Possibly Not Closed Convex Polyhedra and the Parma Polyhedra Library*

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

[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. The domain of convex polyhedra is employed in several systems for the analysis and verification of hardware and software components. Current applications span imperative, functional and logic languages, synchronous languages and synchronization protocols, real-time and hybrid systems. Since the seminal work of P. Cousot and N. Halbwachs, convex polyhedra have thus played an important role in the formal methods community and several critical tasks rely on their software implementations. Despite this, existing libraries for the manipulation of convex polyhedra are still research prototypes and suffer from limitations that make their usage problematic, especially in critical applications. These limitations concern inaccuracies in the documentation of the underlying theory, code and interfaces; numeric overflow and underflow; use of not fully dynamic data-structures and poor mechanisms for error handling and recovery. In addition, there is inadequate support for polyhedra that are not necessarily closed (NNC), i.e., polyhedra that are described by systems of constraints where strict inequalities are allowed to occur. This paper presents the Parma Polyhedra Library, a new, robust and complete implementation of NNC convex polyhedra, concentrating on the distinctive features of the library and on the novel theoretical underpinnings.

Dedicated to the memory of Hervé Le Verge

1 Introduction

Convex polyhedra are regions of some n-dimensional space that are bounded by a finite set of hyperplanes. A convex polyhedron in $\mathbb{R}^n$ describes a relation between n real-valued quantities. The class of all such relations turns out to be useful for the representation of the abstract properties of various kinds of complex systems.

* 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" and by EPSRC grant GR/R53401/01.

The seminal work of P. Cousot and N. Halbwachs [14] introduced the use of convex polyhedra as a domain of descriptions to solve, by *abstract interpretation* [12], a number of 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 [5, 16] and of linear hybrid automata (an extension of finite-state machines that models time requirements) [18, 19], for the computer-aided formal verification of concurrent and reactive systems based on temporal specifications [24], for inferring argument size relationships in logic programs [3], and for the automatic parallelization of imperative programs [27]. Since the work of Cousot and Halbwachs, convex polyhedra have thus played an important role in the formal methods community and new uses continue to emerge (see, e.g., [10, 15]). As a consequence, several critical tasks, such as checking the correctness of synchronization protocols or verifying the absence of run-time errors of systems whose failure can cause serious damage, rely on the software implementations of convex polyhedra.

Traditionally, convex polyhedra are assumed to be topologically closed and described by constraint systems containing linear equations and non-strict linear inequalities. However, some tasks need to use convex polyhedra that are not necessarily closed (NNC), i.e., polyhedra that are described by constraint systems possibly containing strict linear inequalities (in addition to equations and non-strict inequalities). Strict inequalities are important, for instance, in order to directly represent non-intersecting temporal regions, as is often the case when modeling applications where synchronization protocols, asynchronous interactions and temporal constraints come into play. Recently, they have also been used for the automatic computation of linear ranking functions [10].

Prior to the release of the Parma Polyhedra Library (PPL) [2] which is the subject of this paper, four libraries for the manipulation of convex polyhedra were (and continue to be) available:[3]

1. Polylib, designed and written by H. Le Verge and D. K. Wilde [22, 29];
2. *PolyLib*, the successor of the library by Le Verge and Wilde [23];
3. New Polka, by B. Jeannet [21];
4. the polyhedra library that comes with the HyTech tool [19].

All libraries,[4] including the PPL, use the same basic technique for representing convex polyhedra (the Double Description method described in Section 3) and the same core algorithm for manipulating that representation (the implementation and extension of N. V. Chernikova's algorithms [7–9] by Le Verge [22]).

[3] Note that we restrict ourselves to those libraries that provide the services required by applications in static analysis and computer-aided verification. There are other libraries for convex polyhedra, such as those developed in the field of computational geometry, that, as far as we can tell, are unsuitable for the tasks of interest in this paper. Moreover, we only consider the libraries that are freely available.

[4] We refer to the following versions that, at the time of writing, are the latest available: Polylib 2.1, *PolyLib* 5.0.4, New Polka 1.1.3c, HyTech 1.04f.

Apart from the PPL, only New Polka supports NNC polyhedra. However, this support is incomplete, incurring avoidable inefficiencies and leaving the client application with the non-trivial task of a correct interpretation of the results.

Libraries 1 and 4 may incur overflow problems whereas 2 and 3 can use unbounded integers as coefficients. Libraries 2 and 3 can be configured to use finite integral types for extra speed but this, of course, comes with the possibility of overflows. In addition, 2 (and, according to [4], 4) can use floating point values, so that overflow, underflow and rounding errors can affect the results.

In libraries 1–4, matrices of coefficients, which are the main data structures used to represent polyhedra, cannot grow dynamically and the client application is ultimately responsible for specifying their dimensions. Since the worst case space complexity of the methods employed is exponential,[5] in general the client application cannot make a safe and practical choice: specifying small dimensions may provoke a run-time failure; generous dimensions may waste significant amounts of memory and, again, result in unnecessary run-time failures.

The problems caused by run-time errors such as overflow and memory allocation failure could be mitigated or even solved by suitable mechanisms for error detection, handling and recovery. This requires the ability to detect the problem, releasing any affected data-structure whether it be completely or only partially constructed,[6] and continue the computation with an alternative method (e.g., by reverting to an interval-based approximation). Library 1 detects some errors and sets an error flag whereas libraries 3 and 4 detect some errors, print an error message and abort. Library 2 detects more errors, sometimes setting a flag and sometimes printing a message and aborting. A somewhat drastic approach to error recovery is taken by STeP, the Stanford Temporal Prover [24]. STeP uses the Polka polyhedra library [18] by Halbwachs and Y.-E. Proy for the automatic generation of invariants.[7] The manual for the latest released version of STeP [6] specifies that this facility is in an *"experimental" state*[8] and run as an external process, completely independent from the STeP environment. The user can set three environment variables controlling the maximum memory and time resources allowed for the invariant computation process [6, Section 7.3]. If one or both these limits are trespassed, the invariant generation will fail. The user can also explicitly interrupt the invariant generation by clicking a button. This terminates the invariant generation, but [6, page 95]

[5] For instance, an n-dimensional closed hypercube has $2n$ facets (that are described by as many constraints) and 2^n vertices (which are individually stored in a matrix).

[6] Leaving useless data-structures around is a simple memory leak problem; the presence and reachability of partially constructed objects is a more serious issue since it can result in unpredictable behavior.

[7] The Polka polyhedra library is not available in source format and binaries are distributed under rather restrictive conditions (until sometimes around 1996 they could be freely downloaded.

[8] Double quotes in the original.

this may leave a separate UNIX process running, which users may have to terminate independently from a UNIX prompt (see the UNIX `kill` command man pages).

All in all, this possibility does not look very attractive.

Libraries 1–3 are free software released under the GNU General Public License (GPL, see `http://www.gnu.org/`). Thus, when faced with the need to overcome the above mentioned limitations, anyone can freely take and use them as the basis for further development. However, it appears that none of the libraries provide documentation for the interfaces and code that is adequate for an outsider to make such improvements with any real confidence. This feeling of insecurity is aggravated by the discovery of some errors and imprecisions in the theoretical sections of the documentation of some libraries (see Section A for more information). Moreover, a complete solution to issues such as error recovery and fully dynamic memory allocation requires a somewhat radical departure from the existing code bases.

For all these reasons we decided to write the PPL, a robust and complete implementation of convex polyhedra. This paper describes the library concentrating on some of its distinctive features and the novel theoretical underpinnings.

The plan of the paper is as follows: Section 2 recalls some basic notation and terminology; Section 3 introduces convex polyhedra and the Double Description method paying attention to common pitfalls; Section 4 presents the NNC polyhedra and describes (from a theoretical perspective) how they are handled in the PPL; Section 5 briefly describes the design and implementation of the PPL and how it addresses all the limitations we have just discussed; Section 6 concludes.

2 Preliminaries

In this paper, all topological arguments refer to the topological space $\mathbb{R}^n$ with the standard topology. The *topological closure* of $S \subseteq \mathbb{R}^n$ is denoted by $\mathbb{C}(S)$ and defined as $\mathbb{C}(S) \stackrel{\mathrm{def}}{=} \bigcap\{\, C \subseteq \mathbb{R}^n \mid S \subseteq C \text{ and } C \text{ is closed} \,\}$. We denote the set of all non-negative reals 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 $\mathbb{R}^n$, called *the origin*, having all components equal to zero. A vector $\boldsymbol{v} \in \mathbb{R}^n$ can be also interpreted as a matrix in $\mathbb{R}^{n \times 1}$ and manipulated accordingly using the usual definitions for addition, multiplication (both by a scalar and by another matrix), and transposition, which is denoted by $\boldsymbol{v}^{\mathrm{T}}$. The *scalar product* of $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^n$, denoted $\langle \boldsymbol{v}, \boldsymbol{w} \rangle$, is the real number $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{w} = \sum_{i=1}^{n} v_i w_i$. For any relational operator $\bowtie \in \{=, \geq, \leq, <, >\}$, we write $\boldsymbol{v} \bowtie \boldsymbol{w}$ to denote the conjunctive proposition $\bigwedge_{i=1}^{n} (v_i \bowtie w_i)$. In contrast, $\boldsymbol{v} \neq \boldsymbol{w}$ will denote the proposition $\neg(\boldsymbol{v} = \boldsymbol{w})$. We will sometimes use the convenient notation $a \bowtie_1 b \bowtie_2 c$ to denote the conjunction $a \bowtie_1 b \wedge b \bowtie_2 c$ and we will not distinguish conjunctions of propositions from sets of propositions. For each set $S \subseteq \mathbb{R}^n$ of finite cardinality m, we denote by $\operatorname{matrix}(S) \subseteq \mathbb{R}^{n \times m}$ the set of all matrices having S as the set of their columns.

3 Convex Polyhedra and the Double Description Method

For each vector $\boldsymbol{a} \in \mathbb{R}^n$ and scalar $b \in \mathbb{R}$, where $\boldsymbol{a} \neq \boldsymbol{0}$, the linear (non-strict) inequality constraint $\langle \boldsymbol{a}, \boldsymbol{x} \rangle \geq b$ defines a closed affine half-space. The linear equality constraint $\langle \boldsymbol{a}, \boldsymbol{x} \rangle = b$ defines an affine hyperplane. Convex polyhedra are usually described as finite systems of linear equality and inequality constraints. When working at the theoretical level, it is simpler to express each equality constraint as the intersection of the two half-spaces $\langle \boldsymbol{a}, \boldsymbol{x} \rangle \geq b$ and $\langle -\boldsymbol{a}, \boldsymbol{x} \rangle \geq -b$.

Definition 1. (Closed polyhedron.) *The set $\mathcal{P} \subseteq \mathbb{R}^n$ is a* closed 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 $\mathcal{P} = \varnothing$.*

Alternatively, the definition of a convex polyhedron can be based on some of the geometric features of the set of solutions of such a system of constraints. A vector $\boldsymbol{r} \in \mathbb{R}^n$ such that $\boldsymbol{r} \neq \boldsymbol{0}$ is a *ray* (or *direction of infinity*) of a non-empty polyhedron $\mathcal{P} \subseteq \mathbb{R}^n$ if, for every point $\boldsymbol{p} \in \mathcal{P}$ and every $\lambda \in \mathbb{R}_+$, it holds $\boldsymbol{p}+\lambda\boldsymbol{r} \in \mathcal{P}$; a vector $\boldsymbol{l} \in \mathbb{R}^n$ is a *line* of $\mathcal{P}$ if both $\boldsymbol{l}$ and $-\boldsymbol{l}$ are rays of $\mathcal{P}$. The empty polyhedron has no rays and no lines. As was the case for equality constraints, the theory can dispense with the use of lines by using the corresponding pair of opposite rays. The following theorem is a simple consequence of well known theorems by Minkowski and Weyl [28].

Theorem 1. *The set $\mathcal{P} \subseteq \mathbb{R}^n$ is a closed polyhedron if and only if there exist finite sets $R, P \subseteq \mathbb{R}^n$ of cardinality k and ℓ, respectively, such that $\boldsymbol{0} \notin R$ and, for any matrices $K \in \mathbb{R}^{n \times k}$ and $L \in \mathbb{R}^{n \times \ell}$ where $K \in \operatorname{matrix}(R)$ and $L \in \operatorname{matrix}(P)$,*

$$\mathcal{P} = \left\{ K\boldsymbol{\mu} + L\boldsymbol{\nu} \in \mathbb{R}^n \;\middle|\; \boldsymbol{\mu} \in \mathbb{R}^k_+, \boldsymbol{\nu} \in \mathbb{R}^\ell_+, \textstyle\sum_{i=1}^{\ell} \nu_i = 1 \right\}.$$

When $\mathcal{P} \neq \varnothing$, we say that $\mathcal{P}$ is described by the *generator system* $\mathcal{G} = (R, P)$. In particular, the vectors of R and P are rays and points of $\mathcal{P}$, respectively. Informally speaking, if no "supporting" point is provided then an empty polyhedron is obtained; formally, $\mathcal{P} = \varnothing$ if and only if $P = \varnothing$. By convention, the empty system (i.e., the system with $R = \varnothing$ and $P = \varnothing$) is the only generator system for the empty polyhedron. It is worth stressing that, in general, the vectors in R and P are not the *extreme* rays and the *vertices* of the polyhedron (see Section A): for instance, any half-space of $\mathbb{R}^2$ has two extreme rays and no vertices, but any generator system describing it will contain at least three rays and one point.

The combination of the two approaches outlined above is the basis of the Double Description (DD) method [26], which exploits the duality principle to compute each representation starting from the other one, possibly minimizing the descriptions. We will write $\operatorname{con}(\mathcal{C})$ and $\operatorname{gen}(\mathcal{G})$ to denote the polyhedra described by the finite constraint system $\mathcal{C}$ and generator system $\mathcal{G}$, respectively.

