Precise Widening Operators for Convex Polyhedra*

Roberto Bagnara[1], Patricia M. Hill[2], Elisa Ricci[1], and Enea Zaffanella[1]

[1] Department of Mathematics, University of Parma, Italy
{bagnara,ericci,zaffanella}@cs.unipr.it
[2] School of Computing, University of Leeds, UK
hill@comp.leeds.ac.uk

Abstract. Convex polyhedra constitute the most used abstract domain among those capturing numerical relational information. Since the domain of convex polyhedra admits infinite ascending chains, it has to be used in conjunction with appropriate mechanisms for enforcing and accelerating convergence of the fixpoint computation. Widening operators provide a simple and general characterization for such mechanisms. For the domain of convex polyhedra, the original widening operator proposed by Cousot and Halbwachs amply deserves the name of *standard widening* since most analysis and verification tools that employ convex polyhedra also employ that operator. Nonetheless, there is demand for more precise widening operators that still has not been fulfilled. In this paper, after a formal introduction to the standard widening where we clarify some aspects that are often overlooked, we embark on the challenging task of improving on it. We present a framework for the systematic definition of new and precise widening operators for convex polyhedra. The framework is then instantiated so as to obtain a new widening operator that combines several heuristics and uses the standard widening as a last resort so that it is never less precise. A preliminary experimental evaluation has yielded promising results. We also suggest an improvement to the well-known widening delay technique that allows to gain precision while preserving its overall simplicity.

1 Introduction

An ability to reason about numerical quantities is crucial for increasing numbers of applications in the field of automated analysis and verification of complex systems. Of particular interest are representations that capture *relational* information, that is, information relating different quantities such as, for example, the length of a buffer and the contents of a program variable, or the number of agents in different states in the modeling of a distributed protocol.

* This work has been partly supported by MURST projects “Aggregate- and number-reasoning for computing: from decision algorithms to constraint programming with multisets, sets, and maps” and “Constraint Based Verification of Reactive Systems”.

Convex polyhedra, since the work of Cousot and Halbwachs [18], constitute the most used abstract domain among those capturing numerical, relational information. They have been used to solve, by abstract interpretation [15], several important data-flow analysis problems such as array bound checking, compile-time overflow detection, loop invariant computations and loop induction variables. Convex polyhedra are also used, among many other applications, for the analysis and verification of synchronous languages [6, 23] and of linear hybrid automata (an extension of finite-state machines that models time requirements) [24, 27], for the computer-aided formal verification of concurrent and reactive systems based on temporal specifications [29], for inferring argument size relationships in logic languages [4, 5], for the automatic parallelization of imperative programs [31], for detecting buffer overflows in C [21], and for the automatic generation of the ranking functions needed to prove progress properties [10].

Since the domain of convex polyhedra admits infinite ascending chains, it has to be used in conjunction with appropriate mechanisms for enforcing and accelerating convergence of the fixpoint computation. *Widening operators* [14, 15, 17] provide a simple and general characterization for such mechanisms. In its simplest form, a widening operator on a poset $(L, \sqsubseteq)$ is defined as a partial function $\nabla \colon L \times L \rightarrowtail L$ satisfying:

1. for each $x, y \in L$ such that $x \mathbin{\nabla} y$ is defined, we have $x \sqsubseteq x \mathbin{\nabla} y$ and $y \sqsubseteq x \mathbin{\nabla} y$;
2. for all increasing chains $y_0 \sqsubseteq y_1 \sqsubseteq \cdots$, the increasing chain defined by $x_0 \stackrel{\mathrm{def}}{=} y_0, \ldots, x_{i+1} \stackrel{\mathrm{def}}{=} x_i \mathbin{\nabla} y_{i+1}, \ldots$ is not strictly increasing.

It must be observed that a widening operator may serve different purposes, besides forcing the stabilization of approximated iteration sequences after a finite number of iterations: it may be used to speed up the convergence of iteration sequences and to ensure the existence of the approximations of concrete elements when considering abstract domains that are algebraically weak [16]. Thus a widening does not need to be a total function, the only requirement is that its domain of definition be compatible with the intended application. The application will also affect the required trade-off between precision and efficiency: when speeding up convergence of an (perhaps intrinsically finite) iteration sequence, precision is more willingly given away; in other cases, the objective is to ensure termination without compromising precision too much. As a consequence, it is meaningful to have two or more widening operators, each one tuned with a different compromise between precision and efficiency. The different widenings can be used in different applications or even in the same application, with the system dynamically switching from one to another [12].

For the domain of convex polyhedra, the first widening operator was proposed by Cousot and Halbwachs in [18] and further refined in [22]. It amply deserves the name of *standard widening* since most analysis and verification tools that employ convex polyhedra also employ that operator.

There are a number of applications of convex polyhedra in the field of systems' analysis and verification that are particularly sensitive to the precision of the deduced numerical information. The importance of precision in the field of automated verification has led to the use of *extrapolation operators*, that is, binary operators satisfying condition 1 in the definition of widening but not condition 2 (i.e., without convergence guarantees). For instance, in [26], Henzinger and Ho propose a new extrapolation operator for use in the HyTech model checker since “Halbwachs's widening operator [...] is sometimes too coarse for [their] purposes” (symbolic model checking of linear hybrid systems). A further step toward more precision is proposed in [28], where the authors present another extrapolation operator used in the HyTech system: “This operator is tighter than (and therefore less aggressive than) both the widening operator of [23] and the extrapolation operator of [26], which is not monotone in its second argument.” Other extrapolation operators based on similar approaches have been sketched in [6]. Still in the field of automatic verification, the need for more precision than warranted by the standard widening is remarked in both [9] and [19]; and a new extrapolation operator on sets of convex polyhedra is defined in each of these papers.

If giving up convergence guarantees is acceptable (though not desirable) for semi-automatic, human-operated verifiers, this is certainly not the case for fully-automatic program analyzers. In this field, the request for more precision has traditionally been satisfied by the delayed application of the widening, a general idea suggested in [12]. This amounts to delaying the application of the widening operator k times for some fixed parameter $k \in \mathbb{N}$. A study of the effect of alternative values for k in the automatic determination of linear size relations between the arguments of logic programs has been conducted in [4, 5]. One application of this idea is in termination inference [30]. In order to achieve reasonable precision, the cTI analyzer runs with $k = 3$ as a default, but there are simple programs (such as *mergesort*) whose termination can only be established with $k > 3$. On the other hand, setting $k = 4$ as the default can have a sensible impact on performance of cTI [F. Mesnard, personal communication, 2003].

In this paper, after a formal introduction to the standard widening where we clarify some important aspects that are often overlooked, we embark on the challenging task of improving on it. Elaborating on an idea originally proposed in [6], we present a framework for the systematic definition of new and precise widening operators for convex polyhedra, which is based on the definition of a suitable relation on convex polyhedra satisfying the ascending chain condition. The framework makes it particularly easy to combine several heuristics and prove that the resulting operator is indeed a widening. Here we instantiate it with a selection of extrapolation operators —some of which embody improvements of heuristics already proposed in the literature— and the standard widening so that the new widening operator is always at least as precise as the standard one for a single application. An experimental evaluation of the new widening shows that, for the analysis problem considered, it captures common growth patterns and obtains precision improvements in as many as 33% of the benchmarks.

The paper is structured as follows: Section 2 recalls the required concepts and notations; Section 3 introduces the standard widening, highlighting a few important aspects of its formal definition that are often overlooked; Section 4 presents a framework for the systematic definition of new widenings operators improving upon the standard widening; Section 5 instantiates this framework by considering several variants of extrapolations techniques proposed in the literature, as well as one that is new to this paper; Section 6 summarizes the results of our experimental evaluation of the new widening; Section 7 proposes an improvement of the well-known widening delay technique. Section 8 concludes.

2 Preliminaries

The cardinality of a set S is denoted by $\# S$. If M and N are finite multisets over $\mathbb{N}$, $\#(n, M)$ denotes the number of occurrences of $n \in \mathbb{N}$ in M and $M \gg N$ means that there exists $j \in \mathbb{N}$ such that $\#(j, M) > \#(j, N)$ and, for each $k \in \mathbb{N}$ with $k > j$, we have $\#(k, M) = \#(k, N)$. The relation $\gg$ is well-founded [20]. The set of non-negative reals is denoted by $\mathbb{R}_+$.

Any vector $\boldsymbol{v} \in \mathbb{R}^n$ is also regarded as a matrix in $\mathbb{R}^{n \times 1}$ so that it can be manipulated with the usual matrix operations of addition, multiplication (both by a scalar and by another matrix), and transposition, which is denoted by $\boldsymbol{v}^{\mathrm{T}}$. For each $i \in \{1, \ldots, n\}$, the i-th component of the vector $\boldsymbol{v} \in \mathbb{R}^n$ is denoted by v_i. The *scalar product* of $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^n$, denoted $\langle \boldsymbol{v}, \boldsymbol{w} \rangle$, is $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{w} = \sum_{i=1}^{n} v_i w_i$. The vector of $\mathbb{R}^n$ having all components equal to zero is denoted by $\boldsymbol{0}$. We write $\boldsymbol{v} = \boldsymbol{w}$ to denote the conjunctive proposition $\bigwedge_{i=1}^{n}(v_i = w_i)$. In contrast, $\boldsymbol{v} \neq \boldsymbol{w}$ will denote the proposition $\neg(\boldsymbol{v} = \boldsymbol{w})$.

Let $V = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\} \subseteq \mathbb{R}^n$ be a finite set of vectors. For all scalars λ_1, …, $\lambda_k \in \mathbb{R}$, the vector $\boldsymbol{v} = \sum_{i=1}^{k} \lambda_i \boldsymbol{v}_i$ is said to be a *linear combination* of the vectors in V. Such a combination is said to be

- a *positive* (or *conic*) combination, if $\forall i \in \{1, \ldots, k\} : \lambda_i \in \mathbb{R}_+$;
- an *affine* combination, if $\sum_{i=1}^{k} \lambda_i = 1$;
- a *convex* combination, if it is both positive and affine.

The vectors in V are said *linearly independent* if the only solution of the equation $\sum_{i=1}^{k} \lambda_i \boldsymbol{v}_i = \boldsymbol{0}$ is $\lambda_i = 0$, for each $i = 1$, …, k; they are said *affinely independent* if the only solution of the system of equations $\bigl\{ \sum_{i=1}^{k} \lambda_i \boldsymbol{v}_i = \boldsymbol{0},\ \sum_{i=1}^{k} \lambda_i = 0 \bigr\}$ is $\lambda_i = 0$, for each $i = 1$, …, k.

Let $V \subseteq \mathbb{R}^n$. The subspace of $\mathbb{R}^n$ defined by the set of all affine combinations of finite subsets of V is called the *affine hull* of V and denoted by $\mathrm{aff.hull}(V)$; the *orthogonal* of V is $V^{\perp} = \bigl\{ \boldsymbol{w} \in \mathbb{R}^n \bigm| \forall \boldsymbol{v} \in V : \langle \boldsymbol{v}, \boldsymbol{w} \rangle = 0 \bigr\}$; the set $\{ -\boldsymbol{v} \in \mathbb{R}^n \mid \boldsymbol{v} \in V \}$ is denoted by $-V$.

For each vector $\boldsymbol{a} \in \mathbb{R}^n$ and scalar $b \in \mathbb{R}$, where $\boldsymbol{a} \neq \boldsymbol{0}$, the linear inequality constraint $\langle \boldsymbol{a}, \boldsymbol{x} \rangle \geq b$ defines a topologically closed affine half-space of $\mathbb{R}^n$. We do not distinguish between syntactically different constraints defining the same affine half-space so that, for example, $x \geq 2$ and $2x \geq 4$ are the same constraint. The set $\mathcal{P} \subseteq \mathbb{R}^n$ is a (*closed* and *convex*) *polyhedron* if and only if either $\mathcal{P}$ can be expressed as the intersection of a finite number of closed affine half-spaces of $\mathbb{R}^n$, or $n = 0$ and $\mathcal{P} = \varnothing$. The set of all closed polyhedra on $\mathbb{R}^n$ is denoted by $\mathbb{CP}_n$. In this paper, we only consider polyhedra in $\mathbb{CP}_n$ when $n > 0$. The set

$\mathbb{CP}_n$, when partially ordered by subset inclusion, is a lattice where the binary meet operation is set-intersection; the binary join operation, denoted $\uplus$, is called *convex polyhedral hull*, *poly-hull* for short.

