A New Encoding of Not Necessarily Closed Convex Polyhedra*

Extended Abstract

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

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

1 Introduction

Many applications of static analysis and verification compute on some abstract domain based on convex polyhedra. Traditionally, most of these applications are restricted to convex polyhedra that are topologically closed. When adopting the *Double Description* (DD) method [8], a closed convex polyhedron can be specified in two ways, using a *constraint system* or a *generator system*: the constraint system contains a finite set of linear non-strict inequality constraints; the generator system contains two finite sets of vectors, collectively called *generators*, which are rays and points of the polyhedron.

Some applications of static analysis and verification, including recent proposals such as [3], need to compute on the domain of *not necessarily closed* (NNC) convex polyhedra. By definition, any NNC polyhedron can be represented by a so-called *mixed constraint system*, that is, a constraint system where a further finite set of linear *strict* inequality constraints is allowed to occur. The usual approach for implementing NNC polyhedra is to embed them into closed polyhedra in a vector space with one extra dimension. While this idea, originally proposed in [6] and also described in [7], proved to be quite effective, its direct application results in a low-level user interface where most of the geometric intuition of the DD method gets lost under the "implementation details".

A much cleaner approach was proposed in [1, 2], where the concept of generator of an NNC polyhedron is extended to also account for the *closure points* of the polyhedron. In particular, it is shown that any NNC polyhedron can be elegantly and intuitively represented by means of *extended generator systems*. The combined use of mixed constraint systems and extended generator systems provides a higher level interface to the domain of NNC polyhedra, allowing for simpler definitions of some of the corresponding operators.

* This work has been partly supported by MURST projects "Abstract interpretation, type systems and control-flow analysis" and "Aggregate- and number-reasoning for computing: from decision algorithms to constraint programming with multisets, sets, and maps".

The use of closure points in the approach proposed in [1, 2] provides a two-fold improvement over the proposal in [6, 7]: first, an NNC polyhedron can be presented to the client application directly in terms of its defining strict and non-strict constraints or its generating rays, points and closure points; second, the implementation becomes separate from the user interface.

In this paper we exploit the latter possibility by introducing an alternative class of closed polyhedra for representing the NNC polyhedra. The basis of this representation is a simple generalization of the class of polyhedra used in [6, 7] and also in [2]. The new class continues to employ an additional dimension to encode whether or not each affine half-space defining the NNC polyhedron is closed and rely on the same semantic function given in [2] for extracting the NNC polyhedron it embeds. We describe two alternative specializations of this class for representing the NNC polyhedra. One of these, shown to be biased for the use of the constraint representation, corresponds to the embedding defined in [2] while the other, which is biased for the use of the generator representation, is new to this paper.

An interesting and potentially useful consequence of having the option of these alternative implementations is that, depending on the number of strict constraints in the constraint system compared with the number of closure points that are also points in the generator system, the choice of representation will affect the efficiency of the polyhedral operations.

2 Preliminaries

We first define some necessary terminology. The reader is referred to [2] for a more formal introduction to the required concepts and notations.

In the paper, all topological arguments refer to the usual topological space $\mathbb{R}^n$ where $n \in \mathbb{N}$. The set of non-negative reals is denoted by $\mathbb{R}_+$. For each $i \in \{1, \dots, n\}$, v_i denotes the i-th component of the (column) vector $\boldsymbol{v} \in \mathbb{R}^n$. We denote by $\mathbf{0}$ the vector of $\mathbb{R}^n$ having all components equal to zero. The scalar product of $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^n$ is denoted by $\langle \boldsymbol{v}, \boldsymbol{w} \rangle$.

A subset $\mathcal{P}$ of $\mathbb{R}^n$ is called a *closed polyhedron* 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 $\mathbb{CP}_n$.

A subset $\mathcal{P}$ of $\mathbb{R}^n$ is called an *NNC polyhedron* if either $\mathcal{P}$ can be expressed as the intersection of a finite number of (not necessarily closed) affine half-spaces of $\mathbb{R}^n$ or $n = 0$ and $\mathcal{P} = \varnothing$. The set of all NNC polyhedra on $\mathbb{R}^n$ is denoted by $\mathbb{P}_n$. Obviously, we have $\mathbb{CP}_n \subseteq \mathbb{P}_n$ (note that $\mathbb{CP}_n = \mathbb{P}_n$ if and only if $n = 0$).