Definition 2. (DD pair and minimal forms.) *If $\operatorname{con}(\mathcal{C}) = \operatorname{gen}(\mathcal{G}) = \mathcal{P}$, then $(\mathcal{C}, \mathcal{G})$ is said to be a* DD pair *for $\mathcal{P}$, and we write $(\mathcal{C}, \mathcal{G}) \equiv \mathcal{P}$. We say that*

- $\mathcal{C}$ *is in* minimal form *if there does not exist* $\mathcal{C}' \subset \mathcal{C}$ *such that* $\mathrm{con}(\mathcal{C}') = \mathcal{P}$*;*
- $\mathcal{G} = (R, P)$ *is in* minimal form *if there does not exist* $\mathcal{G}' = (R', P') \neq \mathcal{G}$ *such that* $R' \subseteq R$*,* $P' \subseteq P$ *and* $\mathrm{gen}(\mathcal{G}') = \mathcal{P}$*;*
- *the DD pair* $(\mathcal{C}, \mathcal{G})$ *is in* minimal form *if* $\mathcal{C}$ *and* $\mathcal{G}$ *are both in minimal form.*

The set of all closed polyhedra on the vector space $\mathbb{R}^n$, denoted $\mathbb{CP}_n$, can be partially ordered by set-inclusion to form a lattice, having the emptyset and $\mathbb{R}^n$ as the bottom and top element, respectively. The binary meet operation is thus given by set-intersection, which is easily implemented by taking the union of the constraint systems representing the two arguments. Similarly, the binary join operation, denoted $\uplus$ and called *convex polyhedral hull* (poly-hull, for short), is implemented by taking the union of the two arguments' generator systems. Note that, in general, the poly-hull of two polyhedra is different from their convex hull [28] (see Section A).

4 Handling Not Necessarily Closed Polyhedra

For each vector $\boldsymbol{a} \in \mathbb{R}^n$ and scalar $b \in \mathbb{R}$, where $\boldsymbol{a} \neq \boldsymbol{0}$, the linear strict inequality constraint $\langle \boldsymbol{a}, \boldsymbol{x} \rangle > b$ defines an open affine half-space. By allowing strict inequalities to occur in the system of constraints, it is possible to define polyhedra that are not necessarily closed (NNC polyhedra, for short).

Definition 3. (NNC polyhedron.) *The set* $\mathcal{P} \subseteq \mathbb{R}^n$ *is a* NNC polyhedron *if and only 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* $\mathcal{P} = \varnothing$*.*

We denote by $\mathbb{P}_n$ the set of all NNC polyhedra on the vector space $\mathbb{R}^n$. Obviously, we have $\mathbb{CP}_n \subseteq \mathbb{P}_n$ (note that $\mathbb{CP}_n = \mathbb{P}_n$ if and only if $n = 0$). When partially ordered by set-inclusion, $\mathbb{P}_n$ is a lattice and $\mathbb{CP}_n$ is a sublattice of $\mathbb{P}_n$.

To the best of the authors' knowledge, the first software library (based on the DD method) allowing for the computation over the domain $\mathbb{P}_n$ was the Polka library [18], where each NNC polyhedron $\mathcal{P} \in \mathbb{P}_n$ is embedded into a closed polyhedron $\mathcal{R} \in \mathbb{CP}_{n+1}$. The additional dimension of the vector space, usually labeled by the letter ϵ, encodes the topological closedness of each affine half-space in the constraint description for $\mathcal{P}$. Namely, if $\mathcal{P} = \mathrm{con}(\mathcal{C})$, where

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

then the representation polyhedron is defined as $\mathcal{R} = \mathrm{con}\bigl(\mathrm{con\_repr}(\mathcal{C})\bigr)$, where

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

It should be stressed that the choice of the value -1 for the coefficients of the additional variable ϵ in the constraints representing strict inequalities is rather arbitrary: any other negative value will do. Similarly, the side constraint $\epsilon \leq 1$ could be replaced by any other ϵ-*upper-bound* constraint, i.e., by any constraint $\epsilon \leq \delta$ such that $\delta > 0$. Different, though equivalent, constraint systems $\mathcal{C}_j$ describing $\mathcal{P}$ may be embedded into different representation polyhedra $\mathcal{R}_j = \mathrm{con}\bigl(\mathrm{con\_repr}(\mathcal{C}_j)\bigr)$. We shall abuse notation by writing $\mathcal{R} = \mathrm{con\_repr}(\mathcal{P})$ as a shorthand for $\mathcal{R} = \mathrm{con}\bigl(\mathrm{con\_repr}(\mathcal{C})\bigr)$, provided the constraint system $\mathcal{C}$ describing $\mathcal{P}$ is clear from context.

As far as we know, no available implementation of the double description method offers *full* support for NNC polyhedra. Partial support is provided by the New Polka library [21], which can be initialized to also work with strict inequalities. However, the correct encoding of the different kinds of constraints, the addition of the required side conditions on the ϵ dimension and the correct interpretation of the obtained results, are all under the user's responsibility. This kind of approach, which requires the user to know so many implementation details, is far from being satisfactory.

4.1 The Generators of NNC Polyhedra

One of the fundamental features of the DD method, and the very reason for its name, is the ability to represent a closed polyhedron using a system of constraints or a system of generators. While being equivalent, there are contexts where each of these descriptions is the most appropriate, so that a good library should provide the client application with both possibilities.

Any NNC polyhedron can be easily described by using constraint systems containing strict inequalities, but a similar generalization of the concept of generator system seems to be missing. This causes an asymmetry in the handling of NNC polyhedra using the DD method that is reflected in existing software libraries. For instance, the following sentence comes from the documentation of New Polka [21, Section 1.1.4, page 10] (where s denotes the ϵ coefficient):

> Don't ask me the intuitive meaning of $s \neq 0$ in rays and vertices !

The problem is discussed in some more detail in [17, Section 4.5, pp. 10–11]:

> While strict inequations handling is transparent for constraints (being displayed accurately), the extra dimension added to the variables space is apparent when it comes to generators : one extra coefficient, resp. extra vertices (as `epsilon` is bounded), materialize this dimension in every generator, resp. generators system.
> This makes more difficult to define polyhedra with the only help of generators : one should carefully study the extra vertices with non null `epsilon` coefficients added to constraints defined polyhedra, in the case of large inequations, and the case of strict inequations.

We will now show how, by decoupling the user interface from the details of the particular implementation, it is possible to provide an intuitive generalization of the concept of generator system, so that the geometric features of any NNC

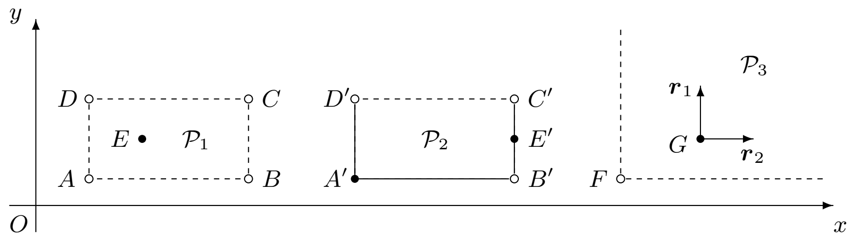

Fig. 1. Using closure points to define NNC polyhedra.

polyhedron could be accurately represented. The key step is the introduction of a new kind of generators.

Definition 4. (Closure point.) *A vector* $\boldsymbol{c} \in \mathbb{R}^n$ *is a* closure point *of* $S \subseteq \mathbb{R}^n$ *if and only if* $\boldsymbol{c} \in \mathbb{C}(S)$.

When considering NNC polyhedra, closure points can be characterized by a property which is similar to the one used when defining rays.

Proposition 1. *A vector* $\boldsymbol{c} \in \mathbb{R}^n$ *is a closure point of the NNC polyhedron* $\mathcal{P} \in \mathbb{P}_n$ *if and only if* $\mathcal{P} \neq \varnothing$ *and for every point* $\boldsymbol{p} \in \mathcal{P}$ *and* $\lambda \in \mathbb{R}$ *such that* $0 < \lambda < 1$, *it holds* $\lambda \boldsymbol{p} + (1-\lambda)\boldsymbol{c} \in \mathcal{P}$.

We are now able to provide a parametric description for any NNC polyhedron.

Theorem 2. *The set* $\mathcal{P} \subseteq \mathbb{R}^n$ *is a NNC polyhedron if and only if there exist finite sets* $R, P, C \subseteq \mathbb{R}^n$ *of cardinality* k, ℓ *and* m, *respectively, such that* $\mathbf{0} \notin R$ *and, for any matrices* $K \in \mathbb{R}^{n \times k}$, $L \in \mathbb{R}^{n \times \ell}$, $M \in \mathbb{R}^{n \times m}$ *where* $K \in \mathrm{matrix}(R)$, $L \in \mathrm{matrix}(P)$ *and* $M \in \mathrm{matrix}(C)$,

$$\mathcal{P} = \left\{ K\boldsymbol{\mu} + L\boldsymbol{\nu} + M\boldsymbol{\eta} \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \boldsymbol{\mu} \in \mathbb{R}^k_+, \boldsymbol{\nu} \in \mathbb{R}^\ell_+, \boldsymbol{\nu} \neq \mathbf{0}, \boldsymbol{\eta} \in \mathbb{R}^m_+, \\ \sum_{i=1}^{\ell} \nu_i + \sum_{i=1}^{m} \eta_i = 1 \end{array} \right\}.$$

When $\mathcal{P} \neq \varnothing$, we say that $\mathcal{P}$ is described by the *extended* generator system $\mathcal{G} = (R, P, C)$. As was the case for closed polyhedra, the vectors in R and P are rays and points of $\mathcal{P}$, respectively. The condition $\boldsymbol{\nu} \neq \mathbf{0}$ ensures that at least one of the points of P plays an active role in any convex combination of the vectors of P and C. It follows from Proposition 1 that the vectors of C are closure points of $\mathcal{P}$. Since rays and closure points need a supporting point, we have $\mathcal{P} = \varnothing$ if and only if $P = \varnothing$.

In Figure 1, we provide a few examples on the use of extended generator systems for the description of NNC polyhedra: (closure) points are represented by small (un-) filled circles, whereas rays are represented by vectors that, for notational convenience, are applied to points.

The NNC polyhedron $\mathcal{P}_1$ is an open rectangle and is described by the closure points A, B, C, D and the point E; note that E could have been replaced by any other point of $\mathcal{P}_1$, whereas all the four closure points have to be included in any generator system for $\mathcal{P}_1$. The NNC polyhedron $\mathcal{P}_2$ is another rectangle which is neither closed nor open: since A' is a point, the open segments $]A', B'[$ and $]A', D'[$ are included in $\mathcal{P}_2$; similarly, the open segment $]B', C'[$ is included in $\mathcal{P}_2$ because E' is a point of the generator system (note that E' is needed, since both B' and C' are not in $\mathcal{P}_2$, but it could have been replaced by any other point lying on this open segment); in contrast, the closed segment $[C', D']$ is disjoint from $\mathcal{P}_2$, because neither C' nor D' are points of $\mathcal{P}_2$. Finally, the NNC polyhedron $\mathcal{P}_3$ can be regarded as the translation by F of the open positive orthant. Thus the generator system includes the closure point F, the rays $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ and the point G; again, the latter could have been replaced by any other point of $\mathcal{P}_3$.

We will now show how the high level description of a NNC polyhedron provided by an extended generator system can be mapped into an implementation based on the ϵ dimension approach. Namely, if $\mathcal{G} = (R, P, C)$ is the extended generator system describing $\mathcal{P} \in \mathbb{P}_n$, the corresponding closed representation $\mathcal{R} \in \mathbb{CP}_{n+1}$ is described by the generator system $\operatorname{gen\_repr}(\mathcal{G}) \stackrel{\text{def}}{=} (R', P')$ where

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

Even in this case, the value 1 for the coordinate of the ϵ dimension in the translation of points is almost arbitrary: any other positive value could be chosen. Different though equivalent extended generator systems $\mathcal{G}_j$ describing $\mathcal{P} \in \mathbb{P}_n$ may result in different representation polyhedra $\mathcal{R}_j = \operatorname{gen}\bigl(\operatorname{gen\_repr}(\mathcal{G}_j)\bigr)$. It is worth noting that the closure points of $\mathcal{P}$ are mapped to points of $\mathcal{R}$ having a zero ϵ coordinate. In contrast, the points of $\mathcal{P}$ are mapped to a pair of points of $\mathcal{R}$, having a zero and a strictly positive ϵ coordinate, respectively; by doing this, we explicitly enforce in the closed representation the key invariant saying that any point of $\mathcal{P}$ is also a closure point of $\mathcal{P}$.

Definition 5. (ϵ-representation.) *A polyhedron $\mathcal{R} \in \mathbb{CP}_{n+1}$ is said to be an* ϵ-representation *if and only if*

$$\exists \delta > 0 \,.\, \mathcal{R} \subseteq \operatorname{con}\bigl(\{0 \le \epsilon \le \delta\}\bigr); \tag{1}$$

$$\bigl(\exists \epsilon > 0 \,.\, (\boldsymbol{x}^{\mathrm{T}}, \epsilon)^{\mathrm{T}} \in \mathcal{R}\bigr) \implies (\boldsymbol{x}^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}. \tag{2}$$

$\mathcal{R}$ is said to be an ϵ-representation for *$\mathcal{P} \in \mathbb{P}_n$, denoted $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P}$, if $\mathcal{R}$ is an ϵ-representation and*

$$\mathcal{P} = [\![\mathcal{R}]\!] \stackrel{\text{def}}{=} \bigl\{\, \boldsymbol{x} \in \mathbb{R}^n \bigm| \exists \epsilon > 0 \,.\, (\boldsymbol{x}^{\mathrm{T}}, \epsilon)^{\mathrm{T}} \in \mathcal{R} \,\bigr\}. \tag{3}$$

Proposition 2. *Let $(\mathcal{C}, \mathcal{G}) \equiv \mathcal{P} \in \mathbb{P}_n$. Then we have* $\operatorname{con}\bigl(\operatorname{con\_repr}(\mathcal{C})\bigr) \Rrightarrow_\epsilon \mathcal{P}$ *and* $\operatorname{gen}\bigl(\operatorname{gen\_repr}(\mathcal{G})\bigr) \Rrightarrow_\epsilon \mathcal{P}$.

Operations such as the intersection of NNC polyhedra and the application of affine transformations can be safely performed on any of the ϵ-representations of the arguments; the same holds for the poly-hull operation, provided none of the arguments is an empty NNC polyhedron. Some care has to be taken when testing the emptyness of a NNC polyhedron or the inclusion of a NNC polyhedron into another one [18]. For instance, any ϵ-representation included in the hyperplane $\epsilon = 0$ actually encodes the empty NNC polyhedron.