If $k+1 \leq n+1$ is the maximum number of affinely independent points of a polyhedron $\mathcal{P} \in \mathbb{CP}_n$, then we write $\dim(\mathcal{P}) = k$ and we say that $\mathcal{P}$ has dimension k. If $\mathcal{P} \neq \varnothing$, the *characteristic cone* of $\mathcal{P}$ is given by the set $\operatorname{char.cone}(\mathcal{P}) \stackrel{\mathrm{def}}{=} \{\, \boldsymbol{w} \in \mathbb{R}^n \mid \forall \boldsymbol{v} \in \mathcal{P} : \boldsymbol{v} + \boldsymbol{w} \in \mathcal{P} \,\}$ whereas the *lineality space* of $\mathcal{P}$ is $\operatorname{lin.space}(\mathcal{P}) \stackrel{\mathrm{def}}{=} \operatorname{char.cone}(\mathcal{P}) \cap -\operatorname{char.cone}(\mathcal{P})$.

The linear equality constraint $\langle \boldsymbol{a}, \boldsymbol{x} \rangle = b$ defines an affine hyperplane of $\mathbb{R}^n$ (i.e., the intersection of the affine half-spaces $\langle \boldsymbol{a}, \boldsymbol{x} \rangle \geq b$ and $\langle -\boldsymbol{a}, \boldsymbol{x} \rangle \geq -b$). Each polyhedron $\mathcal{P} \in \mathbb{CP}_n$ can therefore be represented by a finite set of linear equality and inequality constraints $\mathcal{C}$ called a *constraint system*. We write $\mathcal{P} = \operatorname{con}(\mathcal{C})$. The subsets of equality and inequality constraints in system $\mathcal{C}$ are denoted by $\operatorname{eq}(\mathcal{C})$ and $\operatorname{ineq}(\mathcal{C})$, respectively. When $\mathcal{P} = \operatorname{con}(\mathcal{C}) \neq \varnothing$, we say that constraint system $\mathcal{C}$ is in *minimal form* if $\#\operatorname{eq}(\mathcal{C}) = n - \dim(\mathcal{P})$ and there does not exist $\mathcal{C}' \subset \mathcal{C}$ such that $\operatorname{con}(\mathcal{C}') = \mathcal{P}$. All the constraint systems in minimal form describing a given polyhedron have the same cardinality.

Let $\mathcal{P} \in \mathbb{CP}_n$. A vector $\boldsymbol{p} \in \mathcal{P}$ is called a *point* of $\mathcal{P}$; a vector $\boldsymbol{r} \in \mathbb{R}^n$, where $\boldsymbol{r} \neq \boldsymbol{0}$, is called a *ray* of $\mathcal{P}$ if $\mathcal{P} \neq \varnothing$ and $\boldsymbol{p} + \lambda \boldsymbol{r} \in \mathcal{P}$, for all points $\boldsymbol{p} \in \mathcal{P}$ and all $\lambda \in \mathbb{R}_+$; a vector $\boldsymbol{l} \in \mathbb{R}^n$ is called a *line* of $\mathcal{P}$ if both $\boldsymbol{l}$ and $-\boldsymbol{l}$ are rays of $\mathcal{P}$. We do not distinguish between rays (resp., lines) differing by a positive (resp., non-null) factor so that, for example, $(1,3)^{\mathrm{T}}$ and $(2,6)^{\mathrm{T}}$ are the same ray.

Given three finite sets of vectors $L, R, P \subseteq \mathbb{R}^n$ such that $L = \{\boldsymbol{l}_1, \ldots, \boldsymbol{l}_\ell\}$, $R = \{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_r\}$, $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_p\}$ and $\boldsymbol{0} \notin L \cup R$, then the triple $\mathcal{G} = (L, R, P)$ is called a *generator system* for the polyhedron

$$\operatorname{gen}(\mathcal{G}) = \left\{ \sum_{i=1}^{\ell} \lambda_i \boldsymbol{l}_i + \sum_{i=1}^{r} \rho_i \boldsymbol{r}_i + \sum_{i=1}^{p} \pi_i \boldsymbol{p}_i \;\middle|\; \begin{array}{l} \boldsymbol{\lambda} \in \mathbb{R}^\ell, \boldsymbol{\rho} \in \mathbb{R}^r_+, \boldsymbol{\pi} \in \mathbb{R}^p_+, \\ \sum_{i=1}^{p} \pi_i = 1 \end{array} \right\}.$$

The polyhedron $\operatorname{gen}(\mathcal{G})$ is empty if and only if $P = \varnothing$. If $P \neq \varnothing$, the vectors in L, R and P are lines, rays and points of $\operatorname{gen}(\mathcal{G})$, respectively. We define an ordering '$\preceq$' on generator systems such that, for any generator systems $\mathcal{G}_1 = (L_1, R_1, P_1)$ and $\mathcal{G}_2 = (L_2, R_2, P_2)$, $\mathcal{G}_1 \preceq \mathcal{G}_2$ if and only if $L_1 \subseteq L_2$, $R_1 \subseteq R_2$ and $P_1 \subseteq P_2$; if, in addition, $\mathcal{G}_1 \neq \mathcal{G}_2$, we write $\mathcal{G}_1 \prec \mathcal{G}_2$. When $\operatorname{gen}(\mathcal{G}) \neq \varnothing$, the generator system $\mathcal{G} = (L, R, P)$ is said to be in *minimal form* if $\# L = \dim\bigl(\operatorname{lin.space}(\mathcal{P})\bigr)$ and there does not exist a generator system $\mathcal{G}' \prec \mathcal{G}$ such that $\operatorname{gen}(\mathcal{G}') = \operatorname{gen}(\mathcal{G})$.

Let $c = \bigl(\langle \boldsymbol{a}, \boldsymbol{x} \rangle \bowtie b\bigr)$ be a linear constraint, where $\bowtie \in \{\geq, =\}$. We say that a point (resp., a ray or a line) $\boldsymbol{v}$ *saturates* constraint c if and only if $\langle \boldsymbol{a}, \boldsymbol{v} \rangle = b$ (resp., $\langle \boldsymbol{a}, \boldsymbol{v} \rangle = 0$). For each point $\boldsymbol{p}$ and constraint system $\mathcal{C}$, we define the constraint system

$$\operatorname{sat\_con}(\boldsymbol{p}, \mathcal{C}) \stackrel{\mathrm{def}}{=} \{\, c \in \mathcal{C} \mid \boldsymbol{p} \text{ saturates } c \,\};$$

for each constraint c and generator system $\mathcal{G} = (L, R, P)$, we define the generator system $\mathrm{sat\_gen}(c, \mathcal{G}) = (L', R', P')$, where

$$L' \stackrel{\mathrm{def}}{=} \{\, \boldsymbol{l} \in L \mid \boldsymbol{l} \text{ saturates } c \,\},$$
$$R' \stackrel{\mathrm{def}}{=} \{\, \boldsymbol{r} \in R \mid \boldsymbol{r} \text{ saturates } c \,\},$$
$$P' \stackrel{\mathrm{def}}{=} \{\, \boldsymbol{p} \in P \mid \boldsymbol{p} \text{ saturates } c \,\}.$$

A generator system $\mathcal{G} = (L, R, P)$ is in *orthogonal form* if it is in minimal form and $R \cup P \subseteq L^{\perp}$. All generator systems in orthogonal form describing a given polyhedron have identical sets of rays and points. A generator system in minimal form can be tranformed into an equivalent system in orthogonal form by means of the well-known Gram-Shmidt method. By duality, orthogonal forms can also be defined for constraint systems. For each linear constraint $c = \bigl(\langle \boldsymbol{a}, \boldsymbol{x} \rangle \bowtie b\bigr)$, let $c_{\boldsymbol{a}} = \boldsymbol{a}$. A constraint system $\mathcal{C}$ is in orthogonal form if it is in minimal form and $I \subseteq E^{\perp}$, where $I = \bigl\{\, c_{\boldsymbol{a}} \in \mathbb{R}^n \bigm| c \in \mathrm{ineq}(\mathcal{C}) \,\bigr\}$ and $E = \bigl\{\, c_{\boldsymbol{a}} \in \mathbb{R}^n \bigm| c \in \mathrm{eq}(\mathcal{C}) \,\bigr\}$. All constraint systems in orthogonal form describing a given polyhedron have identical sets of inequality constraints.

3 The Standard Widening

The first widening on polyhedra was introduced in [18]. Intuitively, if $\mathcal{P}_1$ is the polyhedron obtained in the previous step of the upward iteration sequence and the current step yields polyhedron $\mathcal{P}_2$, then the widening of $\mathcal{P}_2$ with respect to $\mathcal{P}_1$ is the polyhedron defined by all the constraints of $\mathcal{P}_1$ that are satisfied by all the points of $\mathcal{P}_2$. An improvement on the above idea was defined in [22]. This operator, termed *standard widening*, has indeed been used almost universally.

The formal specification of the standard widening requires that each equality constraint is split into the two corresponding linear inequalities; thus, for each constraint system $\mathcal{C}$, we define

$$\mathrm{repr}_{\geq}(\mathcal{C}) \stackrel{\mathrm{def}}{=} \Bigl\{\, \langle -\boldsymbol{a}, \boldsymbol{x} \rangle \geq -b \Bigm| \bigl(\langle \boldsymbol{a}, \boldsymbol{x} \rangle = b\bigr) \in \mathcal{C} \,\Bigr\} \\ \cup \Bigl\{\, \langle \boldsymbol{a}, \boldsymbol{x} \rangle \geq b \Bigm| \bigl(\langle \boldsymbol{a}, \boldsymbol{x} \rangle \bowtie b\bigr) \in \mathcal{C}, \bowtie \in \{\geq, =\} \,\Bigr\}.$$

Definition 1. (Standard widening.) [22, Définition 5.3.3, p. 57] *For $i = 1$, 2, let $\mathcal{P}_i \in \mathbb{CP}_n$ be such that $\mathcal{P}_i = \mathrm{con}(\mathcal{C}_i)$* [and let $\mathcal{C}_1$ be either inconsistent or in minimal form]*. Then, the polyhedron $\mathcal{P}_1 \mathbin{\nabla} \mathcal{P}_2 \in \mathbb{CP}_n$ is defined as*

$$\mathcal{P}_1 \mathbin{\nabla} \mathcal{P}_2 = \begin{cases} \mathcal{P}_2, & \textit{if } \mathcal{P}_1 = \varnothing; \\ \mathit{con}(\mathcal{C}'_1 \cup \mathcal{C}'_2), & \textit{otherwise;} \end{cases}$$

where

$$\mathcal{C}'_1 = \Bigl\{\, \beta \in \mathrm{repr}_{\geq}(\mathcal{C}_1) \Bigm| \mathcal{P}_2 \subseteq \mathrm{con}\bigl(\{\beta\}\bigr) \,\Bigr\},$$
$$\mathcal{C}'_2 = \Bigl\{\, \gamma \in \mathrm{repr}_{\geq}(\mathcal{C}_2) \Bigm| \exists \beta \in \mathrm{repr}_{\geq}(\mathcal{C}_1) \mathrel{.} \mathcal{P}_1 = \mathrm{con}\Bigl(\bigl(\mathrm{repr}_{\geq}(\mathcal{C}_1) \setminus \{\beta\}\bigr) \cup \{\gamma\}\Bigr) \,\Bigr\}.$$

The constraints in $\mathcal{C}'_1$ are those that would have been selected when using the original proposal of [18], whereas the constraints in $\mathcal{C}'_2$ are added to ensure that this widening is a well-defined operator on the domain of polyhedra (i.e., it does not depend on the particular constraint representations).

The condition in square brackets that $\mathcal{C}_1$, when consistent, should be in minimal form, was implicit from the context of [22, Définition 5.3.3, p. 57], though not explicitly present in the definition itself. Such a requirement has been sometimes neglected in later papers discussing the standard widening (and also in some implementations), but it is actually needed in order to obtain a correct definition. In fact, the following two examples show that if a redundant (i.e., not minimal) constraint description is taken into account, then not only is the widening operator not well defined (see Example 1), but also the chain condition may be violated (see Example 2).