The set $\mathbb{P}_n$, when partially ordered by subset inclusion, is a lattice and $\mathbb{CP}_n$ is a sublattice of $\mathbb{P}_n$. The binary meet operation is given by set-intersection, whereas the binary join operation, denoted $\uplus$, is called *convex polyhedral hull*.

For each vector $\boldsymbol{a} \in \mathbb{R}^n$ and scalar $b \in \mathbb{R}$, where $\boldsymbol{a} \neq \mathbf{0}$, the linear inequality constraint $\langle \boldsymbol{a}, \boldsymbol{x} \rangle \geq b$ (resp., $\langle \boldsymbol{a}, \boldsymbol{x} \rangle > b$) defines a topologically closed (resp., open) affine half-space of $\mathbb{R}^n$. A *mixed constraint system* $\mathcal{C}$ is a finite set of linear inequality constraints and we will write $\mathrm{con}(\mathcal{C})$ to denote the polyhedron described by $\mathcal{C}$.

A vector $\boldsymbol{r} \in \mathbb{R}^n$ such that $\boldsymbol{r} \neq \boldsymbol{0}$ is a *ray* of a non-empty polyhedron $\mathcal{P} \in \mathbb{P}_n$ if, for every point $\boldsymbol{p} \in \mathcal{P}$ and every $\mu \in \mathbb{R}_+$, it holds $\boldsymbol{p} + \mu\boldsymbol{r} \in \mathcal{P}$; a vector $\boldsymbol{c} \in \mathbb{R}^n$ is a *closure point* of $\mathcal{P} \in \mathbb{P}_n$ if $\boldsymbol{c} \in \mathbb{C}(\mathcal{P})$. Given three finite sets of vectors $R, P, C \subseteq \mathbb{R}^n$, where $R = \{\boldsymbol{r}_1, \dots, \boldsymbol{r}_k\}$ and $\boldsymbol{0} \notin R$, $P = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_\ell\}$ and $C = \{\boldsymbol{c}_1, \dots, \boldsymbol{c}_m\}$, then the triple $\mathcal{G} = (R, P, C)$ is called an *extended generator system* for the NNC polyhedron

$$\mathrm{gen}(\mathcal{G}) \stackrel{\mathrm{def}}{=} \left\{ \sum_{i=1}^{k} \mu_i \boldsymbol{r}_i + \sum_{i=1}^{\ell} \nu_i \boldsymbol{p}_i + \sum_{i=1}^{m} \eta_i \boldsymbol{c}_i \;\middle|\; \begin{array}{l} \boldsymbol{\mu} \in \mathbb{R}_+^k, \boldsymbol{\nu} \in \mathbb{R}_+^\ell, \boldsymbol{\eta} \in \mathbb{R}_+^m, \\ \boldsymbol{\nu} \neq \boldsymbol{0}, \sum_{i=1}^{\ell} \nu_i + \sum_{i=1}^{m} \eta_i = 1 \end{array} \right\}.$$

The polyhedron $\mathrm{gen}(\mathcal{G})$ is empty if and only if $P = \varnothing$. For a non-empty polyhedron, vectors in R, P, and C are rays, points and closure points, respectively. When $C = \varnothing$, we will omit it from the generator system and simply write $\mathcal{G} = (R, P)$.

3 Representing NNC Polyhedra

We will define two alternative representations for NNC polyhedra. The two classes of closed polyhedra used for these representations are instances of a more general class of closed polyhedra. In the following, we denote by ϵ the variable corresponding to the $(n+1)$-st Cartesian axis of $\mathbb{R}^{n+1}$.

Definition 1. (ϵ-polyhedron.) *A polyhedron $\mathcal{R} \in \mathbb{CP}_{n+1}$ is said to be an* ϵ-polyhedron *if and only if*

$$\exists \delta \in \mathbb{R} \;.\; \Bigl(\delta > 0 \wedge \mathcal{R} \subseteq \mathrm{con}\bigl(\{\epsilon \leq \delta\}\bigr)\Bigr); \tag{1}$$

$$\forall \boldsymbol{v} \in \mathbb{R}^n, e \in \mathbb{R} : (\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R} \implies (\boldsymbol{v}^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}. \tag{2}$$

Condition (2) that every point in the ϵ-polyhedron $\mathcal{R}$ has a projection on the hyperplane defined by the constraint $(\epsilon = 0)$ corresponds to a dual property concerning the constraints for $\mathcal{R}$.