Proposition 3. *Let $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P}$, $\mathcal{R}_1 \Rrightarrow_\epsilon \mathcal{P}_1$ and $\mathcal{R}_2 \Rrightarrow_\epsilon \mathcal{P}_2$. Then*

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

4.2 The Issue of Minimization

When adopting the ϵ dimension approach proposed in [18], the computed representation $\mathcal{R} \in \mathbb{CP}_{n+1}$ of a NNC polyhedron $\mathcal{P} \in \mathbb{P}_n$ will depend not only on the particular constraint system considered, but also on the sequence of operations (intersections, poly-hulls, affine transformations, etc.) performed on the polyhedron. If not properly handled, such an abundance of possible representations may cause problems when trying to provide a non-redundant description of $\mathcal{P}$.

The reason is that libraries such as New Polka compute the minimal forms of the closed representation $\mathcal{R}$. Very often, such an approach results in a redundant description of the represented NNC polyhedron: there may be a different ϵ-representation for $\mathcal{P}$ that is characterized by a smaller number of constraints (generators). The following example illustrates this point.

Consider the two NNC polyhedra $\mathcal{P}_1, \mathcal{P}_2 \in \mathbb{P}_1$ defined as

$$\mathcal{P}_1 \stackrel{\text{def}}{=} \operatorname{con}\bigl(\{0 < x < 2\}\bigr), \qquad \mathcal{P}_2 \stackrel{\text{def}}{=} \operatorname{con}\bigl(\{2 < x < 3\}\bigr).$$

These polyhedra are encoded by the closed polyhedra $\mathcal{R}_1, \mathcal{R}_2 \in \mathbb{CP}_2$ such that[9]

$$\mathcal{R}_1 \stackrel{\text{def}}{=} \text{con\_repr}(\mathcal{P}_1) = \left\{ (x, \epsilon)^{\mathrm{T}} \in \mathbb{R}^2 \;\middle|\; \begin{array}{r} \epsilon \geq 0 \\ x - \epsilon \geq 0 \\ -x - \epsilon \geq -2 \end{array} \right\},$$
$$\mathcal{R}_2 \stackrel{\text{def}}{=} \text{con\_repr}(\mathcal{P}_2) = \left\{ (x, \epsilon)^{\mathrm{T}} \in \mathbb{R}^2 \;\middle|\; \begin{array}{r} \epsilon \geq 0 \\ x - \epsilon \geq 2 \\ -x - \epsilon \geq -3 \end{array} \right\}.$$

[9] In both cases, we do not explicitly include the ϵ-upper-bound constraint $\epsilon \leq 1$, which happens to be redundant.

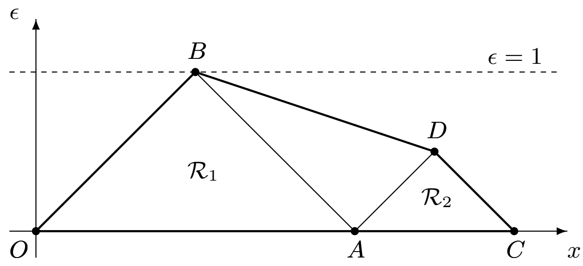

Fig. 2. The ϵ-representations of $\mathcal{P}_1$ and $\mathcal{P}_2$ and of their poly-hull.

Suppose now that the user wants to compute the poly-hull of the two original polyhedra, therefore obtaining the NNC polyhedron

$$\mathcal{P}_3 \stackrel{\text{def}}{=} \operatorname{con}\bigl(\{0 < x < 3\}\bigr).$$

At the representation level, the situation will be the one described in Figure 2: $\mathcal{P}_3$ is represented by the closed polyhedron generated by the four vertices O, C, D, and B, whereas point A is identified as redundant. Formally,

$$\mathcal{R}_3 \stackrel{\text{def}}{=} \left\{ (x, \epsilon)^{\mathrm{T}} \in \mathbb{R}^2 \;\middle|\; \begin{array}{r} \epsilon \geq 0 \\ x - \epsilon \geq 0 \\ -x - \epsilon \geq -3 \\ -x - 3\epsilon \geq -4 \end{array} \right\}.$$

The last non-strict inequality, which corresponds to the segment $[B, D]$ in Figure 2, is not redundant as far as the ϵ-representation $\mathcal{R}_3$ is concerned. However, this non-strict inequality stands for the strict inequality $x < 4$, which is clearly redundant when considering the represented polyhedron $\mathcal{P}_3$.

The problem outlined above is even more critical when dealing with higher dimension vector spaces: it is straightforward to devise examples where more than half of the constraints in the "minimized" representation happen to be redundant. Even when disregarding these pathological cases, redundancy could have a serious negative impact on the efficiency of some of the operations computed on the polyhedron, characterized by a worst case complexity which is exponential in the size of the description.

Besides efficiency issues, the presence of redundant constraints may also cause headaches to the users of the library. For instance, suppose one wants to know if a given NNC polyhedron is not topologically closed. A common user (i.e., all the users but the experts) may be tempted to implement such a test by checking whether the constraint system in minimal form contains any strict inequality. Unfortunately, such an approach would be unsound, as can be easily observed by considering the scenario proposed in Figure 3. Here, the NNC polyhedron $\mathcal{P}_1$ is intersected with the NNC polyhedron

$$\mathcal{P}_4 \stackrel{\text{def}}{=} \operatorname{con}\bigl(\{1 \leq 4x \leq 3\}\bigr),$$

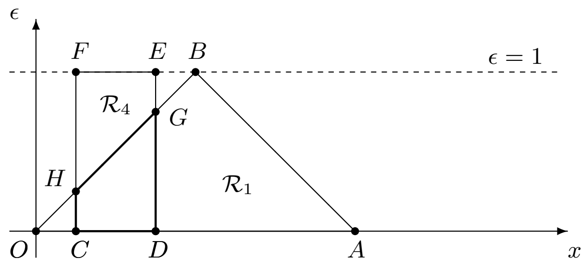

Fig. 3. The "minimized" trapezium $CDGH$, obtained by intersecting $\mathcal{R}_1$ and $\mathcal{R}_4$ and still representing the topologically closed NNC polyhedron $\mathcal{P}_4$, also encodes the strict inequality $x > 0$.

whose representation $\mathcal{R}_4 = \text{con\_repr}(\mathcal{P}_4)$ is the rectangle having vertices C, D, E and F. The resulting trapezium is another ϵ-representation for the NNC polyhedron $\mathcal{P}_4$, which is topologically closed. However, any constraint system describing the trapezium will also encode the strict inequality $x > 0$, corresponding to the closed segment $[G, H]$.

It is therefore meaningful to address the problem of providing a minimization procedure that is able to remove all of these redundancies. To this end, the introduction of some notation will be helpful.

We say that vector $\boldsymbol{v}$ *saturates* constraint $\langle \boldsymbol{a}, \boldsymbol{x} \rangle \bowtie b$ if and only if $\langle \boldsymbol{a}, \boldsymbol{v} \rangle = b$. For any constraint system $\mathcal{C}$ and generator system $\mathcal{G} = (R, P)$, we define

$$\text{sat\_con}(\boldsymbol{v}, \mathcal{C}) \stackrel{\text{def}}{=} \{\, c \in \mathcal{C} \mid \boldsymbol{v} \text{ saturates } c \,\};$$
$$\text{sat\_gen}(c, \mathcal{G}) \stackrel{\text{def}}{=} \{\, \boldsymbol{v} \in R \cup P \mid \boldsymbol{v} \text{ saturates } c \,\}.$$

Let $(\mathcal{C}, \mathcal{G}) \equiv \mathcal{R} \in \mathbb{CP}_{n+1}$ be such that $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P}$. The set of *strict* and *non-strict inequality encodings* $\mathcal{C}_>$ and $\mathcal{C}_\geq$ of constraint system $\mathcal{C}$ are defined as

$$\mathcal{C}_> \stackrel{\text{def}}{=} \Big\{ \big(\langle \boldsymbol{a}, \boldsymbol{x} \rangle + s \cdot \epsilon \geq b\big) \in \mathcal{C} \;\Big|\; \boldsymbol{a} \neq \boldsymbol{0}, s < 0 \Big\};$$
$$\mathcal{C}_\geq \stackrel{\text{def}}{=} \Big\{ \big(\langle \boldsymbol{a}, \boldsymbol{x} \rangle + s \cdot \epsilon \geq b\big) \in \mathcal{C} \;\Big|\; \boldsymbol{a} \neq \boldsymbol{0}, s = 0 \Big\};$$

we also define the set of *ϵ-upper-bounds* as

$$\mathcal{C}_\epsilon \stackrel{\text{def}}{=} \Big\{ \big(\langle \boldsymbol{a}, \boldsymbol{x} \rangle + s \cdot \epsilon \geq b\big) \in \mathcal{C} \;\Big|\; \boldsymbol{a} = \boldsymbol{0}, s < 0 \Big\}.$$

For ease of notation, a constraint $c \in \mathcal{C}_\epsilon$ will be usually denoted as $\epsilon \leq \delta$, where $\delta \stackrel{\text{def}}{=} \frac{b}{s}$. Note that, whenever $\mathcal{P} \neq \varnothing$, it will hold $\delta > 0$.

Similarly, the set of *closure point encodings* $\mathcal{G}_C$, the set of *point encodings* $\mathcal{G}_P$, and the set of *unmatched point encodings* $\mathcal{G}_U \subseteq \mathcal{G}_P$ of the generator system

$\mathcal{G} = (R, P)$ are defined as follows:

$$\begin{aligned}
\mathcal{G}_C &\stackrel{\text{def}}{=} \bigl\{ (\boldsymbol{p}^{\mathrm{T}}, e)^{\mathrm{T}} \in P \bigm| e = 0 \bigr\};\\
\mathcal{G}_P &\stackrel{\text{def}}{=} \bigl\{ (\boldsymbol{p}^{\mathrm{T}}, e)^{\mathrm{T}} \in P \bigm| e > 0 \bigr\};\\
\mathcal{G}_U &\stackrel{\text{def}}{=} \bigl\{ (\boldsymbol{p}^{\mathrm{T}}, e)^{\mathrm{T}} \in P \bigm| e > 0, (\boldsymbol{p}^{\mathrm{T}}, 0)^{\mathrm{T}} \notin P \bigr\}.
\end{aligned}$$

Definition 6. (Strong minimal form.) *Let $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P}$, where $(\mathcal{C}, \mathcal{G}) \equiv \mathcal{R}$ is a DD pair in minimal form. Then, we say that*

- *$\mathcal{C}$ is in* strong minimal form *if there does not exist a constraint system $\mathcal{C}'$ in minimal form such that $(\mathcal{C}'_> \cup \mathcal{C}'_\geq) \subset (\mathcal{C}_> \cup \mathcal{C}_\geq)$ and $\mathrm{con}(\mathcal{C}') \Rrightarrow_\epsilon \mathcal{P}$;*
- *$\mathcal{G} = (R, P)$ is in* strong minimal form *if there does not exist a generator system $\mathcal{G}' = (R', P') \neq \mathcal{G}$ such that $R' \subseteq R$, $P' \subseteq P$ and $\mathrm{gen}(\mathcal{G}') \Rrightarrow_\epsilon \mathcal{P}$.*

For the computation of strong minimal forms (smf's, for short), the key step is the identification of ϵ-redundant constraints and generators.

Definition 7. (ϵ-redundancy.) *Let $(\mathcal{C}, \mathcal{G}) \equiv \mathcal{R} \in \mathbb{CP}_{n+1}$. A constraint c is* ϵ-redundant *in $\mathcal{C}$ if $c \in \mathcal{C}_>$ and any of the following conditions holds:*

$$\mathrm{sat\_gen}(c, \mathcal{G}) \cap \mathcal{G}_C = \varnothing;$$
$$\exists c' \in \mathcal{C}_> \setminus \{c\} \,.\, \mathrm{sat\_gen}(c, \mathcal{G}) \setminus \mathcal{G}_P \subseteq \mathrm{sat\_gen}(c', \mathcal{G}).$$

A generator $\boldsymbol{p}$ is ϵ-redundant *in $\mathcal{G}$ if $\boldsymbol{p} \in \mathcal{G}_U$ and*

$$\exists \boldsymbol{p}' \in \mathcal{G}_P \setminus \{\boldsymbol{p}\} \,.\, \mathrm{sat\_con}(\boldsymbol{p}, \mathcal{C}) \cap \mathcal{C}_\geq \subseteq \mathrm{sat\_con}(\boldsymbol{p}', \mathcal{C}).$$

Proposition 4. *Let $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P} \neq \varnothing$, where $(\mathcal{C}, \mathcal{G}) \equiv \mathcal{R}$ is a DD pair in minimal form. Then, the following hold:*

1. *If c is ϵ-redundant in $\mathcal{C}$, then $\mathrm{con}\bigl(\mathcal{C} \setminus \{c\} \cup \{\epsilon \leq 1\}\bigr) \Rrightarrow_\epsilon \mathcal{P}$;*
2. *If $\boldsymbol{p}$ is ϵ-redundant in $\mathcal{G} = (R, P)$, then $\mathrm{gen}\Bigl(\bigl(R, P \setminus \{\boldsymbol{p}\}\bigr)\Bigr) \Rrightarrow_\epsilon \mathcal{P}$;*
3. *If $\mathcal{C}$ contains no ϵ-redundant constraint, then it is in* smf*;*
4. *If $\mathcal{G}$ contains no ϵ-redundant generator, then it is in* smf*.*

It is worth stressing that, even though $(\mathcal{C}, \mathcal{G}) \equiv \mathcal{R}$ is a DD pair for $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P}$, in general the independent application of the strong minimization process to both systems will result in descriptions $\mathcal{C}'$ and $\mathcal{G}'$ such that $\mathrm{con}(\mathcal{C}') \neq \mathrm{gen}(\mathcal{G}')$, so that $(\mathcal{C}', \mathcal{G}')$ is not a DD pair. The same happens if strong minimization is applied to only one of the two descriptions.

Even though a given (constraint or generator) system is in smf, the corresponding dual description is not necessarily in smf. However, in these cases a DD pair with both components in smf can be computed quite easily.

