *The least solution of a disjunctive system $\mathcal{E}$ of rational equations can be computed by solving linearly many LPs of polynomial sizes.* □

This theorem results in a polynomial algorithm if we apply interior point methods for solving the occurring LPs [14, 21, 1]. Note, however, that the run-time then crucially depends on the sizes of occurring numbers. At the danger of an exponential run-time in contrived cases, we can also rely on the simplex algorithm instead: the advantage of the latter algorithm is that its run-time is *uniform*, i.e., independent of the sizes of occurring numbers (given that arithmetic operations, comparison, storage and retrieval for numbers are counted for $\mathcal{O}(1)$).

3 Least Solutions of Systems of Rational Equations

In this section we provide techniques for computing least solutions of systems of rational equations. Our techniques are based on (max-) strategy improvement. Let $M_\vee(\mathcal{E})$ denote the set of all maximum subexpressions occurring in $\mathcal{E}$. A *(max-)strategy* π is a function mapping every expression $e_1 \vee e_2$ in $M_\vee(\mathcal{E})$ to one of the subexpressions e_1, e_2. Given a max-strategy π together with an expression e, we write $e\,\pi$ for the expression obtained by recursively replacing every maximum expression in $\mathcal{E}$ by the respective subexpression selected by π. Assuming that $\mathcal{E}$ is the system $\mathbf{x}_i = e_i, i = 1, \ldots, n$, we write $\mathcal{E}(\pi)$ for the system $\mathbf{x}_i = e_i\,\pi, i = 1, \ldots, n$. Thus $\mathcal{E}(\pi)$ is extracted from $\mathcal{E}$ via the strategy π. Note that $\mathcal{E}(\pi)$ is conjunctive.

Example 1. Consider the system $\mathcal{E}$ of rational equations given by the equation $\mathbf{x} = (2 \cdot \mathbf{x} - 2 \wedge 10) \vee 4$. Consider the max-strategy π which maps the top-level expression $(2 \cdot \mathbf{x} - 2 \wedge 10) \vee 4$ to the expression 4. Then the system $\mathcal{E}(\pi)$ of conjunctive equations is given by the equation $\mathbf{x} = 4$. □

Assume that μ^* denotes the least solution of the system $\mathcal{E}$ of rational equations. Our goal is to construct a strategy improvement algorithm for computing μ^*. The algorithm maintains a current max-strategy π and a current variable assignment μ. The current variable assignment μ is a pre-solution of $\mathcal{E}$ which is less than or equal to μ^*. For a current max-strategy π and a current variable assignment μ, the algorithm performs an accelerated least fixpoint computation on the system $\mathcal{E}(\pi)$ which starts with μ. This fixpoint computation results in a variable assignment μ' which is a a solution of $\mathcal{E}(\pi)$ and a pre-solution of $\mathcal{E}$ and moreover is still less than or equal to μ^*. If μ' is not a solution of $\mathcal{E}$, a new improved max-strategy π' is determined and the algorithm re-starts with π' as current max-strategy and μ' as current variable assignment. These steps are repeated until the least fixpoint of $\mathcal{E}$ is reached.

Given a current max-strategy π and a solution μ of $\mathcal{E}(\pi)$, we pursue the policy to improve π at all expressions $e_1' \vee e_2'$ where $[\![e_1' \vee e_2']\!]\mu > [\![(e_1' \vee e_2')\,\pi]\!]\mu$ simultaneously. Formally, we introduce an improvement operator $P_\vee$ by:

$$P_\vee(\pi, \mu)(e_1 \vee e_2) = \begin{cases} e_1 & \text{if } [\![e_1]\!]\mu > [\![e_2]\!]\mu \\ e_2 & \text{if } [\![e_1]\!]\mu < [\![e_2]\!]\mu \\ \pi(e_1 \vee e_2) & \text{if } [\![e_1]\!]\mu = [\![e_2]\!]\mu \end{cases}$$

Note that the strategy $P_\vee(\pi, \mu)$ differs from π only if μ is not a solution of $\mathcal{E}$.

Algorithm 1 Least Solution of The System $\mathcal{E}$ of Rational Equations

$\pi \leftarrow \pi_{-\infty}$; $\mu \leftarrow \underline{-\infty}$;

while (μ is not a solution of $\mathcal{E}$) {

  $\pi \leftarrow P_\vee(\pi, \mu)$; $\mu \leftarrow$ least solution of $\mathcal{E}(\pi)$ that is greater than or equal to μ;

}

return μ

Example 2. Consider the system $\mathcal{E}$ and the max-strategy π from example 1. Let μ denote the unique solution of $\mathcal{E}(\pi)$, i.e., $\mu(\mathbf{x}) = 4$. The variable assignment μ is less than or equal to the least solution of $\mathcal{E}$ and the max-strategy $\pi' := P_\vee(\pi, \mu) \neq \pi$ leads to the system $\mathcal{E}(\pi')$ given by the equation $\mathbf{x} = (2 \cdot \mathbf{x} - 2 \wedge 10)$. □

In order to formulate our strategy improvement algorithm, we do not consider the original system $\mathcal{E}$. Instead, we replace every equation $\mathbf{x}_i = e_i$ of $\mathcal{E}$ by $\mathbf{x}_i = e_i \vee -\infty$. For simplicity, we denote the resulting system again by $\mathcal{E}$. Our algorithm starts with the max-strategy that maps every top-level expression to $-\infty$. We denote this max-strategy by $\pi_{-\infty}$. Then, our strategy improvement algorithm is given as algorithm 1. Clearly, if algorithm 1 terminates, it returns a solution of $\mathcal{E}$. It returns the *least* one, since for every strategy π the least solution μ' of $\mathcal{E}(\pi)$ with $\mu' \geq \mu$ is less than or equal to the least solution μ'' of $\mathcal{E}$ with $\mu'' \geq \mu$. Therefore the value of the program variable μ is always less than or equal to μ^*.

Two things remain to be explained. First, we need an algorithm for computing the least solution μ' of a conjunctive system such as $\mathcal{E}(\pi)$ with $\mu' \geq \mu$ for a given variable assignment μ. Here, we will exploit that every μ to be considered is not arbitrary but a *consistent pre-solution* (see below) of $\mathcal{E}(\pi)$. Secondly, we must prove that every strategy π occurs only finitely often during the strategy iteration. Before going further, we illustrate algorithm 1 by an example.

Example 3. Consider the system $\mathcal{E}$ of rational equations shown on the right. Algorithm 1 computes the least solution μ^* using 4 max-strategies $\pi_1, \ldots, \pi_4$. The strategies π_i lead to the systems $\mathcal{E}(\pi_i)$ shown on the right. Let us consider the system $\mathcal{E}(\pi_3)$. The only solution maps every variable to 2. Thus, the improvement step leads to the system $\mathcal{E}(\pi_4)$ for which we must compute the least solution which maps every variable to values greater than or equal to 2. This solution maps $\mathbf{x}_1$ to 500 and $\mathbf{x}_2$ to 100 and is also the least solution of $\mathcal{E}$. □

$\mathcal{E}$	$\equiv$	$\mathbf{x}_1 = 0.8{\cdot}\mathbf{x}_1{+}\mathbf{x}_2 \vee 2 \vee -\infty$	$\mathbf{x}_2 = (\mathbf{x}_2{+}1 \wedge 100) \vee \mathbf{x}_1 \vee -\infty$
$\mathcal{E}(\pi_1)$	$\equiv$	$\mathbf{x}_1 = -\infty$	$\mathbf{x}_2 = -\infty$
$\mathcal{E}(\pi_2)$	$\equiv$	$\mathbf{x}_1 = 2$	$\mathbf{x}_2 = -\infty$
$\mathcal{E}(\pi_3)$	$\equiv$	$\mathbf{x}_1 = 2$	$\mathbf{x}_2 = \mathbf{x}_1$
$\mathcal{E}(\pi_4)$	$\equiv$	$\mathbf{x}_1 = 0.8{\cdot}\mathbf{x}_1{+}\mathbf{x}_2$	$\mathbf{x}_2 = \mathbf{x}_2{+}1 \wedge 100$