Proposition 1. *Let $\mathcal{R} \in \mathbb{CP}_{n+1}$ be such that, for some $\delta \in \mathbb{R}$ with $\delta > 0$, $\mathcal{R} \subseteq \mathrm{con}\bigl(\{\epsilon \leq \delta\}\bigr)$. Then $\mathcal{R}$ is an ϵ-polyhedron if and only if, for each $s \in \mathbb{R}$,*

$$\mathcal{R} \subseteq \mathrm{con}\Bigl(\bigl\{\langle \boldsymbol{a}, \boldsymbol{x} \rangle + s \cdot \epsilon \geq b\bigr\}\Bigr) \implies \mathcal{R} \subseteq \mathrm{con}\Bigl(\bigl\{\langle \boldsymbol{a}, \boldsymbol{x} \rangle \geq b\bigr\}\Bigr).$$

Each ϵ-polyhedron in $\mathbb{CP}_{n+1}$ denotes an NNC polyhedron in $\mathbb{P}_n$. In particular, points in an ϵ-polyhedron with a strictly positive ϵ-coordinate, correspond to points in the NNC polyhedron.

Definition 2. (Represented NNC polyhedron.) *Let $\mathcal{R} \in \mathbb{CP}_{n+1}$ be a closed polyhedron. $\mathcal{R}$ is said to* represent *the NNC polyhedron $\mathcal{P} \in \mathbb{P}_n$ if and only if*

$$\mathcal{P} = [\![\mathcal{R}]\!] \stackrel{\mathrm{def}}{=} \Bigl\{ \boldsymbol{v} \in \mathbb{R}^n \;\Bigm|\; \exists e \in \mathbb{R} \;.\; \bigl(e > 0 \wedge (\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}\bigr) \Bigr\}. \tag{3}$$

The polyhedron $\mathcal{R}$ is said to be an ϵ-polyhedron for $\mathcal{P} \in \mathbb{P}_n$*, denoted* $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P}$*, if* $\mathcal{R}$ *is an ϵ-polyhedron and* $\mathcal{P} = [\![\mathcal{R}]\!]$.

We now consider two special subclasses of the class of ϵ-polyhedra. The first of these assumes the presence of the constraint $(\epsilon \geq 0)$ providing a lower bound for the ϵ dimension.

Definition 3. (C-ϵ-polyhedron.) *An ϵ-polyhedron* $\mathcal{R} \in \mathbb{CP}_{n+1}$ *is said to be* constraint-biased *and called a* C-ϵ-polyhedron *if and only if*

$$\mathcal{R} \subseteq \operatorname{con}\bigl(\{\epsilon \geq 0\}\bigr).$$

We write $\mathcal{R} \Rrightarrow_C \mathcal{P}$ *if* $\mathcal{R}$ *is a C-ϵ-polyhedron and* $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P}$*.*

Thus the set of constraint-biased ϵ-polyhedra is the subset of ϵ-polyhedra on the vector space $\mathbb{R}^{n+1}$ that, when adopting the proposal of [6,7], represent an NNC polyhedron on the vector space $\mathbb{R}^n$. Note that, in [2], a constraint-biased ϵ-polyhedron is called an ϵ-representation. Thus many of the definitions and results below concerning C-ϵ-polyhedra and the embedding of the NNC polyhedra in them are taken from [2].

In [2], we have shown how a C-ϵ-polyhedron for an NNC polyhedron $\mathcal{P}$ may be constructed directly from the constraint and generator systems for $\mathcal{P}$.

