Precise Interval Analysis vs. Parity Games

Thomas Gawlitza[1] and Helmut Seidl[1]

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

Abstract. In [**?**], a practical algorithm for precise interval analysis is provided for which, however, no non-trivial upper complexity bound is known. Here, we present a lower bound by showing that precise interval analysis is at least as hard as computing the sets of winning positions in parity games. Our lower-bound proof relies on an encoding of parity games into systems of particular integer equations. Moreover, we present a simplification of the algorithm for integer systems from [**?**]. For the given encoding of parity games, the new algorithm provides another algorithm for parity games which is almost as efficient as the discrete strategy improvement algorithm by Vöge and Jurdziński [**?**].

1 Introduction

Interval analysis as introduced by Cousot and Cousot [**?**,**?**] tries to determine at compile-time for each variable $\mathtt{x}$ and program point v in a program an as tight interval as possible which is guaranteed to contain all values of $\mathtt{x}$ when reaching program point v. This problem is of fundamental importance for program optimizations such as safe removal of array bound checks as well as the certification of absence of arithmetic overflows. The problem with interval analysis, though, is that the lattice of all intervals has infinite ascending chains implying that acceleration techniques are needed to enforce fixpoint iteration to terminate. One such acceleration technique is the widening and narrowing approach of Cousot and Cousot [**?**,**?**] which, however, results in algorithms which may fail to return the least solution of the given system of equations extracted from the program.

Recently, the problem of interval analysis has attracted new attention. In [**?**] Su and Wagner identified a class of polynomial solvable range constraints for interval analysis which can be solved *precisely*. This class admits full addition. Multiplication and intersection are restricted in such a way that at least one of the arguments must be a constant interval. Leroux and Sutre [**?**] extend this result by providing an acceleration-based algorithm for solving interval constraints with *full* multiplication and restricted intersection in cubic time *precisely*. In [**?**], Gaubert et al. suggest strategy iteration as an alternative method for computing solutions of interval equations with full intersections. Their method still fails to return the least solution in some cases. Computing the *least* solution to the interval equations introduced for interval analysis will be called *precise* interval analysis in the sequel. In [**?**], we reduce precise interval analysis to solving systems of integer equations for which we propose another variant of strategy iteration which is guaranteed to return the least solution. The practical efficiency of any

algorithm based on strategy iteration depends on the number of strategies encountered during the iteration. Although we never have observed more than a linear number of strategies, no non-trivial upper bound to this number is known. Thus, one might think of other methods to obtain not only a practical, but also provably polynomial algorithm for precise interval analysis. Here we show that, if such an algorithm exists, it also solves a long standing open problem, namely, to compute the winning regions of a parity game in polynomial time.

This lower-bound proof uses a reduction similar to the reductions of parity games to mean payoff games and discounted payoff games [**?**,**?**]. A different class of interval constraints is considered in [**?**] where Bordeaux et al. prove that computing the least fixpoint is **NP**-hard. This strong lower bound, however, relies on the explicit use of a square-root operator and thus cannot easily be carried over to our class where only linear operations on intervals are allowed.

Our encoding of parity games does not only give a lower-bound argument for precise interval analysis, but also allows to use methods for integer systems to solve parity games. As our second contribution, we therefore present a new version of the algorithm from [**?**] for integer systems which is significantly simpler. Similar to the algorithm in [**?**], the new algorithm is based on strategy iteration. The original algorithm, however, relies on an instrumentation of the underlying lattice to guide strategy improvement. This extra overhead is now avoided. Via our encoding, the new method for integer systems also provides a very simple algorithm for parity games. Compared to the discrete strategy improvement algorithm of Vöge [**?**,**?**], the valuations to determine the next strategy needed by our algorithm are just mappings from positions to integers.

The paper is organized as follows. In section 2, we introduce basic notions and the concepts of parity games and systems of integer equations. In section **??**, we show how one can reduce the computation of the winning regions and the winning strategies for a parity game to the computation of the least solution of systems of particular integer equations. In section **??**, we show how computing least solutions of these integer equations can be reduced to precise interval analysis — thus completing the lower-bound proof for interval analysis. In section **??**, we present the novel strategy iteration algorithm for solving systems of integer equations. Moreover, we organize the strategy iteration in such a way that, for simple integer equations, i.e., for equations with addition of constants only, the number of maxima with constants no longer affects the asymptotic complexity. Since the systems obtained from our reduction from parity games are simple, the reduction together with the new algorithm for integer equations provides another strategy iteration algorithm for parity games. Each improvement step of this algorithm requires at most quadratically many operations on integers of length $\mathcal{O}(d \cdot \log\ n)$ where n is the number of positions and d is the maximal rank of the parity game.

2 Notation and Basic Concepts

As usual, $\mathbb{N}$ and $\mathbb{Z}$ denote the set of natural numbers excluding 0 and the set of integers, respectively. We write $\mathbb{N}_0$ for $\mathbb{N} \cup \{0\}$. Given a relation $R \subseteq A \times B$ and a subset $A' \subseteq A$ we write $A'R$ for the set $\{b \in B \mid \exists a \in A' : (a, b) \in R\}$. Our complexity results will

be stated w.r.t. a uniform cost measure where we count memory accesses and arithmetic operations for $\mathcal{O}(1)$.

Parity Games. A *parity game* is a tuple $G = (V_\vee, V_\wedge, E, r)$. $V_\vee$ and $V_\wedge$ are disjoint finite sets of *positions* owned by the $\vee$-player and the $\wedge$-player, respectively. We will always write V for the set $V_\vee \cup V_\wedge$. The set $E \subseteq V^2$ is a finite set of possible *moves* with $\{v\}E \neq \emptyset$ for every position $v \in V$, i.e., there is no sink. Finally, $r : V \to \mathbb{N}_0$ is the *rank function* which assigns a *rank* $r(v)$ to every position v.

A *play* over G is an infinite word $w = v_1 v_2 \cdots$ with $(v_i, v_{i+1}) \in E$ for $i \in \mathbb{N}$. Let $m(w) := \max\{r(v) \mid v \in V \text{ occurs infinitely often in } w\}$. The play w is won by the $\vee$-player (resp. $\wedge$-player) iff $m(w)$ is odd (resp. even). A position $v \in V$ is called $\vee$-*winning* (resp. $\wedge$-*winning*) iff the $\vee$-player (resp. $\wedge$-player) can enforce that every play starting at v is won by the $\vee$-player (resp. $\wedge$-player). The set of all $\vee$-winning (resp. $\wedge$-winning) positions is called the $\vee$-*winning region* (resp. $\wedge$-*winning region*).

A mapping $\sigma_\vee : V_\vee \to V$ with $\sigma_\vee(v) \in \{v\}E$ for every $v \in V_\vee$ is called a *positional* $\vee$-*strategy*. Dually, a mapping $\sigma_\wedge : V_\wedge \to V$ with $\sigma_\wedge(v) \in \{v\}E$ for every $v \in V_\wedge$ is called a *positional* $\wedge$-*strategy*. A play w is *consistent* with the positional $\vee$-strategy $\sigma_\vee$ iff $\sigma_\vee(v_\vee) = v$ for every finite prefix $w' v_\vee v$ of w with $v_\vee \in V_\vee$. Dually, a play w is *consistent* with the positional $\wedge$-strategy $\sigma_\wedge$ iff $\sigma_\wedge(v_\wedge) = v$ for every finite prefix $w' v_\wedge v$ of w with $v_\wedge \in V_\wedge$. It is well-known that positional strategies are sufficient (memoryless determinacy) [**?**]. This means: there exists a positional $\vee$-strategy $\sigma_\vee$ such that every play w which starts at a $\vee$-winning position and which is consistent with $\sigma_\vee$ is won by the $\vee$-player. Such a positional $\vee$-strategy is called *winning*. Dually, there exists a positional $\wedge$-strategy $\sigma_\wedge$ (called *winning*) such that every play w which starts at a $\wedge$-winning position and which is consistent with $\sigma_\wedge$ is won by the $\wedge$-player.

Given a positional $\vee$-strategy $\sigma_\vee$ (resp. $\wedge$-strategy $\sigma_\wedge$) we write $G(\sigma_\vee)$ (resp. $G(\sigma_\wedge)$) for the parity game $(V_\vee, V_\wedge, (E \cap V_\wedge \times V) \cup \sigma_\vee, r)$ (resp. $(V_\vee, V_\wedge, (E \cap V_\vee \times V) \cup \sigma_\wedge, r)$) [1]. Thus, the parity game $G(\sigma_\vee)$ (resp. $G(\sigma_\wedge)$) is obtained from G by removing all moves which cannot be used in any play which is consistent with $\sigma_\vee$ (resp. $\sigma_\wedge$). A $\vee$-strategy $\sigma_\vee$ (resp. $\wedge$-strategy $\sigma_\wedge$) is winning iff every play w in $G(\sigma_\vee)$ (resp. $G(\sigma_\wedge)$) which starts from a $\vee$-winning position (resp. $\wedge$-winning position) is won by the $\vee$-player (resp. $\wedge$-player).