Assume that $\mathcal{E}$ denotes the conjunctive system $\mathbf{x}_i = e_i$, $i = 1, \ldots, n$ and that μ is a pre-fixpoint of $\mathcal{E}$. We define the set $\mathcal{D}_\mu(\mathcal{E})$ of *derived constraints* as the smallest set of constraints of the form $\mathbf{x} \leq e$ such that

- $\mathbf{x}_i \leq e' \in \mathcal{D}_\mu(\mathcal{E})$ whenever $\mathbf{x}_i = e_i$ with $\mu(\mathbf{x}_i) < \infty$ can be rewritten (using distributivity) into $\mathbf{x}_i = e' \wedge e''$ where e' does not contain $\wedge$-operators;
- $\mathbf{x}_i \leq \frac{1}{1-c} \cdot e \in \mathcal{D}_\mu(\mathcal{E})$ whenever $\mathbf{x}_i \leq c \cdot \mathbf{x}_i + e \in \mathcal{D}_\mu(\mathcal{E})$ where $0 < c < 1$; and
- $\mathbf{x}_i \leq c \cdot e' + e \in \mathcal{D}_\mu(\mathcal{E})$ whenever $\mathbf{x}_i \leq c \cdot \mathbf{x}_j + e \in \mathcal{D}_\mu(\mathcal{E})$ and $\mathbf{x}_j \leq e' \in \mathcal{D}_\mu(\mathcal{E})$.

Lemma 1. *Assume that μ is a pre-solution of the conjunctive system $\mathcal{E}$. Then $\mu(\mathbf{x}) \leq [\![e]\!]\mu$ for every $\mathbf{x} \leq e \in \mathcal{D}_\mu(\mathcal{E})$.* □

We call the pre-solution μ of $\mathcal{E}$ *($\mathcal{E}$-)consistent* iff

- $[\![e]\!]\mu = -\infty$ implies $e = -\infty$ for every expression e occurring in $\mathcal{E}$.
- $\mathcal{D}_\mu(\mathcal{E})$ does not contain a derived constraint $\mathbf{x}_i \leq c \cdot \mathbf{x}_i + e \in \mathcal{D}_\mu(\mathcal{E})$ with $c \geq 1$ and $\mu(\mathbf{x}_i) = [\![c \cdot \mathbf{x}_i + e]\!]\mu$. (We call such a constraint μ-critical).

Example 4. The pre-solution $\mu = \{\mathbf{x}_1 \mapsto 2, \mathbf{x}_2 \mapsto 2\}$ is *not* $\mathcal{E}$-consistent for the conjunctive system $\mathcal{E}$ given by the equations $\mathbf{x}_1 = 0.75 \cdot \mathbf{x}_1 + 0.25 \cdot \mathbf{x}_2$ and $\mathbf{x}_2 = 4 \cdot \mathbf{x}_1 - 6$, because $\mathbf{x}_1 \leq 1.75 \cdot \mathbf{x}_1 - 1.5 \in \mathcal{D}_\mu(\mathcal{E})$ and $\mu(\mathbf{x}_1) = 2 = [\![1.75 \cdot \mathbf{x}_1 - 1.5]\!]\mu$. However, the pre-solution $\mu' = \{\mathbf{x}_1 \mapsto 3, \mathbf{x}_2 \mapsto 4\}$ is $\mathcal{E}$-consistent. □

We claim that algorithm 1 computes least solutions μ' of $\mathcal{E}(\pi)$ with $\mu' \geq \mu$ for variable assignments μ which are consistent pre-solutions of $\mathcal{E}$, only. Since $\underline{-\infty}$ is a consistent pre-solution of $\mathcal{E}(\pi_{-\infty})$, this follows inductively using the following two lemmas.

Lemma 2. *Let $\mathcal{E}$ be a conjunctive system and μ be a consistent pre-solution of $\mathcal{E}$. Every pre-solution $\mu' \geq \mu$ of $\mathcal{E}$ is consistent.* □

Lemma 3. *Assume that $\mathcal{E}$ is a system, π a max-strategy, μ a consistent pre-solution of $\mathcal{E}(\pi)$ and $\pi' = P_\vee(\pi, \mu)$. Then μ is a consistent pre-solution of $\mathcal{E}(\pi')$.* □

It remains to provide a method for computing the least solution μ' with $\mu' \geq \mu$ of a conjunctive system $\mathcal{E}$ for a *consistent* pre-solution μ of $\mathcal{E}$.

3.1 Systems of Conjunctive Equations

In this subsection we consider conjunctive systems $\mathcal{E}$ of rational equations. Of a particular interest are *feasible* systems. We call $\mathcal{E}$ *feasible* iff there exists a consistent pre-solution $\mu \ll \underline{\infty}$ of $\mathcal{E}$. It turns out that feasible systems enjoy the property to have a *least consistent* solution. The main challenge and the goal of this section therefore is to derive a method for computing the least consistent solution of feasible systems. This method then will be used to compute the least solution μ' of $\mathcal{E}$ with $\mu' \geq \mu$ provided that μ is a consistent pre-solution of $\mathcal{E}$ with $\mu \ll \underline{\infty}$.

The restriction to consistent pre-solutions with $\mu \ll \underline{\infty}$ can be lifted as follows. Assume that μ denotes an arbitrary consistent pre-solution of $\mathcal{E}$. Let $\mathbf{X}^\infty$ be the set of variables $\mathbf{x}$ with $\mu(\mathbf{x}) = \infty$. Let $\mathcal{E}'$ denote the system obtained from $\mathcal{E}$ by (1) removing every equation $\mathbf{x} = e$ with $\mathbf{x} \in \mathbf{X}^\infty$ and (2) replacing every variable $\mathbf{x} \in \mathbf{X}^\infty$ by the constant ∞. Then $\mu|_{\mathbf{X} \setminus \mathbf{X}^\infty}$ is a consistent pre-solution of $\mathcal{E}'$ with $\mu|_{\mathbf{X} \setminus \mathbf{X}^\infty} \ll \underline{\infty}$ and thus the least solution of $\mathcal{E}'$ with $\mu' \geq \mu|_{\mathbf{X} \setminus \mathbf{X}^\infty}$ is the least consistent solution of $\mathcal{E}'$. Finally, the least solution μ^* of $\mathcal{E}$ with $\mu^* \geq \mu$ is then given by $\mu^*(\mathbf{x}) = \infty$ for $\mathbf{x} \in \mathbf{X}^\infty$ and $\mu^*(\mathbf{x}) = \mu'(\mathbf{x})$ for $\mathbf{x} \notin \mathbf{X}^\infty$. In the following, we only consider *feasible* systems of conjunctive rational equations. Furthermore, we assume that the constant $-\infty$ does not occur in the systems under consideration. In a first step we consider systems of *basic* equations, i.e., systems in which neither $\vee$ nor $\wedge$ occur. The following lemma implies that every feasible system of *basic* equations has a least consistent solution.

Lemma 4. *Assume that $\mathcal{E}$ is a feasible system of basic equations. Assume that $\mu \ll \underline{\infty}$ is a pre-solution of $\mathcal{E}$ and μ' a consistent solution of $\mathcal{E}$. Then $\mu \leq \mu'$.*

Proof. Assume that $\mathcal{E}$ denotes the system $\mathbf{x}_i = e_i$, $i = 1, \ldots, n$. We proceed by induction on the number of variables occurring in right-hand sides of $\mathcal{E}$. If no variable occurs in a right-hand side of $\mathcal{E}$, then μ' is the only solution of $\mathcal{E}$. Thus $\mu \leq \mu'$, since otherwise μ would not be a pre-solution of $\mathcal{E}$. For the induction step, consider an equation $\mathbf{x}_i = e_i$ of $\mathcal{E}$ where $\mathbf{x}_i$ occurs in a right-hand side e_j of $\mathcal{E}$.

*C*ase 1: e_i does not contain $\mathbf{x}_i$

We obtain a system $\mathcal{E}'$ from $\mathcal{E}$ by replacing all occurrences of $\mathbf{x}_i$ in right-hand sides with e_i. Since $\mathcal{D}_{\mu'}(\mathcal{E}') \subseteq \mathcal{D}_{\mu'}(\mathcal{E})$, μ' is a consistent solution of $\mathcal{E}'$. Since μ is also a pre-solution of $\mathcal{E}'$ and the system $\mathcal{E}'$ contains one variable less in right-hand sides we get $\mu \leq \mu'$ by induction hypothesis.