Example 1. For $i = 1, 2$, let $\mathcal{P}_i = \mathrm{con}(\mathcal{C}_i) \in \mathbb{CP}_2$, where

$$\begin{aligned} \mathcal{C}_1 &= \{x \geq 0, y \geq 0, x - y \geq 2\}, \\ \mathcal{C}_2 &= \{x \geq 2, y \geq 0\}. \end{aligned}$$

Note that the constraint $x \geq 0$ is redundant in $\mathcal{C}_1$. By applying [22, Définition 5.3.3, p. 57] verbatim, without enforcing minimization, we would obtain the polyhedron

$$\mathcal{P} = \mathrm{con}\bigl(\{x \geq 0, y \geq 0\}\bigr).$$

In contrast, by applying Definition 1, i.e., by enforcing minimization, we obtain the polyhedron

$$\mathcal{P}' = \mathrm{con}\bigl(\{y \geq 0\}\bigr).$$

Example 2. Consider, for each $k \in \mathbb{N}$, the polyhedron $\mathcal{P}_k \overset{\mathrm{def}}{=} \mathrm{con}(\mathcal{C}_k) \in \mathbb{CP}_1$, where

$$\mathcal{C}_k \overset{\mathrm{def}}{=} \left\{0 \leq x, x \leq \frac{k}{k+1}\right\} \cup \{x \leq 2\},$$

and note that no $\mathcal{C}_k$ is minimal since the constraint $x \leq 2$ is redundant in all of them. Moreover, the infinite chain constituted by the $\mathcal{P}_k$'s, that is, using an interval notation,

$$\mathcal{P}_0 = [0, 0],\ \mathcal{P}_1 = \left[0, \frac{1}{2}\right],\ \mathcal{P}_2 = \left[0, \frac{2}{3}\right],\ \mathcal{P}_3 = \left[0, \frac{3}{4}\right],\ \ldots,$$

is strictly increasing. We will now show that for the infinite chain $\mathcal{Q}_0 = \mathcal{P}_0$, $\ldots$, $\mathcal{Q}_{k+1} = \mathcal{Q}_k \mathbin{\nabla} \mathcal{P}_{k+1}$, $\ldots$ we have $\mathcal{Q}_n = \mathcal{P}_n$ for each $n \in \mathbb{N}$, so that the chain condition is violated.

For each $n \in \mathbb{N}$ we have $\mathcal{Q}_n = \operatorname{con}(\mathcal{D}_n)$, where $\mathcal{D}_0 \stackrel{\text{def}}{=} \mathcal{C}_0$ and

$$\begin{aligned} \mathcal{D}_{k+1} \stackrel{\text{def}}{=} \Bigl\{ \beta \in \mathcal{D}_k \Bigm| \mathcal{P}_{k+1} \subseteq \operatorname{con}\bigl(\{\beta\}\bigr) \Bigr\} \\ \cup \Bigl\{ \gamma \in \mathcal{C}_{k+1} \Bigm| \exists \beta \in \mathcal{D}_k \mathrel{.} \mathcal{Q}_k = \operatorname{con}\Bigl(\bigl(\mathcal{D}_k \setminus \{\beta\}\bigr) \cup \{\gamma\}\Bigr) \Bigr\}. \end{aligned}$$

We will show by induction that $\mathcal{D}_n = \mathcal{C}_n$ for each $n \in \mathbb{N}$. First we note that $\{0 \leq x, x \leq 2\} \subseteq \mathcal{D}_0 = \mathcal{C}_0$ and thus $\{0 \leq x, x \leq 2\} \subseteq \mathcal{D}_k$ for each $k \in \mathbb{N}$, since $\mathcal{P}_{k+1} \subseteq \operatorname{con}\bigl(\{0 \leq x\}\bigr)$ and $\mathcal{P}_{k+1} \subseteq \operatorname{con}\bigl(\{x \leq 2\}\bigr)$. Now assume $\mathcal{D}_k = \mathcal{C}_k$ and note that, taking $\gamma = \bigl(x \leq \frac{k+1}{k+2}\bigr) \in \mathcal{C}_{k+1}$ and $\beta = (x \leq 2) \in \mathcal{D}_k = \mathcal{C}_k$, we have

$$\begin{aligned} \operatorname{con}\Bigl(\bigl(\mathcal{D}_k \setminus \{\beta\}\bigr) \cup \{\gamma\}\Bigr) &= \operatorname{con}\biggl(\Bigl\{0 \leq x, x \leq \frac{k}{k+1}, x \leq \frac{k+1}{k+2}\Bigr\}\biggr) \\ &= \operatorname{con}\biggl(\Bigl\{0 \leq x, x \leq \frac{k}{k+1}\Bigr\}\biggr) \\ &= \operatorname{con}(\mathcal{D}_k). \end{aligned}$$

We thus have $\mathcal{D}_{k+1} = \bigl\{0 \leq x, x \leq \frac{k+1}{k+2}, x \leq 2\bigr\} = \mathcal{C}_{k+1}$.

3.1 Implementation of the Standard Widening

The following proposition provides a formal justification for the correctness of an algorithm implementing the standard widening when the inclusion hypothesis $\mathcal{P}_1 \subseteq \mathcal{P}_2$ is satisfied. The main idea, which has been proposed in [22] and later reported in [25], is to replace the expensive test in the specification of $\mathcal{C}'_2$ in Definition 1 with an appropriate saturation condition to be checked on any generator system for $\mathcal{P}_1$. We provide an improved version of the above result showing that neither the addition of the set of constraints $\mathcal{C}'_1$ as given in Definition 1 nor the splitting of equality constraints into corresponding pairs of inequalities are needed. A similar result, but without the use of saturation conditions, can be found in [5, Chapter 6].

Proposition 1. *Let $\mathcal{P}_1 = \operatorname{con}(\mathcal{C}_1) = \operatorname{gen}(\mathcal{G}_1) \in \mathbb{CP}_n$ and $\mathcal{P}_2 = \operatorname{con}(\mathcal{C}_2) \in \mathbb{CP}_n$, where $\mathcal{C}_1$ is in minimal form and $\mathcal{P}_1 \subseteq \mathcal{P}_2$. Then $\mathcal{P}_1 \nabla \mathcal{P}_2 = \operatorname{con}(\mathcal{C}_s)$, where*

$$\mathcal{C}_s \stackrel{\text{def}}{=} \bigl\{ \gamma \in \mathcal{C}_2 \bigm| \exists \beta \in \mathcal{C}_1 \mathrel{.} \operatorname{sat\_gen}(\gamma, \mathcal{G}_1) = \operatorname{sat\_gen}(\beta, \mathcal{G}_1) \bigr\}.$$

Proof. In this proof, to simplify the notation, if $\mathcal{C}$ is any constraint system and β and γ are constraints, we will write $\mathcal{C}[\gamma/\beta]$ to denote the constraint system $\bigl(\mathcal{C} \setminus \{\beta\}\bigr) \cup \{\gamma\}$.

We prove that $\mathcal{P}_1 \nabla \mathcal{P}_2 = \operatorname{con}(\mathcal{C}_s)$ by considering the two inclusions separately. First we show that $\mathcal{P}_1 \nabla \mathcal{P}_2 \subseteq \operatorname{con}(\mathcal{C}_s)$. Suppose $\gamma \in \operatorname{repr}_{\geq}(\mathcal{C}_s)$; then, by the hypothesis, there exists $\beta \in \mathcal{C}_1$ such that $\operatorname{sat\_gen}(\gamma, \mathcal{G}_1) = \operatorname{sat\_gen}(\beta, \mathcal{G}_1)$. If $\beta \in \operatorname{eq}(\mathcal{C}_1)$, $\operatorname{sat\_gen}(\gamma, \mathcal{G}_1) = \mathcal{G}_1$. Let $\gamma^{=}$ be the equality constraint with the same coefficients as γ. Then $\operatorname{sat\_gen}(\gamma^{=}, \mathcal{G}_1) = \mathcal{G}_1$ and, as $\mathcal{C}_1$ is in minimal form, for some $\beta_1 \in \mathrm{eq}(\mathcal{C}_1)$, $\mathcal{P}_1 = \mathrm{con}\bigl(\mathcal{C}_1[\gamma^=/\beta_1]\bigr)$. Thus, for some $\beta_1' \in \mathrm{repr}_{\geq}\bigl(\{\beta_1\}\bigr)$ (so that $\beta_1' \in \mathrm{repr}_{\geq}(\mathcal{C}_1)$), $\mathcal{P}_1 = \mathrm{con}\bigl(\mathrm{repr}_{\geq}(\mathcal{C}_1)[\gamma/\beta_1']\bigr)$. On the other hand, as $\mathcal{C}_1$ is in minimal form, if $\beta \in \mathrm{ineq}(\mathcal{C}_1)$, $\mathrm{sat\_gen}(\beta, \mathcal{G}_1) \neq \mathcal{G}_1$ and, as $\mathcal{P}_1 \subseteq \mathcal{P}_2 \subseteq \mathrm{con}\bigl(\{\gamma\}\bigr)$, $\mathcal{P}_1 = \mathrm{con}\bigl(\mathrm{repr}_{\geq}(\mathcal{C}_1)[\gamma/\beta]\bigr)$.

We now prove that $\mathrm{con}(\mathcal{C}_s) \subseteq \mathcal{P}_1 \nabla \mathcal{P}_2$. Let $\mathcal{C}_1', \mathcal{C}_2'$ be as defined in Definition 1. First we will show that $\mathrm{con}(\mathcal{C}_s) \subseteq \mathrm{con}(\mathcal{C}_1')$. Suppose $\beta \in \mathcal{C}_1'$. Then $\beta \in \mathrm{repr}_{\geq}(\mathcal{C}_1)$ and $\mathcal{P}_2 \subseteq \mathrm{con}\bigl(\{\beta\}\bigr)$. Let $\boldsymbol{v} \in \mathrm{gen}\bigl(\mathrm{sat\_gen}(\beta, \mathcal{G}_1)\bigr)$. Suppose there exists a closed ball $B \subseteq \mathbb{R}^n$ centered at $\boldsymbol{v}$ of radius $\lambda > 0$ that is contained in $\mathcal{P}_2$. Then, as $\mathcal{P}_2 \subseteq \mathrm{con}\bigl(\{\beta\}\bigr)$, $B \subseteq \mathrm{con}\bigl(\{\beta\}\bigr)$ contradicting the assumption that $\boldsymbol{v}$ saturates β. Thus there exists a constraint in $\mathrm{repr}_{\geq}(\mathcal{C}_2)$ that is saturated by $\boldsymbol{v}$. As $\boldsymbol{v} \in \mathrm{gen}\bigl(\mathrm{sat\_gen}(\beta, \mathcal{G}_1)\bigr)$ was arbitrary, there exists $\gamma \in \mathrm{repr}_{\geq}(\mathcal{C}_2)$ that is saturated by every point in $\mathrm{gen}\bigl(\mathrm{sat\_gen}(\beta, \mathcal{G}_1)\bigr)$ so that $\mathrm{gen}\bigl(\mathrm{sat\_gen}(\beta, \mathcal{G}_1)\bigr) \subseteq \mathrm{gen}\bigl(\mathrm{sat\_gen}(\gamma, \mathcal{G}_1)\bigr)$. Thus $\mathrm{sat\_gen}(\beta, \mathcal{G}_1) \preceq \mathrm{sat\_gen}(\gamma, \mathcal{G}_1)$. As $\mathcal{C}_1$ is in minimal form, $\mathrm{sat\_gen}(\beta, \mathcal{G}_1) = \mathrm{sat\_gen}(\gamma, \mathcal{G}_1)$ so that $\gamma \in \mathcal{C}_s$. As this holds for all $\beta \in \mathcal{C}_1'$, we have $\mathrm{con}(\mathcal{C}_s) \subseteq \mathrm{con}(\mathcal{C}_1')$. Finally, we show that $\mathrm{con}(\mathcal{C}_s) \subseteq \mathrm{con}(\mathcal{C}_2')$. Suppose $\gamma \in \mathcal{C}_2'$. Then there exists $\beta \in \mathrm{repr}_{\geq}(\mathcal{C}_1)$ such that $\mathrm{con}\bigl(\mathcal{C}_1[\gamma/\beta]\bigr) = \mathcal{P}_1$. Thus $\mathrm{sat\_gen}(\gamma, \mathcal{G}_1) = \mathrm{sat\_gen}(\beta, \mathcal{G}_1)$ and hence $\mathrm{con}(\mathcal{C}_s) \subseteq \mathrm{con}\bigl(\{\gamma\}\bigr)$. As this holds for all $\gamma \in \mathcal{C}_2'$, we have $\mathrm{con}(\mathcal{C}_s) \subseteq \mathrm{con}(\mathcal{C}_2')$. Thus $\mathrm{con}(\mathcal{C}_s) \subseteq \mathrm{con}(\mathcal{C}_1') \cap \mathrm{con}(\mathcal{C}_2')$ and hence, $\mathrm{con}(\mathcal{C}_s) \subseteq \mathrm{con}(\mathcal{C}_1' \cup \mathcal{C}_2')$. By Definition 1, $\mathcal{P}_1 \nabla \mathcal{P}_2 = \mathrm{con}(\mathcal{C}_1' \cup \mathcal{C}_2')$ so that $\mathrm{con}(\mathcal{C}_s) \subseteq \mathcal{P}_1 \nabla \mathcal{P}_2$, as required. □

It is worth stressing that the correctness of the above proposition relies on the inclusion hypothesis $\mathcal{P}_1 \subseteq \mathcal{P}_2$, which was only implicitly present in [22, 25]. The following example shows that, when $\mathcal{P}_1 \nsubseteq \mathcal{P}_2$, the result is not guaranteed to be an upper approximation of the arguments. Note that this is independent of the two improvements mentioned above.