Systems of Integer Equations. We briefly introduce systems of integer equations (cf. [**?**]). Let $\overline{\mathbb{Z}}$ denote the complete lattice $\mathbb{Z} \cup \{-\infty, \infty\}$ equipped with the natural ordering. We extend the operations addition $+ : \overline{\mathbb{Z}} \times \overline{\mathbb{Z}} \to \overline{\mathbb{Z}}$ and multiplication $\cdot : \overline{\mathbb{Z}} \times \overline{\mathbb{Z}} \to \overline{\mathbb{Z}}$ to the operands $-\infty$ and ∞:

$$
\begin{array}{llll}
x + (-\infty) = -\infty & \text{for all } x \in \overline{\mathbb{Z}} & x + \infty = \infty & \text{for all } x > -\infty \\
0 \cdot x = 0 & \text{for all } x > -\infty & x \cdot (-\infty) = -\infty & \text{for all } x > 0 \\
x \cdot \infty = \infty & \text{for all } x > 0 & x \cdot (-\infty) = \infty & \text{for all } x < 0 \\
x \cdot \infty = -\infty & \text{for all } x < 0 & &
\end{array}
$$

A system $\mathcal{E}$ of integer equations is a sequence of equations $\mathbf{x}_i = e_i$ for $i = 1, \ldots, n$, where the variables $\mathbf{x}_i$ on the left-hand sides are pairwise distinct and the right-hand

[1] Here a mapping $f : A \to B$ is considered as the relation $\{(a, f(a)) \mid a \in A\}$.

sides e_i are expressions e built up from constants and variables by means of addition, multiplication with constants as well as minimum ("$\wedge$") and maximum ("$\vee$"):

$$e \ ::= \ a \mid \mathbf{x} \mid e_1 + e_2 \mid b \cdot e_1 \mid e_1 \wedge e_2 \mid e_1 \vee e_2$$

where e_1, e_2 are expressions, $\mathbf{x}$ is a variable, $a, b \in \overline{\mathbb{Z}}$, $b \geq 1$. We assume that $b\cdot$ has the highest operator precedence followed by $+$, $\wedge$ and $\vee$ which has the lowest operator precedence. We write $|\mathcal{E}|$ for the number of subexpressions occurring in right-hand sides of $\mathcal{E}$. Thus, $|\mathcal{E}|$ is independent of the sizes of numbers occurring in $\mathcal{E}$. We denote the set of variables of $\mathcal{E}$ by $\mathbf{X}_\mathcal{E}$. We drop the subscript whenever $\mathcal{E}$ is clear from the context. The system $\mathcal{E}$ is called *disjunctive*, if it does not contain $\wedge$-expressions, and it is called *conjunctive*, if it does not contain $\vee$-expressions. A system without $\vee$- and $\wedge$-expressions is called *basic*. If $\mathcal{E}$ denotes the system $\mathbf{x}_i = e_i$, $i = 1, \ldots, n$, then, for $a, b \in \overline{\mathbb{Z}}$ with $a \leq b$, $\mathcal{E}|_{[a,b]}$ denotes the system $\mathbf{x}_i = (e_i \wedge b) \vee a$, $i = 1, \ldots, n$.

Under a variable assignment μ, i.e., a function which maps variables from $\mathbf{X}$ to values from $\overline{\mathbb{Z}}$, an expression e evaluates to a value $[\![e]\!]\mu \in \overline{\mathbb{Z}}$:

$$\begin{array}{ccc} [\![a]\!]\mu = a & [\![\mathbf{x}]\!]\mu = \mu(\mathbf{x}) & [\![e_1 + e_2]\!]\mu = [\![e_1]\!]\mu + [\![e_2]\!]\mu \\ [\![b \cdot e]\!]\mu = b \cdot [\![e]\!]\mu & [\![e_1 \vee e_2]\!]\mu = [\![e_1]\!]\mu \vee [\![e_2]\!]\mu & [\![e_1 \wedge e_2]\!]\mu = [\![e_1]\!]\mu \wedge [\![e_2]\!]\mu \end{array}$$

where e, e_1, e_2 are expressions, $\mathbf{x}$ is a variable, $a, b \in \overline{\mathbb{Z}}$, $b \geq 1$. Together with the point-wise ordering the set of variable assignments $\mathbf{X} \to \overline{\mathbb{Z}}$ forms a complete lattice. A *solution* of $\mathcal{E}$ is a variable assignment μ which satisfies all equations of a system $\mathcal{E}$, i.e. $\mu(\mathbf{x}_i) = [\![e_i]\!]\mu$ for all i. A variable assignment μ with $\mu(\mathbf{x}_i) \leq [\![e_i]\!]\mu$ (resp. $\mu(\mathbf{x}_i) \geq [\![e_i]\!]\mu$) is called a pre-solution (resp. post-solution) of $\mathcal{E}$. Since every right-hand side e_i induces a monotonic function $[\![e_i]\!]$, Knaster-Tarski's fixpoint Theorem implies that every system $\mathcal{E}$ of integer equations has a least solution μ^*, i.e., $\mu^* \leq \mu$ for every solution μ of $\mathcal{E}$. The least solution μ^* is the greatest lower bound of all post-solutions. We refer to computing the least solution of a system $\mathcal{E}$ as solving the system $\mathcal{E}$.

We will also define strategies for systems of integer equations. Let $M(\mathcal{E})$ denote the set of all $\vee$-expressions occurring in $\mathcal{E}$. Moreover, let $M_c(\mathcal{E}) \subseteq M(\mathcal{E})$ denote the set of $\vee$-expression $e \vee e'$ occurring in $\mathcal{E}$ where at least one of the arguments e, e' is constant, i.e. it does not contain any variable. Let $M_{nc}(\mathcal{E}) := M(\mathcal{E}) \setminus M_c(\mathcal{E})$. A $\vee$-*strategy* π for $\mathcal{E}$ is a function mapping every expression $e_1 \vee e_2$ in $M(\mathcal{E})$ to one of the subexpressions e_1, e_2. For an expression e we write $e\pi$ for the expression obtained from e by recursively replacing every $\vee$-expression with the respective subexpression selected by the $\vee$-strategy π, i.e.:

$$\begin{array}{ccc} a\pi = a & \mathbf{x}\pi = \mathbf{x} & (e_1 + e_2)\pi = e_1\pi + e_2\pi \\ (b \cdot e)\pi = b \cdot e\pi & (e_1 \vee e_2)\pi = (\pi(e_1 \vee e_2))\pi & (e_1 \wedge e_2)\pi = e_1\pi \wedge e_2\pi \end{array}$$

where e, e_1, e_2 are expressions, $\mathbf{x}$ is a variable, $a, b \in \overline{\mathbb{Z}}$, $b \geq 1$. Assuming that $\mathcal{E}$ is the system $\mathbf{x}_i = e_i$, $i = 1, \ldots, n$, we write $\mathcal{E}(\pi)$ for the system $\mathbf{x}_i = e_i\pi$, $i = 1, \ldots, n$. The definitions for $\wedge$-strategies are dual.

Systems of *simple* integer equations are of a particular interest. We call an expression e *simple* iff it is of the following form:

$$e \ ::= \ c \mid \mathbf{x} \mid e + a \mid e_1 \vee e_2 \mid e_1 \wedge e_2$$

where e, e_1, e_2 are simple expressions, $\mathbf{x}$ is a variable, $a \in \mathbb{Z}$, $c \in \overline{\mathbb{Z}}$. I.e., at least one argument of every $+$-expression is a constant. An integer equation $\mathbf{x} = e$ is called *simple* iff e is simple.

We define the relation $\rightarrow$ between expressions of $\mathcal{E}$ by $e \rightarrow e'$ iff e' is an immediate subexpression of e or e is a variable and e' is the right-hand side of e, i.e., $e = e'$ is an equation of $\mathcal{E}$. A sequence $p = e_1, \ldots, e_k$ of expressions occurring in $\mathcal{E}$ is called a *path* in $\mathcal{E}$ iff $e_i \rightarrow e_{i+1}$ for $i = 1, \ldots, k-1$. The path is called *simple* iff no expression occurs twice in it. The path $e_1, \ldots, e_k$ is called a *cycle* iff $e_k \rightarrow e_1$. The weight $w(p)$ of a path $p = e_1, \ldots, e_k$ is the sum $\sum_{i=1}^{k} w(e_i)$ where $w(e)$ equals a if $e \equiv e' + a$ for some expression e' and $a \in \mathbb{Z}$, and $w(e)$ equals 0 otherwise. We call a system $\mathcal{E}$ of simple integer equations *non-zero* iff $w(c) \neq 0$ for every simple cycle c in $\mathcal{E}$.

Example 1. Consider the following systems of simple integer equations:

$$\mathcal{E}_1 = \mathbf{x}_1 = \mathbf{x}_2 + 2, \mathbf{x}_2 = \mathbf{x}_1 + (-1) \quad \mathcal{E}_2 = \mathbf{x}_1 = \mathbf{x}_2 + 2 \vee \mathbf{x}_2 + 1, \mathbf{x}_2 = \mathbf{x}_1 + (-1)$$

The system $\mathcal{E}_1$ is non-zero, because the only simple cycle in $\mathcal{E}_1$ (up to cyclic permutations) is $\mathbf{x}_1, \mathbf{x}_2 + 2, \mathbf{x}_2, \mathbf{x}_1 + (-1)$ which has weight 1. The system $\mathcal{E}_2$ is not non-zero, because the simple cycle $\mathbf{x}_1, \mathbf{x}_2 + 1, \mathbf{x}_2, \mathbf{x}_1 + (-1)$ has weight 0. □

A variable assignment μ with $-\infty < \mu(\mathbf{x}) < \infty$, $\mathbf{x} \in \mathbf{X}$ is called *finite*. We have:

Lemma 1. *Every non-zero system $\mathcal{E}$ of simple equations has at most one finite solution.*

Proof. Note that, if we rewrite an expression in $\mathcal{E}$ using distributivity, then the resulting system is still non-zero. Let $\mathbf{X}_{\mathcal{E}}^{rhs}$ denote the set of variables occurring in right-hand sides of $\mathcal{E}$. We proceed by induction on $|\mathbf{X}_{\mathcal{E}}^{rhs}|$. If $|\mathbf{X}_{\mathcal{E}}^{rhs}| = 0$, then the statement is fulfilled, since there is exactly one solution.