*C*ase 2: e_i contains $\mathbf{x}_i$

Using distributivity, we rewrite the equation $\mathbf{x}_i = e_i$ equivalently into

$$\mathbf{x}_i = c \cdot \mathbf{x}_i + e$$

where $c \in \mathbb{R}^{>0}$ and e does not contain $\mathbf{x}_i$. Then we obtain the systems $\mathcal{E}_1$ and $\mathcal{E}_2$ from $\mathcal{E}$ by replacing the equation $\mathbf{x}_i = c \cdot \mathbf{x}_i + e$ by $\mathbf{x}_i = \infty$ and $\mathbf{x}_i = \frac{1}{1-c} \cdot e$, respectively. Then we obtain systems $\mathcal{E}_1'$ and $\mathcal{E}_2'$ from $\mathcal{E}_1$ and $\mathcal{E}_2$ by replacing all occurrences of the variable $\mathbf{x}_i$ in right-hand sides with ∞ and $\frac{1}{1-c} \cdot e$, respectively.

First consider the case $c < 1$. Since μ' is consistent we get that $\mu'(\mathbf{x}_i) > -\infty$. Thus, $\mu'(\mathbf{x}_i) \in \{[\![\frac{1}{1-c} \cdot e]\!]\mu', \infty\}$. If $\mu'(\mathbf{x}_i) = \infty$, we conclude that, since $\mathcal{D}_{\mu'}(\mathcal{E}_1') \subseteq \mathcal{D}_{\mu'}(\mathcal{E}_1) \subseteq \mathcal{D}_{\mu'}(\mathcal{E})$, μ' is a consistent solution of $\mathcal{E}_1'$. Since μ is a pre-solution of $\mathcal{E}_1'$ and $\mathcal{E}_1'$ has at least one variable less in right-hand sides than $\mathcal{E}$, we get $\mu \leq \mu'$ by induction hypothesis. If $\mu'(\mathbf{x}_i) = [\![\frac{1}{1-c} \cdot e]\!]\mu'$, we conclude that since $\mathcal{D}_{\mu'}(\mathcal{E}_2') \subseteq \mathcal{D}_{\mu'}(\mathcal{E}_2) \subseteq \mathcal{D}_{\mu'}(\mathcal{E})$, μ' is a consistent solution of $\mathcal{E}_2'$. Since μ is a pre-solution of $\mathcal{E}_2'$ and $\mathcal{E}_2'$ has at least one variable less in right-hand sides than $\mathcal{E}$, we get $\mu \leq \mu'$ by induction hypothesis.

Now consider the case $c \geq 1$. Again, $\mu'(\mathbf{x}_i) > -\infty$, since μ' is consistent. It follows $\mu'(\mathbf{x}_i) = \infty$. Otherwise μ' would not be consistent, since then $\mu'(\mathbf{x}_i) = [\![c \cdot \mathbf{x}_i + e]\!]\mu'$ and thus $\mathbf{x}_i \leq c \cdot \mathbf{x}_i + e \in \mathcal{D}_{\mu'}(\mathcal{E})$ would be μ'-critical. Note that, since $\mathcal{D}_{\mu'}(\mathcal{E}_1') \subseteq \mathcal{D}_{\mu'}(\mathcal{E}_1) \subseteq \mathcal{D}_{\mu'}(\mathcal{E})$, μ' is a consistent solution of $\mathcal{E}_1'$. Since μ is a pre-solution of $\mathcal{E}_1'$ and $\mathcal{E}_1'$ has at least one variable less in right-hand sides than $\mathcal{E}$, we get $\mu \leq \mu'$ by induction hypothesis. □

We now extend this result to systems of conjunctive equations.

Lemma 5. *Assume that $\mathcal{E}$ is a feasible system of conjunctive equations. Assume that $\mu \ll \underline{\infty}$ is a pre-solution of $\mathcal{E}$ and μ' is a consistent solution of $\mathcal{E}$. Then $\mu \leq \mu'$. Moreover, there exists at most one consistent solution μ' with $\mu' \ll \underline{\infty}$.*

Proof. There exists a min-strategy (min-strategies are defined analog to max-strategies) π s.t. μ' is a consistent solution of the system $\mathcal{E}(\pi)$ of basic equations. Then $\mu \ll \underline{\infty}$ is a pre-solution of $\mathcal{E}(\pi)$ by monotonicity. Thus, $\mu \leq \mu'$ by lemma 4. In order to show the second statement, assume that $\mu' \ll \underline{\infty}$ and let $\mu'' \ll \underline{\infty}$ denote a consistent solution of $\mathcal{E}$. Then $\mu' \leq \mu''$ and $\mu'' \leq \mu'$ implying $\mu' = \mu''$. □

Using lemma 5 we conclude that every feasible conjunctive system has a least consistent solution. The following theorem states this fact and moreover observes that the least consistent solution is given by the least solution which is bounded below by a consistent pre-solution μ with $\mu \ll \underline{\infty}$.

Theorem 2. *Assume that $\mathcal{E}$ is a feasible conjunctive system, and $\mu \ll \underline{\infty}$ is a consistent pre-solution of $\mathcal{E}$. Then there exists a least consistent solution μ^* of $\mathcal{E}$ which equals the least solution μ' of $\mathcal{E}$ with $\mu' \geq \mu$.* □

In order to simplify complexity estimations, we state the following corollary explicitly.

Corollary 1. *Assume that $\mathcal{E}$ denotes a conjunctive system with n variables. Let $(\mu_i)_{i\in\mathbb{N}}$ denote an increasing sequence of consistent pre-solutions of $\mathcal{E}$. Let μ'_i denote the least solution of $\mathcal{E}$ with $\mu'_i \geq \mu_i$ for $i \in \mathbb{N}$. Then $|\{\mu'_i \mid i \in \mathbb{N}\}| \leq n$.* □

We now use the results above in order to compute the least consistent solution μ^* of the feasible conjunctive system $\mathcal{E}$. We first restrict our consideration to the case $\mu^* \ll \underline{\infty}$. Since, by lemma 5, μ^* is the *only* solution of $\mathcal{E}$ with $\mu \leq \mu^* \ll \underline{\infty}$, μ^* is in particular the *greatest* solution of $\mathcal{E}$ with $\mu^* \ll \underline{\infty}$. We compute μ^* by solving a linear program which maximizes the sum of the values of the variables occurring in $\mathcal{E}$. Assume w.l.o.g. that $\mathcal{E}$ is given by $\mathbf{x}_i = e_i^{(1)} \wedge \cdots \wedge e_i^{(k_i)}$ for $i = 1, \ldots, n$ where $e_i^{(j)}$ do not contain $\wedge$-operators, i.e., $\mathcal{E}$ is in normal form. (This form can be achieved from a general form in linear time by introducing at most $m_\wedge$ auxiliary variables and equations, where $m_\wedge$ denotes the number of $\wedge$-subexpressions.) We define $\mathcal{C}_\mathcal{E}$ as the following system of rational *constraints*:

$$\mathbf{x}_i \leq e_i^{(j)} \qquad \text{for } i = 1, \ldots, n,\ j = 1, \ldots, k_i.$$

Then we must maximize $\sum_{\mathbf{x}\in\mathbf{X}} \mu(\mathbf{x})$ under the restriction that μ is a solution of $\mathcal{C}_\mathcal{E}$.