Example 3. Let $\mathcal{P}_1 = \mathrm{con}(\mathcal{C}_1) \in \mathbb{CP}_2$ and $\mathcal{P}_2 = \mathrm{con}(\mathcal{C}_2) \in \mathbb{CP}_2$, where

$$\begin{aligned}\mathcal{C}_1 &= \{x = 0, 0 \leq y \leq 2\},\\ \mathcal{C}_2 &= \{y \geq 2\}.\end{aligned}$$

Then $\mathcal{P}_1 = \mathrm{gen}(\mathcal{G}_1)$, where $\mathcal{G}_1 = (\varnothing, \varnothing, P)$ and $P = \bigl\{(0,0)^{\mathrm{T}}, (2,0)^{\mathrm{T}}\bigr\}$. Note that $\mathcal{P}_1 \nsubseteq \mathcal{P}_2$. By Definition 1, we obtain $\mathcal{C}_1' = \mathcal{C}_2' = \varnothing$, so that $\mathcal{P}_1 \nabla \mathcal{P}_2 = \mathbb{R}^2$. Considering the constraints $\beta = (-y \geq -2) \in \mathcal{C}_1$ and $\gamma = (y \geq 2) \in \mathcal{C}_2$, we have

$$\mathrm{sat\_gen}(\beta, \mathcal{G}_1) = \Bigl(\varnothing, \varnothing, \bigl\{(2,0)^{\mathrm{T}}\bigr\}\Bigr) = \mathrm{sat\_gen}(\gamma, \mathcal{G}_1),$$

so that $\gamma \in \mathcal{C}_s$. Thus, the result of the algorithm is $\mathcal{P}_2$, which is different from $\mathcal{P}_1 \nabla \mathcal{P}_2$ and it is not an upper approximation of $\mathcal{P}_1$.

To avoid problems such as the one above, in the following we adopt a minor variant of the classical definition of widening operator given in Section 1 (see the footnote in [17, p. 275]).

Definition 2. *Let $L(\sqsubseteq, \sqcup)$ be a join-semi-lattice (i.e., the least upper bound $x \sqcup y$ exists for all $x, y \in L$). The operator $\nabla \colon L \times L \rightarrowtail L$ is a* widening *if*

1. $\forall x, y \in L : x \sqsubseteq y \implies y \sqsubseteq x \nabla y$;

2. for all increasing chains $y_0 \sqsubseteq y_1 \sqsubseteq \cdots$*, the increasing chain defined by* $x_0 \overset{\text{def}}{=} y_0, \ldots, x_{i+1} \overset{\text{def}}{=} x_i \nabla y_{i+1}, \ldots$ *is not strictly increasing.*

It can be proved [17] that, for any continuous operator $\mathcal{F} \colon L \to L$, the upward iteration sequence with widenings starting from any element $x_0 \in L$ and defined by

$$x_{i+1} = \begin{cases} x_i, & \text{if } \mathcal{F}(x_i) \sqsubseteq x_i; \\ x_i \nabla \bigl(x_i \sqcup \mathcal{F}(x_i)\bigr), & \text{otherwise;} \end{cases}$$

converges after a finite number of iterations. Note that the widening is always applied to arguments $x = x_i$ and $y = x_i \sqcup \mathcal{F}(x_i)$ satisfying $x \sqsubseteq y$ and $x \neq y$. Thus, without loss of generality, in the following we will assume that the two argument polyhedra satisfy the strict inclusion hypothesis $\mathcal{P}_1 \subset \mathcal{P}_2$.

As far as the implementation of the standard widening is concerned, it is worth noting the following result, which provides the justification for an alternative algorithm based on the original proposal in [18]. A similar result has also been proved in [5, Chapter 6].

Proposition 2. *Let* $\mathcal{P}_1, \mathcal{P}_2 \in \mathbb{CP}_n$*, where* $\mathcal{P}_1 \subseteq \mathcal{P}_2$ *and* $\dim(\mathcal{P}_1) = \dim(\mathcal{P}_2)$*. Let also* $\mathcal{P}_1 = \mathrm{con}(\mathcal{C}_1) \neq \varnothing$*, where the constraint system* $\mathcal{C}_1$ *is in minimal form. Then* $\mathcal{P}_1 \nabla \mathcal{P}_2 = \mathrm{con}(\mathcal{C})$*, where* $\mathcal{C} \overset{\text{def}}{=} \bigl\{\, \beta \in \mathcal{C}_1 \bigm| \mathcal{P}_2 \subseteq \mathrm{con}\bigl(\{\beta\}\bigr) \,\bigr\}$*.*

Proof. Since, by hypothesis, $\dim(\mathcal{P}_1) = \dim(\mathcal{P}_2)$ and $\mathcal{P}_1 \subseteq \mathcal{P}_2$, we also have $\mathrm{aff.hull}(\mathcal{P}_1) = \mathrm{aff.hull}(\mathcal{P}_2)$. As $\mathcal{C}_1$ is in minimal form, $\mathrm{aff.hull}(\mathcal{P}_1) = \mathrm{con}\bigl(\mathrm{eq}(\mathcal{C}_1)\bigr)$. Moreover, for all $\beta \in \mathrm{eq}(\mathcal{C}_1)$, $\mathrm{aff.hull}(\mathcal{P}_1) \subseteq \mathrm{con}\bigl(\{\beta\}\bigr)$ so that $\mathcal{P}_2 \subseteq \mathrm{con}\bigl(\{\beta\}\bigr)$. Thus, by definition of $\mathcal{C}$, $\mathrm{aff.hull}(\mathcal{P}_1) = \mathrm{aff.hull}\bigl(\mathrm{con}(\mathcal{C})\bigr)$. As $\mathcal{P}_1 \nabla \mathcal{P}_2$ is well defined and hence, does not depend on the constraint system used to represent $\mathcal{P}_2$, we assume $\mathcal{C}_2$ is in minimal form and that $\mathrm{eq}(\mathcal{C}_1) = \mathrm{eq}(\mathcal{C}_2)$.

Since $\mathrm{repr}_{\geq}(\mathcal{C}) \subseteq \bigl\{\, \beta \in \mathrm{repr}_{\geq}(\mathcal{C}_1) \bigm| \mathcal{P}_2 \subseteq \mathrm{con}\bigl(\{\beta\}\bigr) \,\bigr\}$, it follows from Definition 1 that $\mathcal{P}_1 \nabla \mathcal{P}_2 \subseteq \mathrm{con}(\mathcal{C})$. We show that $\mathrm{con}(\mathcal{C}) \subseteq \mathcal{P}_1 \nabla \mathcal{P}_2$. Let $\mathcal{P}_1 = \mathrm{gen}(\mathcal{G}_1)$ for some generator system $\mathcal{G}_1$ and

$$\mathcal{C}_s \overset{\text{def}}{=} \bigl\{\, \gamma \in \mathcal{C}_2 \bigm| \exists \beta \in \mathcal{C}_1 \mathrel{.} \mathrm{sat\_gen}(\gamma, \mathcal{G}_1) = \mathrm{sat\_gen}(\beta, \mathcal{G}_1) \,\bigr\}.$$

By Proposition 1, $\mathcal{P}_1 \nabla \mathcal{P}_2 = \mathrm{con}(\mathcal{C}_s)$. Suppose $\gamma \in \mathcal{C}_s$. Then $\gamma \in \mathcal{C}_2$ and there exists $\beta \in \mathcal{C}_1$ such that $\mathrm{sat\_gen}(\gamma, \mathcal{G}_1) = \mathrm{sat\_gen}(\beta, \mathcal{G}_1)$. Thus

$$\mathrm{con}\bigl(\{\beta\}\bigr) \cap \mathrm{aff.hull}(\mathcal{P}_1) = \mathrm{con}\bigl(\{\gamma\}\bigr) \cap \mathrm{aff.hull}(\mathcal{P}_1). \tag{1}$$

Hence, as $\mathcal{P}_2 \subseteq \mathrm{con}\bigl(\{\gamma\}\bigr)$ and $\mathcal{P}_2 \subseteq \mathrm{aff.hull}(\mathcal{P}_1)$, $\mathcal{P}_2 \subseteq \mathrm{con}\bigl(\{\beta\}\bigr)$ so that $\beta \in \mathcal{C}$. As $\mathrm{aff.hull}\bigl(\mathrm{con}(\mathcal{C})\bigr) = \mathrm{aff.hull}(\mathcal{P}_1)$, it follows from (1) that $\mathrm{con}(\mathcal{C}) \subseteq \mathrm{con}\bigl(\{\gamma\}\bigr)$. As this holds for all $\gamma \in \mathcal{C}_s$, we obtain $\mathrm{con}(\mathcal{C}) \subseteq \mathrm{con}(\mathcal{C}_s) = \mathcal{P}_1 \nabla \mathcal{P}_2$. □

The interesting fact about this alternative algorithm is that its implementation does not require the computation of a constraint system for the polyhedron $\mathcal{P}_2$: any generator system for $\mathcal{P}_2$ can be used to check the hypothesis that the two argument polyhedra have the same dimension and, if this is the case, to select the constraints from $\mathcal{C}_1$; if the two polyhedra have different dimensions, we fall back to the implementation based on Proposition 1. Note that it is almost always the case that polyhedron $\mathcal{P}_2$ has been obtained as the result of a poly-hull operation, so that in a "lazy" implementation the polyhedron will be described by a generator system only (since the poly-hull is implemented by taking the union of two generator system).

4 Defining More Precise Widenings

In this section, elaborating on an idea originally proposed in [6], we will present a framework for the systematic definition of new and precise widening operators for polyhedra. In particular, we will state the theoretical result that will be used to ensure that all the instances of the framework are indeed widening operators. In order to do that, we need the following definition.

Definition 3. (Number of non-null coordinates of a vector.) *Let $\boldsymbol{v} \in \mathbb{R}^n$. We write $\kappa(\boldsymbol{v})$ to denote the number of non-null coordinates of $\boldsymbol{v}$. For each finite set $V \subseteq \mathbb{R}^n$, we define $\kappa(V)$ to be the multiset obtained by applying κ to each of the vectors in V.*

We now define the relation $\curvearrowright \subseteq \mathbb{CP}_n \times \mathbb{CP}_n$ incorporating a notion of, so to speak, "limited growth" or "growth that cannot be indefinite" (graphically, a descending parabola).

Definition 4. ($\curvearrowright \subseteq \mathbb{CP}_n \times \mathbb{CP}_n$.) *Let $\mathcal{P}_1, \mathcal{P}_2 \in \mathbb{CP}_n$ be two polyhedra. Then $\mathcal{P}_1 \curvearrowright \mathcal{P}_2$ if and only if $\mathcal{P}_1 \subset \mathcal{P}_2$ and either $\mathcal{P}_1 = \varnothing$ or at least one of the following conditions holds, where, for $i = 1$, 2, $\mathcal{P}_i$ is given by means of a constraint system $\mathcal{C}_i$ in minimal form and a generator system $\mathcal{G}_i = (L_i, R_i, P_i)$ in orthogonal form:*

$$\dim(\mathcal{P}_1) < \dim(\mathcal{P}_2); \tag{2}$$

$$\dim\bigl(\text{lin.space}(\mathcal{P}_1)\bigr) < \dim\bigl(\text{lin.space}(\mathcal{P}_2)\bigr); \tag{3}$$

$$\#\,\mathcal{C}_1 > \#\,\mathcal{C}_2; \tag{4}$$

$$\#\,\mathcal{C}_1 = \#\,\mathcal{C}_2 \wedge \#\,P_1 > \#\,P_2; \tag{5}$$

$$\#\,\mathcal{C}_1 = \#\,\mathcal{C}_2 \wedge \#\,P_1 = \#\,P_2 \wedge \kappa(R_1) \gg \kappa(R_2). \tag{6}$$

Note that the relation $\curvearrowright$ is well defined, since it does not depend on the particular constraint and generator representations chosen; in particular, the requirement that the $\mathcal{G}_i$ are in orthogonal form ensures that the computation of $\kappa(R_i)$ is not ambiguous (see Section 2).

The next result incorporates the basic idea behind the overall approach.