Let $|\mathbf{X}_{\mathcal{E}}^{rhs}| > 0$ and $\mathbf{x} \in \mathbf{X}_{\mathcal{E}}^{rhs}$. Consider the equation $\mathbf{x} = e$. We consider the case where e contains the variable $\mathbf{x}$. Because of distributivity, we can w.l.o.g. assume that $\mathbf{x} = e$ is of the form $\mathbf{x} = ((\mathbf{x} + c) \wedge e_1) \vee e_2$. where e_1 and e_2 are such that no $\vee$ occurs within a $\wedge$-expression and no $\wedge$-expression occurs within a $+$-expression. We say that such an expression is in disjunctive normal form. Since $\mathcal{E}$ is non-zero, we know that $c \neq 0$. We only consider the case that $c > 0$. The other case is similar. First of all, observe that, for every finite variable assignment μ, the following holds:

$$\mu(\mathbf{x}) = [\![((\mathbf{x} + c) \wedge e_1) \vee e_2]\!]\mu \quad \text{implies} \quad \mu(\mathbf{x}) = [\![e_1 \vee e_2]\!]\mu. \tag{1}$$

Let μ_1 and μ_2 be finite solutions of $\mathcal{E}$. Let $\mathcal{E}'$ denote the system of simple equations obtained from $\mathcal{E}$ by replacing the equation $\mathbf{x} = e$ with the equation $\mathbf{x} = e_1 \vee e_2$. The system $\mathcal{E}'$ is non-zero. (1) implies that μ_1 and μ_2 are finite solutions of $\mathcal{E}'$. Since we can repeat this step, we can w.l.o.g. assume that the variable $\mathbf{x}$ does not occur within $e_1 \vee e_2$. We now replace every occurrence of $\mathbf{x}$ in right-hand sides of $\mathcal{E}'$ by $e_1 \vee e_2$ and obtain a system $\mathcal{E}''$. This system is again non-zero and μ_1 and μ_2 are finite solutions of $\mathcal{E}''$. Thus, since $|\mathbf{X}_{\mathcal{E}''}^{rhs}| = |\mathbf{X}_{\mathcal{E}}^{rhs}| - 1$, the induction hypotheses implies $\mu_1 = \mu_2$. □

3 From Parity Games to Systems of Integer Equations

In this section we reduce computing winning regions and winning strategies for parity games to solving systems of integer equations. Thus, the latter computational problem is as least as hard as solving parity games. It is an intriguing open problem to determine the precise complexity of parity games. What is known is that this problem is in $\mathbf{UP} \cap \mathbf{co}{-}\mathbf{UP}$ [**?**]. A first subexponential algorithm has been presented in [**?**]. Whether or not, however, parity games can be solved in polynomial time, is still unknown.

Let us fix a parity game $G = (V_\vee, V_\wedge, E, r)$. Let $n := |V|$ be the number of positions, $d := \max\ r(V) = \max\ \{r(v) \mid v \in V\}$ the maximal rank and $m := n^{d+1}$. In order to compute the winning regions, we consider the system $\mathcal{E}_G$ of integer equations which we define subsequently. From the least solution μ^* of $\mathcal{E}_G|_{[-m,m]}$ we will deduce the winning regions as well as winning strategies for both players. For every position $v \in V$ we introduce a fresh variable $\mathbf{x}_v$, i.e., $\mathbf{X}_{\mathcal{E}_G} := \{\mathbf{x}_v \mid v \in V\}$. Let

$$\delta_r = -(-n)^r.$$

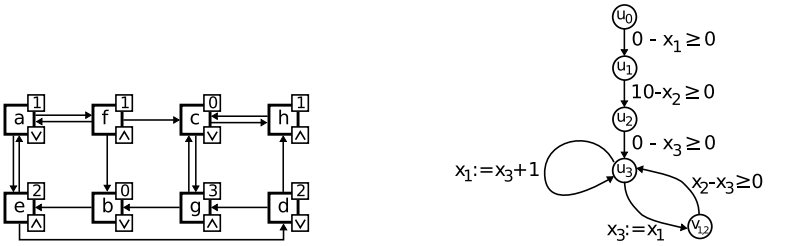
Observe that δ_r is less than 0 whenever r is even and greater than 0 whenever r is odd. Moreover, δ_r is chosen such that $(n-1)|\delta_{r'}| < |\delta_r|$ whenever $r' < r$. This important property ensures that, for $k \leq n$, the sum $\delta_{r_1} + \cdots + \delta_{r_k}$ is greater than 0 iff the most relevant rank within $\{r_1, \ldots, r_k\}$ is odd. We construct $\mathcal{E}_G$ as follows. For every position $v \in V_\vee$ we add the equation

$$\mathbf{x}_v = (\mathbf{x}_{v_1} \vee \cdots \vee \mathbf{x}_{v_k}) + \delta_{r(v)}$$

where $\{v\}E = \{v_1, \ldots, v_k\}$. For every position $v \in V_\wedge$ we add the equation

$$\mathbf{x}_v = (\mathbf{x}_{v_1} \wedge \cdots \wedge \mathbf{x}_{v_k}) + \delta_{r(v)}$$

where $\{v\}E = \{v_1, \ldots, v_k\}$. We illustrate this reduction by an example.



(a) The parity game G of example **??** (b) Affine program $P_{\mathcal{C}}$ of example **??**

Fig. 1.

Example 2. Consider the parity game $G = (V_\vee, V_\wedge, E, r)$ (from [**?**]) where

- $V_\vee = \{a, b, c, d\}$ and $V_\wedge = \{e, f, g, h\}$
- $E = \{(a, f), (a, e), (b, e), (c, g), (c, h), (d, g), (d, h), (e, a), (e, d), (f, a), (f, b), (f, c), (g, b), (g, c), (h, c)\}$

– $r(b) = r(c) = 0, r(a) = r(f) = r(h) = 1, r(d) = r(e) = 2, r(g) = 3$

which is illustrated in figure **??** (a). The system $\mathcal{E}_G|_{[-m,m]}$ is given as

$$\begin{array}{ll} \mathbf{x}_a = (\mathbf{x}_e \vee \mathbf{x}_f) + 8 \wedge m \vee -m & \mathbf{x}_b = \mathbf{x}_e + (-1) \wedge m \vee -m \\ \mathbf{x}_c = (\mathbf{x}_g \vee \mathbf{x}_h) + (-1) \wedge m \vee -m & \mathbf{x}_d = (\mathbf{x}_g \vee \mathbf{x}_h) + (-64) \wedge m \vee -m \\ \mathbf{x}_e = (\mathbf{x}_a \wedge \mathbf{x}_d) + (-64) \wedge m \vee -m & \mathbf{x}_f = (\mathbf{x}_a \wedge \mathbf{x}_b \wedge \mathbf{x}_c) + 8 \wedge m \vee -m \\ \mathbf{x}_g = (\mathbf{x}_b \wedge \mathbf{x}_c) + 512 \wedge m \vee -m & \mathbf{x}_h = \mathbf{x}_c + 8 \wedge m \vee -m \end{array}$$

where $m = 4096$. □

We summarize statements about $\mathcal{E}_G$ and $\mathcal{E}_G|_{[-m,m]}$ in the following Lemma:

Lemma 2. *1.* $|\mathbf{X}_{\mathcal{E}_G}| = |\mathbf{X}_{\mathcal{E}_G|_{[-m,m]}}| = n$*;*
2. $|M(\mathcal{E}_G)| = |E \cap V_\vee \times V| - |V_\vee|$ *and* $|M(\mathcal{E}_G|_{[-m,m]})| = |E \cap V_\vee \times V| - |V_\vee| + n$*;*
3. The size of occurring numbers is bounded by $(d+1)\log_2 n$*;*
4. The systems $\mathcal{E}_G$ *and* $\mathcal{E}_G|_{[-m,m]}$ *of simple equations are non-zero.*

Proof. We only prove the fourth statement. Since there exists a one-to-one mapping f from the set of simple cycles in $\mathcal{E}_G|_{[-m,m]}$ onto the set of simple cycles in $\mathcal{E}_G$ with $w(c) = w(f(c))$ for every simple cycle c in $\mathcal{E}_G|_{[-m,m]}$, we only have to show that $\mathcal{E}_G$ is non-zero. W.l.o.g., let

$$c \;=\; \mathbf{x}_1,\; e_1 + \delta_{r(v_1)},\; \ldots,\; \mathbf{x}_2, e_2 + \delta_{r(v_2)},\; \ldots,\; \mathbf{x}_k,\; e_k + \delta_{r(v_k)}$$

be a simple cycle in $\mathcal{E}_G$ where $\mathbf{x}_1, \ldots, \mathbf{x}_k$ are the only expressions in the sequence c which are variables. Thus $k \leq n$. Let $J := \{j \in \{1, \ldots, k\} \mid |\delta_{r(v_j)}| = \max_{i=1,\ldots,k} |\delta_{r(v_i)}|\}$. Let r denote the only rank in the set $r(\{v_j \mid j \in J\})$. Note that $k - |J| \leq n - 1$ and $|\delta_r| > (n-1)|\delta_{r-1}|$. We get:

$$\begin{aligned} |w(c)| &= |\textstyle\sum_{i=1}^{k} \delta_{r(v_i)}| = |\sum_{i\in J} \delta_{r(v_i)} + \sum_{i\in\{1,\ldots,k\}\setminus J} \delta_{r(v_i)}| \\ &= ||J|\delta_r + \textstyle\sum_{i\in\{1,\ldots,k\}\setminus J} \delta_{r(v_i)}| \geq |\delta_r| - \sum_{i\in\{1,\ldots,k\}\setminus J} |\delta_{r(v_i)}| \\ &\geq |\delta_r| - (k - |J|)|\delta_{r-1}| \geq |\delta_r| - (n-1)|\delta_{r-1}| > 0 \end{aligned}$$

It follows $w(c) \neq 0$. □

Thus, by Lemma 1 and **??**, $\mathcal{E}_G|_{[-m,m]}$ has exactly one solution which is finite.