Lemma 6. *Assume that $\mathcal{E}$ denotes a feasible conjunctive system and that $\mu^* \ll \underline{\infty}$ denotes the least consistent solution of $\mathcal{E}$. Then there exists a solution μ' of $\mathcal{C}_\mathcal{E}$ with $\mu' \ll \underline{\infty}$ which maximizes the sum $\sum_{\mathbf{x}\in\mathbf{X}} \mu'(\mathbf{x})$. Furthermore, $\mu' = \mu^*$. Thus, μ^* can be computed by solving a single LP which can be extracted from $\mathcal{E}$ in linear time.* □

Example 5. Consider the system $\mathcal{E}(\pi_4)$ from example 3. Our goal is to compute the least solution μ' with $\mu' \geq \mu = \{\mathbf{x}_1 \mapsto 2, \mathbf{x}_2 \mapsto 2\}$. Theorem 2 implies that μ' is given as the *least consistent solution*. Assuming that $\mu' \ll \underline{\infty}$, i.e., μ' maps all variables to finite values, lemma 6 implies that μ' is given as the *unique* solution of the LP

$$\bigvee\{\mathbf{x}_1 + \mathbf{x}_2 \mid \mathbf{x}_1 \leq 0.8 \cdot \mathbf{x}_1 + \mathbf{x}_2,\ \mathbf{x}_2 \leq \mathbf{x}_2 + 1,\ \mathbf{x}_2 \leq 100\}.$$

Thus, μ' maps $\mathbf{x}_1$ to 500 and $\mathbf{x}_2$ to 100. □

Until now, we can only deal with feasible systems $\mathcal{E}$ whose least consistent solution μ^* does not map any variable to ∞. In order to lift this restriction, we first have to determine the set $\mathbf{X}^{*\infty} := \{\mathbf{x} \in \mathbf{X} \mid \mu^*(\mathbf{x}) = \infty\}$. Given $\mathbf{X}^{*\infty}$ we can remove each equation $\mathbf{x}_i = e_i$ with $\mathbf{x}_i \in \mathbf{X}^{*\infty}$ and thus obtain a system whose least consistent solution $\mu^{*\prime}$ does not map any variable to ∞. Moreover $\mu * |_{\mathbf{X}\setminus\mathbf{X}^{*\infty}} = \mu^{*\prime}$.

We reduce the problem of determining $\mathbf{X}^{*\infty}$ to the problem of computing the greatest solution of an *abstracted* system of rational equations for which we know that the greatest solution does not map any variable to ∞ or $-\infty$. Therefore, this *abstracted* system can be solved again by linear programming. We first define a transformation $[\cdot]^\infty$ which maps the constant ∞ to 1 and every finite constant to 0 while preserving all multiplicative factors and variable occurrences (recall that $-\infty$ does not occur in the expressions under consideration):

$$\begin{array}{lll} [\mathbf{x}]^\infty = \mathbf{x} & [a]^\infty = 0 & [\infty]^\infty = 1 \\ [c \cdot e]^\infty = c \cdot [e]^\infty & [e_1 + e_2]^\infty = [e_1]^\infty + [e_2]^\infty & [e_1 \wedge e_2]^\infty = [e_1]^\infty \wedge [e_2]^\infty \end{array}$$

where $a < \infty$, $0 < c < \infty$, $\mathbf{x}$ is a variable and e, e_1, e_2 are expressions. Assuming that $\mathcal{E}$ denotes the system $\mathbf{x}_1 = e_1, \ldots, \mathbf{x}_n = e_n$ we write $[\mathcal{E}]^\infty$ for the system

$$\mathbf{x}_1 = [e_1]^\infty \wedge 1, \ldots, \mathbf{x}_n = [e_n]^\infty \wedge 1.$$

The next lemma states that the set $\mathbf{X}^{*\infty}$ can be read off the greatest solution μ^∞ of $[\mathcal{E}]^\infty$. Thus our problem reduces to computing μ^∞. Since by construction $0 \leq \mu^\infty(\mathbf{x}) \leq 1$ for every variable $\mathbf{x}$, this can be done using linear programming, i.e., we have to compute a solution μ^∞ of $\mathcal{C}_{\mathcal{E}^\infty}$ which maximizes the sum $\sum_{\mathbf{x} \in \mathbf{X}} \mu^\infty(\mathbf{x})$. There exists only one such solution and this solution is the greatest solution of $\mathcal{E}^\infty$. We have:

Lemma 7. *Assume that μ^* denotes the least consistent solution of the feasible conjunctive system $\mathcal{E}$. Let μ^∞ denote the greatest solution of $[\mathcal{E}]^\infty$. Then $\mu^*(\mathbf{x}) = \infty$ iff $\mu^\infty(\mathbf{x}) > 0$ for all variables $\mathbf{x}$. Furthermore, μ^∞ and thus $\{\mathbf{x} \in \mathbf{X} \mid \mu^*(\mathbf{x}) = \infty\}$ can be computed by solving a single LP which can be extracted from $\mathcal{E}$ in linear time.* □

Example 6. Consider again the system $\mathcal{E}(\pi_4)$ from example 3. As we already know, the system is feasible and we are interested in computing the least consistent solution μ^*. In order to compute the set of variables which μ^* maps to ∞, we construct the abstracted system $\quad x_1 = 0.8 \cdot x_1 + x_2 \wedge 1, \quad x_2 = x_2 \wedge 0 \wedge 1 \quad$ for which we must compute the greatest solution μ^∞. Then μ^∞ can be computed using linear programming. More exactly, μ^∞ is given as the *unique determined* solution which maximizes the sum $\sum_{\mathbf{x} \in \mathbf{X}} \mu^\infty(\mathbf{x})$. Here, obviously, μ^∞ maps every variable to 0. Thus, according to lemma 7, μ^* maps all variables to finite values — implying that the finiteness assumption in example 5 is justified. □

In conclusion, our method for computing the least consistent solution μ^* of a feasible conjunctive system $\mathcal{E}$ works as follows. Using lemma 7, we first determine the set $\mathbf{X}^{*\infty}$ of variables $\mathbf{x}$ with $\mu^*(\mathbf{x}) = \infty$. After that we obtain a system $\mathcal{E}'$ of conjunctive equations from $\mathcal{E}$ by (1) removing all equations $\mathbf{x} = e$ with $\mu^*(\mathbf{x}) = \infty$ and (2) replacing all expressions e with $[\![e]\!]\mu$ by ∞. Then $\mu^*|_{\mathbf{X} \setminus \mathbf{X}^{*\infty}}$ is the least consistent solution of $\mathcal{E}'$ and moreover $\mu^*|_{\mathbf{X} \setminus \mathbf{X}^{*\infty}} \ll \underline{\infty}$. By lemma 6, $\mu^*|_{\mathbf{X} \setminus \mathbf{X}^{*\infty}}$ and thus μ^* can be determined by solving an appropriate LP. We arrive at our result for feasible conjunctive systems:

Theorem 3. *The least consistent solution of a feasible conjunctive system $\mathcal{E}$ can be computed by solving two LPs each of which can be extracted from $\mathcal{E}$ in linear time.* □

3.2 The Result

Consider again algorithm 1. Assume w.l.o.g. that $\mathcal{E}$ denotes the system $\mathbf{x}_i = e_i \vee -\infty$, $i = 1, \ldots, n$ with least solution μ^* and $m_\vee = m + n$ $\vee$-expressions. In order to give a precise characterization of the run-time, let $\Pi(m_\vee)$ denote the maximal number of updates of strategies necessary for systems with $m_\vee$ maximum expressions.

Let π_i denote the max-strategy π after the execution of the first statement in the i-th iteration. Accordingly, let μ_i denote the variable assignment μ at this point and let μ_i' denote the variable assignment μ after the i-th iteration. It remains to show that algorithm 1 always terminates. Lemmas 2 and 3 imply that μ_i is a consistent pre-solution of $\mathcal{E}(\pi_i)$ with $\mu_i \leq \mu^*$ for every i. By theorem 3 μ_i' can be computed by solving two appropriate LP problems extracted from $\mathcal{E}$. The sequence (μ_i) is strictly increasing until the least solution is reached. Moreover, every strategy π is contained at most n times in the sequence (π_i). Otherwise, there would be more than n least solutions of the conjunctive system $\mathcal{E}(\pi)$ exceeding some consistent pre-solution contained in (μ_i). This would be a contradiction to corollary 1. Therefore, the number of iterations of the loop executed by algorithm 1 is bounded by $n \cdot \Pi(m + n)$. Summarizing, we have:

Theorem 4. *The least solution of a system $\mathcal{E}$ of rational equations with n variables and m maximum expressions can be computed by solving $2n \cdot \Pi(m+n)$ LPs each of which can be extracted from $\mathcal{E}$ in linear time.* □

All practical experiments with strategy iteration we know of seem to indicate that the number of strategy improvements $\Pi(m+n)$ (at least practically) grows quite slowly in the number of maximums m and the number of variables n. Interestingly, though, it is still open whether (or: under which circumstances) the trivial upper bound of 2^{m+n} for $\Pi(m + n)$ can be significantly improved [22, 2]. For a small improvement, we notice that for expressions $e_1 \vee e_2$ in which e_2 is an expression without variables, all strategies considered by algorithm 1 after e_1 evaluates to a greater value than e_2 will always select e_1. This in particular holds for the n $\vee$-expressions $e \vee -\infty$ at the top-level introduced in order to deal with $-\infty$. Thus, $\Pi(m_\vee + n)$ in our complexity estimation can be replaced with $n \cdot 2^{m_\vee}$.