Claim. Let $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P} \neq \varnothing$, where $(\mathcal{C}, \mathcal{G}) \equiv \mathcal{R}$ is a minimal DD pair. Then

1. If $\mathcal{C}$ is in smf and either $\mathcal{C}_\epsilon = \varnothing$ or $\mathrm{con}(\mathcal{C} \setminus \mathcal{C}_\epsilon) \not\Rrightarrow_\epsilon \mathcal{P}$, then $\mathcal{G}$ is in smf;

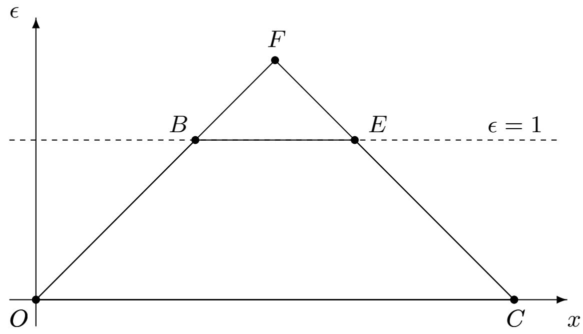

Fig. 4. The trapezium $OCEB$ is obtained by applying the strong minimization process to the constraint system describing $\mathcal{R}_3$. The triangle OCF is yet another representation for the NNC polyhedron $\mathcal{P}_3$, which can be characterized by a constraint system and a generator system that are both in smf.

2. If $\mathcal{G}$ is in smf and all the points encodings in $\mathcal{G}_P$ have the same value for the ϵ coordinate, then $\mathcal{C}$ is in smf.

As an example, we now compute smf's for the polyhedron $\mathcal{R}_3$ represented in Figure 2. Let us first consider the constraint system. The two strict inequality encodings $x - \epsilon \geq 0$ and $-x - \epsilon \geq -3$, which correspond to segments $[O, B]$ and $[C, D]$, are not ϵ-redundant, because they are saturated by the closure point encodings O and C, respectively. In contrast, the constraint $x - 3\epsilon \geq 4$, corresponding to segment $[B, D]$, is identified as ϵ-redundant (no closure point encoding saturates it) and will be removed, while adding the constraint $\epsilon \leq 1$. The resulting constraint system, which is in smf, defines the trapezium of vertices O, C, E, and B represented in Figure 4. Note that the generator system for this trapezium is not in smf: as suggested by item 1 of the above claim, this additional property can be enforced by removing the ϵ-upper-bound constraint $\epsilon \leq 1$, therefore obtaining the triangle of vertices O, C, and F.

Starting again from polyhedron $\mathcal{R}_3$, let us now consider the strong minimization of its generator system, which is made up of the four points O, C, D, and B. It is easy to observe that each one of the two unmatched point encodings is made ϵ-redundant by the other one (they both saturate the empty set of non-strict inequality encodings); as a consequence, one of them can be removed, obtaining either one of the triangles OCB and OCD represented in Figure 5, which are both in smf.

It is worth noting that, after removing an ϵ-redundant constraint, the addition of the ϵ-upper-bound constraint $\epsilon \leq 1$ is in general required to obtain another ϵ-representation. For instance, this happens when computing the smf of the constraint system describing the trapezium $CDGH$ of Figure 3: the simple removal of the constraint corresponding to segment $[G, H]$ would yield a stripe which is unbounded from above, so that it would not satisfy condition 1

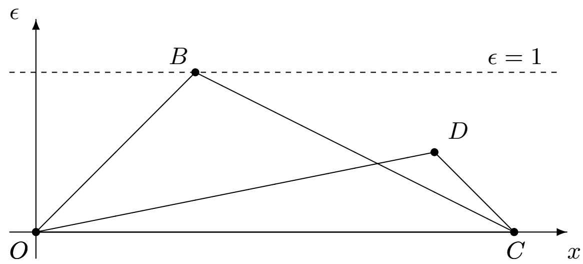

Fig. 5. Other two different ϵ-representations for $\mathcal{P}_3$, which can be obtained by applying the strong minimization process to the generator system describing $\mathcal{R}_3$.

of Definition 5; the addition of the ϵ-bounding constraint results in the rectangle $CDEF$ (i.e., the ϵ-representation $\mathcal{R}_4$ of $\mathcal{P}_4$).

5 The Parma Polyhedra Library

The Parma Polyhedra Library (PPL, `http://www.cs.unipr.it/ppl/`) is a collaborative project started in January 2001 at the Department of Mathematics of the University of Parma. It aims at becoming a truly professional library for the handling of approximations based on (not necessarily closed) convex polyhedra targeted at abstract interpretation and computer-aided verification. In this section we briefly review some of the key features of the library.[10] Before continuing, it is perhaps worth stressing that the theoretical treatment of previous sections applies to polyhedra in $\mathbb{R}^n$. Not surprisingly, the implementation only deals with rational polyhedra, that is, polyhedra that can be defined by constraints with rational coefficients or, dually, generators with rational coordinates. It is easy to prove that the property of being rational is preserved by all the operations of interest for the applications the library aims at.

The library is written in standard C++ and this ensures, among other things, maximum portability across different computing platforms. Using C++ for the development made it easier to adopt a number of programming techniques that are the key to the library's robustness, generality, efficiency and usability. However, for the sake of maximal code reuse and utility, care has been taken not to require the client application to be written in C++. The library includes a complete C interface and thus can be interfaced to all programming languages' implementations (there are many) that provide a C interface. There is also a Prolog interface supporting several Prolog systems and direct interfaces to other languages, such as Objective Caml and Mercury, are planned for the near future.

[10] The features described here are present either in release 0.3 or in the CVS version of the library available at `http://www.cs.unipr.it/ppl/`. All of them will be in the official release 0.4 due by May 2002.

One of the key features of the library is robustness. In particular, this means that failure is avoided whenever possible and, in all other cases, failure is recognized and handled properly. The PPL uses arbitrary precision integer arithmetic to implement coefficients and coordinates, and is thus immune from both rounding and overflow problems. In addition, all the data structures used in the implementation are fully dynamic and can expand automatically (in amortized constant time) to any dimension allowed by the available virtual memory. Two other aspects of failure avoidance are hiding and systematic checking of the interface invariants. In contrast to other libraries, the PPL hides the implementation details almost completely. For instance, the internal representation of constraints, generators and systems thereof need not concern the client application. Similarly, implementation devices such as the *positivity constraint* [29] and all the matters regarding the ϵ-representation encoding of NNC polyhedra are completely invisible from outside. The client application is provided with more natural interfaces, allowing the direct manipulation of higher level concepts, such as inequalities, lines and closure points. For instance, in the appropriate contexts, '`X < 5*Z`' and '`X + 2*Y + 5*Z >= 7`' is valid syntax expressing a strict and a non-strict inequality, both in the C++ and the Prolog interfaces. Even the C interface, which is at a considerably lower level of abstraction, refers to concepts like linear expression, constraint and constraint system and not to their possible implementations such as vectors and matrices. As usual, this is important for error prevention and to allow maximum latitude for the implementation.

Forbidding access to the internal structures manipulated by the library and systematic checking of the interface invariants could impede the overall system performance. However, the resulting overhead can be completely repaid (to the point of giving rise to a speedup) if the implementation exploits the freedom it has from the user interface. Even though, at the time of writing, the PPL only exploits a small part of this freedom, everything is in place to take full advantage of it. On a related aspect, the library has been designed to support the systematic application of incremental and lazy computation techniques, from which considerable efficiency improvements can be expected.[11]

In the design of the library, particular care has been taken concerning the possibility of interfacing the library with other constraint domains and with respect to scalability issues. It is clear that any user of any polyhedra library must face the possibility of excessive CPU time consumption and/or excessive memory usage [20]. As we have seen, one far from satisfactory but possible solution is to hope for the best but eventually "kill" those processes requiring more than the available resources [6]. In contrast, what we aim at is support for the dynamic and automatic composition of the trade-off between expressivity and efficiency. In this scenario, the static analysis or verification procedures should, by default, use the more descriptive (and costly) domains, such as the

[11] For the expert: the issues of incrementality and laziness include the use of *saturation matrices* (that were already present in Polylib [29]) and their efficient handling, delayed and incremental minimization of systems of constraints and generators, incremental sorting of matrices and so forth.

```
typedef Parma_Polyhedra_Library::NNC_Polyhedron PH;
void complex_function(const PH& ph1, const PH& ph2, PH& result) {
  try {
    start_timer(max_time_for_complex_function);
    complex_function_on_polyhedra(ph1, ph2, result ...);
    stop_timer();
  }
  catch (Exception& e) {
    // Virtual memory exhausted, or timeout expired, or any other error.
    ...
    BoundingBox bb1, bb2, bb_result;
    ph1.shrink_bounding_box(bb1);
    ph2.shrink_bounding_box(bb2);
    complex_function_on_bounding_boxes(bb1, bb2, bb_result ...);
    result = Polyhedron(bb_result);
  }
}
```

Fig. 6. Falling back to bounding boxes when the analysis with polyhedra *is* too costly.

convex polyhedra. When the computation of an operation on these descriptions requires too much time, or memory space, the system should detect this and perform a change of representation, for instance by approximating the input polyhedra with enclosing *bounding boxes* (rectangular regions with sides parallel to the axes); after the execution of the requested operation on this less precise (but much less computationally expensive) domain, the result is converted back into a convex polyhedron. Such a scenario is feasible only if the polyhedra library is implemented so that it can consistently react, without any loss of system resources, to events such as a timeout or an exception thrown by the memory allocation routines. As far as we know, the Parma Polyhedra Library is the first one to provide this facility. The idea is exemplified in Figure 6, where a robust C++ implementation of `complex_function` is sketched. The objective is to compute the polyhedron `result` starting from the polyhedra `ph1` and `ph2`. The operation is first tried on the polyhedra themselves and, in case of failure, the computation is done on the bounding box approximations of `ph1` and `ph2`. In the latter case the result of the bounding box computation, `bb_result` is converted into a convex polyhedron object and assigned to the output polyhedron `result`. The trade-off between efficiency and precision can also be controlled by means of the widening operations provided by the PPL, which are based on those originally proposed in [14] and improved in [13].

Concerning the efficiency of the PPL, at present no application can use both the PPL and another convex polyhedra library, so comparisons are not possible. However, the PPL has been integrated with the CHINA analyzer [1] for the purpose of detecting linear argument size relations [3]. The performance of the combined system has been compared, on the same task, with the performance of the cTI analyzer [25], which uses an implementation of convex polyhedra based on the SICStus CLP(Q) package. The combined system China+PPL outperformed that version of cTI in a significant way, exhibiting termination instead of thrashing and speedups of one or more orders of magnitude.

The Parma Polyhedra Library is free software released under the GPL: code and documentation can be downloaded and its development can be followed at `http://www.cs.unipr.it/ppl/`.

6 Conclusion

Convex polyhedra are the basis for several abstractions used in static analysis and computer-aided verification of complex and sometimes mission critical systems. For that purposes an implementation of convex polyhedra must be firmly based on a clear theoretical framework and written in accordance with sound software engineering principles. In this paper we have presented some of the most important ideas that are behind the Parma Polyhedra Library. In particular, we have provided a novel theoretical framework for the representation and manipulation of not necessarily closed convex polyhedra. We have also briefly sketched some important features of the design and implementation of the library. We believe that all this work constitutes a significant improvement in the state of the art and an encouragement to the wider adoption of abstract interpretation techniques in static analysis and computer-aided verification.

Acknowledgments. We wish to thank Costantino Medori for the many useful discussions we had on the subject of this paper. Thanks also to Fred Mesnard for letting us play with cTI's analyzer.

References

1. R. Bagnara. *Data-Flow Analysis for Constraint Logic-Based Languages.* PhD thesis, Dipartimento di Informatica, Università di Pisa, Pisa, Italy, 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 0.3 edition, February 2002. Available at `http://www.cs.unipr.it/ppl/`.
3. F. Benoy and A. King. Inferring argument size relationships with CLP(R). In J. P. Gallagher, editor, *Logic Programming 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.
4. B. Bérard and L. Fribourg:. Reachability analysis of (timed) Petri nets using real arithmetic. In J. C. M. Baeten and S. Mauw, editors, *CONCUR'99: Concurrency Theory, Proceedings of the 10th International Conference*, volume 1664 of *Lecture Notes in Computer Science*, pages 178–193, Eindhoven, The Netherlands, 1999. Springer-Verlag, Berlin.
5. 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.

6. N. S. Bjørner, A. Browne, M. Colón, B. Finkbeiner, Z. Manna, M. Pichora, H. B. Sipma, and T. E. Uribe. *STeP: The Stanford Temporal Prover (Educational Release) User's Manual*. Computer Science Department, Stanford University, Stanford, California, version 1.4-α edition, July 1998. Available at `http://www-step.stanford.edu/`.
7. N. V. Chernikova. Algorithm for finding a general formula for the non-negative solutions of system of linear equations. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 4(4):151–158, 1964.
8. N. V. Chernikova. Algorithm for finding a general formula for the non-negative solutions of system of linear inequalities. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 5(2):228–233, 1965.
9. N. V. Chernikova. Algorithm for discovering the set of all solutions of a linear programming problem. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 8(6):282–293, 1968.
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. P. Cousot, editor. *Static Analysis: 8th International Symposium, SAS 2001*, volume 2126 of *Lecture Notes in Computer Science*, Paris, France, 2001. Springer-Verlag, Berlin.
12. 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, 1977.
13. 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.
14. 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.
15. N. Dor, M. Rodeh, and S. Sagiv. Cleanness checking of string manipulations in C programs via integer analysis. In Cousot [11], pages 194–212.
16. 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.
17. N. Halbwachs, A. Kerbrat, and Y.-E. Proy. *POLyhedra INtegrated Environment*. Verimag, France, version 1.0 of POLINE edition, September 1995. Documentation taken from source code.
18. 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.
19. 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.

20. 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.
21. B. Jeannet. *Convex Polyhedra Library*, release 1.1.3c edition, March 2002. Documentation of the "New Polka" library available at `http://www.irisa.fr/prive/Bertrand.Jeannet/newpolka.html`.
22. H. Le Verge. A note on Chernikova's algorithm. *Publication interne* 635, IRISA, Campus de Beaulieu, Rennes, France, 1992.
23. V. Loechner. *PolyLib*: A library for manipulating parameterized polyhedra. Available at `http://icps.u-strasbg.fr/~loechner/polylib/`, March 1999. Declares itself to be a continuation of [29].
24. 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.
25. F. Mesnard and U. Neumerkel. Applying static analysis techniques for inferring termination conditions of logic programs. In Cousot [11], pages 93–110.
26. 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.
27. W. Pugh. A practical algorithm for exact array dependence analysis. *Communications of the ACM*, 35(8):102–114, 1992.
28. J. Stoer and C. Witzgall. *Convexity and Optimization in Finite Dimensions I.* Springer-Verlag, Berlin, 1970.
29. D. K. Wilde. A library for doing polyhedral operations. Master's thesis, Oregon State University, Corvallis, Oregon, December 1993. Also published as IRISA *Publication interne* 785, Rennes, France, 1993.