Example 3. The unique solution μ^* of $\mathcal{E}_G|_{[-4096,4096]}$ in example **??** is given by

$$\begin{array}{l} \mu^*(\mathbf{x}_a) = -4080, \mu^*(\mathbf{x}_b) = -4096, \mu^*(\mathbf{x}_c) = 4095, \mu^*(\mathbf{x}_d) = 4032, \\ \mu^*(\mathbf{x}_e) = -4096, \mu^*(\mathbf{x}_f) = -4088, \mu^*(\mathbf{x}_g) = -3584, \mu^*(\mathbf{x}_h) = 4096. \end{array}$$ □

The next Lemma states that we can reassemble the unique solution of $\mathcal{E}_G|_{[-m,m]}$ by a $\vee$-strategy for $\mathcal{E}_G$. This is simlar to the memoryless determinacy of parity games.

Lemma 3. *Let* μ^* *denote the unique finite solution of* $\mathcal{E}_G|_{[-m,m]}$*. There exists a* $\vee$*-strategy (resp.* $\wedge$*-strategy)* π *for* $\mathcal{E}_G$ *such that* μ^* *is the unique solution of* $\mathcal{E}_G(\pi)|_{[-m,m]}$*. Moreover,* π *can be computed from* μ^* *in time* $\mathcal{O}(|\mathcal{E}_G|)$*.*

Proof. We only prove the $\vee$-strategy case. Let π be the $\vee$-strategy defined by

$$\pi(e_1 \vee e_2) = \begin{cases} e_1 & \text{if } [\![e_1]\!]\mu^* \geq [\![e_2]\!]\mu^* \\ e_2 & \text{if } [\![e_1]\!]\mu^* < [\![e_2]\!]\mu^* \end{cases}$$

for every expression $e_1 \vee e_2$ occurring in $\mathcal{E}_G$. The system $\mathcal{E}_G(\pi)|_{[-m,m]}$ is non-zero and μ^* is a solution of $\mathcal{E}_G(\pi)|_{[-m,m]}$. Thus, Lemma 1 implies that μ^* is the only solution of $\mathcal{E}_G(\pi)|_{[-m,m]}$. The complexity statement follows from the fact that the $\vee$-strategy π can be computed by evaluating each right-hand side once. □

Before going further we consider the special case that no player has a choice.

Lemma 4. *Let $G = (V_\vee, V_\wedge, E, r)$ be a parity game where only one move is possible for every position, i.e., $|\{v\}E| = 1$ for every $v \in V_\vee \cup V_\wedge$. Let μ^* be the unique finite solution of $\mathcal{E}_G|_{[-m,m]}$. Then $\mu^*(\mathbf{x}_v) > 0$ iff v is a $\vee$-winning position.*

Proof. Since the winning regions partition the set of positions, we only have to show that $\mu^*(\mathbf{x}_v) > 0$ for every $\vee$-winning position v. Let v be a $\vee$-winning position. Let

$$w = v'_1 \cdot \dots \cdot v'_{k'} \cdot (v_1 \cdot \dots \cdot v_k)^\omega$$

denote the only game which can be played on G starting at v. We can assume that $v'_1, \dots, v'_{k'}, v_1, \dots, v_k$ are pair-wise distinct. Then $k + k' \leq n$ and $k \geq 1$. Since w is won by the $\vee$-player, the highest rank h which occurs in $r(v_1), \dots, r(v_k)$ is odd. Thus $\delta_h > 0$. Let j be the smallest $j \in \{1, \dots, k\}$ with $r(v_j) = h$. The system $\mathcal{E}_G|_{[-m,m]}$ contains the equations

$$\mathbf{x}_{v_i} = \mathbf{x}_{v_{(i+1) \bmod k}} + \delta_{r(v_i)} \wedge m \vee -m, \quad i = 1, \dots, k.$$

Thus, since $\sum_{i=1}^{k} \delta_{r(v_i)} \geq \delta_h - (k-1)|\delta_{h-1}| > 0$, it follows that $\mu^*(\mathbf{x}_{v_j}) = m$. Since $\sum_{i=1}^{k'} \delta_{r(v'_i)} + \sum_{i=1}^{j-1} \delta_{r(v_i)} \leq (n-1)|\delta_d| = (n-1)n^d < n^{d+1} = m$, we get $\mu^*(\mathbf{x}_{v'_1}) > 0$. □

We establish a one-to-one correspondence between positional strategies for G and strategies for $\mathcal{E}_G$. For a positional $\vee$-strategy $\sigma_\vee$ (resp. $\wedge$-strategy $\sigma_\wedge$) for G, we write $\pi(\sigma_\vee)$ (resp. $\pi(\sigma_\wedge)$) for the $\vee$-strategy (resp. $\wedge$-strategy) for $\mathcal{E}_G$ which corresponds to $\sigma_\vee$ (resp. $\sigma_\wedge$). More precisely, the $\vee$-strategy $\pi(\sigma_\vee)$ is defined by

$$\pi(\sigma_\vee)(\mathbf{x}_{v_1} \vee \dots \vee \mathbf{x}_{v_k}) = \mathbf{x}_{v_j} \quad \text{for } \{v\}E = \{v_1, \dots, v_k\} \text{ and } \sigma_\vee(v) = v_j.$$

The $\wedge$-strategy $\pi(\sigma_\wedge)$ is defined analogously. Since the mapping π is one-to-one, the inverse π^{-1} exists which maps strategies for $\mathcal{E}_G$ to positional strategies for G. By construction, $\mathcal{E}_{G(\sigma)} = \mathcal{E}_G(\pi(\sigma))$ and thus $\mathcal{E}_{G(\sigma)}|_{[-m,m]} = \mathcal{E}_G(\pi(\sigma))|_{[-m,m]}$ for every $\vee$-strategy (resp. $\wedge$-strategy) σ for G.

Let μ^* denote the unique solution of $\mathcal{E}_G|_{[-m,m]}$. By Lemma **??** we can compute a $\vee$-strategy $\pi_\vee$ for $\mathcal{E}_G$ such that μ^* is the unique solution of $\mathcal{E}_G(\pi_\vee)|_{[-m,m]}$. The next Lemma in particular states that $\pi^{-1}(\pi_\vee)$ is a $\vee$-winning strategy for G.

Lemma 5. *Let $G = (V_\vee, V_\wedge, E, r)$ be a parity game. Let μ^* be the unique solution of $\mathcal{E}_G|_{[-m,m]}$. Then $\mu^*(\mathbf{x}_v) > 0$ (resp. $\mu^*(\mathbf{x}_v) \leq 0$) iff v is a $\vee$-winning (resp. $\wedge$-winning) position. Moreover, winning strategies for both players can be computed from μ^* in time $\mathcal{O}(|E|)$. More precisely, if $\pi_\vee$ (resp. $\pi_\wedge$) is a $\vee$-strategy (resp. $\wedge$-strategy) for $\mathcal{E}_G$ such that μ^* is the unique solution of $\mathcal{E}_G(\pi_\vee)|_{[-m,m]}$ (resp. $\mathcal{E}_G(\pi_\wedge)|_{[-m,m]}$), then $\pi^{-1}(\pi_\vee)$ (resp. $\pi^{-1}(\pi_\wedge)$) is $\vee$-winning (resp. $\wedge$-winning).*

Proof. We only show the statement for the $\vee$-player. The statement for the $\wedge$-player can be shown dually. Let W denote the $\vee$-winning region in G. Let $\Sigma_\vee$ (resp. $\Sigma_\wedge$) denote the set of $\vee$-strategies (resp. $\wedge$-strategies) for G. Given some $\sigma_\vee \in \Sigma_\vee$ and some $\sigma_\wedge \in \Sigma_\wedge$, we write $W_{\sigma_\vee}$ (resp. $W_{\sigma_\vee\sigma_\wedge}$) for the $\vee$-winning region in $G(\sigma_\vee)$ (resp. $G(\sigma_\vee)(\sigma_\wedge)$). Let $\Pi_\vee$ (resp. $\Pi_\wedge$) denote the set of $\vee$-strategies (resp. $\wedge$-strategies) for $\mathcal{E}_G$. Given some $\pi_\vee \in \Pi_\vee$ and some $\pi_\wedge \in \Pi_\wedge$, we write $\mu_{\pi_\vee}$ (resp. $\mu_{\pi_\vee\pi_\wedge}$) for the unique solution of $\mathcal{E}_G(\pi_\vee)|_{[-m,m]}$ (resp. $\mathcal{E}_G(\pi_\vee)(\pi_\wedge)|_{[-m,m]}$). Lemma **??** implies

$$W_{\sigma_\vee\sigma_\wedge} = \{v \in V \mid \mu_{\pi(\sigma_\vee)\pi(\sigma_\wedge)}(\mathbf{x}_v) > 0\} \text{ for all } \sigma_\vee \in \Sigma_\vee \text{ and all } \sigma_\wedge \in \Sigma_\wedge. \quad (2)$$