4 Analyzing Affine Programs

In this section we discuss affine programs, their collecting semantics as well as their abstract semantics over the template constraint matrix domain [20] which subsumes the interval as well as the zone- and octagon domains [16, 15]. We use similar notations as in [17]. Let $\mathbf{X}_G = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be the set of variables the program operates on and let $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ denote the vector of variables. We assume that the variables take values in $\mathbb{R}$. Then in a concrete semantics a *state* assigning values to the variables is conveniently modeled by a vector $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$; x_i is the value assigned to variable $\mathbf{x}_i$. Note that we distinguish variables and their values by using a different font. Statements in affine programs are of the following forms:

$$(1) \quad \mathbf{x} := A\mathbf{x} + b \qquad (2) \quad \mathbf{x}_j :=? \qquad (3) \quad A\mathbf{x} + b \geq 0$$

where $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$. Statements of the form (1), (2) and (3) are called *affine assignments*, *non-deterministic assignments* and *guards*, respectively. Non-deterministic assignments are necessary to model input routines returning unknown values or variable assignments whose right-hand sides are not affine expressions. Such a statement may update $\mathbf{x}_i$ in the current state with any possible value. We denote the set of all statements by Stmt.

As common in flow analysis, we use the program's collecting semantics which associates a set of vectors $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ to each program point. Each statement $s \in \mathsf{Stmt}$ induces a transformation $[\![s]\!] : 2^{\mathbb{R}^n} \to 2^{\mathbb{R}^n}$, given by

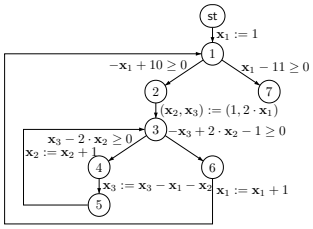
$$[\![\mathbf{x} := A\mathbf{x} + b]\!]X = \{Ax{+}b \mid x \in X\} \qquad [\![A\mathbf{x} + b \geq 0]\!]X = \{x \in X \mid Ax{+}b \geq 0\}$$
$$[\![\mathbf{x}_k :=?]\!]X \qquad = \{x{+}\delta\mathbf{1}_k \mid x \in X, \delta \in \mathbb{R}\}$$

for $X \subseteq \mathbb{R}^n$ where $\mathbf{1}_k$ denotes the vector whose components are zero beside the k-th component which is 1. The branching of an *affine program* is non-deterministic. Formally, an *affine program* is given by a *control flow graph* $G = (N, E, \mathsf{st})$ that consists of a set N of *program points*, a set $E \subseteq N \times \mathsf{Stmt} \times N$ of *(control flow) edges* and a special *start point* $\mathsf{st} \in N$. Then, the collecting semantics V is characterized as the least solution of the constraint system

$$\mathbf{V}[\mathsf{st}] \supseteq \mathbb{R}^n \qquad \mathbf{V}[v] \supseteq [\![s]\!](\mathbf{V}[u]) \quad \text{for each } (u, s, v) \in E$$

where the variables $\mathbf{V}[v]$, $v \in N$ take values in $2^{\mathbb{R}^n}$. We denote the components of the collecting semantics V by $V[v]$ for $v \in N$.

Example 7. Let $G = (N, E, \mathsf{st})$ denote the affine program shown on the right and let V denotes the collecting semantics of G. For simplicity, we do not use matrices in the control-flow graph. However, all statements can be considered as affine assignments and guards, respectively. The statement $(\mathbf{x}_2, \mathbf{x}_3) := (1, 2 \cdot \mathbf{x}_1)$, for instance, represents the affine assignment



$$\mathbf{x} := \begin{pmatrix} 1\,0\,0 \\ 0\,0\,0 \\ 2\,0\,0 \end{pmatrix} \mathbf{x} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

A program analysis could, for instance, aim to answer the question whether at program point 5 the program variable $\mathbf{x}_3$ takes values within the interval $[0, 9]$, only. Formally, this is the question whether $V[5] \subseteq \{(x_1, x_2, x_3) \mid 0 \leq x_3 \leq 9,\ x_1, x_2 \in \mathbb{R}\}$ — which is the case here. □

We now consider an abstract semantics which is an over-approximation of the *collecting semantics*. We assume that we are given a complete lattice $\mathbb{D}$ of abstract values (with partial ordering $\sqsubseteq$). Assume that we are given a function $\alpha_{\mathbb{D}} : 2^{\mathbb{R}^n} \to \mathbb{D}$ (the abstraction) and a function $\gamma_{\mathbb{D}} : \mathbb{D} \to 2^{\mathbb{R}^n}$ (the concretization) which form a Galois-connection. The elements in $\alpha_{\mathbb{D}}(2^{\mathbb{R}^n})$ are called *open* (see e.g. [9]). The best abstract transformer $[\![s]\!]^\sharp_{\mathbb{D}} : \mathbb{D} \to \mathbb{D}$ for a statement s (see, e.g., [6]) is given by

$$[\![s]\!]^\sharp_{\mathbb{D}} = \alpha_{\mathbb{D}} \circ [\![s]\!] \circ \gamma_{\mathbb{D}}.$$

In particular, $[\![s]\!]^\sharp_{\mathbb{D}}$ always returns open elements. We emphasize that we are concerned with *best abstract transformers* only. The *abstract semantics* $V^\sharp_{\mathbb{D}}$ of the affine program $G = (N, E, \mathsf{st})$ over $\mathbb{D}$ is given as the least solution of the system of constraints

$$\mathbf{V}^\sharp_{\mathbb{D}}[\mathsf{st}] \sqsupseteq \top_{\mathbb{D}} \qquad \mathbf{V}^\sharp_{\mathbb{D}}[v] \sqsupseteq [\![s]\!]^\sharp(\mathbf{V}^\sharp_{\mathbb{D}}[u]) \quad \text{for each } (u, s, v) \in E$$

where the variables $\mathbf{V}^\sharp_{\mathbb{D}}[v]$, $v \in N$ take values in $\mathbb{D}$ and $\top_{\mathbb{D}}$ denotes the greatest element of $\mathbb{D}$. We denote the components of the abstract semantics $V^\sharp_{\mathbb{D}}$ by $V^\sharp_{\mathbb{D}}[v]$ for $v \in N$. $V^\sharp_{\mathbb{D}}$ represents an over-approximation of the collecting semantics V [7], i.e., $V^\sharp_{\mathbb{D}}[v] \sqsupseteq \alpha_{\mathbb{D}}(V[v])$ and $\gamma_{\mathbb{D}}(V^\sharp_{\mathbb{D}}[v]) \supseteq V[v]$ for every $v \in N$. Since every transformer $[\![s]\!]^\sharp_{\mathbb{D}}$ always returns open elements, we deduce from the theory of Galois-connections (see e.g. [9]) that $V^\sharp_{\mathbb{D}}[v], v \in N$ are open.

In this paper we consider the complete lattice introduced in [20]. For that, we consider a fixed *template constraints matrix* $T \in \mathbb{R}^{m\times n}$. Each row in this matrix represents a linear combination of variables of interest. Special cases of this domain are intervals, zones and octagons [16, 15, 20]. All these domains represent subclasses of convex polyhedra in the vector space $\mathbb{R}^n$ (n the number of variables). Let us w.l.o.g. assume that T does not contain rows consisting of zeros only. The set $\mathcal{T}_T := \overline{\mathbb{R}}^m$ together with the component-wise partial ordering $\leq$ forms a complete lattice. The *concretization* $\gamma_{\mathcal{T}_T} : \mathcal{T}_T \to 2^{\mathbb{R}^n}$ and the *abstraction* $\alpha_{\mathcal{T}_T} : 2^{\mathbb{R}^n} \to \mathcal{T}_T$ are defined by

$$\gamma_{\mathcal{T}_T}(c) = \{x \in \mathbb{R}^n \mid Tx \leq c\} \qquad \alpha_{\mathcal{T}_T}(X) = \bigwedge\{c \in \overline{\mathbb{R}}^m \mid \gamma_{\mathcal{T}_T}(c) \supseteq X\}$$

for $c \in \overline{\mathbb{R}}^m, X \subseteq \mathbb{R}^n$. As shown in [20], $\alpha_{\mathcal{T}_T}$ and $\gamma_{\mathcal{T}_T}$ form a Galois-connection. Thus, the abstract semantics $V^\sharp_{\mathcal{T}_T}$ of an affine program $G = (N, E, \mathsf{st})$ is well-defined.