A Some Errors and Imprecisions in the Literature

Before deciding to write the Parma Polyhedra Library we have conducted an extensive study of the available software and publications dealing with the manipulation of convex polyhedra using the Double Description method [26]. During this work we have discovered several small errors and imprecisions that sometimes cover the treated matter with a thin layer of ambiguity and uncertainty. We list here the problems we spotted so as to explain why our definitions and results are sometimes different from those in the literature and to save others some frustration.

The most widespread imprecision concerns the parametric representation of convex polyhedra: vertices[12] and extreme rays[13] are often mentioned improperly.

[12] A *vertex* (or extreme point) of a polyhedron $\mathcal{P}$ is any point of $\mathcal{P}$ which cannot be expressed as the convex combination of other points of $\mathcal{P}$.

[13] An *extreme ray* of a polyhedron $\mathcal{P}$ is any ray of $\mathcal{P}$ which cannot be expressed as a positive combination of other rays of $\mathcal{P}$; note that two rays differing by a positive scalar factor (i.e., $\boldsymbol{r} = \mu \boldsymbol{r}'$, where $\mu > 0$) are considered equivalent.

For instance, consider the polyhedron corresponding to the full vector space $\mathbb{R}^2$: this has no vertex and no extreme ray and any generator system describing it will include at least one point and four rays (or, better, one point and two lines). In general, a polyhedron has a vertex if and only if it has no lines.

Such a problem is first found in [22, Section 6, pp. 10–11], where it is said that the parametric representation of a convex polyhedron is composed by vertices and extremal rays,[14] implicitly excluding other kinds of generators. The example above shows that, in general, the generator system of a polyhedron $\mathcal{P}$ may have to include points of $\mathcal{P}$ which are not vertices of $\mathcal{P}$.

In [29, Sections 2.4.1 and 2.6.1, pp. 10–11] the abuse of the word 'vertices' is explained in a footnote, but the problem with extreme rays remains. When Motzkin's theorem is used to decompose a polyhedron, it is said that its characteristic cone can be described by a combination of lines and extreme rays of the polyhedron, thus disregarding examples such as the one mentioned above.

Another imprecision concerns the concept of *polar* of a polyhedron. In [29, Definition 2.25, page 12], the polar $\mathcal{K}^*$ is defined for any closed convex set $\mathcal{K} \subseteq \mathbb{R}^n$ such that $\mathbf{0} \in \mathcal{K}$ as follows:[15]

$$\mathcal{K}^* \stackrel{\text{def}}{=} \bigl\{\, \boldsymbol{y} \in \mathbb{R}^n \bigm| \forall \boldsymbol{x} \in \mathcal{K} : \langle \boldsymbol{x}, \boldsymbol{y} \rangle \leq 0 \,\bigr\}.$$

Then it is said that $\mathcal{K}$ and $\mathcal{K}^*$ are *dual* of each other [29, Property 2.7, page 12], even though the concept of duality has only been defined for polytopes [29, Definition 2.24, page 12]. Note that, even when restricting ourselves to the case where $\mathcal{K}$ is a polytope containing the origin, $\mathcal{K}^*$ is not guaranteed to be a polytope. For an example, let $\mathcal{P}$ be the triangle in $\mathbb{R}^2$ having vertices $O = (0,0)^{\mathrm{T}}$, $A = (1,0)^{\mathrm{T}}$ and $B = (0,1)^{\mathrm{T}}$; then the cone $\mathcal{P}^*$ happens to be equal to the third orthant, so that the concept of dual polytopes is not applicable. Moreover, $\mathcal{P}^{**}$ is equal to the first orthant, so that $\mathcal{P} \neq \mathcal{P}^{**}$ (contrary to what stated in [29, property (iii), page 12]). As far as we can tell, the cause of all these problems is the systematic confusion between the concept of *polar cone* $\mathcal{K}^*$ (defined as above) as opposed to the concept of *polar set* $\mathcal{K}^\pi$, which is defined, for an arbitrary set $\mathcal{K} \subseteq \mathbb{R}^n$, as follows [28, (2.14.6), page 76]:

$$\mathcal{K}^\pi \stackrel{\text{def}}{=} \bigl\{\, \boldsymbol{y} \in \mathbb{R}^n \bigm| \forall \boldsymbol{x} \in \mathcal{K} : \langle \boldsymbol{x}, \boldsymbol{y} \rangle \leq 1 \,\bigr\}.$$

Namely, the polar of a polyhedron is the polar set of the polyhedron, and not its polar cone. Most of the stated results do hold when the two definitions are swapped; for example, if $\mathcal{P} \in \mathbb{CP}_n$ is a polyhedron, we have $\mathcal{P} = \mathcal{P}^{\pi\pi}$ if and only if $\mathbf{0} \in \mathcal{P}$. However, some other statements are still wrong, such as the one

[14] Note that the definition of *extremal ray* by Le Verge, provided for a cone in [22, Section 2, p. 4] and then extended to a polyhedron in [22, Section 6, p. 11], is significantly different from the definition of *extreme ray* used in [29], which is the one reported before.

[15] In fact, the textual definition states the condition $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \geq 0$. We believe that the relation sign '$\leq$' was intended; in any case, these two definitions return sets that are isomorphic via a systematic change of signs.

above regarding the duality of polars. As a matter of fact, requiring $\mathbf{0} \in \mathcal{P}$ is not enough to ensure that $\mathcal{P}^\pi$ is a polytope; as stated in [28, (2.14.11), page 77], this property holds if and only if $\mathcal{P}$ is a fully dimensional polyhedron and $\mathbf{0}$ is an *inner* point of $\mathcal{P}$. Because of these problems, all the observations in [29] concerning the polar of a polyhedron cannot really be trusted (besides confusing the reader to a large extent).

In [29, Definition 2.14, page 8], the convex hull of a set $\mathcal{K}$ is defined as the smallest convex set that contains $\mathcal{K}$. Then the *convex union* of two convex polyhedra $\mathcal{P}_1$ and $\mathcal{P}_2$ is defined as the convex hull of $\mathcal{P}_1 \cup \mathcal{P}_2$ and convex polyhedra are said to be closed with respect to the convex union operation [29, Section 3.4 and Table 3.3, page 17]. However, this is false. A simple counter-example is obtained in the vector space $\mathbb{R}^2$ by letting $\mathcal{P}_1 = \{ (x,y)^\mathrm{T} \in \mathbb{R}^2 \mid x = 0 \}$ and $\mathcal{P}_2 = \{ (1,0)^\mathrm{T} \}$. Then, as observed in [28], the convex hull of $\mathcal{P}_1 \cup \mathcal{P}_2$ is the set

$$\mathcal{K} = \{ (x,y)^\mathrm{T} \in \mathbb{R}^2 \mid 0 \le x < 1 \} \cup \mathcal{P}_2,$$

which cannot be written as the intersection of a finite number of half-spaces (and thus is not a convex polyhedron).

Some of these imprecisions seems to have percolated the subsequent literature. For instance, consider [5], where a new widening operator for the domain of polyhedra is defined. In [5, Definition 7, p. 63] it is stated that

> The set of extreme rays form a basis which describes all directions in which the convex polyhedron is open.

The statement of [5, Theorem 2, p. 63], which is meant to be a variant of the well known theorem by Minkowski, contains the usual error concerning vertices. As a consequence, even when accepting the (somehow standard) abuse of the word "convex hull", the definition of the *convex hull based widening* [5, Section 6.1, pp. 64-65] has to be taken with some care, since it is defined in terms of the cardinalities of the sets of lines, extreme rays and vertices describing the polyhedron.

B Proofs

As already observed in the paper, Theorem 1 is a simple consequence of well known theorems by Minkowski (stating the "only if" part) and Weyl (stating the "if" part). We here provide the proofs of the other formal results stated.

Lemma 1. *Let $\mathcal{P} = \mathrm{con}(\mathcal{C}) \in \mathbb{P}_n$ and $\boldsymbol{v} \in \mathbb{C}(\mathcal{P})$. Let also $\big(\langle \boldsymbol{a}, \boldsymbol{x} \rangle \bowtie b\big) \in \mathcal{C}$, where $\bowtie \in \{\ge, >\}$. Then $\langle \boldsymbol{a}, \boldsymbol{v} \rangle \ge b$.*

Proof. Let $\mathcal{H}$ be the set of affine half-spaces corresponding to the set of constraints $\mathcal{C}$. Since $\mathbb{C}(\cdot)$ is an upper closure operator,

$$\mathbb{C}(\mathcal{P}) = \mathbb{C}\big(\bigcap \mathcal{H}\big) \subseteq \mathbb{C}\Big(\bigcap \{ \mathbb{C}(H) \mid H \in \mathcal{H} \}\Big) = \bigcap \{ \mathbb{C}(H) \mid H \in \mathcal{H} \}.$$

As $\boldsymbol{v} \in \mathbb{C}(\mathcal{P})$, we have $\boldsymbol{v} \in \mathbb{C}(H)$, for all $H \in \mathcal{H}$. Let $H_c \in \mathcal{H}$ denote the affine half-space corresponding to $c = \bigl(\langle \boldsymbol{a}, \boldsymbol{x} \rangle \bowtie b\bigr)$. Hence $\boldsymbol{v} \in \mathbb{C}(H_c)$. If $\bowtie \in \{\geq\}$, then $\mathbb{C}(H_c) = \operatorname{con}\bigl(\{c\}\bigr)$. On the other hand, if $\bowtie \in \{>\}$, then $\mathbb{C}(H_c) = \operatorname{con}\bigl(\{c'\}\bigr)$, where $c' = \bigl(\langle \boldsymbol{a}, \boldsymbol{x} \rangle \geq b\bigr)$. Thus, as $\boldsymbol{v} \in \mathbb{C}(H_c)$, we obtain the thesis $\langle \boldsymbol{a}, \boldsymbol{v} \rangle \geq b$. □

Proof (Proof of Proposition 1). Let $\mathcal{P} = \operatorname{con}(\mathcal{C}) \in \mathbb{P}_n$, where $\mathcal{C} = \{c_1, \dots, c_m\}$.

To prove the "only if" branch, suppose that $\boldsymbol{c}$ is a closure point of $\mathcal{P}$, so that $\boldsymbol{c} \in \mathbb{C}(\mathcal{P})$. Then $\mathcal{P} \neq \varnothing$, because the topological closure of the empty set is still empty. Considering an arbitrary point $\boldsymbol{p} \in \mathcal{P}$ and a scalar λ such that $0 < \lambda < 1$, we have to prove that vector $\boldsymbol{v} \stackrel{\text{def}}{=} \lambda \boldsymbol{p} + (1-\lambda)\boldsymbol{c}$ is such that $\boldsymbol{v} \in \mathcal{P}$. To this end, we show that $\boldsymbol{v}$ satisfies all constraints $c_i = \bigl(\langle \boldsymbol{a}_i, \boldsymbol{x} \rangle \bowtie_i b_i\bigr) \in \mathcal{C}$. Since $\boldsymbol{p} \in \mathcal{P}$, it holds $\langle \boldsymbol{a}_i, \boldsymbol{p} \rangle \bowtie_i b_i$; moreover, by applying Lemma 1, we also have $\langle \boldsymbol{a}_i, \boldsymbol{c} \rangle \geq b_i$. Therefore, we obtain

$$
\begin{aligned}
\langle \boldsymbol{a}_i, \boldsymbol{v} \rangle &= \bigl\langle \boldsymbol{a}_i, \lambda \boldsymbol{p} + (1-\lambda)\boldsymbol{c} \bigr\rangle \\
&= \lambda \langle \boldsymbol{a}_i, \boldsymbol{p} \rangle + (1-\lambda) \langle \boldsymbol{a}_i, \boldsymbol{c} \rangle \\
&\bowtie_i \lambda b_i + (1-\lambda) \langle \boldsymbol{a}_i, \boldsymbol{c} \rangle \\
&\bowtie_i \lambda b_i + (1-\lambda) b_i \\
&= b_i.
\end{aligned}
$$

To prove the "if" branch, suppose now that $\mathcal{P} \neq \varnothing$ and $\lambda \boldsymbol{p} + (1-\lambda)\boldsymbol{c} \in \mathcal{P}$, for all $\boldsymbol{p} \in \mathcal{P}$ and $0 < \lambda < 1$. We have to show that $\boldsymbol{c} \in \mathbb{C}(\mathcal{P})$. To this end, for each $i \in \mathbb{N}$, let $\lambda_i = \frac{1}{i}$. For all $i > 1$, we have $0 < \lambda_i < 1$ and thus $\boldsymbol{v}_i \stackrel{\text{def}}{=} \lambda_i \boldsymbol{p} + (1-\lambda_i)\boldsymbol{c} \in \mathcal{P}$. It is easy to observe that, for any open ball centered in $\boldsymbol{c}$ having ray $\delta = \frac{1}{i}$, the intersection of $\mathcal{P}$ with the ball is not empty, because all the $\boldsymbol{v}_j \in \mathcal{P}$ such that $j > i$ belong to the ball. Therefore, $\boldsymbol{c} \in \mathbb{C}(\mathcal{P})$. □

A direct proof of Theorem 2 would require a generalization of Minkowski's and Weyl's theorems. In contrast, we will provide an indirect proof, which will be based on Proposition 2 as well as on some other lemmas stating properties holding for polyhedra that are ϵ-representations. The formal proof of Proposition 2 is facilitated by the introduction of a more rigorous definition for the function 'gen'.

Definition 8. *Let $\mathcal{G} = (R, P, C)$ be an extended generator system, where*

$$
R = \{\boldsymbol{r}_1, \dots, \boldsymbol{r}_k\}, \quad P = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_\ell\}, \quad C = \{\boldsymbol{c}_1, \dots, \boldsymbol{c}_m\}
$$

are finite subsets of $\mathbb{R}^n$ and $\mathbf{0} \notin R$. The NNC polyhedron $\operatorname{gen}(\mathcal{G})$ generated by $\mathcal{G}$ is defined as

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

For a standard generator system $\mathcal{G} = (R, P)$, we define $\operatorname{gen}(\mathcal{G}) \stackrel{\text{def}}{=} \operatorname{gen}\bigl((R, P, \varnothing)\bigr)$.