Let us fix some $\sigma_\vee \in \Sigma_\vee$. Lemma **??** implies that there exists some $\pi_\wedge \in \Pi_\wedge$ such that $\mu_{\pi(\sigma_\vee)\pi_\wedge} = \mu_{\pi(\sigma_\vee)}$. Let $\sigma'_\wedge \in \Sigma_\wedge$. We have $\mu_{\pi(\sigma_\vee)\pi(\sigma'_\wedge)} \geq \mu_{\pi(\sigma_\vee)} = \mu_{\pi(\sigma_\vee)\pi_\wedge}$. Thus (**??**) implies $W_{\sigma_\vee\sigma'_\wedge} \supseteq W_{\sigma_\vee\pi^{-1}(\pi_\wedge)}$. Since $\sigma'_\wedge$ was chosen arbitrarily, we have $W_{\sigma_\vee} = W_{\sigma_\vee\pi^{-1}(\pi_\wedge)}$. Since $\sigma_\vee$ was also chosen arbitrarily, (**??**) implies

$$W_{\sigma_\vee} = \{v \in V \mid \mu_{\pi(\sigma_\vee)}(\mathbf{x}_v) > 0\} \text{ for all } \sigma_\vee \in \Sigma_\vee. \quad (3)$$

Lemma **??** implies that there exists some $\pi_\vee \in \Pi_\vee$ such that $\mu_{\pi_\vee} = \mu^*$. Let $\sigma'_\vee \in \Sigma_\vee$. We have $\mu_{\pi(\sigma'_\vee)} \leq \mu^* = \mu_{\pi_\vee}$. Thus (**??**) implies $W_{\sigma'_\vee} \subseteq W_{\pi^{-1}(\pi_\vee)}$. Since $\sigma'_\vee$ was chosen arbitrarily, we have $W = W_{\pi^{-1}(\pi_\vee)}$ which means that $\pi^{-1}(\pi_\vee)$ is a $\vee$-winning strategy in G. Using (**??**) we get $W = \{v \in V \mid \mu^*(\mathbf{x}_v) > 0\}$. The complexity statement is obvious. □

Example 4. Consider again example **??** and example **??**. Positions c, d and h are $\vee$-winning positions, since $\mu^*(\mathbf{x}_c), \mu^*(\mathbf{x}_d), \mu^*(\mathbf{x}_h) > 0$. Conversely, a, b, e, f, g are $\wedge$-winning positions, since $\mu^*(\mathbf{x}_a), \mu^*(\mathbf{x}_b), \mu^*(\mathbf{x}_e), \mu^*(\mathbf{x}_f), \mu^*(\mathbf{x}_g) < 0$. A $\vee$-strategy $\pi_\vee$ for $\mathcal{E}_G$ such that μ^* is the unique solution of $\mathcal{E}_G(\pi_\vee)|_{[-m,m]}$ is given by

$$\pi_\vee(\mathbf{x}_e \vee \mathbf{x}_f) = \mathbf{x}_f \qquad \pi_\vee(\mathbf{x}_g \vee \mathbf{x}_h) = \mathbf{x}_h.$$

Thus $\sigma := \pi^{-1}(\pi_\vee)$, given by $\sigma(a) = f, \sigma(c) = h, \sigma(d) = h$ is $\vee$-winning. □

Thus we get the main result for this section as a corollary of Lemma **??**.

Theorem 1. *The problem of computing winning regions for parity games is* **P***-time reducible to solving systems of integer equations.* □

4 From Systems of Integer Equations to Interval Analysis

We now reduce solving systems of integer equations to precise interval analysis for affine programs (cf. e.g. [**?**]). Let $\mathcal{I}$ denote the set of closed intervals in $\mathbb{Z}$, i.e.,

$$\mathcal{I} = \{\emptyset\} \cup \{[a, b] \subseteq \mathbb{Z} \mid a, b \in \overline{\mathbb{Z}} \text{ and } \infty > a \leq b > -\infty\}.$$

Let $\mathcal{B} := \{I_1 \times \cdots \times I_n \mid I_i \in \mathcal{I}, i = 1, \ldots, n\} \subseteq 2^{\mathbb{Z}^n}$. $(\mathcal{B}, \subseteq)$ is a complete lattice. Elements from $\mathcal{B}$ are called boxes. We define $\alpha : 2^{\overline{\mathbb{Z}}^n} \to \mathcal{B}$ by

$$\alpha(X) = \bigcap_{B \in \mathcal{B}, B \supseteq X} B \in \mathcal{B}, \qquad X \subseteq \mathbb{Z}^n.$$

The box $\alpha(X)$ is the smallest box which is a super-set of X.

Subsequently we discuss affine programs. Let us fix a set $\mathbf{X}_P = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ of program variables.Then a *state* in the concrete semantics which assigns values to the variables is conveniently modeled by a vector $x = (x_1, \ldots, x_n) \in \mathbb{Z}^n$; x_i is the value assigned to variable $\mathbf{x}_i$. Note that we distinguish variables and their values by using a different font. In this paper, we only consider statements of the following forms:

$$(1) \quad \mathbf{x}_j := a + \textstyle\sum_{i=1}^n a_i \cdot \mathbf{x}_i \qquad (2) \quad a + \textstyle\sum_{i=1}^n a_i \cdot \mathbf{x}_i \geq 0$$

where $a, a_1, \ldots, a_n \in \mathbb{Z}$. We use an abstract fixpoint semantics which associates a box $B = I_1 \times \cdots \times I_n \in \mathcal{B}$ to each program point. Each statement $s \in \mathsf{Stmt}$ induces a transformation $[\![s]\!] : \mathcal{B} \to \mathcal{B}$, given by

$$[\![\mathbf{x}_j := a + \textstyle\sum_{i=1}^n a_i \cdot \mathbf{x}_i]\!]B = \alpha(\{(x_1, \ldots, x_{j-1}, a + \textstyle\sum_{i=1}^n a_i \cdot x_i, x_{j+1}, \ldots, x_n) \mid (x_1, \ldots, x_n) \in B\})$$

$$[\![a + \textstyle\sum_{i=1}^n a_i \cdot \mathbf{x}_i \geq 0]\!]B = \alpha(\{(x_1, \ldots, x_n) \in B \mid a + \textstyle\sum_{i=1}^n a_i \cdot x_i \geq 0\})$$

where $B \in \mathcal{B}$. We emphasize that $[\![s]\!]$ is the best abstract transformer w.r.t. the natural concrete semantics (cf. [**?**]). The branching of an *affine program* is non-deterministic. Formally, an *affine program* is given by a *control flow graph* $P = (N, T, \mathsf{st})$ that consists of a set N of *program points*, a set $T \subseteq N \times \mathsf{Stmt} \times N$ of *(control flow) edges* and a special *start point* $\mathsf{st} \in N$. Then, the abstract fixpoint semantics V of P is characterized as the least solution of the following system of constraints:

$$(1) \quad \mathbf{V}[\mathsf{st}] \supseteq \mathbb{Z}^n \qquad (2) \quad \mathbf{V}[v] \supseteq [\![s]\!](\mathbf{V}[u]) \quad \text{for each } (u, s, v) \in T$$

where the variables $\mathbf{V}[v]$, $v \in N$ take values in $\mathcal{B}$. We denote the components of the abstract fixpoint semantics V by $V[v]$ for $v \in N$. We emphasize that we focus on *precise* interval analysis which means that it is not sufficient to compute a small solution of the above constraint system. We in fact want to compute the least solution.

Assume that $\mathcal{E}$ denotes a system of integer equations. In place of $\mathcal{E}$ we consider a system $\mathcal{C}$ of integer constraints where each constraint is of one of the following forms

$$(1) \quad \mathbf{x} \geq c \qquad (2) \quad \mathbf{x} \geq a + \textstyle\sum_{i=1}^k a_i \cdot \mathbf{x}_i \qquad (3) \quad \mathbf{x} \geq \mathbf{x}_1 \wedge \mathbf{x}_2$$

where $c \in \overline{\mathbb{Z}} \setminus \{-\infty\}$, $a, a_1, \ldots, a_k > 0$, $\mathbf{x}, \mathbf{x}_1, \mathbf{x}_2$ are variables. This can be done w.o.l.g. since, for every system $\mathcal{E}$ of integer equations, we can compute a system $\mathcal{C}$ of integer constraints of the above form whose least solution gives us the least solution of $\mathcal{E}$ in linear time. Furthermore, we assume w.l.o.g. that, for every variable $\mathbf{x}$, there exists exactly one constraint of the form (1). This can be done w.o.l.g., since we can identify the set of variables $\mathbf{x}$ with $\mu^*(\mathbf{x}) = -\infty$ in time $\mathcal{O}(n \cdot |\mathcal{E}|)$. We can remove these variables and obtain a system whose least solution maps every variable to a value

strictly greater than $-\infty$. Additionally, we can compute a lower bound $c_{\mathbf{x}} \in \mathbb{Z}$ for each variable $\mathbf{x}$, i.e. $\mu^*(\mathbf{x}) \geq c_{\mathbf{x}}$, in time $\mathcal{O}(n \cdot |\mathcal{E}|)$ by performing n lock-step fixpoint computation steps.