In [20] the author allows one template constraint matrix for each program point. For simplicity and similar to [10], we consider one global template constraint matrix only. Note also that open elements of $\mathcal{T}_T$ are called *canonical* in [20].

We now show how to compute the abstract semantics $V^\sharp_{\mathcal{T}_T}$ of the affine program $G = (N, E, \mathsf{st})$ which uses variables $\mathbf{X}_G = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. First of all we have to describe the abstract effect $[\![s]\!]^\sharp_{\mathcal{T}_T}$ for each statement s by a linear program. We have:

Lemma 8. *Let $c \in \mathcal{T}_T$, $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ and $i = 1, \ldots, m$. Then:*

1. $([\![\mathbf{x} := A\mathbf{x} + b]\!]^\sharp_{\mathcal{T}_T}\, c)_{i\cdot} = T_{i\cdot}b + LP_{T,(T_{i\cdot}A)^T}(c)$
2. $([\![A\mathbf{x} + b \geq 0]\!]^\sharp_{\mathcal{T}_T}\, c)_{i\cdot} = LP_{A',T^T_{i\cdot}}(c')$ *where* $A' := \begin{pmatrix} T \\ -A \end{pmatrix}$ *and* $c' := \begin{pmatrix} c \\ b \end{pmatrix}$.
3. $[\![\mathbf{x}_k :=?]\!]^\sharp_{\mathcal{T}_T}\, c \leq forget_{T,k} + c$. *Moreover* $[\![\mathbf{x}_k :=?]\!]^\sharp_{\mathcal{T}_T}\, c = forget_{T,k} + c$ *whenever c is open. Thereby the vector $forget_{T,k} \in \mathcal{T}_T$ is defined by*

$$(forget_{T,k})_{i\cdot} = \begin{cases} \infty & \text{if } T_{i\cdot k} \neq 0 \\ 0 & \text{if } T_{i\cdot k} = 0. \end{cases}$$

□

Note that the post operator in [20] combines an affine assignment and a guard. In order to compute the abstract semantics $V^\sharp_{\mathcal{T}_T}$ of G over $\mathcal{T}_T$, we rely on our methods for

systems of rational equations presented in section 3. We additionally allow the LP operator to occur in right-hand sides, i.e., we additionally allow subexpressions of the form: $LP_{A,b}(e_1, \ldots, e_m)$ where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^n$ and e_i are expressions. We call such expressions and equations *with LP*. We define:

$$[\![LP_{A,b}(e_1, \ldots, e_m)]\!]\mu = LP_{A,b}\left(([\![e_1]\!]\mu, \ldots, [\![e_m]\!]\mu)^T\right)$$

Since again all operators in expressions with LP are monotone, every system of rational equations with LP has a least solution. For the computation of $V^\sharp_{\mathcal{T}_T}$, we construct a system $\mathcal{C}_G$ of rational constraints with LP which uses variables $\mathbf{X} = \{\mathbf{x}_{v,i} \mid v \in N, i = 1, \ldots, m\}$ (m is the number of rows of T) as follows. For the start point st of the affine program we introduce the constraints $\mathbf{x}_{\mathsf{st},i} \geq \infty$ for $i = 1, \ldots, m$. According to lemma 8 we introduce a constraint for every control flow edge $(u, s, v) \in E$ and every $i = 1, \ldots, m$ as shown in the following table.

control flow edge	constraint
$(u,\ \mathbf{x} := A\mathbf{x} + b,\ v)$	$\mathbf{x}_{v,i} \ \geq\ T_{i\cdot}b + LP_{T_{\cdot}, (T_{i\cdot}A)^T}(\mathbf{x}_{u,1}, \ldots, \mathbf{x}_{u,m})$
$(u,\ A\mathbf{x} + b \geq 0,\ v)$	$\mathbf{x}_{v,i} \ \geq\ LP_{\begin{pmatrix} T \\ -A \end{pmatrix}, T_{i\cdot}^T}(\mathbf{x}_{u,1}, \ldots, \mathbf{x}_{u,m}, b_{1\cdot}, \ldots, b_{n\cdot})$
$(u,\ \mathbf{x}_k :=?,\ v)$	$\mathbf{x}_{v,i} \ \geq\ (forget_{T,k})_{i\cdot} + \mathbf{x}_{u,i}$

The correctness follows from lemma 8 and the fact that $V^\sharp_{\mathcal{T}_T}[v], v \in N$ are open.

Theorem 5. *Let $V^\sharp_{\mathcal{T}_T}$ be the abstract semantics of the affine program $G = (N, E, \mathsf{st})$ over $\mathcal{T}_T$ and let μ^* be the least solution of the corresponding system $\mathcal{C}_G$ of rational constraints with LP. Then $(V^\sharp_{\mathcal{T}_T}[v])_{i\cdot} = \mu^*(\mathbf{x}_{v,i})$ for $v \in N, i = 1, \ldots, m$.* □

Example 8.

Let V denote the collecting semantics of the affine program $G = (N, E, \mathsf{st})$ of example 7. For our analysis we choose the set of constraints shown on the right which lead to the template constraint matrix T. Our goal is to determine for every program point v a vector $(c_1, \ldots, c_8)$ which is as small as possible and for which every vector $(x_1, x_2, x_3) \in V[v]$ fulfills the constraints. Let us consider the edge $(1, -\mathbf{x}_1 + 10 \geq 0, 2) \in E$ which is an abbreviation for $(1, A\mathbf{x} + b \geq \mathbf{0}, 2)$ (using the matrices above). This edge leads amongst others to the constraint

set of constraints:

$$\begin{aligned} x_1 &\leq c_1 \\ -x_1 &\leq c_2 \\ 2x_2 &\leq x_3 + c_3 \\ -x_2 &\leq c_4 \\ x_3 &\leq 2x_1 + c_5 \\ -x_3 &\leq -2x_1 + c_6 \\ x_3 &\leq c_7 \\ -x_3 &\leq c_8 \end{aligned}$$

$$T = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 2 & -1 \\ 0 & -1 & 0 \\ -2 & 0 & 1 \\ 2 & 0 & -1 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{pmatrix} \quad A = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad b = \begin{pmatrix} 10 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{x}_{2,1} \geq LP_{\begin{pmatrix} T \\ -A \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}}(\mathbf{x}_{1,1}, \mathbf{x}_{1,2}, \ldots, \mathbf{x}_{1,8}, 10, 0, 0)$$

Here, for instance, evaluating the right-hand side of the constraint above under the variable assignment $\underline{\infty}$ results in the value 10. Finally, the whole system $\mathcal{C}_G$ describes the abstract semantics $V^\sharp_{\mathcal{T}_T}$. Here, in particular, $(V^\sharp_{\mathcal{T}_T}[5])_{7\cdot} = 9$ and $(V^\sharp_{\mathcal{T}_T}[5])_{8\cdot} = 0$ which means that the value of the program variable $\mathbf{x}_3$ is between 0 and 9 at program point 5. This result is optimal and could not be established using interval analysis. □

By theorem 5, our problem reduces to computing least solutions of systems of rational equations *with LP*. Such systems will be discussed in the next section.

5 Systems of Rational Equations with LP

Our goal is to apply algorithm 1 also for computing the least solution μ^* of a system $\mathcal{E}$ of rational equations *with LP*. In order to use the results from section 3, we state the following lemma which can be shown using the duality theorem for linear programming.