Definition 4. (con_repr$_C$.) *Let $\mathcal{C}$ be a generic mixed constraint system on the vector space $\mathbb{R}^n$, that is,*

$$\mathcal{C} = \bigl\{\, \langle \boldsymbol{a}_i, \boldsymbol{x} \rangle \bowtie_i b_i \bigm| i \in \{1, \ldots, m\}, \boldsymbol{a}_i \in \mathbb{R}^n, \bowtie_i \in \{\geq, >\}, b_i \in \mathbb{R} \,\bigr\}$$

The function con_repr$_C$ *associates $\mathcal{C}$ with the constraint system on the vector space* $\mathbb{R}^{n+1}$

$$\begin{aligned}
\operatorname{con_repr}_C(\mathcal{C}) \stackrel{\text{def}}{=} &\bigl\{0 \leq \epsilon\bigr\} \cup \bigl\{\epsilon \leq 1\bigr\} \\
&\cup \bigl\{\, \langle \boldsymbol{a}_i, \boldsymbol{x} \rangle - 1 \cdot \epsilon \geq b_i \bigm| i \in \{1, \ldots, m\}, \bowtie_i \in \{>\} \,\bigr\} \\
&\cup \bigl\{\, \langle \boldsymbol{a}_i, \boldsymbol{x} \rangle + 0 \cdot \epsilon \geq b_i \bigm| i \in \{1, \ldots, m\}, \bowtie_i \in \{\geq\} \,\bigr\}.
\end{aligned}$$

Definition 5. (gen_repr$_C$.) *Let $\mathcal{G} = (R, P, C)$ be an extended generator system on the vector space $\mathbb{R}^n$. The function* gen_repr$_C$ *associates to $\mathcal{G}$ the generator system* $\operatorname{gen\_repr}_C(\mathcal{G}) \stackrel{\text{def}}{=} (R', P')$ *on the vector space $\mathbb{R}^{n+1}$, where*

$$\begin{aligned}
R' &= \bigl\{\, (\boldsymbol{r}^{\mathrm{T}}, 0)^{\mathrm{T}} \bigm| \boldsymbol{r} \in R \,\bigr\}, \\
P' &= \bigl\{\, (\boldsymbol{p}^{\mathrm{T}}, 1)^{\mathrm{T}} \bigm| \boldsymbol{p} \in P \,\bigr\} \cup \bigl\{\, (\boldsymbol{q}^{\mathrm{T}}, 0)^{\mathrm{T}} \bigm| \boldsymbol{q} \in P \cup C \,\bigr\}.
\end{aligned}$$

Observe that, in the mapping defined by the generator representation function gen_repr$_C$ and using the notation in Definition 5, each point in P corresponds to two distinct points in P'. In general, the above encodings require a constant number of additional constraints versus a linear number of additional generators: this is the reason why ϵ-polyhedra in this subclass are called "constraint-biased".

The second special subclass of ϵ-polyhedra assumes the presence of the ray $-\boldsymbol{e}_\epsilon \stackrel{\text{def}}{=} (\boldsymbol{0}^{\mathrm{T}}, -1)^{\mathrm{T}}$, so that there is no lower bound for the ϵ dimension.

Definition 6. (G-ϵ-polyhedron.) *An ϵ-polyhedron $\mathcal{R} = \mathrm{gen}(R, P) \in \mathbb{CP}_{n+1}$ is said to be* generator-biased *and called a* G-ϵ-polyhedron *if and only if*

$$\mathcal{R} \supseteq \mathrm{gen}\bigl(\{-\boldsymbol{e}_\epsilon\}, P\bigr).$$

We write $\mathcal{R} \Rrightarrow_G \mathcal{P}$ if $\mathcal{R}$ is a G-ϵ-polyhedron and $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P}$.

As for the constraint-biased case, generator-biased ϵ-polyhedra can also be used for representing any NNC polyhedra. In particular, the generator and constraint systems for a G-ϵ-polyhedron for an NNC polyhedron $\mathcal{P}$ may be constructed directly from the constraint and generator systems for $\mathcal{P}$.