Proof (Proof of Proposition 2). We first show that $\mathcal{R} \stackrel{\text{def}}{=} \operatorname{con}\bigl(\text{con\_repr}(\mathcal{C})\bigr) \Rrightarrow_\epsilon \mathcal{P}$.

Condition (1) of Definition 5 follows directly from the definition of con_repr.

To prove condition (2) of Definition 5, suppose that $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}$. Then this satisfies any constraint $c = \bigl(\langle \boldsymbol{a}, \boldsymbol{x}\rangle + s \cdot \epsilon \geq b\bigr) \in \text{con\_repr}(\mathcal{C})$. By condition (1), $e \geq 0$ and, by definition of con_repr, $s \leq 0$. It therefore follows that $\langle \boldsymbol{a}, \boldsymbol{v}\rangle \geq b$ so that $(\boldsymbol{v}^{\mathrm{T}}, 0)^{\mathrm{T}}$ also satisfies c. As the choice of $c \in \text{con\_repr}(\mathcal{C})$ was arbitrary, $(\boldsymbol{v}^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}$.

To prove condition (3) of Definition 5, we have to show $\boldsymbol{v} \in \mathcal{P}$ if and only if there exists $e > 0$ such that $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}$.

Suppose first that there exists $e > 0$ such that $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}$. We will show that $\boldsymbol{v}$ satisfies all constraints $c \in \mathcal{C}$. If $c = \bigl(\langle \boldsymbol{a}, \boldsymbol{x}\rangle \bowtie b\bigr) \in \mathcal{C}$, then, by definition of con_repr, there exists a constraint $c' = \bigl(\langle \boldsymbol{a}, \boldsymbol{x}\rangle + s \cdot \epsilon \geq b\bigr) \in \text{con\_repr}(\mathcal{C})$, where $s \in \{0, -1\}$. Since $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}}$ satisfies c', we have $\langle \boldsymbol{a}, \boldsymbol{v}\rangle + s \cdot e \geq b$. If $s = -1$, then $s \cdot e < 0$, so that $\langle \boldsymbol{a}, \boldsymbol{v}\rangle > b$ and $\boldsymbol{v}$ satisfies the constraint c. Otherwise, if $s = 0$, we have $\bowtie \in \{\geq\}$ and $\langle \boldsymbol{a}, \boldsymbol{v}\rangle \geq b$, which again means that $\boldsymbol{v}$ satisfies the constraint c. As the choice of $c \in \mathcal{C}$ was arbitrary, $\boldsymbol{v} \in \mathcal{P}$.

Suppose next that $\boldsymbol{v} \in \mathcal{P}$ and consider an arbitrary constraint $c' = \bigl(\langle \boldsymbol{a}, \boldsymbol{x}\rangle + s \cdot \epsilon \geq b\bigr) \in \text{con\_repr}(\mathcal{C})$. Then, by definition of con_repr, $s \in \{0, -1\}$. We show that there exists $\delta_{c'} > 0$ such that, for all $e \leq \delta_{c'}$, $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}}$ satisfies c'. There are two cases depending on the value of s. If $s = 0$, then $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}}$ satisfies c' for all $e \in \mathbb{R}$. In this case, we let $\delta_{c'} = 1$. If $s = -1$, by definition of con_repr, there exists a constraint $c = \bigl(\langle \boldsymbol{a}, \boldsymbol{x}\rangle > b\bigr) \in \mathcal{C}$. Then, for all $e \leq \langle \boldsymbol{a}, \boldsymbol{v}\rangle - b$, the vector $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}}$ satisfies c'. In this case, let $\delta_{c'} = \langle \boldsymbol{a}, \boldsymbol{v}\rangle - b$. Therefore, for each constraint $c' \in \text{con\_repr}(\mathcal{C})$, there exists $\delta_{c'} > 0$ such that, for all $e \leq \delta_{c'}$, $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}}$ satisfies c'. Letting δ be the least element of the set $\{\, \delta_{c'} \mid c' \in \mathcal{C} \,\}$, we obtain $\delta > 0$ and $(\boldsymbol{v}^{\mathrm{T}}, \delta)^{\mathrm{T}}$ satisfies all constraints in $\text{con\_repr}(\mathcal{C})$, so that $(\boldsymbol{v}^{\mathrm{T}}, \delta)^{\mathrm{T}} \in \mathcal{R}$.

Secondly, we show that $\mathcal{R} \stackrel{\text{def}}{=} \operatorname{gen}\bigl(\text{gen\_repr}(\mathcal{G})\bigr) \Rrightarrow_\epsilon \mathcal{P}$.

To this end, let $\mathcal{G} = (R, P, C)$, where $R = \{\boldsymbol{r}_1, \dots, \boldsymbol{r}_k\}$, $P = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_\ell\}$ and $C = \{\boldsymbol{c}_1, \dots, \boldsymbol{c}_m\}$. Let also $\mathcal{G}' = (R', P')$ be defined as $\mathcal{G}' \stackrel{\text{def}}{=} \text{gen\_repr}(\mathcal{G})$, so that $R' = \{\boldsymbol{r}'_1, \dots, \boldsymbol{r}'_k\}$, with $\boldsymbol{r}'_i = (\boldsymbol{r}_i^{\mathrm{T}}, 0)^{\mathrm{T}}$, and $P' = \{\boldsymbol{p}'_1, \dots, \boldsymbol{p}'_h\}$, with $h = m + 2\ell$ and

$$\boldsymbol{p}'_i = \begin{cases} (\boldsymbol{c}_i^{\mathrm{T}}, 0)^{\mathrm{T}}, & \text{if } 1 \leq i \leq m; \\ (\boldsymbol{p}_i^{\mathrm{T}}, 1)^{\mathrm{T}}, & \text{if } m+1 \leq i \leq m+\ell; \\ (\boldsymbol{p}_i^{\mathrm{T}}, 0)^{\mathrm{T}}, & \text{if } m+\ell+1 \leq i \leq m+2\ell. \end{cases}$$

Then, by Definition 8, we have $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}$ if and only if there exist $\boldsymbol{\mu} \in \mathbb{R}_+^k$ and $\boldsymbol{\nu}' \in \mathbb{R}_+^h$ such that $\sum_{i=1}^h \nu'_i = 1$ and

$$(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} = \sum_{i=1}^{k} \mu_i \boldsymbol{r}'_i + \sum_{i=1}^{h} \nu'_i \boldsymbol{p}'_i. \tag{4}$$

To prove condition (1) of Definition 5, let $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}$. Clearly, $0 \leq e \leq 1$ holds because coefficient e is obtained in (4) as a convex combination of coefficients taken from the set $\{0, 1\}$.

To prove condition (2) of Definition 5, let $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}$, where $e > 0$. Now write $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}}$ according to (4) and let $\boldsymbol{\nu}'' \in \mathbb{R}^h_+$ be defined as

$$\nu''_i \stackrel{\text{def}}{=} \begin{cases} \nu'_i, & \text{if } 1 \le i \le m; \\ 0, & \text{if } m+1 \le i \le m+\ell; \\ \nu'_{i-\ell} + \nu'_i, & \text{if } m+\ell+1 \le i \le m+2\ell. \end{cases}$$

Since we still have $\sum_{i=1}^{h} \nu''_i = 1$, by Definition 8 the combination

$$\sum_{i=1}^{k} \mu_i \boldsymbol{r}'_i + \sum_{i=1}^{h} \nu''_i \boldsymbol{p}'_i = (\boldsymbol{v}^{\mathrm{T}}, 0)^{\mathrm{T}}$$

is another point of $\mathcal{R}$, as required.

To prove condition (3) of Definition 5, we have to show $\boldsymbol{v} \in \mathcal{P}$ if and only if there exists $e > 0$ such that $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}$.

Suppose first that there exists $e > 0$ such that $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}$. By expressing $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}}$ according to (4), we also obtain

$$\boldsymbol{v} = \sum_{i=1}^{k} \mu_i \boldsymbol{r}_i + \sum_{i=1}^{m} \nu'_i \boldsymbol{c}_i. + \sum_{i=m+1}^{m+\ell} \nu'_i \boldsymbol{p}_i + \sum_{i=m+\ell+1}^{m+2\ell} \nu'_i \boldsymbol{p}_i.$$

Let now $\boldsymbol{\eta} = (\nu'_1, \ldots, \nu'_m) \in \mathbb{R}^m_+$ and $\boldsymbol{\nu} \in \mathbb{R}^\ell_+$ be such that $\nu_i = \nu'_{m+i} + \nu'_{m+\ell+i}$, for all $i \in \{1, \ldots, \ell\}$. Then it holds $\sum_{i=1}^{\ell} \nu_i + \sum_{i=1}^{m} \eta_i = \sum_{i=1}^{h} \nu'_i = 1$ and $\boldsymbol{\nu} \neq \boldsymbol{0}$ so that, by Definition 8, we obtain that the combination

$$\sum_{i=1}^{k} \mu_i \boldsymbol{r}_i + \sum_{i=1}^{\ell} \nu_i \boldsymbol{p}_i + \sum_{i=1}^{m} \eta_i \boldsymbol{c}_i = \boldsymbol{v}$$

is a point of $\mathcal{P} = \mathrm{gen}\big((R, P, C)\big)$, as required.

Suppose next that $\boldsymbol{v} \in \mathcal{P}$. By Definition 8, there exist $\boldsymbol{\mu} \in \mathbb{R}^m_+$, $\boldsymbol{\nu} \in \mathbb{R}^\ell_+$ and $\boldsymbol{\eta} \in \mathbb{R}^m_+$ such that $\boldsymbol{\nu} \neq \boldsymbol{0}$, $\sum_{i=1}^{\ell} \nu_i + \sum_{i=1}^{m} \eta_i = 1$ and

$$\boldsymbol{v} = \sum_{i=1}^{k} \mu_i \boldsymbol{r}_i + \sum_{i=1}^{\ell} \nu_i \boldsymbol{p}_i + \sum_{i=1}^{m} \eta_i \boldsymbol{c}_i.$$

Let $h = m + 2\ell$ and $\boldsymbol{\nu}' \in \mathbb{R}^h_+$ be such that

$$\nu'_i = \begin{cases} \eta_i, & \text{if } 1 \le i \le m; \\ \nu_{i-m}, & \text{if } m+1 \le i \le \ell; \\ 0, & \text{if } m+\ell+1 \le i \le m+2\ell. \end{cases}$$

Note that $\boldsymbol{\nu}' \neq \boldsymbol{0}$ and $\sum_{i=1}^{h} \nu'_i = 1$. Thus, by Definition 8, the combination

$$\sum_{i=1}^{k} \mu_i \boldsymbol{r}'_i + \sum_{i=1}^{h} \nu'_i \boldsymbol{p}'_i = (\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}}$$

is a point of $\mathcal{R} = \mathrm{gen}\big((R', P')\big)$. Note that $\boldsymbol{\nu}' \neq \boldsymbol{0}$ implies $e > 0$, as required. □

We now provide lemmas showing that there is no loss of generality in the choice of ϵ coefficients made by function 'con_repr' (resp., the choice of ϵ coordinates made by function 'gen_repr') and that all rays of an ϵ-representation have a zero ϵ coordinate.

Lemma 2. *Let $\mathcal{P} \in \mathbb{P}_n$ and $\mathcal{C}_1$ be a constraint system in $\mathbb{R}^{n+1}$ containing only non-strict inequalities such that $\operatorname{con}(\mathcal{C}_1) \Rrightarrow_\epsilon \mathcal{P}$. Then, there exists a mixed constraint system $\mathcal{C}$ such that $\operatorname{con}\bigl(\operatorname{con\_repr}(\mathcal{C})\bigr) \Rrightarrow_\epsilon \mathcal{P}$.*

Proof. Let $\mathcal{R}_1 = \operatorname{con}(\mathcal{C}_1)$. Consider the constraint system $\mathcal{C} = \mathcal{C}' \cup \mathcal{C}''$, where

$$\mathcal{C}' \stackrel{\text{def}}{=} \Bigl\{\, \langle \boldsymbol{a}, \boldsymbol{x} \rangle > b \Bigm| \bigl(\langle \boldsymbol{a}, \boldsymbol{x} \rangle + s \cdot \epsilon \geq b\bigr) \in \mathcal{C}_1, s < 0, \boldsymbol{a} \neq \boldsymbol{0} \,\Bigr\},$$
$$\mathcal{C}'' \stackrel{\text{def}}{=} \Bigl\{\, \langle \boldsymbol{a}, \boldsymbol{x} \rangle \geq b \Bigm| \bigl(\langle \boldsymbol{a}, \boldsymbol{x} \rangle \geq b\bigr) \in \mathcal{C}_1, \boldsymbol{a} \neq \boldsymbol{0} \,\Bigr\}.$$

Let $\mathcal{C}_2 \stackrel{\text{def}}{=} \operatorname{con\_repr}(\mathcal{C})$ and $\mathcal{R}_2 = \operatorname{con}(\mathcal{C}_2)$. Then, by definition of 'con_repr', we obtain $\mathcal{C}_2 = \{0 \leq \epsilon \leq 1\} \cup \mathcal{C}_2' \cup \mathcal{C}''$, where

$$\mathcal{C}_2' = \Bigl\{\, \langle \boldsymbol{a}, \boldsymbol{x} \rangle - \epsilon \geq b \Bigm| \bigl(\langle \boldsymbol{a}, \boldsymbol{x} \rangle + s \cdot \epsilon \geq b\bigr) \in \mathcal{C}_1, s < 0, \boldsymbol{a} \neq \boldsymbol{0} \,\Bigr\}.$$

By Proposition 2, $\mathcal{R}_2$ satisfies conditions (1) and (2) of Definition 5. To complete the proof, it remains to show that $[\![\mathcal{R}_1]\!] = [\![\mathcal{R}_2]\!]$.