Theorem 1. *Let $\mathcal{P}_0 \curvearrowright \mathcal{P}_1 \curvearrowright \cdots \curvearrowright \mathcal{P}_i \curvearrowright \cdots$ be a chain of polyhedra in $\mathbb{CP}_n$. Then the chain is finite.*

Proof. Let us consider the polyhedron $\mathcal{P}_j \neq \varnothing$ for any $j \geq 0$. Then both the notions of constraint system $\mathcal{C}_j$ in minimal form and generator system $\mathcal{G}_j = (L_j, R_j, P_j)$ in orthogonal form are well defined for $\mathcal{P}_j$. Since $\dim(\mathcal{P}_j) \leq n$ and $\dim\bigl(\mathrm{lin.space}(\mathcal{P}_j)\bigr) \leq n$ and the conditions (2) and (3) of Definition 4 prescribe an increase in these values, respectively, the conditions can be regarded as defining a strict decrease on the negations of these values with lower bound $-n$. As $\mathcal{P}_j \neq \varnothing$, we have $\#\,\mathcal{C}_j \geq 0$, $\#\,P_j \geq 1$ and $\#\,R_j \geq 0$. If $\mathcal{P}_i \curvearrowright \mathcal{P}_{i+1}$ is a link in the chain, conditions (4) and (5) prescribe $\#\,\mathcal{C}_i > \#\,\mathcal{C}_{i+1}$ and $\#\,P_i > \#\,P_{i+1}$, respectively. If $\boldsymbol{r}$ is any ray in $\mathbb{R}^n$, then we must have $\kappa(\boldsymbol{r}) < n$. Moreover, '$\gg$' is well-founded [20] and, if $\mathcal{P}_i \curvearrowright \mathcal{P}_{i+1}$ is a link in the chain, condition (6) prescribes $\kappa(R_i) \gg \kappa(R_{i+1})$.

Thus, each condition (2), (3), (4), (5), and (6) of Definition 4 requires a strict decrease with respect to a well-founded ordering. Note that, by the hypothesis, if $\mathcal{P}_i \curvearrowright \mathcal{P}_{i+1}$ is a link in the chain, then $\mathcal{P}_i \subset \mathcal{P}_{i+1}$. Thus, when the conditions (2) and (3) are not met, we have both $\dim(\mathcal{P}_i) = \dim(\mathcal{P}_{i+1})$ and $\dim\bigl(\mathrm{lin.space}(\mathcal{P}_i)\bigr) = \dim\bigl(\mathrm{lin.space}(\mathcal{P}_{i+1})\bigr)$. Thus, as condition (5) requires $\#\,\mathcal{C}_i = \#\,\mathcal{C}_{i+1}$ and (6) requires $\#\,\mathcal{C}_i = \#\,\mathcal{C}_{i+1}$ and $\#\,P_i = \#\,P_{i+1}$, the relation '$\curvearrowright$' is a lexicographic product of the inverses of well-founded orderings so that '$\curvearrowright$' satisfies the ascending chain condition and hence, the chain is finite. □

The '$\curvearrowright$' relation is a variant of a similar notion of limited growth defined in [6, Theorem 3]. These two proposals are not formally comparable since neither one of the relations refines the other. On one hand, in Definition 4, there are convergence criteria that were not considered in [6], namely conditions (4) and (6); on the other hand, to ensure that the relation satisfies the ascending chain condition, condition (5) also requires that the number of constraints is not increasing.

From a more practical point of view, the relation defined in [6] is unsatisfactory, since neither the standard widening ∇, nor the heuristics informally sketched in [6] ensure that consecutive iterates satisfy the given notion of limited growth. In summary, the overall approach does not define a widening operator in the precise sense of Definition 2 [F. Besson, personal communication, 2002]. By contrast, the introduction of condition (4) ensures that applications of ∇ always yield polyhedra that are related to previous iterates by the '$\curvearrowright$' relation.

Theorem 2. *Let $\mathcal{P}_1 \subset \mathcal{P}_2 \in \mathbb{CP}_n$ and $\mathcal{P}_1 \mathbin{\nabla} \mathcal{P}_2 = \mathcal{P}$. Then $\mathcal{P}_2 \subseteq \mathcal{P}$ and $\mathcal{P}_1 \curvearrowright \mathcal{P}$.*

Proof. Let $\mathcal{P}_1 = \mathrm{con}(\mathcal{C}_1)$, where $\mathcal{C}_1$ is a constraint system in minimal form, and $\mathcal{P}_2 = \mathrm{con}(\mathcal{C}_2)$. We have $\mathcal{P} = \mathrm{con}(\mathcal{C}'_1 \cup \mathcal{C}'_2)$, where $\mathcal{C}'_1$ and $\mathcal{C}'_2$ are as specified in Definition 1. Thus, for all $\beta \in \mathcal{C}'_1$, $\mathcal{P}_2 \subseteq \mathrm{con}\bigl(\{\beta\}\bigr)$ so that $\mathcal{P}_2 \subseteq \mathrm{con}(\mathcal{C}'_1)$. Also, for all $\gamma \in \mathcal{C}'_2$, we have $\gamma \in \mathrm{repr}_{\geq}(\mathcal{C}_2)$ and hence $\mathcal{P}_2 \subseteq \mathrm{con}\bigl(\{\gamma\}\bigr)$ so that $\mathcal{P}_2 \subseteq \mathrm{con}(\mathcal{C}'_2)$. Therefore, $\mathcal{P}_2 \subseteq \mathcal{P}$. Since by hypothesis we have $\mathcal{P}_1 \subset \mathcal{P}_2$, we also obtain $\mathcal{P}_1 \subset \mathcal{P}$, so that $\dim(\mathcal{P}_1) \leq \dim(\mathcal{P})$. If $\dim(\mathcal{P}_1) < \dim(\mathcal{P})$, then, by Definition 4, $\mathcal{P}_1 \curvearrowright \mathcal{P}$. Suppose next that $\dim(\mathcal{P}_1) = \dim(\mathcal{P})$.

Let $\mathcal{P} = \mathrm{con}(\mathcal{C})$ where $\mathcal{C}$ is in minimal form. Then we show that $\#\,\mathcal{C} < \#\,\mathcal{C}_1$. Let

$$\mathcal{C}_\nabla = \Bigl\{\, \beta \in \mathcal{C}_1 \Bigm| \mathcal{P}_2 \subseteq \mathrm{con}\bigl(\{\beta\}\bigr) \Bigr\}.$$

Thus $\mathcal{C}_\nabla \subseteq \mathcal{C}_1$. As $\mathcal{P}_1 \subset \mathcal{P}$, $\mathcal{C}_\nabla \subset \mathcal{C}_1$. Therefore $\#\,\mathcal{C}_\nabla < \#\,\mathcal{C}_1$. By hypothesis $\dim(\mathcal{P}_1) = \dim(\mathcal{P})$ and $\mathcal{P}_1 \subset \mathcal{P}_2$, so that Proposition 2 applies and we obtain $\mathcal{P} = \mathrm{con}(\mathcal{C}) = \mathrm{con}(\mathcal{C}_\nabla)$. As $\mathcal{C}$ is in minimal form, $\#\,\mathcal{C} \leq \#\,\mathcal{C}_\nabla$ so that $\#\,\mathcal{C} < \#\,\mathcal{C}_1$. Hence, by Definition 4, $\mathcal{P}_1 \curvearrowright \mathcal{P}$. □

This simple result provides a secure foundation for the use of new widening operators such as the ones proposed here. In fact, because of Theorems 1 and 2, the following result shows how the definition of a widening operator that can improve on the standard widening is greatly simplified.

Theorem 3. *Let $h \colon \mathbb{CP}_n^2 \to \mathbb{CP}_n$ be an upper bound operator and*

$$\mathcal{P}_1 \mathbin{\tilde{\nabla}} \mathcal{P}_2 \stackrel{\mathrm{def}}{=} \begin{cases} h(\mathcal{P}_1, \mathcal{P}_2), & \text{if } \mathcal{P}_1 \curvearrowright h(\mathcal{P}_1, \mathcal{P}_2) \subset \mathcal{P}_1 \mathbin{\nabla} \mathcal{P}_2; \\ \mathcal{P}_1 \mathbin{\nabla} \mathcal{P}_2, & \text{otherwise.} \end{cases}$$

Then the $\tilde{\nabla}$ operator is a widening at least as precise as ∇.

Proof. By hypothesis, h is an upper bound operator and, by Theorem 2, the same holds for the standard widening. Thus, in all cases $\mathcal{P}_2 \subseteq \mathcal{P}_1 \mathbin{\tilde{\nabla}} \mathcal{P}_2$. By definition of $\tilde{\nabla}$ and, in the case that $\mathcal{P}_1 \mathbin{\tilde{\nabla}} \mathcal{P}_2 = \mathcal{P}_1 \mathbin{\nabla} \mathcal{P}_2$, also by Theorem 2, $\mathcal{P}_1 \curvearrowright \mathcal{P}_1 \mathbin{\tilde{\nabla}} \mathcal{P}_2$ holds. By Theorem 1, any increasing chain of polyhedra with respect to the '$\curvearrowright$' relation is finite. Thus, by Definition 2, the $\tilde{\nabla}$ operator is a widening. Finally, the fact that the result of $\tilde{\nabla}$ is at least as precise as the standard widening holds trivially by construction. □

The above scheme is easily extended to any finite set of such techniques, still obtaining a widening operator. In the following section we will consider several of these possible heuristics: the simplest one, also adopted in [6], was actually suggested in [17]; the second one is based on an idea informally sketched in [6]; the third one is a minor variant of the extrapolation operator of [26]; the fourth and last one is new to this paper.

5 Improving the Standard Widening by Heuristics

The simplest heuristics, already suggested in [17], is the one saying 'do not widen': if we are along an iteration chain having finite length, there is no need to provide further approximations, so that we can safely return the most precise upper bound $\mathcal{P}_2$ (remember that we assume $\mathcal{P}_1 \subset \mathcal{P}_2$). In our context, this is the case whenever $\mathcal{P}_1 \curvearrowright \mathcal{P}_2$. As a consequence, all the other widening techniques considered here including the standard widening, are only applied to a pair of polyhedra such that $\mathcal{P}_1 \not\curvearrowright \mathcal{P}_2$, so that $\dim(\mathcal{P}_1) \geq \dim(\mathcal{P}_2)$ and $\dim\bigl(\mathrm{lin.space}(\mathcal{P}_1)\bigr) \geq \dim\bigl(\mathrm{lin.space}(\mathcal{P}_2)\bigr)$; by the inclusion hypothesis, these imply $\mathrm{aff.hull}(\mathcal{P}_1) = \mathrm{aff.hull}(\mathcal{P}_2)$ and $\mathrm{lin.space}(\mathcal{P}_1) = \mathrm{lin.space}(\mathcal{P}_2)$, respectively.

When defining a widening operator on an abstract domain, a common tactic is to split the current abstract description into several components and look at each one in isolation so as to identify what has changed with respect to the previous iteration. Intuitively, the information provided by stable components should be propagated to the next iteration, whereas the information of components that have changed should be extrapolated according to a hypothetical "change pattern". For instance, in the case of the widening in [18], each element of a constraint system is regarded as a separate component and the extrapolation just forgets about the constraints that have changed. The second heuristics, which is a variant of a similar one sketched in [6], can be seen as an application of the above approach, where instead of the constraints we consider the points in the generator system describing the polyhedron of the previous iteration. When using the standard widening it may happen that points that are common to the boundaries[3] of $\mathcal{P}_1$ and $\mathcal{P}_2$ (and, hence, likely to be an invariant feature along the chain of polyhedra) will not lie on the boundary of the widened polyhedron. This is the case, for instance, for the two points $\boldsymbol{p}$ and $\boldsymbol{q}$ in Figure 1. For each of such points, the technique forces the presence of an inequality constraint that is saturated by the point, so that they will lie on the boundary of the result.