Definition 7. (con_repr$_G$.) *Let $\mathcal{C}$ be a mixed constraint system as in* Definition 4. *The function* con_repr$_G$ *associates $\mathcal{C}$ with the constraint system on the vector space $\mathbb{R}^{n+1}$*

$$\begin{aligned}\mathrm{con\_repr}_G(\mathcal{C}) \stackrel{\mathrm{def}}{=} \bigl\{\epsilon \leq 1\bigr\} & \\ \cup \bigl\{\, \langle \boldsymbol{a}_i, \boldsymbol{x}\rangle - 1 \cdot \epsilon \geq b_i \bigm| i \in \{1, \ldots, m\}, \bowtie_i \in \{>\} \,\bigr\} & \\ \cup \bigl\{\, \langle \boldsymbol{a}_i, \boldsymbol{x}\rangle + 0 \cdot \epsilon \geq b_i \bigm| i \in \{1, \ldots, m\}, \bowtie_i \in \{\geq, >\} \,\bigr\}. &\end{aligned}$$

Definition 8. (gen_repr$_G$.) *Let $\mathcal{G} = (R, P, C)$ be an extended generator system on the vector space $\mathbb{R}^n$. The function* gen_repr$_G$ *associates to $\mathcal{G}$ the generator system* gen_repr$_G(\mathcal{G}) \stackrel{\mathrm{def}}{=} (R', P')$ *on the vector space $\mathbb{R}^{n+1}$, where*

$$\begin{aligned}R' &= \bigl\{-\boldsymbol{e}_\epsilon\bigr\} \cup \bigl\{\, (\boldsymbol{r}^{\mathrm{T}}, 0)^{\mathrm{T}} \bigm| \boldsymbol{r} \in R \,\bigr\}, \\ P' &= \bigl\{\, (\boldsymbol{p}^{\mathrm{T}}, 1)^{\mathrm{T}} \bigm| \boldsymbol{p} \in P \,\bigr\} \cup \bigl\{\, (\boldsymbol{q}^{\mathrm{T}}, 0)^{\mathrm{T}} \bigm| \boldsymbol{q} \in C \,\bigr\}.\end{aligned}$$

It can be seen that, for each strict inequality contained in $\mathcal{C}$, the representation function con_repr$_G$ adds both the strict and the non-strict inequality encodings. This ensures that condition (2) of Definition 1 is met.

In contrast, for each point in the generator system, the function gen_repr$_G$ no longer adds the corresponding closure point. As a matter of fact, these closure points are no longer needed, because they can be generated by combining the corresponding point with the ray $-\boldsymbol{e}_\epsilon$, which is always added. As a consequence, the G-ϵ-polyhedron for a non-empty NNC polyhedron is always unbounded. Since the encodings for ϵ-polyhedra in this subclass require a linear number of additional constraints versus a constant number of additional generators, they are called "generator-biased".

The following result states the correctness of the four encoding functions.

Proposition 2. *Let $(\mathcal{C}, \mathcal{G}) \equiv \mathcal{P} \in \mathbb{P}_n$. Then*

1. $\mathrm{con}\bigl(\mathrm{con\_repr}_C(\mathcal{C})\bigr) \Rrightarrow_C \mathcal{P}$, $\quad \mathrm{gen}\bigl(\mathrm{gen\_repr}_C(\mathcal{G})\bigr) \Rrightarrow_C \mathcal{P}$;
2. $\mathrm{con}\bigl(\mathrm{con\_repr}_G(\mathcal{C})\bigr) \Rrightarrow_G \mathcal{P}$, $\quad \mathrm{gen}\bigl(\mathrm{gen\_repr}_G(\mathcal{G})\bigr) \Rrightarrow_G \mathcal{P}$.

Any ϵ-polyhedron included in the half-space defined by $\epsilon \leq 0$ actually encodes the empty NNC polyhedron. Operations such as the intersection of NNC polyhedra and the application of affine transformations can be safely performed on any of the constraint-biased or generator-biased ϵ-polyhedra for the arguments; the same holds for the convex polyhedral hull operation, provided neither of the arguments is empty.

Proposition 3. *Let* $\Rrightarrow_Y \in \{\Rrightarrow_C, \Rrightarrow_G\}$. *Suppose* $\mathcal{R} \Rrightarrow_Y \mathcal{P}$, *and* $\mathcal{R}_1 \Rrightarrow_Y \mathcal{P}_1$ *and* $\mathcal{R}_2 \Rrightarrow_Y \mathcal{P}_2$. *Then*