Let $\boldsymbol{v} \in [\![\mathcal{R}_1]\!]$, so that there exists $e > 0$ such that $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}_1$. By definition of 'con', $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}}$ satisfies all the constraints in $\mathcal{C}_1$. Thus $(\boldsymbol{v}^{\mathrm{T}}, e')^{\mathrm{T}}$ satisfies all the constraints in $\mathcal{C}''$ for any $e' \in \mathbb{R}$. Suppose $c_1 \stackrel{\text{def}}{=} \bigl(\langle \boldsymbol{a}, \boldsymbol{x} \rangle + s \cdot \epsilon \geq b\bigr) \in \mathcal{C}_1$, where $s < 0$ and $\boldsymbol{a} \neq \boldsymbol{0}$. Then $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}}$ satisfies c_1 and $-s \cdot e > 0$. Moreover, if $c' \stackrel{\text{def}}{=} \bigl(\langle \boldsymbol{a}, \boldsymbol{x} \rangle - \epsilon \geq b\bigr)$, then $c' \in \mathcal{C}_2'$. Thus, for any $0 < e' \leq -s \cdot e$, $(\boldsymbol{v}, e')$ satisfies c'. Thus, if e_0 is the greatest lower bound of the set

$$\{1\} \cup \Bigl\{\, -s \cdot e \Bigm| \bigl(\langle \boldsymbol{a}, \boldsymbol{x} \rangle + s \cdot \epsilon \geq b\bigr) \in \mathcal{C}_1, \boldsymbol{a} \neq \boldsymbol{0} \,\Bigr\},$$

we obtain $0 < e_0 \leq 1$ and $(\boldsymbol{v}^{\mathrm{T}}, e_0)^{\mathrm{T}}$ satisfies all the constraints in $\mathcal{C}_2$, so that $(\boldsymbol{v}^{\mathrm{T}}, e_0)^{\mathrm{T}} \in \mathcal{R}_2$ and hence, $\boldsymbol{v} \in [\![\mathcal{R}_2]\!]$, as required.

Let $\boldsymbol{v} \in [\![\mathcal{R}_2]\!]$, so that there exists $0 < e \leq 1$ such that $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}_2$. By definition of 'con', $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}}$ satisfies all the constraints in $\mathcal{C}_2$. Thus $(\boldsymbol{v}^{\mathrm{T}}, e')^{\mathrm{T}}$ satisfies all the constraints in $\mathcal{C}''$ for any $e' \in \mathbb{R}$. Let $c' \stackrel{\text{def}}{=} \bigl(\langle \boldsymbol{a}, \boldsymbol{x} \rangle - \epsilon \geq b\bigr) \in \mathcal{C}_2'$, where $\boldsymbol{a} \neq \boldsymbol{0}$. Then $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}}$ satisfies c'. Moreover, if $c_1 \stackrel{\text{def}}{=} \bigl(\langle \boldsymbol{a}, \boldsymbol{x} \rangle + s \cdot \epsilon \geq b\bigr)$, where $s < 0$, then $c_1 \in \mathcal{C}_1$ and $-\frac{e}{s} > 0$. Thus, for any $0 < e' \leq -\frac{e}{s}$, $(\boldsymbol{v}, e')$ satisfies c_1. Thus, if e_0 is the greatest lower bound of the set

$$\bigl\{\, \delta \bigm| (\epsilon \leq \delta) \in \mathcal{C}_1 \,\bigr\} \cup \Bigl\{\, -\tfrac{e}{s} \Bigm| \bigl(\langle \boldsymbol{a}, \boldsymbol{x} \rangle + s \cdot \epsilon \geq b\bigr) \in \mathcal{C}_1, \boldsymbol{a} \neq \boldsymbol{0} \,\Bigr\},$$

we obtain $e_0 > 0$ and $(\boldsymbol{v}^{\mathrm{T}}, e_0)^{\mathrm{T}}$ satisfies all the constraints in $\mathcal{C}_1$, so that $(\boldsymbol{v}^{\mathrm{T}}, e_0)^{\mathrm{T}} \in \mathcal{R}_1$ and hence, $\boldsymbol{v} \in [\![\mathcal{R}_1]\!]$, as required. ☐

Lemma 3. *Let* $\mathrm{gen}\big((R,P)\big) \Rrightarrow_\epsilon \mathcal{P}$*, where* $\mathcal{P} \neq \varnothing$*. Then, for all* $(\boldsymbol{r}^{\mathrm{T}}, e)^{\mathrm{T}} \in R$*,* $e = 0$*.*

Proof. By contraposition, suppose there exists $(\boldsymbol{r}^{\mathrm{T}}, e)^{\mathrm{T}} \in R$ such that $e \neq 0$. As $\mathcal{R}$ is an ϵ-representation, condition (1) holds so that, there exists $\delta > 0$ such that for any point $(\boldsymbol{x}^{\mathrm{T}}, \epsilon)^{\mathrm{T}} \in \mathcal{R}$, $0 \leq \epsilon \leq \delta$. As $\mathcal{R}$ is non-empty, there exists a point $(\boldsymbol{v}, e_0) \in \mathcal{R}$ for some $0 \leq e_0 \leq \delta$. Thus, for all $\mu \in \mathbb{R}_+$,

$$(\boldsymbol{v}_\mu^{\mathrm{T}}, e_\mu)^{\mathrm{T}} \stackrel{\mathrm{def}}{=} \mu(\boldsymbol{r}^{\mathrm{T}}, e)^{\mathrm{T}} + (\boldsymbol{v}^{\mathrm{T}}, e_0)^{\mathrm{T}} \in \mathcal{R}.$$

If $e < 0$, by taking $\mu > -\frac{e_0}{e} \in \mathbb{R}_+$, we obtain $e_\mu < 0$. Similarly, if $e > 0$, by taking $\mu \geq \frac{\delta}{e} \in \mathbb{R}_+$, we obtain $e_\mu \geq e_0 + \delta > \delta$. Thus, in both cases we contradict condition (1) of Definition 5. Therefore, it must hold $e = 0$. □

Lemma 4. *Let* $\mathcal{P} \in \mathbb{P}_n$ *and* $\mathcal{G}_1$ *be a (standard) generator system in* $\mathbb{R}^{n+1}$ *such that* $\mathrm{gen}(\mathcal{G}_1) \Rrightarrow_\epsilon \mathcal{P}$*. Then there exists an extended generator system* $\mathcal{G}$ *such that* $\mathrm{gen}\big(\mathrm{gen\_repr}(\mathcal{G})\big) \Rrightarrow_\epsilon \mathcal{P}$*.*

Proof. Let $\mathcal{G}_1 = (R_1, P_1)$ and $\mathcal{R}_1 = \mathrm{gen}(\mathcal{G}_1)$. Consider the extended generator system $\mathcal{G} = (R, P, C)$ such that

$$\begin{aligned} R &= \big\{\, \boldsymbol{r} \in \mathbb{R}^n \bigm| (\boldsymbol{r}^{\mathrm{T}}, 0)^{\mathrm{T}} \in R_1 \,\big\}, \\ P &= \big\{\, \boldsymbol{p} \in \mathbb{R}^n \bigm| (\boldsymbol{p}^{\mathrm{T}}, e)^{\mathrm{T}} \in P_1, e > 0 \,\big\}, \\ C &= \big\{\, \boldsymbol{c} \in \mathbb{R}^n \bigm| (\boldsymbol{c}^{\mathrm{T}}, e)^{\mathrm{T}} \in P_1, e = 0 \,\big\}. \end{aligned}$$

Let $\mathcal{G}_2 = \mathrm{gen\_repr}(\mathcal{G}) = (R_2, P_2)$ and $\mathcal{R}_2 = \mathrm{gen}(\mathcal{G}_2)$; by definition of 'gen_repr' and Lemma 3, we have $R_2 = R_1$ and

$$\begin{aligned} P_2 = {} & \big\{\, (\boldsymbol{p}^{\mathrm{T}}, 1)^{\mathrm{T}} \in \mathbb{R}^{n+1} \bigm| (\boldsymbol{p}^{\mathrm{T}}, e)^{\mathrm{T}} \in P_1, e > 0 \,\big\} \\ & \cup \big\{\, (\boldsymbol{p}^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathbb{R}^{n+1} \bigm| (\boldsymbol{p}^{\mathrm{T}}, e)^{\mathrm{T}} \in P_1 \,\big\}. \end{aligned}$$

Let k, ℓ and m be the cardinalities of R_1, P_1 and P_2, respectively.

By Proposition 2, $\mathcal{R}_2$ satisfies conditions (1) and (2) of Definition 5. To complete the proof, it remains to show that $[\![\mathcal{R}_1]\!] = [\![\mathcal{R}_2]\!]$.

Let $\boldsymbol{p} \in [\![\mathcal{R}_1]\!]$, so that there exists $e > 0$ such that $(\boldsymbol{p}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}_1$. By definition of 'gen', there exist $\boldsymbol{\mu} \in \mathbb{R}_+^k$ and $\boldsymbol{\nu} \in \mathbb{R}_+^\ell$, where $\sum_{i=1}^{\ell} \nu_i = 1$, such that

$$(\boldsymbol{p}^{\mathrm{T}}, e)^{\mathrm{T}} = \boldsymbol{\mu} R_1 + \boldsymbol{\nu} P_1.$$

Now, by definition of P_2, for each $(\boldsymbol{p}_i^{\mathrm{T}}, e_i)^{\mathrm{T}} \in P_1$ such that $e_i > 0$ (resp., $e_i = 0$) there exists $(\boldsymbol{p}_i^{\mathrm{T}}, e_i')^{\mathrm{T}} \in P_2$ such that $e_i' > 0$ (resp., $e_i' = 0$). Therefore, there exists also $\boldsymbol{\eta} \in \mathbb{R}_+^m$, where $\sum_{i=1}^{m} \eta_i = 1$, such that

$$(\boldsymbol{p}^{\mathrm{T}}, e')^{\mathrm{T}} = \boldsymbol{\mu} R_1 + \boldsymbol{\nu} P_2,$$

where $e' > 0$. Thus, $(\boldsymbol{p}^{\mathrm{T}}, e')^{\mathrm{T}} \in \mathcal{R}_2$ and $\boldsymbol{p} \in [\![\mathcal{R}_2]\!]$, as required.

The other inclusion is proved by a symmetric argument. □

Proof (Proof of Theorem 2). To prove the 'only if' branch, letting $\mathcal{P} \in \mathbb{P}_n$ we will prove that there exists an extended generator system $\mathcal{G} = (R, P, C)$ such that $\mathcal{P} = \mathrm{gen}(\mathcal{G})$. If $\mathcal{P} = \varnothing$, then we simply take $R = P = C = \varnothing$. Otherwise, let $\mathcal{P} \neq \varnothing$. By definition of NNC polyhedron, there exists a constraint system $\mathcal{C}$ such that $\mathcal{P} = \mathrm{con}(\mathcal{C})$. Let $\mathcal{R} \stackrel{\mathrm{def}}{=} \mathrm{con}\bigl(\mathrm{con\_repr}(\mathcal{C})\bigr) \in \mathbb{CP}_{n+1}$ so that, by Proposition 2, $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P}$. By Theorem 1, there exists a (standard) generator system $\mathcal{G}' = (R', P')$ such that $\mathcal{R} = \mathrm{gen}\bigl((R', P')\bigr)$. Then the thesis follows by Lemma 4.

To prove the 'if' branch, letting $\mathcal{G} = (R, P, C)$ be an extended generator system, we will show that $\mathcal{P} = \mathrm{gen}(\mathcal{G})$ is a NNC polyhedron. If $P = \varnothing$, then we obtain $\mathcal{P} = \varnothing$ and the empty set is a NNC polyhedron. Otherwise, let $P \neq \varnothing$, so that $\mathcal{P} \neq \varnothing$. Let $\mathcal{R} = \mathrm{gen}\bigl(\mathrm{gen\_repr}(\mathcal{G})\bigr)$ so that, by Proposition 2, $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P}$. By Theorem 1, there exist a constraint system $\mathcal{C}'$, containing non-strict linear inequalities only, such that $\mathcal{R} = \mathrm{con}(\mathcal{C}')$. Then the thesis follows by Lemma 2. □

Lemma 5. *If $\mathcal{P} \in \mathbb{P}_n$ is a non-empty NNC polyhedron and $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P}$, then*

$$\mathbb{C}(\mathcal{P}) = \bigl\{\, \boldsymbol{x} \in \mathbb{R}^n \bigm| (\boldsymbol{x}^\mathrm{T}, 0)^\mathrm{T} \in \mathcal{R} \,\bigr\}.$$

Proof. Letting $\mathcal{P}' \stackrel{\mathrm{def}}{=} \bigl\{\, \boldsymbol{x} \in \mathbb{R}^n \bigm| (\boldsymbol{x}^\mathrm{T}, 0)^\mathrm{T} \in \mathcal{R} \,\bigr\}$, we will prove $\mathcal{P}' = \mathbb{C}(\mathcal{P})$.

First, we show that $\mathcal{P}' \subseteq \mathbb{C}(\mathcal{P})$. Let $\boldsymbol{x} \in \mathcal{P}'$, so that $(\boldsymbol{x}^\mathrm{T}, 0)^\mathrm{T} \in \mathcal{R}$, and consider $\boldsymbol{p} \in \mathcal{P}$: since $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P}$, by applying property (3) of Definition 5, there exists $\epsilon > 0$ such that $(\boldsymbol{p}^\mathrm{T}, \epsilon)^\mathrm{T} \in \mathcal{R}$. Since $\mathcal{R}$ is a convex set, for all $\lambda \in \mathbb{R}$ such that $0 < \lambda < 1$ we have

$$\lambda(\boldsymbol{p}^\mathrm{T}, \epsilon)^\mathrm{T} + (1-\lambda)(\boldsymbol{x}^\mathrm{T}, 0)^\mathrm{T} = \bigl(\lambda\boldsymbol{p}^\mathrm{T} + (1-\lambda)\boldsymbol{x}^\mathrm{T}, \lambda\epsilon\bigr)^\mathrm{T} \in \mathcal{R}.$$

Note that, since $\lambda\epsilon > 0$, by applying again property (3) of Definition 5, we obtain $\lambda\boldsymbol{p} + (1-\lambda)\boldsymbol{x} \in \mathcal{P}$. Since $\mathcal{P} \neq \varnothing$ and the choices of $\boldsymbol{p} \in \mathcal{P}$ and λ where both arbitrary, we can apply Proposition 1 and conclude $\boldsymbol{x} \in \mathbb{C}(\mathcal{P})$.