Definition 5. (Combining constraints.) *Let $\mathcal{P}_1, \mathcal{P}_2 \in \mathbb{CP}_n$ be two polyhedra such that $\mathcal{P}_1 \subset \mathcal{P}_2$, $\operatorname{aff.hull}(\mathcal{P}_1) = \operatorname{aff.hull}(\mathcal{P}_2)$ and $\operatorname{lin.space}(\mathcal{P}_1) = \operatorname{lin.space}(\mathcal{P}_2)$. Let $\mathcal{P}_1 = \operatorname{gen}(\mathcal{G}_1)$, $\mathcal{P}_2 = \operatorname{con}(\mathcal{C}_2)$ and $\mathcal{P}_1 \nabla \mathcal{P}_2 = \operatorname{con}(\mathcal{C}_\nabla)$, where the constraint systems $\mathcal{C}_2$, $\mathcal{C}_\nabla$ and the generator system $\mathcal{G}_1 = (L_1, R_1, P_1)$ are in orthogonal form. Let also*

$$\mathcal{C}_\oplus \stackrel{\text{def}}{=} \left\{ \oplus(\mathcal{C}_{\boldsymbol{p}}) \;\middle|\; \begin{array}{l} \boldsymbol{p} \in P_1, \operatorname{sat\_con}\bigl(\boldsymbol{p}, \operatorname{ineq}(\mathcal{C}_\nabla)\bigr) = \varnothing, \\ \mathcal{C}_{\boldsymbol{p}} = \operatorname{sat\_con}\bigl(\boldsymbol{p}, \operatorname{ineq}(\mathcal{C}_2)\bigr) \neq \varnothing \end{array} \right\},$$

where the operator $\oplus$ computes a convex combination of a non-empty set of linear inequality constraints (i.e., of the corresponding coefficients), returning another linear inequality constraint. Then $h_c(\mathcal{P}_1, \mathcal{P}_2) \stackrel{\text{def}}{=} \operatorname{con}(\mathcal{C}_\nabla \cup \mathcal{C}_\oplus)$.

Since the operator h_c is only defined for arguments having the same affine hull and lineality space, by requiring orthogonal forms we ensure that the result does not depend on the particular representations considered.

Note that the particular convex combination encoded by function $\oplus$ is deliberately left unspecified so as to allow for a very liberal definition of h_c that still possesses the required properties. For instance, in [6] it was argued that a good heuristics could be obtained by letting $\oplus$ compute a normed linear combination (i.e., a sort of average) of the chosen constraints. Another legitimate choice would be to "bless" one of the constraints in $\mathcal{C}_{\boldsymbol{p}}$ and forget all the others. In both cases, by keeping just one constraint for each point $\boldsymbol{p}$, we hopefully reduce the cardinality of the constraint system describing the result, so that it is more likely that condition (4) of Definition 4 will be met. Actually, this attempt at reducing the number of constraints is the main difference between the technique presented in Definition 5 and the extrapolation operator proposed in [28,

[3] In this context, a "boundary point" is any point of $\mathcal{P} \cap \operatorname{lin.space}(\mathcal{P})^\perp$ which is not a relatively interior point for $\mathcal{P}$. Namely, we abstract from both the affine hull and the lineality space of the polyhedron.

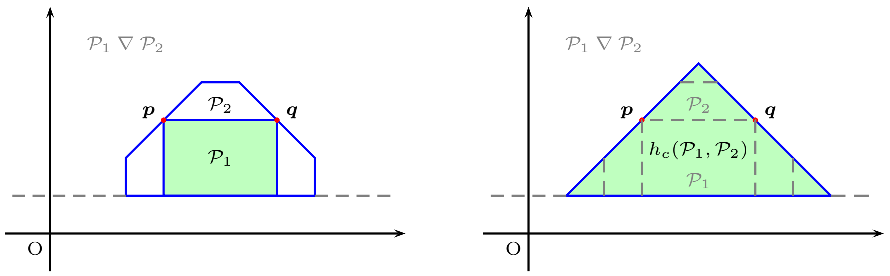

Fig. 1. The heuristics h_c improving on the standard widening.

Section 3.3], which could itself be included in the current framework as a more refined widening heuristics.

Our third heuristic technique is a variant of the extrapolation operator '$\propto$' defined in [26]. The technique examines each new point $\boldsymbol{p}_2$ of the polyhedron $\mathcal{P}_2$ as if it was obtained from each old point $\boldsymbol{p}_1$ of the polyhedron $\mathcal{P}_1$: we say that $\boldsymbol{p}_2$ is an evolution of $\boldsymbol{p}_1$. The extrapolation is defined as continuing this evolution towards infinity, therefore generating the ray having direction $\boldsymbol{p}_2 - \boldsymbol{p}_1$. The new ray will subsume point $\boldsymbol{p}_2$, so that it is likely that the convergence condition (5) of Definition 4 will be met. Notice that any ray that violates a constraint of the standard widening is dropped.

Definition 6. (Evolving points.) *Let* $\mathcal{P}_1, \mathcal{P}_2 \in \mathbb{CP}_n$ *be such that* $\mathcal{P}_1 \subset \mathcal{P}_2$ *and* $\operatorname{lin.space}(\mathcal{P}_1) = \operatorname{lin.space}(\mathcal{P}_2)$. *For each* $i = 1, 2$, *consider a generator system* $\mathcal{G}_i = (L_i, R_i, P_i)$ *in orthogonal form such that* $\mathcal{P}_i = \operatorname{gen}(\mathcal{G}_i)$ *and let*

$$R \stackrel{\text{def}}{=} \bigl\{\, \boldsymbol{p}_2 - \boldsymbol{p}_1 \bigm| \boldsymbol{p}_1 \in P_1, \boldsymbol{p}_2 \in P_2 \setminus P_1 \,\bigr\}.$$

Then we define $h_p(\mathcal{P}_1, \mathcal{P}_2) = \operatorname{gen}\bigl((L_2, R_2 \cup R, P_2)\bigr) \cap (\mathcal{P}_1 \nabla \mathcal{P}_2)$.

Since the operator h_p is only defined for arguments having the same lineality space, by requiring orthogonal forms we ensure that the result does not depend on the particular generator system representations considered.

The difference with respect to the extrapolation operator '$\propto$' is that we do not require the two points to lie on the same 1-dimensional face of $\mathcal{P}_2$; moreover, the result of '$\propto$' may be less precise than the standard widening. Note that, as in the "combining constraints" technique, it is possible to add just a single ray which is a convex combination of the rays in R instead of the complete set R; yielding a more precise widening technique. However, this technique and the one defined by the h_p operator are incomparable with respect to the '$\curvearrowright$' relation and one can fail the '$\curvearrowright$' convergence criteria when the other succeeds.

We now introduce a fourth widening heuristics that tries to extrapolate the way rays have evolved since the last iteration. The technique examines each new ray $\boldsymbol{r}_2$ of the polyhedron $\mathcal{P}_2$ as if it was generated by rotation of each old ray $\boldsymbol{r}_1$ of the polyhedron $\mathcal{P}_1$: we say that $\boldsymbol{r}_2$ is an evolution of $\boldsymbol{r}_1$. The extrapolation is defined as continuing this evolution until one or more of the non-null coordinates of ray $\boldsymbol{r}_2$ become zero. This way, it is likely that the convergence condition (6) of Definition 4 will be met. Intuitively, the new ray will reach one of the boundaries of the orthant where $\boldsymbol{r}_2$ lies, without trespassing it.

Definition 7. (evolve.) *The function* evolve: $\mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ *is defined, for each* $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^n$*, as* $\operatorname{evolve}(\boldsymbol{v}, \boldsymbol{w}) \overset{\text{def}}{=} \boldsymbol{v}'$*, where*

$$v'_i \overset{\text{def}}{=} \begin{cases} 0, & \text{if } \exists j \in \{1, \dots, n\} \mathrel{.} (v_i \cdot w_j - v_j \cdot w_i) \cdot v_i \cdot v_j < 0; \\ v_i, & \text{otherwise.} \end{cases}$$

To understand this definition consider a pair of coordinates i and j and suppose that the vectors $\boldsymbol{v}$ and $\boldsymbol{w}$ are projected onto the two-dimensional plane defined by i (for the first coordinate) and j (for the second coordinate). Then, we identify the direction of the rotation of the vector $(v_i, v_j)^{\mathrm{T}}$ with respect to the vector $(w_i, w_j)^{\mathrm{T}}$ by using the well-known cross-product test [11, Chapter 35]; the direction is clockwise if $cw = v_i \cdot w_j - v_j \cdot w_i > 0$ and anti-clockwise when $cw < 0$. Moreover, vector $(v_i, v_j)^{\mathrm{T}}$ lies inside the first or third quadrant when $q = v_i \cdot v_j > 0$ and it lies inside the second or fourth quadrant when $q < 0$. Then, the condition $cw \cdot q < 0$ states that the evolution is clockwise and $(v_i, v_j)^{\mathrm{T}}$ is in the second or fourth quadrant or the evolution is anti-clockwise and $(v_i, v_j)^{\mathrm{T}}$ is in the first or third quadrant: in all these cases, the evolution is towards the j axis. Thus, for a fixed i, if there exists j such that the evolution is towards the j axis, then we define $v'_i = 0$. Otherwise, we let $v'_i = v_i$. We are now ready to define our last widening heuristics.

Definition 8. (Evolving rays.) *Let* $\mathcal{P}_1, \mathcal{P}_2 \in \mathbb{CP}_n$ *be such that* $\mathcal{P}_1 \subset \mathcal{P}_2$ *and* $\operatorname{lin.space}(\mathcal{P}_1) = \operatorname{lin.space}(\mathcal{P}_2)$*. For each* $i = 1, 2$*, consider a generator system* $\mathcal{G}_i = (L_i, R_i, P_i)$ *in orthogonal form such that* $\mathcal{P}_i = \operatorname{gen}(\mathcal{G}_i)$ *and let*

$$R \overset{\text{def}}{=} \bigl\{ \operatorname{evolve}(\boldsymbol{r}_2, \boldsymbol{r}_1) \bigm| \boldsymbol{r}_1 \in R_1, \boldsymbol{r}_2 \in R_2 \setminus R_1 \bigr\}.$$

Then we define $h_r(\mathcal{P}_1, \mathcal{P}_2) \overset{\text{def}}{=} \operatorname{gen}\bigl((L_2, R_2 \cup R, P_2)\bigr) \cap (\mathcal{P}_1 \nabla \mathcal{P}_2)$*.*

Figure 2 shows an example where the evolving rays technique is able to improve on the standard widening. It should be noted that the boundary of $\mathcal{P}_1 \nabla \mathcal{P}_2$ contains the intersection of the boundaries of $\mathcal{P}_1$ and $\mathcal{P}_2$, so that the “combining constraints” technique is not applicable. Neither the “evolving points” technique can be applied, since $\mathcal{P}_1$ and $\mathcal{P}_2$ have the same set of irredundant points. Besides having the same affine hull and lineality space, polyhedra $\mathcal{P}_1$, $\mathcal{P}_2$ and $h_r(\mathcal{P}_1, \mathcal{P}_2)$, are defined by the same number of irredundant constraints and points, so that $\mathcal{P}_1 \curvearrowright h_r(\mathcal{P}_1, \mathcal{P}_2)$ holds by condition (6) of Definition 4.

To use these heuristics techniques in the general framework we have defined in the previous section, each of them needs to be an upper bound operator. This is trivial for the first technique. The same result holds, by construction, for the other three heuristics.

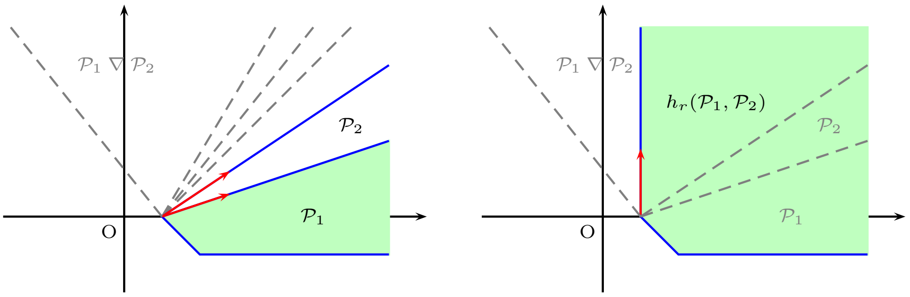
$\mathcal{P}_1 \nabla \mathcal{P}_2$
$\mathcal{P}_2$
$\mathcal{P}_1$
O
$\mathcal{P}_1 \nabla \mathcal{P}_2$
$h_r(\mathcal{P}_1, \mathcal{P}_2)$
$\mathcal{P}_2$
$\mathcal{P}_1$
O

Fig. 2. The heuristics h_r improving on the standard widening.