1. $\mathcal{P} = \varnothing$ *if and only if* $\mathcal{R} \subseteq \mathrm{con}\bigl(\{\epsilon \leq 0\}\bigr)$;
2. $\mathcal{R}_1 \cap \mathcal{R}_2 \Rrightarrow_Y \mathcal{P}_1 \cap \mathcal{P}_2$;
3. $(\mathcal{P}_1 \neq \varnothing \land \mathcal{P}_2 \neq \varnothing) \implies (\mathcal{R}_1 \uplus \mathcal{R}_2 \Rrightarrow_Y \mathcal{P}_1 \uplus \mathcal{P}_2)$;
4. *let* $f \stackrel{\mathrm{def}}{=} \lambda \boldsymbol{x} \in \mathbb{R}^n \mathrel{.} A\boldsymbol{x} + \boldsymbol{b}$ *be any affine transformation defined on* $\mathbb{P}_n$; *then* $g(\mathcal{R}) \Rrightarrow_Y f(\mathcal{P})$, *where*
$$g \stackrel{\mathrm{def}}{=} \lambda \begin{pmatrix} \boldsymbol{x} \\ \epsilon \end{pmatrix} \in \mathbb{R}^{n+1} \mathrel{.} \begin{pmatrix} A & \boldsymbol{0} \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{x} \\ \epsilon \end{pmatrix} + \begin{pmatrix} \boldsymbol{b} \\ 0 \end{pmatrix}$$
is the corresponding affine transformation on $\mathbb{CP}_{n+1}$.

4 Discussion and Future Work

The encoding based on G-ϵ-polyhedra has dual properties with respect to the one based on C-ϵ-polyhedra. In particular, using a C-ϵ-polyhedron, the encoding of an NNC polyhedron may require a similar number of constraints but about twice the number of generators, while, using a G-ϵ-polyhedron, it may require a similar number of generators but twice the number of constraints.

Increasing the number of constraints or generators, besides affecting the space efficiency, will have an impact on the running time of many of the various polyhedral operations required by applications in data-flow analysis and verification. In implementations based on the double description method, the most expensive operation is deriving a constraint (generator) representation from a generator (constraint) representation: this is, in the worst case, exponential in the number of generators (constraints). These conversions are performed because some of the other operations are more efficiently expressed and implemented on constraint systems (imposing new constraints and computing the relation between a polyhedron and a generator), some are better done on generator systems (adding new generators, convex polyhedral hull, projection onto a lower dimensional space, computing the relation between a polyhedron and a generator, testing for finiteness, time-elapse [6, 7]), and some require both the constraint and the generator systems (such as testing for inclusion and widenings [5, 4]). For these reasons, it seems likely that the performance of one encoding with respect to the other will depend on the particular application and, more specifically, on the kind of polyhedra and operations that are more common in that application. An implementation of the proposed techniques is ongoing and this will be useful to conduct experiments on realistic examples. It would also be interesting to investigate whether efficient techniques can be devised so as to use both constraint- and generator-biased encodings, switching dynamically from one to the other in an attempt to maximize performance.

As pointed out in [2], even if a constraint or generator system defining the constraint-biased ϵ-polyhedron is minimized (using the usual minimization for closed polyhedra), the system may still encode redundant strict inequality constraints or redundant points, respectively. The solution proposed in [2] is the definition of a stronger form of minimization, ensuring that no subset of the constraint/generator system still defines a constraint-biased ϵ-polyhedron for the same NNC polyhedron. It would be interesting to define the same notion of strong minimal form, but this time for any ϵ-polyhedron being careful that if it is constraint- or generator-biased before minimization, it remains constraint- or generator-biased, respectively, after the minimization.

References

1. 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/`.
2. R. Bagnara, E. Ricci, E. Zaffanella, and P. M. Hill. Possibly not closed convex polyhedra and the Parma Polyhedra Library. In M. Hermenegildo and G. Puebla, editors, *Static Analysis: 9th International Symposium, SAS 2002*, volume 2477 of *Lecture Notes in Computer Science*, Madrid, Spain, 2002. Springer-Verlag, Berlin. To appear.
3. 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.
4. 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.
5. 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.
6. 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.
7. 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.
8. T. S. Motzkin, H. Raiffa, G. L. Thompson, and R. M. Thrall. The double description method. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games – Volume II*, number 28 in Annals of Mathematics Studies, pages 51–73. Princeton University Press, Princeton, New Jersey, 1953.