Lemma 9. *Let $A \in \mathbb{R}^{m\times n}$ with $A_{i\cdot} \neq (0,\ldots,0)$ for $i = 1,\ldots,m$ and $b \in \mathbb{R}^n$. There exists a finite (possibly empty) set $mult(LP_{A,b}) = \{y_1,\ldots,y_k\} \subseteq \mathbb{R}^m$ with $y_1,\ldots,y_k \geq 0$ and $A^T y_1 = \cdots = A^T y_k = b$ such that for every $c \in \overline{\mathbb{R}}^m$ with $LP_{A,b}(c) > -\infty$ it holds that $LP_{A,b}(c) = \bigwedge_{y\in mult(LP_{A,b})} c^T y$.* □

We also extent the definition of $\mathcal{E}(\pi)$ for a strategy π and the definition of the improvement operator $P_\vee$ in the natural way. Moreover, for a *conjunctive* system $\mathcal{E}$, we extend the notion of *consistency* to a notion of *LP-consistency*.

In order to define *LP-consistency* let us, for every $LP_{A,b}$-operator, fix a finite set $mult(LP_{A,b})$ of vectors which satisfies the claims of lemma 9. Let $\mathcal{E}$ denote a conjunctive system of rational equations with LP. We first define the transformation $[\cdot]$ by:

$$[a] = a \quad [\mathbf{x}] = \mathbf{x} \quad [e_1 + e_2] = [e_1] + [e_2] \quad [c \cdot e] = c \cdot [e] \quad [e_1 \wedge e_2] = [e_1] \wedge [e_2]$$
$$[LP_{A,b}(e_1,\ldots,e_m)] = \textstyle\bigwedge_{y\in mult(LP_{A,b})}([e_1],\ldots,[e_m])y$$

where $a \in \overline{\mathbb{R}}$, $c \in \mathbb{R}^{>0}$, $\mathbf{x}$ is a variable and e_i are expressions. Thereby $([e_1],\ldots,[e_m])y$ denotes the expression $y_{1\cdot}\cdot[e_1]+\cdots+y_{m\cdot}\cdot[e_m]$ and we assume that an expression $0\cdot e_i$ is simplified to 0 (This is correct, since e_i does not evaluate to $-\infty$ in the cases which have to be considered). Assuming that $\mathcal{E}$ denotes the system $\mathbf{x}_i = e_i, i = 1,\ldots,n$, we write $[\mathcal{E}]$ for the system $\mathbf{x}_i = [e_i], i = 1,\ldots,n$. Then, we call a pre-solution μ of $\mathcal{E}$ *LP-consistent* iff $[\![LP_{A,b}(e_1,\ldots,e_m)]\!]\mu > -\infty$ for every subexpression $LP_{A,b}(e_1,\ldots,e_m)$ and μ is a consistent pre-solution of $[\mathcal{E}]$.

We have to ensure that μ will be a LP-consistent pre-solution of $\mathcal{E}$ whenever algorithm 1 computes the least solution μ' of $[\mathcal{E}]$ with $\mu' \geq \mu$. This is fulfilled, since lemmas 2 and 3 can be formulated literally identical for LP-consistency instead of consistency.

Assume that μ is a LP-consistent pre-solution of $\mathcal{E}$. It remains to compute the least solution μ' of $\mathcal{E}$ with $\mu' \geq \mu$. Since $\mathcal{E}$ is LP-consistent and thus in particular $[\![LP_{A,b}(e_1,\ldots,e_m)]\!]\mu > -\infty$ for every subexpression $LP_{A,b}(e_1,\ldots,e_m)$, lemma 9 implies that μ' is the least solution of $[\mathcal{E}]$ with $\mu' \geq \mu$. Since $[\mathcal{E}]$ denotes a conjunctive system *without LP*, we can compute it and moreover corollary 1 implies that every conjunctive system $\mathcal{E}$ is considered at most n times in algorithm 1. We find:

Lemma 10. *Assume that $\mathcal{E}$ denotes a conjunctive system with LP which uses n variables and m $\vee$-expressions. Algorithm 1 computes at most $n \cdot \Pi(m+n)$ times the least solution μ' of $\mathcal{E}(\pi)$ with $\mu' \geq \mu$ for some π and some LP-consistent pre-solution μ of $\mathcal{E}(\pi)$. After that, it returns the least solution of $\mathcal{E}$.* □

We want to compute the least solution μ' of $\mathcal{E}$ with $\mu' \geq \mu$ which is also the least solution of $[\mathcal{E}]$ with $\mu' \geq \mu$. Recall from section 3 that for this purpose we essentially have to compute least consistent solutions of feasible systems of conjunctive equations. Writing down the system $[\mathcal{E}]$ explicitly and solving it after this would be too inefficient.

Therefore we aim at computing the least consistent solution of the feasible system $[\mathcal{E}]$ without explicit representation. For that purpose, assume w.l.o.g. that $\mathcal{E}$ is given by

$$\begin{array}{ll} \mathbf{x}_i = e_i^{(1)} \wedge \cdots \wedge e_i^{(k_i)} & \text{for } i = 1, \ldots, n' \\ \mathbf{x}_i = LP_{A_i,b_i}(\mathbf{x}'_1, \ldots, \mathbf{x}'_{m_i}) & \text{for } i = n'+1, \ldots, n \end{array}$$

where the $e_i^{(j)}$ do neither contain $\wedge$- nor $LP_{A,b}$-operators. This form can be achieved by introducing variables. Recall from section 3 that the system $\mathcal{C}_{[\mathcal{E}]}$ is given by

$$\begin{array}{ll} \mathbf{x}_i \leq [e_i^{(j)}] & \text{for } i = 1, \ldots, n',\ j = 1, \ldots, k_i \\ \mathbf{x}_i \leq \bigwedge_{y \in mult(LP_{A_i,b_i})} (\mathbf{x}'_1, \ldots, \mathbf{x}'_{m_i}) y & \text{for } i = n'+1, \ldots, n \end{array}$$

We define the system $\mathcal{C}_{\mathcal{E}}^{LP}$ of rational constraints as the system:

$$\begin{array}{ll} \mathbf{x}_i \leq e_i^{(j)} & \text{for } i = 1, \ldots, n',\ j = 1, \ldots, k_i \\ \mathbf{x}_i \leq LP_{A_i,b_i}(\mathbf{x}'_1, \ldots, \mathbf{x}'_{m_i}) & \text{for } i = n'+1, \ldots, n \end{array}$$

Using lemma 9 we conclude that the sets of solutions μ' with $\mu' \geq \mu$ of $\mathcal{C}_{\mathcal{E}}^{LP}$ and of $\mathcal{C}_{[\mathcal{E}]}$ are equal. In particular, the sets of solutions μ' with $\mu' \geq \mu$ which maximize the sum $\sum_{\mathbf{x} \in \mathbf{X}} \mu'(\mathbf{x})$ are equal.

As in section 3 we first assume that the least consistent solution μ^* of $\mathcal{E}$ does not map any variable to ∞. In this situation, the above considerations and lemma 6 implies, that μ^* is the uniquely determined solution of $\mathcal{C}_{\mathcal{E}}^{LP}$ which maximizes the sum $\sum_{\mathbf{x} \in \mathbf{X}} \mu^*(\mathbf{x})$. In order to compute it using linear programming, we have to eliminate all occurrences of $LP_{A,b}$-operators. Therefore, consider a constraint

$$\mathbf{x} \leq LP_{A,b}(\mathbf{x}_1, \ldots, \mathbf{x}_m)$$

occurring in $\mathcal{C}_{\mathcal{E}}^{LP}$. Using the definition of the $LP_{A,b}$-operator we can conceptually replace the right-hand side with $\bigvee \{b^T y \mid y \in \mathbb{R}^n, Ay \leq (\mathbf{x}_1, \ldots, \mathbf{x}_m)^T\}$. Since we are interested in maximizing the value of the variable $\mathbf{x}$ anyway we replace the above constraint with the constraints

$$\mathbf{x} \leq b_{1\cdot} \cdot \mathbf{y}_1 + \cdots + b_{n\cdot} \cdot \mathbf{y}_n, \qquad A_{i\cdot 1} \cdot \mathbf{y}_1 + \cdots + A_{i\cdot n} \cdot \mathbf{y}_n \leq \mathbf{x}_i \quad \text{for } i = 1, \ldots, m$$

where $\mathbf{y}_1, \ldots, \mathbf{y}_n$ are fresh variables. This replacement step preserves the solution μ^* which maximizes the sum $\sum_{\mathbf{x} \in \mathbf{X}} \mu^*$. Doing this for every $LP_{A,b}$-expression in $\mathcal{E}$ we obtain a system of constraints without $LP_{A,b}$-expressions. Thus, we can compute μ^* by linear programming. We have:

Lemma 11. *Assume that $\mu \ll \underline{\infty}$ is a LP-consistent pre-solution of the conjunctive system $\mathcal{E}$ with LP. Assume that $\mu^* \ll \underline{\infty}$ is the least consistent solution of $[\mathcal{E}]$. Then μ^* can be computed by solving one LP which can be extracted from $\mathcal{E}$ in linear time.* □

Until now we have assumed that the least consistent solution μ^* of $[\mathcal{E}]$ maps every variable to values strictly smaller then ∞. As in section 3, we have to identify the variables $\mathbf{x}$ with $\mu^*(\mathbf{x}) = \infty$ in order to lift this restriction. For that, by lemma 7, we must

compute the greatest solution μ^∞ of the system $[\ [\mathcal{E}]\ ]^\infty$. For this purpose we extend the abstraction $[\cdot]^\infty$ by setting $[LP_{A,b}(e_1,\ldots,e_m)]^\infty = LP_{A,b}([e_1]^\infty,\ldots,[e_m]^\infty)$. It turns out that μ^∞ is also the greatest solution of $[\mathcal{E}]^\infty$. Since μ^∞ maps every variable to a finite value, μ^∞ is the only solution finite solution of $\mathcal{C}_{[\mathcal{E}]^\infty}$ which maximizes the sum $\sum_{\mathbf{x}\in\mathbf{X}} \mu^\infty(\mathbf{x})$. Thus, μ^∞ can again be computed using linear programming. Since we can identify the set $\{\mathbf{x}\in\mathbf{X} \mid \mu^*(\mathbf{x}) = \infty\}$ in this way, we can lift the restriction to systems with finite least consistent solutions in lemma 11. We have:

Lemma 12. *Assume that $\mu \ll \underline{\infty}$ denotes a LP-consistent pre-solution of the conjunctive system $\mathcal{E}$ with LP. Let μ^* denote the least consistent solution of $[\mathcal{E}]$. Then μ^* can be computed by solving two LP problems which can be extracted from $\mathcal{E}$ in linear time.* □

In conclusion, we obtain our main result for systems of rational equations with LP:

Theorem 6. *The least solution of a system $\mathcal{E}$ of rational equations with LP which uses n variables and m maximum expressions can be computed by solving $3n \cdot \Pi(m+n)$ LPs each of which can be extracted from $\mathcal{E}$ in linear time.* □

Finally, combining theorem 5 and 6, we derive our main result for the analysis of affine programs:

Theorem 7. *Assume that $G = (N, E, \mathsf{st})$ denotes an affine program. Let $T \in \mathbb{R}^{m\times n}$, $indeg(v) := \{(u,s,v') \in E \mid v' = v\}$ and $m_\vee := m \cdot \sum_{v\in N} max(indeg(v) - 1, 0)$. The abstract fixpoint semantics of G over $\mathcal{T}_T$ can be computed by solving at most $3m|N| \cdot \Pi(m_\vee + m|N|)$ LPs.* □

It remains to emphasize that all least solutions (resp. abstract semantics) computed by our methods are rational whenever all numbers occurring in the input are rational.

6 Conclusion

We presented a practical strategy improvement algorithm for computing exact least solutions of systems of equations over the rationals with addition, multiplication with positive constants, maximum and minimum. The algorithm is based on strategy improvement combined with LP solving for each selected strategy where each strategy can be selected only linearly often. We extended the method in order to deal a special LP-operator in right-hand sides of equations. We applied our techniques to compute the abstract least fixpoint semantics of affine programs over the template constraint matrix domain. In particular, we thus obtain practical algorithms for dealing with zones and octagons. It remains for future work to experiment with practical implementations of the proposed approaches.

References

1. GNU Linear Programming Kit. Technical report. http://www.gnu.org/software/glpk.
2. Henrik Bjorklund, Sven Sandberg, and Sergei Vorobyov. Complexity of Model Checking by Iterative Improvement: the Pseudo-Boolean Framework . In *Proc. 5th Int. Andrei Ershov Memorial Conf. Perspectives of System Informatics*, pages 381–394. LNCS 2890, Springer, 2003.

3. Jean Cochet-Terrasson, Stéphane Gaubert, and Jeremy Gunawardena. A Constructive Fixed Point Theorem for Min-Max Functions. *Dynamics and Stability of Systems*, 14(4):407–433, 1999.
4. Alexandru Costan, Stephane Gaubert, Eric Goubault, Matthieu Martel, and Sylvie 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.
5. P. Cousot and R. Cousot. Static Determination of Dynamic Properties of Recursive Procedures. In E.J. Neuhold, editor, *IFIP Conf. on Formal Description of Programming Concepts*, pages 237–277. North-Holland, 1977.
6. P. Cousot and R. Cousot. Systematic Design of Program Analysis Frameworks. In *6th ACM Symp. on Principles of Programming Languages (POPL)*, pages 238–352, 1979.
7. Patrick Cousot and Radhia Cousot. Static Determination of Dynamic Properties of Programs. In *Second Int. Symp. on Programming*, pages 106–130. Dunod, Paris, France, 1976.
8. Patrick Cousot and Radhia Cousot. Comparison of the Galois Connection and Widening/Narrowing Approaches to Abstract Interpretation. JTASPEFL '91, Bordeaux. *BIGRE*, 74:107–110, October 1991.
9. M. Erné, J. Koslowski, A. Melton, and G. E. Strecker. A Primer On Galois Connections, 1992.
10. Stephane Gaubert, Eric Goubault, Ankur Taly, and Sarah Zennou. Static Analysis by Policy Iteration on Relational Domains. In *European Symposium on Programming (ESOP)*, pages 237–252. Springer Verlag, LNCS 4421, 2007.
11. Thomas Gawlitza and Helmut Seidl. Precise Fixpoint Computation Through Strategy Iteration. In *European Symposium on Programming (ESOP)*, pages 300–315. Springer Verlag, LNCS 4421, 2007.
12. A.J. Hoffman and R.M. Karp. On Nonterminating Stochastic Games. *Management Sci.*, 12:359–370, 1966.
13. R. Howard. *Dynamic Programming and Markov Processes*. Wiley, New York, 1960.
14. Nimrod Megiddo. On the Complexity of Linear Programming. In T. Bewley, editor, *Advances in Economic Theory: 5th World Congress*, pages 225–268. Cambridge University Press, 1987.
15. A. Miné. The Octagon Abstract Domain in Analysis, Slicing and Transformation. In *IEEE Working Conf. on Reverse Engineering*, pages 310–319, 2001.
16. Antoine Miné. A new numerical abstract domain based on difference-bound matrices. In Olivier Danvy and Andrzej Filinski, editors, *PADO*, volume 2053 of *Lecture Notes in Computer Science*, pages 155–172. Springer, 2001.
17. Markus Müller-Olm and Helmut Seidl. Precise Interprocedural Analysis through Linear Algebra. In *31st ACM Symp. on Principles of Programming Languages (POPL)*, pages 330–341, 2004.
18. Anuj Puri. *Theory of Hybrid and Discrete Systems*. PhD thesis, University of California, Berkeley, 1995.
19. Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, New York, 1994.
20. Sriram Sankaranarayanan, Henny B. Sipma, and Zohar Manna. Scalable analysis of linear systems using mathematical programming. In Radhia Cousot, editor, *VMCAI*, volume 3385 of *Lecture Notes in Computer Science*, pages 25–41. Springer, 2005.
21. Alexandeer Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, Inc., New York, NY, USA, 1986.
22. Jens Vöge and Marcin Jurdzinski. A Discrete Strategy Improvement Algorithm for Solving Parity Games. In *Computer Aided Verification, 12th Int. Conf. (CAV)*, pages 202–215. LNCS 1855, Springer, 2000.