Proposition 3. *Let $\mathcal{P}_1, \mathcal{P}_2 \in \mathbb{CP}_n$, where $\mathcal{P}_1 \subset \mathcal{P}_2$,* $\operatorname{aff.hull}(\mathcal{P}_1) = \operatorname{aff.hull}(\mathcal{P}_2)$ *and* $\operatorname{lin.space}(\mathcal{P}_1) = \operatorname{lin.space}(\mathcal{P}_2)$*. Then, for each technique $h \in \{h_c, h_p, h_r\}$, $\mathcal{P}_2 \subseteq h(\mathcal{P}_1, \mathcal{P}_2) \subseteq \mathcal{P}_1 \nabla \mathcal{P}_2$.*

Proof. Let $\mathcal{P}_t = h(\mathcal{P}_1, \mathcal{P}_2)$. Consider first the case when $h = h_c$ and assume the notation introduced in Definition 5. The proof for $\mathcal{P}_t \subseteq \mathcal{P}_1 \nabla \mathcal{P}_2$ is immediate, since $\mathcal{P}_t$ is defined by a constraint system $\mathcal{C}_\nabla \cup \mathcal{C}_\oplus$ including all of the constraints defining $\mathcal{P}_1 \nabla \mathcal{P}_2$. To prove that $\mathcal{P}_2 \subseteq \mathcal{P}_t$ we show that $\mathcal{P}_2 \subseteq \operatorname{con}\bigl(\{\beta\}\bigr)$, for each constraint $\beta \in \mathcal{C}_\nabla \cup \mathcal{C}_\oplus$ defining $\mathcal{P}_t$. Clearly, if $\beta \in \mathcal{C}_\nabla$ then the inclusion holds by the fact that the standard widening is an upper bound operator, i.e., by Theorem 2. If otherwise $\beta \in \mathcal{C}_\oplus$, then, for some $\mathcal{C}_{\boldsymbol{p}} \subseteq \operatorname{ineq}(\mathcal{C}_2)$, $\beta = \oplus(\mathcal{C}_{\boldsymbol{p}})$, so that $\mathcal{P}_2 \subseteq \operatorname{con}(\mathcal{C}_{\boldsymbol{p}}) \subseteq \operatorname{con}\bigl(\{\beta\}\bigr)$.

Next, consider the cases when $h \in \{h_p, h_r\}$ and assume the notation introduced in Definitions 6 and 8. Let $\mathcal{G}' = (L_2, R_2 \cup R, P_2)$ and $\mathcal{P}' = \operatorname{gen}(\mathcal{G}')$; then $\mathcal{P}_t = \mathcal{P}' \cap (\mathcal{P}_1 \nabla \mathcal{P}_2)$. Thus $\mathcal{P}_t \subseteq \mathcal{P}_1 \nabla \mathcal{P}_2$. As $\mathcal{G}_2 \preceq \mathcal{G}'$, we obtain $\mathcal{P}_2 \subseteq \mathcal{P}'$. Moreover, by Theorem 2, we also have $\mathcal{P}_2 \subseteq \mathcal{P}_1 \nabla \mathcal{P}_2$. Therefore, by the monotonicity of set intersection, we conclude $\mathcal{P}_2 \subseteq \mathcal{P}_t$. □

The new widening operator is obtained by instantiating the framework of the previous section using the four heuristic techniques presented above.

Definition 9. (The $\hat{\nabla}$ widening.) *Let $\mathcal{P}_1, \mathcal{P}_2 \in \mathbb{CP}_n$, where $\mathcal{P}_1 \subset \mathcal{P}_2$. Then*

$$\mathcal{P}_1 \,\hat{\nabla}\, \mathcal{P}_2 \stackrel{\text{def}}{=} \begin{cases} \mathcal{P}_2, & \text{if } \mathcal{P}_1 \curvearrowright \mathcal{P}_2; \\ h_c(\mathcal{P}_1, \mathcal{P}_2), & \text{if } \mathcal{P}_1 \curvearrowright h_c(\mathcal{P}_1, \mathcal{P}_2) \subset \mathcal{P}_1 \nabla \mathcal{P}_2; \\ h_p(\mathcal{P}_1, \mathcal{P}_2), & \text{if } \mathcal{P}_1 \curvearrowright h_p(\mathcal{P}_1, \mathcal{P}_2) \subset \mathcal{P}_1 \nabla \mathcal{P}_2; \\ h_r(\mathcal{P}_1, \mathcal{P}_2), & \text{if } \mathcal{P}_1 \curvearrowright h_r(\mathcal{P}_1, \mathcal{P}_2) \subset \mathcal{P}_1 \nabla \mathcal{P}_2; \\ \mathcal{P}_1 \nabla \mathcal{P}_2, & \text{otherwise.} \end{cases}$$

It can be seen that $\hat{\nabla}$ is an instance of the framework proposed in the previous section: in particular, when applying the first heuristics, the omission of the applicability condition $\mathcal{P}_2 \subset \mathcal{P}_1 \nabla \mathcal{P}_2$ is a simple and inconsequential optimization. Thus the following result is a direct consequence of Theorem 3 and Proposition 3.

Proposition 4. *The $\hat{\nabla}$ operator is a widening at least as precise as ∇.*

Proof. Suppose that $\mathcal{P}_1, \mathcal{P}_2 \in \mathbb{CP}_n$, where $\mathcal{P}_1 \subset \mathcal{P}_2$, so that Definition 9 applies. If $\mathcal{P}_2 = \mathcal{P}_1 \nabla \mathcal{P}_2$, then $\mathcal{P}_1 \hat{\nabla} \mathcal{P}_2 = \mathcal{P}_1 \nabla \mathcal{P}_2$. Therefore, in order to apply Theorem 3, we have to show that when $\mathcal{P}_2 \subset \mathcal{P}_1 \nabla \mathcal{P}_2$, all the heuristic techniques used in Definition 9 are upper bound operators. This trivially holds for the first technique, which returns the least upper bound $\mathcal{P}_2$; for the other techniques, it is a consequence of Proposition 3. □

Proposition 4 is not strong enough to ensure that the final results of upward iteration sequences using the new widening are *uniformly* more precise than those obtained by using the standard widening. For this to hold for any widening that is an instance of our framework, the standard widening needs to be monotonic on both its arguments. In fact, as shown in the next proposition, the standard widening is monotonic on its second argument.

Proposition 5. *Let $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}'_2 \in \mathbb{CP}_n$ be such that $\mathcal{P}_1 \subseteq \mathcal{P}_2 \subseteq \mathcal{P}'_2$. Then we have $\mathcal{P}_1 \nabla \mathcal{P}_2 \subseteq \mathcal{P}_1 \nabla \mathcal{P}'_2$.*

Proof. Let $\mathcal{P}_1 = \operatorname{con}(\mathcal{C}_1) = \operatorname{gen}(\mathcal{G}_1)$ where $\mathcal{C}_1$ is in minimal form, $\mathcal{P}_2 = \operatorname{con}(\mathcal{C}_2)$ and $\mathcal{P}'_2 = \operatorname{con}(\mathcal{C}'_2)$. By hypothesis, $\mathcal{P}_2 \subseteq \mathcal{P}'_2$ so that $\mathcal{P}_2 = \operatorname{con}(\mathcal{C}_2 \cup \mathcal{C}'_2)$. Since, by hypothesis, $\mathcal{P}_1 \subseteq \mathcal{P}_2$ and $\mathcal{P}_1 \subseteq \mathcal{P}'_2$, we can apply Proposition 1. Thus we have $\mathcal{P}_1 \nabla \mathcal{P}_2 = \operatorname{con}(\mathcal{C}''_s)$ and $\mathcal{P}_1 \nabla \mathcal{P}'_2 = \operatorname{con}(\mathcal{C}'_s)$, where

$$\begin{aligned}
\mathcal{C}'_s &= \bigl\{\, \gamma \in \mathcal{C}'_2 \bigm| \exists \beta \in \mathcal{C}_1 \mathrel{.} \operatorname{sat\_gen}(\gamma, \mathcal{G}_1) = \operatorname{sat\_gen}(\beta, \mathcal{G}_1) \,\bigr\},\\
\mathcal{C}''_s &= \bigl\{\, \gamma \in \mathcal{C}_2 \cup \mathcal{C}'_2 \bigm| \exists \beta \in \mathcal{C}_1 \mathrel{.} \operatorname{sat\_gen}(\gamma, \mathcal{G}_1) = \operatorname{sat\_gen}(\beta, \mathcal{G}_1) \,\bigr\}\\
&= \mathcal{C}'_s \cup \bigl\{\, \gamma \in \mathcal{C}_2 \bigm| \exists \beta \in \mathcal{C}_1 \mathrel{.} \operatorname{sat\_gen}(\gamma, \mathcal{G}_1) = \operatorname{sat\_gen}(\beta, \mathcal{G}_1) \,\bigr\}.
\end{aligned}$$

Hence $\mathcal{C}''_s \supseteq \mathcal{C}'_s$. Therefore we have the thesis $\mathcal{P}_1 \nabla \mathcal{P}_2 \subseteq \mathcal{P}_1 \nabla \mathcal{P}'_2$. □

On the other hand, as illustrated in the following example, the standard widening (and thus also the new widening) is not monotonic on its first argument [18].

Example 4. Consider the polyhedral domain $\mathbb{CP}_2$ and let

$$\begin{aligned}
\mathcal{P}_1 &= \operatorname{con}\bigl(\{1 \leq x \leq 2, 0 \leq y \leq 2\}\bigr),\\
\mathcal{P}'_1 &= \operatorname{con}\bigl(\{0 \leq x \leq 2, 0 \leq y \leq 2\}\bigr),\\
\mathcal{P}_2 &= \operatorname{con}\bigl(\{x \geq 0\}\bigr),
\end{aligned}$$

so that $\mathcal{P}_1 \subseteq \mathcal{P}'_1 \subseteq \mathcal{P}_2$. By Definition 1 we have

$$\begin{aligned}
\mathcal{P}_1 \nabla \mathcal{P}_2 &= \operatorname{con}(\varnothing) = \mathbb{R}^2,\\
\mathcal{P}'_1 \nabla \mathcal{P}_2 &= \mathcal{P}_2.
\end{aligned}$$

Thus, we obtain $\mathcal{P}_1 \nabla \mathcal{P}_2 \nsubseteq \mathcal{P}'_1 \nabla \mathcal{P}_2$.

Note that in spite of this lack of monotonicity the experimental evaluation of the next section shows that precision degradations are very rare in practice.

6 Experimental Evaluation

We have extended the *Parma Polyhedra Library* (PPL) [2, 3], a modern C++ library for the manipulation of convex polyhedra, with a prototype implementation of the widening of Definition 9. The PPL has been integrated with the China analyzer [1] for the purpose of detecting linear argument size relations [4]. Our benchmark suite consists of 361 Prolog programs, ranging from small synthetic benchmarks to real-world applications. They define 23279 predicates whose analysis with China requires the direct use of a widening and about as many predicates for which no widening is used. In this respect, it must be noted that China employs a sophisticate chaotic iteration strategy proposed in [7, 8] that, among other benefits, allows to greatly reduce the number of widenings' applications.[4] This is an important point, since it would be quite easy to improve on an iteration strategy applying widenings "everywhere or improperly" [7]. The results of this experimental evaluation are summarized in Table 1, where each row corresponds to a different choice for the value of the extrapolation threshold k, controlling the delay before the applications of both the standard and the new widening operators.

	Precision						Time			
	# programs			# predicates			std ∇_k		new $\hat{\nabla}_k$	
k (delay)	improve	degr	incomp	improve	degr	incomp	all	top 20	all	top 20
0	121	0	2	1340	3	2	1.00	0.72	1.05	0.77
1	34	0	0	273	0	0	1.09	0.79	1.11	0.80
2	29	0	0	222	0	0	1.16	0.83	1.18	0.84
3	28	0	0	160	0	0	1.23	0.88	1.25	0.89
4	25	0	2	126	2	0	1.32	0.95	1.34	0.95
10	25	0	0	124	0	0	1.82	1.23	1.85	1.24

Table 1. Precision and time comparisons.

The part of the table headed 'Precision' shows the obtained precision improvements and degradations, both in terms of the number of programs and the number of predicates affected; in the columns labeled 'incomp' we report those cases where incomparable results have been obtained. For $k = 0$, we observe a precision improvement on one third of the considered programs; not surprisingly, fewer improvements are obtained for higher values of k, but we still have an improvement on 7% of the benchmarks when considering $k = 10$. While confirming, as informally argued in [4], that for this particular analysis there is little incentive in using values of k greater than 4, our experiments show that the new widening captures growth patterns that do happen in practice and that for the

[4] China uses the recursive fixpoint iteration strategy on the weak topological ordering defined by partitioning of the call graph into strongly-connected subcomponents [8].

standard widening (no matter how delayed) are out of reach. This is important since the results obtained in practice are, besides correctness, what really matters when evaluating widening operators. The experimentation also shows that the idea of delaying the widening [12] maintains its validity: even though the new widening is less sensitive to the amount of delay applied, delaying still improves some of the results.