We construct the affine program $P_{\mathcal{C}} = (N, T, \mathsf{st})$ as follows. Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ denote the set of variables used in $\mathcal{C}$. We choose

$$N := \{\mathsf{st}, u_1, \ldots, u_n\} \cup \{v_{k_1,k_2} \mid \mathbf{x}_j \geq \mathbf{x}_{k_1} \wedge \mathbf{x}_{k_2} \text{ is a constraint of } \mathcal{C}\}$$

as the set of program points and identify st with u_0. We construct the set T of control-flow edges as follows. For every constraint $\mathbf{x}_j \geq c$ of $\mathcal{C}$ we add the control-flow edge

$$(u_{j-1}, c - \mathbf{x}_j \geq 0, u_j).$$

For every constraint $\mathbf{x}_j \geq a + \sum_i a_i \cdot \mathbf{x}_{k_i}$ of $\mathcal{C}$ we add the control-flow edge

$$(u_n, \mathbf{x}_j := a + \textstyle\sum_i a_i \cdot \mathbf{x}_{k_i}, u_n).$$

For every constraint $\mathbf{x}_j \geq \mathbf{x}_{k_1} \wedge \mathbf{x}_{k_2}$ of $\mathcal{C}$ we add the control-flow edges

$$(u_n, \mathbf{x}_j := \mathbf{x}_{k_1}, v_{k_1,k_2}) \quad \text{and} \quad (v_{k_1,k_2}, \mathbf{x}_{k_2} - \mathbf{x}_j \geq 0, u_n).$$

Then we can obtain the least solution of $\mathcal{C}$ from the abstract fixpoint semantics V of P:

Lemma 6. *Let μ^* denote the least solution of $\mathcal{C}$ and $(I_1, \ldots, I_n) := V[u_n]$. Then, for every $i = 1, \ldots, n$, $\mu^*(\mathbf{x}_i)$ equals the upper bound of the interval I_i.* □

Example 5. Consider the following system $\mathcal{E}$ of integer constraints:

$$\mathbf{x}_1 = 0 \vee \mathbf{x}_3 + 1 \qquad \mathbf{x}_2 = 10 \qquad \mathbf{x}_3 = \mathbf{x}_1 \wedge \mathbf{x}_2$$

By performing 3 rounds of lock-step fixpoint iteration we get that the value of the variable $\mathbf{x}_3$ is as least 0. Thus, in place of $\mathcal{E}$, we consider the following system $\mathcal{C}$ of integer constraints. $\mathcal{E}$ and $\mathcal{C}$ have the same least solution.

$$\mathbf{x}_1 \geq 0 \qquad \mathbf{x}_1 \geq \mathbf{x}_3 + 1 \qquad \mathbf{x}_2 \geq 10 \qquad \mathbf{x}_3 \geq 0 \qquad \mathbf{x}_3 \geq \mathbf{x}_1 \wedge \mathbf{x}_2$$

The least solution μ^* of $\mathcal{E}$ is given by $\mu^*(\mathbf{x}_1) = 11, \mu^*(\mathbf{x}_2) = 10, \mu^*(\mathbf{x}_3) = 10$. Figure **??** (b) shows the corresponding affine program $P_{\mathcal{C}}$. Let V denote the abstract fixpoint semantics of $P_{\mathcal{C}}$. Then $V[u_3] = [-\infty, 11] \times [-\infty, 10] \times [-\infty, 10]$. □

Combining Theorem **??** and Lemma **??** we get our lower bound result:

Theorem 2. *The problem of computing winning regions of parity games is* **P***-time reducible to precise interval analysis for affine programs.* □

5 Solving Integer Equations

In this section we present a simplified method for computing least solutions of systems of integer equations. As the algorithm in [**?**], our new algorithm essentially iterates

over suitable $\vee$-strategies where, for each attained strategy, we determine the *greatest* solution of the corresponding conjunctive system. Our key contribution is to show that this idea also works, if instrumentation of the underlying lattice as in [**?**] is abandoned.

Assume that μ^* denotes the least solution of the system $\mathcal{E}$ of integer equations. A *$\vee$-strategy improvement operator* $P_\vee$ is a function which maps a pair (π, μ) to an *improved* $\vee$-strategy $\pi' := P_\vee(\pi, \mu)$, where π is a $\vee$-strategy for $\mathcal{E}$ and $\mu \leq \mu^*$ is a pre-solution of $\mathcal{E}$ and the following holds:

$$\pi' \neq \pi \text{ whenever } \mu < \mu^* \text{ and } \pi'(e_1 \vee e_2) \in \begin{cases} \{e_1, \pi(e_1 \vee e_2))\} & \text{if } [\![e_1]\!]\mu > [\![e_2]\!]\mu \\ \{e_2, \pi(e_1 \vee e_2))\} & \text{if } [\![e_1]\!]\mu < [\![e_2]\!]\mu \\ \{\pi(e_1 \vee e_2)\} & \text{if } [\![e_1]\!]\mu = [\![e_2]\!]\mu \end{cases}$$

If not further specified $P_\vee$ means any $\vee$-strategy improvement operator. We define the $\vee$-strategy improvement operator $P_\vee^{eager}$ by

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

where π is a $\vee$-strategy for $\mathcal{E}$ and $\mu \leq \mu^*$ is a pre-solution of $\mathcal{E}$. This is basically the $\vee$-strategy improvement operator used in [**?**].

Assume that $\mathcal{E}$ is a system of *basic* integer equations. We define the set $\mathcal{D}(\mathcal{E})$ of *derived constraints* as the smallest set of constraints of the form $\mathbf{x} \leq e$ such that (1) $\mathbf{x} \leq e \in \mathcal{D}(\mathcal{E})$ whenever $\mathbf{x} = e$ is an equation of $\mathcal{E}$; and (2) $\mathbf{x} \leq e'' \in \mathcal{D}(\mathcal{E})$ whenever $\mathbf{x} \leq e,\ \mathbf{x}' \leq e' \in \mathcal{D}(\mathcal{E})$ and e'' is obtained from e by replacing $\mathbf{x}'$ with e'. For a system $\mathcal{E}$ of conjunctive equations we define the set $\mathcal{D}(\mathcal{E})$ of derived constraints by $\mathcal{D}(\mathcal{E}) := \bigcup_\pi$ is a $\wedge$-strategy for $\mathcal{E}$ $\mathcal{D}(\mathcal{E}(\pi))$. Let $\mathcal{E}$ be a system of conjunctive equations. For every $\mathbf{x} \leq e \in \mathcal{D}(\mathcal{E})$ and every pre-solution μ of $\mathcal{E}$ we have $[\![\mathbf{x}]\!]\mu \leq [\![e]\!]\mu$. A pre-solution μ of $\mathcal{E}$ is called *($\mathcal{E}$-)feasible* iff (1) $e = -\infty$ whenever $\mathbf{x} = e$ is an equation of $\mathcal{E}$ with $[\![e]\!]\rho = -\infty$; and (2) $[\![\mathbf{x}]\!]\mu = [\![e]\!]\mu$ implies $[\![\mathbf{x}]\!]\mu = \infty$ for all derived constraints $\mathbf{x} \leq e \in \mathcal{D}(\mathcal{E})$ where $\mathbf{x}$ occurs in e.

Example 6 (feasibility). There exists no feasible pre-solution of the system $\mathbf{x}_1 = \mathbf{x}_1 \wedge 10$. Every variable assignment which maps $\mathbf{x}_1$ to values between 1 and 10 is a feasible pre-solution of the system $\mathbf{x}_1 = 2 \cdot \mathbf{x}_1 \wedge 10$. □

Lemma 7. *1. Let $\mathcal{E}$ be a conjunctive system of integer equations and μ be a feasible pre-solution of $\mathcal{E}$. Every pre-solution $\mu' \geq \mu$ of $\mathcal{E}$ is feasible.*

2. Let $\mathcal{E}$ be a system of integer equations, π a $\vee$-strategy for $\mathcal{E}$, μ a feasible pre-solution of $\mathcal{E}(\pi)$ and $\pi' := P_\vee(\pi, \mu)$. Then μ is a feasible pre-solution of $\mathcal{E}(\pi')$. □

Let $\mathcal{E}$ be the system $\mathbf{x}_1 = e_1, \ldots, \mathbf{x}_n = e_n$ and μ^* the least solution of $\mathcal{E}$. Our strategy improvement algorithm is given as algorithm **??**. It starts with a $\vee$-strategy $\overline{\pi}$ for $\mathcal{E}$ and *feasible* pre-solution $\overline{\mu} \leq \mu^*$ of $\mathcal{E}(\pi)$.

Algorithm 1 Computing Least Solutions of Systems of Integer Equations

$\pi \leftarrow \overline{\pi};\ \mu \leftarrow \overline{\mu};$
while (μ is not a solution of $\mathcal{E}$) {
  $\pi \leftarrow P_\vee^{eager}(\pi, \mu)$; $\mu \leftarrow$ least solution of $\mathcal{E}(\pi)$ that is greater than or equal to μ;
}
return μ;

By induction one can show that algorithm **??** returns the least solution μ^* of $\mathcal{E}$ whenever it terminates (cf. [**?**]). In order to obtain an upper bound to the number of iterations, we first show that every system of conjunctive equations has at most one feasible solution.