The part of the table headed 'Time' shows the sum, over all the benchmarks, of the fixpoint computation times. This is expressed as a proportion of the time spent when using the standard widening with $k = 0$. Since smaller benchmarks may affect the outcome of this summarization, in the columns labeled 'top 20' we also show the same values but restricted to the 20 benchmarks whose analysis takes more time. It can be seen that the new widening has a negative, but relatively modest impact on efficiency, which anyway is smaller than the cost of increasing the value of k. When looking at these time results, it should be considered that we are comparing a prototype implementation of the new widening with respect to a rather optimized implementation of the standard widening. It is also important to remark that the good performance degradation observed for both widenings when increasing the value of k is essentially due to the iteration strategy employed by China and should not be expected to automatically carry over to systems using other fixpoint computation techniques.

7 An Improved Delay Technique

The technique of employing an extrapolation threshold k has been traditionally implemented (and our experimental evaluation makes no exception) in a "simple way" [17], as a blind delay in the application of the widening. Namely, for each widening operator $\tilde{\nabla}$, the widening operator $\tilde{\nabla}_k$ is formalized as follows, where each abstract value is a pair recording, in its second component, the iteration in which it has been computed:

$$\langle x, i \rangle \; \tilde{\nabla}_k \; \langle y, i+1 \rangle \stackrel{\text{def}}{=} \begin{cases} \langle x, i+1 \rangle, & \text{if } y \sqsubseteq x; \\ \langle x \sqcup y, i+1 \rangle, & \text{if } i < k; \\ \langle x \; \tilde{\nabla} \; y, i+1 \rangle, & \text{otherwise.} \end{cases}$$

Thus, no matter what abstract value would have been computed by the widening, the widening is never applied in the first k iteration steps and it is always applied in all the following iteration steps.

In our opinion, a better approximation strategy can be obtained by interpreting the value k as the maximum number of iterations for which the computation of the widening can be safely avoided. Thus, an abstract value is a pair carrying a number of "tokens" t, each of them allowing for the replacement of one widening application by the least upper bound. Aiming at an improvement in the final result, each widening operator should be left free to choose when to use the available tokens. For instance, tokens should not be wasted when the widening is able to compute the least upper bound of its arguments. The following definition of $\tilde{\nabla}_T$ (widening with tokens) formalizes this idea:

$$\langle x, t\rangle \; \tilde{\nabla}_T \; \langle y, \cdot\rangle \stackrel{\text{def}}{=} \begin{cases} \langle x, t\rangle, & \text{if } y \sqsubseteq x; \\ \langle x \; \tilde{\nabla} \; y, t\rangle, & \text{if } x \; \tilde{\nabla} \; y = x \sqcup y; \\ \langle x \sqcup y, t-1\rangle, & \text{if } t > 0; \\ \langle x \; \tilde{\nabla} \; y, 0\rangle, & \text{otherwise.} \end{cases}$$

The iteration sequence will begin with abstract values of the form $\langle x_0, k\rangle$, that is, with k tokens where k is a parameter of the analysis; the number of tokens will decrease along the iteration chain and, when there are no tokens left, the underlying widening operator $\tilde{\nabla}$ will always be applied. Notice that, when instantiating the above construction with our new widening operator $\hat{\nabla}$ (and assuming the inclusion hypothesis), the conditional guard for the second case of the definition of $\tilde{\nabla}_T$ becomes $\mathcal{P}_1 \; \hat{\nabla} \; \mathcal{P}_2 = \mathcal{P}_2$, which can be easily implemented by performing the test $\mathcal{P}_1 \curvearrowright \mathcal{P}_2$.

Also note that more general definitions for $\tilde{\nabla}_T$ are possible: for instance, when $x \; \tilde{\nabla} \; y \neq x \sqcup y$ and $t > 0$ (i.e., the widening does not compute the exact upper bound and there still are tokens available), we may nonetheless choose to apply the widening operator, provided the corresponding approximation is good enough. This way, we may preserve the tokens and use them to avoid some later approximations, which could be much coarser than the current one. Clearly, such an approach depends on the particular formalization of the notion of "good enough", which is, along with the value of k, intrinsically application-dependent.

8 Conclusion

For the domain of convex polyhedra, the convergence of the fixpoint computation sequence has been typically obtained thanks to the widening operator proposed by Cousot and Halbwachs. Though remarkably precise, this operator does not fulfill the requirements of a number of applications in the fields of analysis and verification that are particularly sensitive to the precision of the deduced numerical information. In this paper, elaborating on an idea proposed in [6], we have defined a framework for the systematic specification of new widening operators improving on the precision of the standard widening. The framework allows any upper bound operator on the domain of convex polyhedra to be transformed into a proper widening operator, therefore ensuring the termination of the computation. We have instantiated the framework with a selection of extrapolation operators, some of which embody improvements of heuristics already proposed in the literature. A first experimental evaluation has yielded promising results. The experimental work has also suggested that the well-known widening delay technique can be improved, yet retaining its overall simplicity. Our proposal is to delay the widening application only when this prevents *actual* (as opposed to *potential*) precision losses. The resulting widening would thus adapt, to some extent, to the abstract description chain being traversed.

Acknowledgments. We would like to thank Frédéric Besson for his useful comments and observations on the ideas sketched in [6]. Thanks also to Fred Mesnard for the information and discussions about the impact of precision on termination inference for Prolog programs.

References

1. R. Bagnara. *Data-Flow Analysis for Constraint Logic-Based Languages.* PhD thesis, Dipartimento di Informatica, Università di Pisa, Pisa, Italy, March 1997. Printed as Report TD-1/97.
2. R. Bagnara, P. M. Hill, E. Ricci, and E. Zaffanella. *The Parma Polyhedra Library User's Manual.* Department of Mathematics, University of Parma, Parma, Italy, release 0.4 edition, July 2002. Available at `http://www.cs.unipr.it/ppl/`.
3. R. Bagnara, E. Ricci, E. Zaffanella, and P. M. Hill. Possibly not closed convex polyhedra and the Parma Polyhedra Library. In M. V. Hermenegildo and G. Puebla, editors, *Static Analysis: Proceedings of the 9th International Symposium*, volume 2477 of *Lecture Notes in Computer Science*, pages 213–229, Madrid, Spain, 2002. Springer-Verlag, Berlin.
4. F. Benoy and A. King. Inferring argument size relationships with CLP($\mathcal{R}$). In J. P. Gallagher, editor, *Logic Program Synthesis and Transformation: Proceedings of the 6th International Workshop*, volume 1207 of *Lecture Notes in Computer Science*, pages 204–223, Stockholm, Sweden, 1997. Springer-Verlag, Berlin.
5. P. M. Benoy. *Polyhedral Domains for Abstract Interpretation in Logic Programming.* PhD thesis, Computing Laboratory, University of Kent, Canterbury, Kent, UK, January 2002.
6. F. Besson, T. P. Jensen, and J.-P. Talpin. Polyhedral analysis for synchronous languages. In A. Cortesi and G. Filé, editors, *Static Analysis: Proceedings of the 6th International Symposium*, volume 1694 of *Lecture Notes in Computer Science*, pages 51–68, Venice, Italy, 1999. Springer-Verlag, Berlin.
7. F. Bourdoncle. Efficient chaotic iteration strategies with widenings. In D. Bjørner, M. Broy, and I. V. Pottosin, editors, *Proceedings of the International Conference on "Formal Methods in Programming and Their Applications"*, volume 735 of *Lecture Notes in Computer Science*, pages 128–141, Academgorodok, Novosibirsk, Russia, 1993. Springer-Verlag, Berlin.
8. F. Bourdoncle. Sémantiques des langages impératifs d'ordre supérieur et interprétation abstraite. PRL Research Report 22, DEC Paris Research Laboratory, 1993.
9. T. Bultan, R. Gerber, and W. Pugh. Model-checking concurrent systems with unbounded integer variables: Symbolic representations, approximations, and experimental results. *ACM Transactions on Programming Languages and Systems*, 21(4):747–789, 1999.
10. M. A. Colón and H. B. Sipma. Synthesis of linear ranking functions. In T. Margaria and W. Yi, editors, *Proceedings of the 7th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2001)*, volume 2031 of *Lecture Notes in Computer Science*, pages 67–81, Genova, Italy, 2001. Springer-Verlag, Berlin.
11. T. H. Cormen, T. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* The MIT Press, Cambridge, Mass., 1990.

12. P. Cousot. Semantic foundations of program analysis. In S. S. Muchnick and N. D. Jones, editors, *Program Flow Analysis: Theory and Applications*, chapter 10, pages 303–342. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1981.
13. P. Cousot, editor. *Static Analysis: 8th International Symposium, SAS 2001*, volume 2126 of *Lecture Notes in Computer Science*, Paris, France, 2001. Springer-Verlag, Berlin.
14. P. Cousot and R. Cousot. Static determination of dynamic properties of programs. In B. Robinet, editor, *Proceedings of the Second International Symposium on Programming*, pages 106–130. Dunod, Paris, France, 1976.
15. 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 the Fourth Annual ACM Symposium on Principles of Programming Languages*, pages 238–252, New York, 1977. ACM Press.
16. P. Cousot and R. Cousot. Abstract interpretation frameworks. *Journal of Logic and Computation*, 2(4):511–547, 1992.
17. P. Cousot and R. Cousot. Comparing the Galois connection and widening/narrowing approaches to abstract interpretation. In M. Bruynooghe and M. Wirsing, editors, *Proceedings of the 4th International Symposium on Programming Language Implementation and Logic Programming*, volume 631 of *Lecture Notes in Computer Science*, pages 269–295, Leuven, Belgium, 1992. Springer-Verlag, Berlin.
18. P. Cousot and N. Halbwachs. Automatic discovery of linear restraints among variables of a program. In *Conference Record of the Fifth Annual ACM Symposium on Principles of Programming Languages*, pages 84–96, Tucson, Arizona, 1978. ACM Press.
19. G. Delzanno and A. Podelski. Model checking in CLP. In R. Cleaveland, editor, *Tools and Algorithms for Construction and Analysis of Systems, Proceedings of the 5th International Conference*, volume 1579 of *Lecture Notes in Computer Science*, pages 223–239, Amsterdam, The Netherlands, 1999. Springer-Verlag, Berlin.
20. N. Dershowitz and Z. Manna. Proving termination with multiset orderings. *Communications of the ACM*, 22(8):465–476, 1979.
21. N. Dor, M. Rodeh, and S. Sagiv. Cleanness checking of string manipulations in C programs via integer analysis. In Cousot [13], pages 194–212.
22. N. Halbwachs. *Détermination Automatique de Relations Linéaires Vérifiées par les Variables d'un Programme*. Thèse de 3$^{\text{ème}}$ cycle d'informatique, Université scientifique et médicale de Grenoble, Grenoble, France, March 1979.
23. N. Halbwachs. Delay analysis in synchronous programs. In C. Courcoubetis, editor, *Computer Aided Verification: Proceedings of the 5th International Conference*, volume 697 of *Lecture Notes in Computer Science*, pages 333–346, Elounda, Greece, 1993. Springer-Verlag, Berlin.
24. N. Halbwachs, Y.-E. Proy, and P. Raymond. Verification of linear hybrid systems by means of convex approximations. In B. Le Charlier, editor, *Static Analysis: Proceedings of the 1st International Symposium*, volume 864 of *Lecture Notes in Computer Science*, pages 223–237, Namur, Belgium, 1994. Springer-Verlag, Berlin.
25. N. Halbwachs, Y.-E. Proy, and P. Roumanoff. Verification of real-time systems using linear relation analysis. *Formal Methods in System Design*, 11(2):157–185, 1997.
26. T. A. Henzinger and P.-H. Ho. A note on abstract interpretation strategies for hybrid automata. In P. J. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors, *Hybrid Systems II*, volume 999 of *Lecture Notes in Computer Science*, pages 252–264. Springer-Verlag, Berlin, 1995.

27. T. A. Henzinger, P.-H. Ho, and H. Wong-Toi. HyTech: A model checker for hybrid systems. *Software Tools for Technology Transfer*, 1(1+2):110–122, 1997.
28. T. A. Henzinger, J. Preussig, and H. Wong-Toi. Some lessons from the hytech experience. In *Proceedings of the 40th Annual Conference on Decision and Control*, pages 2887–2892. IEEE Computer Society Press, 2001.
29. Z. Manna, N. S. Bjørner, A. Browne, M. Colón, B. Finkbeiner, M. Pichora, H. B. Sipma, and T. E. Uribe. An update on STeP: Deductive-algorithmic verification of reactive systems. In R. Berghammer and Y. Lakhnech, editors, *Tool Support for System Specification, Development and Verification*, Advances in Computing Sciences. Springer-Verlag, Berlin, 1999.
30. F. Mesnard and U. Neumerkel. Applying static analysis techniques for inferring termination conditions of logic programs. In Cousot [13], pages 93–110.
31. W. Pugh. A practical algorithm for exact array dependence analysis. *Communications of the ACM*, 35(8):102–114, 1992.