Lemma 8. *Assume that the greatest solution μ^* of the system $\mathcal{E}$ of conjunctive equations is feasible. Then μ^* is the only feasible solution of $\mathcal{E}$.*

Proof. Assume that $\mathcal{E}$ denotes the system $\mathbf{x}_i = e_i$, $i = 1, \ldots, n$. We first prove the statement for a system $\mathcal{E}$ of *basic* equations. Let $\overline{\mathbf{X}}(\mathcal{E})$ denote the set of variables occurring in right-hand sides of $\mathcal{E}$. Let μ be a feasible solution of $\mathcal{E}$. We show by induction on $|\overline{\mathbf{X}}(\mathcal{E})|$ that $\mu = \mu^*$. This is obviously fulfilled, if $|\overline{\mathbf{X}}(\mathcal{E})| = 0$. Thus, consider an equation $\mathbf{x}_i = e_i$ of $\mathcal{E}$ where $\mathbf{x}_i$ occurs in a right-hand side e_j of $\mathcal{E}$.

Assume that 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 $\mathcal{D}(\mathcal{E}') \subseteq \mathcal{D}(\mathcal{E})$, μ, μ^* are feasible solutions of $\mathcal{E}'$. Since $|\overline{\mathbf{X}}(\mathcal{E}')| = |\overline{\mathbf{X}}(\mathcal{E})| - 1$, the induction hypothesis implies $\mu = \mu^*$.

Assume now that e_i contains $\mathbf{x}_i$. Since $\mathbf{x}_i \leq e_i \in \mathcal{D}(\mathcal{E})$ and μ, μ^* are feasible solutions we get $[\![\mathbf{x}_i]\!]\mu = [\![\mathbf{x}_i]\!]\mu^* = \infty$. Thus μ, μ^* are solutions of the system $\mathcal{E}'$ obtained from $\mathcal{E}$ by replacing the equation $\mathbf{x}_i = e_i$ with $\mathbf{x}_i = \infty$ and then replacing all occurrences of the variable $\mathbf{x}_i$ in right-hand sides with ∞. Since $\mathcal{D}(\mathcal{E}') \subseteq \mathcal{D}(\mathcal{E})$, μ, μ^* are feasible solutions of $\mathcal{E}'$. Since $|\overline{\mathbf{X}}(\mathcal{E}')| = |\overline{\mathbf{X}}(\mathcal{E})| - 1$, the induction hypothesis implies $\mu = \mu^*$. Thus the statement holds for systems of basic equations.

Now assume that $\mathcal{E}$ is a system of *conjunctive* equations. In order to derive a contradiction, assume that $\mu < \mu^*$ is a feasible solution of $\mathcal{E}$. Then μ is a feasible solution of $\mathcal{E}(\pi)$ for some $\wedge$-strategy π. Thus μ is the greatest solution of $\mathcal{E}(\pi)$. The greatest solution of $\mathcal{E}(\pi)$ is greater than or equal to μ^*. Thus, $\mu \geq \mu^*$ — contradiction. □

Consider algorithm **??**. Let π_j be the $\vee$-strategy π after the execution of the first statement in the j-th iteration. Let μ_j be the variable assignment μ at this point and μ'_j the variable assignment μ after the j-th iteration. The sequence (μ'_j) is strictly increasing until the least solution is reached. Lemma **??** implies that, for every j, μ_j and μ'_j is a feasible pre-solution of $\mathcal{E}(\pi_j)$. Thus, Lemma **??** implies that μ'_j is the greatest solution of $\mathcal{E}(\pi_j)$. This has two important consequences. The first consequence is that, since $\mathcal{E}(\pi_j)$ is a system of *conjunctive* equations, the greatest solution μ'_j can be computed in time $\mathcal{O}(|\mathbf{X}_\mathcal{E}| \cdot |\mathcal{E}|)$ using Bellman-Ford's algorithm (cf. [**?**]). The second consequence is that every strategy π_j is considered at most once. Otherwise, there exist $j' > j$ such that $\pi_{j'} = \pi_j$ implying that $\mu'_{j'} = \mu'_j$ which is a contradiction to the fact that (μ'_j) is strictly increasing. Thus, the number of iterations is bounded by the number of $\vee$-strategies.

In order to give a precise characterization of the run-time, let $\mathit{\Pi}(m)$ denote the maximal number of updates of strategies necessary for a system with m $\vee$-subexpressions. Thereby we assume that $\overline{\pi}$ and $\overline{\mu}$ are given. $\mathit{\Pi}(m)$ is trivially bounded by 2^m.

Until now we have assumed that we have a $\vee$-strategy $\overline{\pi}$ and a feasible pre-solution $\overline{\mu} \leq \mu^*$ of $\mathcal{E}(\overline{\pi})$ at hand. In order to lift this restriction, we consider $\mathcal{E}^{\vee-\infty}$ in place of $\mathcal{E}$ which we define to be the system $\mathbf{x}_1 = e_1 \vee -\infty, \ldots, \mathbf{x}_n = e_n \vee -\infty$. Then we can choose $\overline{\pi}$ to be the $\vee$-strategy which maps every top-level $\vee$-expression $e_i \vee -\infty$ of $\mathcal{E}^{\vee-\infty}$ to $-\infty$. Accordingly, we choose $\overline{\mu}$ to be the variable assignment which maps every variable to $-\infty$. Then $\overline{\mu} \leq \mu^*$ is a feasible solution of $\mathcal{E}^{\vee-\infty}(\overline{\pi})$.

We now show that the number of updates of strategies necessary for computing the least solution of $\mathcal{E}^{\vee-\infty}$ is $n + \mathit{\Pi}(m)$ although $|M(\mathcal{E}^{\vee-\infty})| = m + n$. We have:

Lemma 9. *$\mu'_n(\mathbf{x}_i) = -\infty$ iff $\mu^*(\mathbf{x}_i) = -\infty$ for $i = 1, \ldots, n$.* □

Let $i \in \{1, \ldots, n\}$. Lemma **??** implies $\mu^*(\mathbf{x}_i) \geq \mu'_j(\mathbf{x}_i) = -\infty$ for all $j \geq n$ iff $\mu'_n(\mathbf{x}_i) = -\infty$ Since μ'_j is a feasible solution of $\mathcal{E}(\pi_j)$, we get $\pi_j(e_i \vee -\infty) = \pi_n(e_i \vee -\infty)$ for all $j \geq n$. Thus, after n iterations we can consider the following iterations as iterations for the system obtained by replacing every right-hand side $e_i \vee -\infty$ with $\pi_n(e_i \vee -\infty)$. This system has m $\vee$-expressions. Thus, the number of iterations is bounded by $n + \Pi(m)$. Summarizing, we have:

Theorem 3. *The least solution of a system $\mathcal{E}$ of integer equations can be computed in time $\mathcal{O}(|\mathbf{X}_\mathcal{E}| \cdot |\mathcal{E}| \cdot \Pi(|M(\mathcal{E})|))$.* □

In contrast to the algorithm presented in [**?**], our new algorithm no longer relies on an instrumentation of the underlying lattice. For systems of *simple* integer equations we can improve on the number of iteration, if we use a different improvement operator.

Assume now that $\mathcal{E}$ is a system of *simple* integer equations. We now also consider *partial $\vee$-strategies* π, i.e., the domain $dom(\pi)$ of a partial $\vee$-strategy π is a subset of $M(\mathcal{E})$. Then we set

$$(e \vee e')\pi = \begin{cases} (\pi(e \vee e'))\pi & \text{if } e \vee e' \in dom(\pi) \\ e\pi \vee e'\pi & \text{if } e \vee e' \notin dom(\pi). \end{cases}$$

Let $M \subseteq M(\mathcal{E})$. We define the $\vee$-strategy improvement operator $P^M_\vee$ by

$$P^M_\vee(\pi, \mu) = \begin{cases} P^{eager}_\vee(\pi, \mu)|_M \cup \pi|_{M(\mathcal{E}) \setminus M} & \text{if } P^{eager}_\vee(\pi, \mu)|_M \neq \pi|_M \\ P^{eager}_\vee(\pi, \mu) & \text{if } P^{eager}_\vee(\pi, \mu)|_M = \pi|_M. \end{cases}$$

Intuitively, $P^M_\vee$ first tries to improve at $\vee$-expressions from M. Only if such an improvement is not possible, $\vee$-expressions from $M(\mathcal{E}) \setminus M$ are considered.

Assume that $\mathcal{E}$ is a system of *conjunctive simple* equations. All derived constraints in $\mathcal{D}(\mathcal{E})$ can be rewritten to the form $\mathbf{x} \leq \mathbf{y} + a$ or $\mathbf{x} \leq c$ where $\mathbf{x}, \mathbf{y}$ are variables, $a \in \mathbb{Z}$ and $c \in \overline{\mathbb{Z}}$. We call $\mathcal{E}$ *feasible* iff $a > 0$ for all derived constraints $\mathbf{x} \leq \mathbf{x} + a \in \mathcal{D}(\mathcal{E})$ and $\mathbf{x} \leq -\infty \in \mathcal{D}(\mathcal{E})$ implies that $\mathbf{x} = -\infty$ is an equation of $\mathcal{E}$. The greatest solution μ' of a feasible system $\mathcal{E}$ of simple conjunctive equations is feasible.

Assume now that $\mathcal{E}$ denotes a system of *simple* integer equations with least solution μ^*. A $\vee$-strategy π for $\mathcal{E}$ is called *feasible* iff $\mathcal{E}(\pi)$ is feasible. Similar to Lemma **??** it can be shown that algorithm **??** considers feasible strategies, only. For systems of simple equations we have the following property:

Lemma 10. *Let $\mathcal{E}$ be the system $\mathbf{x}_1 = e_1, \ldots, \mathbf{x}_n = e_n$ of simple integer equations and μ a solution of $\mathcal{E}$. Assume that π is a feasible $\vee$-strategy with $e_i\pi = -\infty$ whenever $\mu(\mathbf{x}_i) = -\infty$ for $i = 1, \ldots, n$. Let μ_π be the greatest solution of $\mathcal{E}(\pi)$. Then $\mu_\pi \leq \mu$.*

Proof. Note that μ_π is a feasible solution of $\mathcal{E}(\pi)$ and μ is a post-solution of $\mathcal{E}(\pi)$. Let $\mu^{(0)} := \mu$ and, for $j \in \mathbb{N}$, let $\mu^{(j+1)}$ be defined by $\mu^{(j+1)}(\mathbf{x}_i) = [\![e_i\pi]\!]\mu^{(j)}$. Then $\mu' := \bigwedge_{j \in \mathbb{N}_0} \mu^{(j)} \leq \mu$ is a solution of $\mathcal{E}(\pi)$ and, since $a > 0$ for all derived constraints $\mathbf{x} \leq \mathbf{x} + a \in \mathcal{D}(\mathcal{E}(\pi))$, $\mu'(\mathbf{x}_i) = -\infty$ implies $\mu(\mathbf{x}_i) = -\infty$ which implies $e_i\pi = -\infty$ for $i = 1, \ldots, n$. Thus, μ' is a feasible solution of $\mathcal{E}(\pi)$. Since, by Lemma **??**, $\mu_\pi = \mu'$, we get $\mu' \leq \mu$. □

Consider the sequences (μ_j), (μ'_j) and (π_j) which we obtain from algorithm **??** using the $\vee$-strategy improvement operator $P^M_\vee$. We show that there do not exist indexes $j < k$ with $j \geq n$ such that $\pi_k|_{M(\mathcal{E})\setminus M} = \pi_j|_{M(\mathcal{E})\setminus M} \neq \pi_{j+1}|_{M(\mathcal{E})\setminus M}$ $(*)$. In order to derive a contradiction, assume the opposite. By the definition of $P^M_\vee$, μ'_j is a solution of $\mathcal{E}' := \mathcal{E}(\pi_j|_{M(\mathcal{E})\setminus M})$. Furthermore, μ'_k is the greatest solution of the feasible system $\mathcal{E}(\pi_k) = \mathcal{E}(\pi_k|_{M(\mathcal{E})\setminus M})(\pi_k|_M) = \mathcal{E}(\pi_j|_{M(\mathcal{E})\setminus M})(\pi_k|_M) = \mathcal{E}'(\pi_k|_M)$. Since $k > j \geq n$, we have $e_i\pi_k = -\infty$ whenever $\mu'_j(\mathbf{x}_i) = -\infty$. Thus we can apply Lemma **??** which implies that $\mu'_k \leq \mu'_j$. This contradicts the fact that (μ'_j) is strictly ascending.

We use the $\vee$-improvement operator $P^{M_c(\mathcal{E})}_\vee$, i.e., $M = M_c(\mathcal{E})$. The $\vee$-improvement operator $P^{M_c(\mathcal{E})}_\vee$ first tries to improve at expressions $e \vee e' \in M_c(\mathcal{E})$ and only if this is not possible it tries to improve at expressions $e \vee e' \in M_{nc}(\mathcal{E})$. For $M \subseteq M(\mathcal{E})$, we call j an *update index* on M iff $\pi_{j+1}|_M \neq \pi_j|_M$. Assume that $e \vee e' \in M_c(\mathcal{E})$ where w.l.o.g. e' is a constant expression. Then, since μ_j is ascending, if there is a k such that $\pi_k(e \vee e') = e$, then $\pi_j(e \vee e') = e$ for all $j \geq k$. Thus, there are at most $|M_c(\mathcal{E})|$ update indexes on $M_c(\mathcal{E})$ $(**)$.

Let j_i denote the sequence of update indexes on $M_{nc}(\mathcal{E})$. By $(*)$, these are at most $2^{|M_{nc}(\mathcal{E})|}$. Between two update on $M_{nc}(\mathcal{E})$ there must be updates on $M_c(\mathcal{E})$. By $(**)$ the overall number of updates on $M_c(\mathcal{E})$ is bounded by $|M_c(\mathcal{E})|$, i.e., $\sum_i j_{i+1} - j_i - 1 \leq |M_c(\mathcal{E})|$. Thus, the number of strategies is bounded by $2^{|M_{nc}(\mathcal{E})|} + |M_c(\mathcal{E})|$. We denote the maximal number of updates of strategies on $M_{nc}(\mathcal{E})$ necessary for solving a simple system $\mathcal{E}$ by $\Pi_s(|M_{nc}(\mathcal{E})|)$. We obtain:

Theorem 4. *The least solution of a system $\mathcal{E}$ of simple integer equations can be computed in time $\mathcal{O}(|\mathbf{X}_\mathcal{E}| \cdot |\mathcal{E}| \cdot (\Pi_s(|M_{nc}(\mathcal{E})|) + |M_c(\mathcal{E})|))$.* □

The practical run-time of our algorithm is quite comparable to the discrete strategy improvement algorithm by Vöge and Jurdziński [**?**]. The number $\Pi_s(|M_{nc}(\mathcal{E})|))$ corresponds to the number of strategy improvements for the parity game. For each improvement-step, we need $\mathcal{O}(n \cdot |\mathcal{E}_G|_{[-m,m]}|)$ operations where arithmetic operations are on numbers of size $\mathcal{O}(d \cdot \log n)$. The improvement-step of the discrete strategy improvement algorithm by Vöge and Jurdziński [**?**] uses also $\mathcal{O}(n \cdot |\mathcal{E}_G|_{[-m,m]}|)$ operations — but arithmetic operations are just on numbers of size $\mathcal{O}(\log n)$.

6 Conclusion

By encoding parity games into integer equations, we have provided a lower complexity bound for precise interval analysis of affine programs. Additionally, we provided a simplified version of the algorithm in [**?**] for solving integer equations. As in the algorithm of [**?**] for rational equations, the new version for integers avoids the instrumentation of the underlying lattice. The restriction to integers, on the other hand also allowed to improve on the complicated treatment of conjunctive systems in [**?**] for rationals.

The methods which we have presented here, can be applied to simplify the algorithm for interval equations from [**?**] where also multiplication of arbitrary interval expressions is allowed. By modifying the strategy improvement operator, we also have obtained an algoithm for *simple* integer equations where for the complexity estimation only non-constant maxima must be taken into account. By our encoding of parity

games into integer systems, we thus obtain a simple but efficient strategy improvement algorithm for computing winning regions and winning strategies of parity games.

References

1. L. Bordeaux, Y. Hamadi, and M. Y. Vardi. An Analysis of Slow Convergence in Interval Propagation. In *13th Int. Conf. on Principles and Practice of Constraint Programming (CP)*, volume 4741 of *LNCS*, pages 790–797. Springer, 2007.
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 *Proceedings of 4th ACM Symposium on Principles of Programming Languages (POPL)*, pages 238–252. ACM Press, 1977.
5. 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.
6. E. A. Emerson and C. S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *FOCS*, pages 368–377. IEEE, 1991.
7. T. Gawlitza, J. Reineke, H. Seidl, and R. Wilhelm. Polynomial Exact Interval Analysis Revisited. Technical report, TU München, 2006.
8. T. Gawlitza and H. Seidl. Precise Fixpoint Computation Through Strategy Iteration. In *European Symposium on Programming (ESOP)*, pages 300–315. Springer Verlag, LNCS 4421, 2007.
9. T. Gawlitza and H. Seidl. Precise Relational Invariants Through Strategy Iteration. In *Computer Science Logic, 21st Int. Workshop (CSL)*, LNCS 4646, pages 23–40. Springer, 2007.
10. M. Jurdziński. Deciding the winner in parity games is in UP $\cap$ co-UP. *Inf. Process. Lett.*, 68(3):119–124, 1998.
11. M. Jurdziński, M. Paterson, and U. Zwick. A Deterministic Subexponential Algorithm for Solving Parity Games. In *17th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 117–123, 2006.
12. M. Karr. Affine Relationships Among Variables of a Program. *Acta Informatica*, 6:133–151, 1976.
13. J. Leroux and G. Sutre. Accelerated data-flow analysis. In H. R. Nielson and G. Filé, editors, *SAS*, volume 4634 of *Lecture Notes in Computer Science*, pages 184–199. Springer, 2007.
14. A. Puri. *Theory of Hybrid and Discrete Systems*. PhD thesis, University of California, Berkeley, 1995.
15. Z. Su and D. Wagner. A class of polynomially solvable range constraints for interval analysis without widenings and narrowings. In K. Jensen and A. Podelski, editors, *TACAS*, volume 2988 of *Lecture Notes in Computer Science*, pages 280–295. Springer, 2004.
16. J. Vöge. *Strategiesynthese für Paritätsspiele auf endlichen Graphen*. PhD thesis, RWTH Aachen, 2000.
17. J. Vöge and M. Jurdziński. A Discrete Strategy Improvement Algorithm for Solving Parity Games. In *Computer Aided Verification, 12th Int. Conf. (CAV)*, pages 202–215. LNCS 1855, Springer, 2000.