Now we show that $\mathbb{C}(\mathcal{P}) \subseteq \mathcal{P}'$. Let $\boldsymbol{x} \in \mathbb{C}(\mathcal{P})$. For all $\boldsymbol{p} \in \mathcal{P}$ and $i \in \mathbb{N}$ such that $i > 1$, we have $\lambda_i \stackrel{\mathrm{def}}{=} \frac{1}{i}$, $0 < \lambda_i < 1$ and, by Proposition 1,

$$\boldsymbol{v}_i \stackrel{\mathrm{def}}{=} \lambda_i\boldsymbol{p} + (1-\lambda_i)\boldsymbol{x} \in \mathcal{P}.$$

Since $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P}$, by applying first property (3) and then property (2) of Definition 5, we obtain $(\boldsymbol{v}_i^\mathrm{T}, 0)^\mathrm{T} \in \mathcal{R}$. If $\boldsymbol{p} = \boldsymbol{x}$, then $\boldsymbol{v}_i = \boldsymbol{x}$, so that the thesis holds. Otherwise, let $\boldsymbol{p} \neq \boldsymbol{x}$. For any open ball of $\mathbb{R}^{n+1}$ centered in $(\boldsymbol{x}^\mathrm{T}, 0)^\mathrm{T}$ and having ray $\delta > 0$, there exists a $j \in \mathbb{N}$ such that $\lambda_j < \delta$; thus, $(\boldsymbol{v}_j^\mathrm{T}, 0)^\mathrm{T} \in \mathcal{R}$ belongs to the ball and, as the choice of δ is arbitrary, $(\boldsymbol{x}^\mathrm{T}, 0)^\mathrm{T} \in \mathbb{C}(\mathcal{R})$. However, $\mathcal{R} \in \mathbb{CP}_{n+1}$ is a topologically closed set, so that $\mathcal{R} = \mathbb{C}(\mathcal{R})$ and $(\boldsymbol{x}^\mathrm{T}, 0)^\mathrm{T} \in \mathcal{R}$. Hence, $\boldsymbol{x} \in \mathcal{P}'$, completing the proof. □

Lemma 6. *Let $\mathcal{P} = \mathrm{gen}\bigl((R, P, C)\bigr) \in \mathbb{P}_n$ be a non-empty NNC polyhedron and define $R' = \bigl\{\, (\boldsymbol{r}^\mathrm{T}, 0)^\mathrm{T} \bigm| \boldsymbol{r} \in R \,\bigr\} \subseteq \mathbb{R}^{n+1}$. If $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P}$, then there exists $Q \in \mathbb{R}^{n+1}$ such that $\mathcal{R} = \mathrm{gen}\bigl((R', Q)\bigr)$.*

Proof. In this proof, to simplify notation, if A is a set of vectors we let A also denote an arbitrary element of $\mathrm{matrix}(A)$. The context makes it clear when the symbol denotes a set or a matrix.

Since $\mathcal{R} \in \mathbb{CP}_{n+1}$, by Theorem 1 there exist finite sets $S', Q \subseteq \mathbb{R}^{n+1}$ such that $\mathcal{R} = \mathrm{gen}\big((S', Q)\big)$. By Lemma 3, if $(\boldsymbol{s}^{\mathrm{T}}, e)^{\mathrm{T}} \in S'$, then $e = 0$. Let k and ℓ denote the cardinalities of R' and S', respectively, and define

$$S \stackrel{\mathrm{def}}{=} \{\, \boldsymbol{s} \in \mathbb{R}^n \mid (\boldsymbol{s}^{\mathrm{T}}, 0)^{\mathrm{T}} \in S' \,\}.$$

Suppose $\boldsymbol{\mu} \in \mathbb{R}^k_+$. As $\mathcal{P} = \mathrm{gen}\big((R, P, C)\big)$, if $\boldsymbol{x} \in \mathcal{P}$ then $\boldsymbol{v} \stackrel{\mathrm{def}}{=} \boldsymbol{x} + \boldsymbol{\mu} R \in \mathcal{P}$. Thus, by conditions (2) and (3) of Definition 5, $(\boldsymbol{v}^{\mathrm{T}}, 0)^{\mathrm{T}} = (\boldsymbol{x}^{\mathrm{T}}, 0)^{\mathrm{T}} + \boldsymbol{\mu} R' \in \mathcal{R}$. Since this holds for any $\boldsymbol{\mu} \in \mathbb{R}^k_+$, it follows that R' is a set of rays of $\mathcal{R}$, so that $\mathcal{R} = \mathrm{gen}\big((S' \cup R', Q)\big)$.

Suppose $\boldsymbol{\nu} \in \mathbb{R}^\ell_+$. As $\mathcal{R} = \mathrm{gen}\big((S', Q)\big)$, if $(\boldsymbol{x}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}$ and $e > 0$, then $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \stackrel{\mathrm{def}}{=} (\boldsymbol{x}^{\mathrm{T}}, e)^{\mathrm{T}} + \boldsymbol{\nu} S' \in \mathcal{R}$. Then, by condition (3) of Definition 5, $\boldsymbol{v} = \boldsymbol{x} + \boldsymbol{\nu} S \in \mathcal{P}$. Since this holds for any $\boldsymbol{\nu} \in \mathbb{R}^\ell_+$, it follows that S is a set of rays of $\mathcal{P}$. However, since $\mathcal{P} = \mathrm{gen}\big((R, P, C)\big)$, if $\boldsymbol{s} \in S$ then $\boldsymbol{s} = \boldsymbol{\mu} R$, for some $\boldsymbol{\mu} \in \mathbb{R}^k_+$. As a consequence, $(\boldsymbol{s}^{\mathrm{T}}, 0)^{\mathrm{T}} = \boldsymbol{\mu} R'$. Since this holds for all the vectors in S', then S' is a set of redundant rays, so that $\mathcal{R} = \mathrm{gen}\big((R', Q)\big)$. □

Lemma 7. *For $j \in \{1, 2\}$, let $\mathcal{P}_j = \mathrm{gen}\big((R_j, P_j, C_j)\big)$ be a non-empty NNC polyhedron; let also k_j be the cardinality of R_j and $K_j \in \mathrm{matrix}(R_j)$. Then $\boldsymbol{x} \in \mathcal{P}_1 \uplus \mathcal{P}_2$ if and only if there exist $0 \leq \lambda \leq 1$, $\boldsymbol{x}_1 \in \mathbb{C}(\mathcal{P}_1)$, $\boldsymbol{x}_2 \in \mathbb{C}(\mathcal{P}_2)$, $\boldsymbol{\mu}_1 \in \mathbb{R}^{k_1}_+$ and $\boldsymbol{\mu}_2 \in \mathbb{R}^{k_2}_+$ such that*

$$\boldsymbol{x} = \lambda \boldsymbol{x}_1 + (1 - \lambda) \boldsymbol{x}_2 + \boldsymbol{\mu}_1 K_1 + \boldsymbol{\mu}_2 K_2,$$

where $(\boldsymbol{x}_1 \in \mathcal{P}_1 \wedge \lambda > 0) \vee (\boldsymbol{x}_2 \in \mathcal{P}_2 \wedge \lambda < 1)$.

Proof. For $j \in \{1, 2\}$, let ℓ_j and m_j be the cardinalities of P_j and C_j, respectively. By definition of the poly-hull operation, we have $\mathcal{P}_1 \uplus \mathcal{P}_2 = \mathrm{gen}\big((R, P, C)\big)$, where $R = R_1 \cup R_2$, $P = P_1 \cup P_2$ and $C = C_1 \cup C_2$, having cardinalities k, ℓ and m, respectively.

Suppose first that $\boldsymbol{x} \in \mathcal{P}_1 \uplus \mathcal{P}_2$. Then, by Definition 8,

$$\boldsymbol{x} = \boldsymbol{\mu} K + \boldsymbol{\nu} L + \boldsymbol{\eta} M$$

where $K \in \mathrm{matrix}(R)$, $L \in \mathrm{matrix}(P)$, $M \in \mathrm{matrix}(C)$, $\boldsymbol{\mu} \in \mathbb{R}^k_+$, $\boldsymbol{\nu} \in \mathbb{R}^\ell_+$, $\boldsymbol{\eta} \in \mathbb{R}^m_+$, $\sum \boldsymbol{\nu} + \sum \boldsymbol{\eta} = 1$ and $\boldsymbol{\nu} \neq \mathbf{0}$. Therefore, we can also rewrite it as

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{\mu}_1 K_1 + \boldsymbol{\mu}_2 K_2 + \boldsymbol{\nu}_1 L_1 + \boldsymbol{\nu}_2 L_2 + \boldsymbol{\eta}_1 M_1 + \boldsymbol{\eta}_2 M_2 \\ &= (\boldsymbol{\nu}_1 L_1 + \boldsymbol{\eta}_1 M_1) + (\boldsymbol{\nu}_2 L_2 + \boldsymbol{\eta}_2 M_2) + \boldsymbol{\mu}_1 K_1 + \boldsymbol{\mu}_2 K_2, \end{aligned}$$

where $L_j \in \mathrm{matrix}(P_j)$, $M_j \in \mathrm{matrix}(C_j)$, $\boldsymbol{\mu}_j \in \mathbb{R}^{k_j}_+$, $\boldsymbol{\nu}_j \in \mathbb{R}^{\ell_j}_+$ and $\boldsymbol{\eta}_j \in \mathbb{R}^{m_j}_+$, for $j \in \{1, 2\}$, $\sum \boldsymbol{\nu}_1 + \sum \boldsymbol{\nu}_2 + \sum \boldsymbol{\eta}_1 + \sum \boldsymbol{\eta}_2 = 1$ and $\sum \boldsymbol{\nu}_1 + \sum \boldsymbol{\nu}_2 > 0$. It follows that either $\boldsymbol{\nu}_1 \neq \mathbf{0}$ or $\boldsymbol{\nu}_2 \neq \mathbf{0}$.

If $\sum \boldsymbol{\nu}_1 + \sum \boldsymbol{\eta}_1 = 1$ (so that $\boldsymbol{\nu}_2 = \boldsymbol{\eta}_2 = \mathbf{0}$ and $\boldsymbol{\nu}_1 \neq \mathbf{0}$), then, by Definition 8, we obtain $\boldsymbol{x}_1 \stackrel{\text{def}}{=} \boldsymbol{\nu}_1 L_1 + \boldsymbol{\eta}_1 M_1 \in \mathcal{P}_1$. Taking $\lambda = 1$, we have $1 - \lambda = 0$, so that we can take an arbitrary $\boldsymbol{x}_2 \in \mathbb{C}(\mathcal{P}_2)$ (there must exist one, since $\mathcal{P}_2 \neq \varnothing$). Similarly, if $\sum \boldsymbol{\nu}_2 + \sum \boldsymbol{\eta}_2 = 1$ (so that $\boldsymbol{\nu}_1 = \boldsymbol{\eta}_1 = \mathbf{0}$ and $\boldsymbol{\nu}_2 \neq \mathbf{0}$), we obtain $\boldsymbol{x}_2 \stackrel{\text{def}}{=} \boldsymbol{\nu}_2 L_2 + \boldsymbol{\eta}_2 M_2 \in \mathcal{P}_2$, so that we can take $\lambda = 0$ and an arbitrary $\boldsymbol{x}_1 \in \mathbb{C}(\mathcal{P}_1)$. Otherwise, let both $\sum \boldsymbol{\nu}_1 + \sum \boldsymbol{\eta}_1 \neq 0$ and $\sum \boldsymbol{\nu}_2 + \sum \boldsymbol{\eta}_2 \neq 0$. Then, by taking $\lambda = \sum \boldsymbol{\nu}_1 + \sum \boldsymbol{\eta}_1$ we have $\lambda > 0$ and $1 - \lambda = \sum \boldsymbol{\nu}_2 + \sum \boldsymbol{\eta}_2 > 0$. Therefore we can define $\boldsymbol{x}_1 \stackrel{\text{def}}{=} \frac{1}{\lambda}(\boldsymbol{\nu}_1 L_1 + \boldsymbol{\eta}_1 M_1)$ and $\boldsymbol{x}_2 \stackrel{\text{def}}{=} \frac{1}{1-\lambda}(\boldsymbol{\nu}_2 L_2 + \boldsymbol{\eta}_2 M_2)$. Thus $\boldsymbol{x}_1 \in \mathbb{C}(\mathcal{P}_1)$ and $\boldsymbol{x}_2 \in \mathbb{C}(\mathcal{P}_2)$. Moreover, by Definition 8, as $\boldsymbol{\nu}_1 \neq \mathbf{0}$ or $\boldsymbol{\nu}_2 \neq \mathbf{0}$, either $\boldsymbol{x}_1 \in \mathcal{P}_1$ or $\boldsymbol{x}_2 \in \mathcal{P}_2$. In all the three cases above, we obtain

$$\boldsymbol{x} = \lambda \boldsymbol{x}_1 + (1 - \lambda)\boldsymbol{x}_2 + \boldsymbol{\mu}_1 K_1 + \boldsymbol{\mu}_2 K_2,$$

with $\boldsymbol{x}_j \in \mathbb{C}(\mathcal{P}_j)$ for $j = \{1, 2\}$ where either $\boldsymbol{x}_1 \in \mathcal{P}_1$ and $\lambda > 0$ or $\boldsymbol{x}_2 \in \mathcal{P}_2$ and $\lambda < 1$, as required.

To prove the other direction, suppose that $\boldsymbol{x}_j \in \mathbb{C}(\mathcal{P}_j)$ and $\boldsymbol{\mu}_j \in \mathbb{R}_+^{k_j}$, for $j = \{1, 2\}$, and there exist $0 \leq \lambda \leq 1$ such that

$$\boldsymbol{x} = \lambda \boldsymbol{x}_1 + (1 - \lambda)\boldsymbol{x}_2 + \boldsymbol{\mu}_1 K_1 + \boldsymbol{\mu}_2 K_2,$$

where either $\boldsymbol{x}_1 \in \mathcal{P}_1$ and $\lambda > 0$ or $\boldsymbol{x}_2 \in \mathcal{P}_2$ and $\lambda < 1$.

For $j \in \{1, 2\}$, since $\boldsymbol{x}_j \in \mathbb{C}(\mathcal{P}_j)$, letting $L_j \in \text{matrix}(P_j)$ and $M_j \in \text{matrix}(C_j)$, there exist $\boldsymbol{\mu}_j \in \mathbb{R}_+^{k_j}$, $\boldsymbol{\nu}_j \in \mathbb{R}_+^{\ell_j}$ and $\boldsymbol{\eta}_j \in \mathbb{R}_+^{m_j}$ such that

$$\boldsymbol{x}_j = \boldsymbol{\mu}_j K_j + \boldsymbol{\nu}_j L_j + \boldsymbol{\eta}_j M_j,$$

where $\sum \boldsymbol{\nu}_j + \sum \boldsymbol{\eta}_j = 1$ (note that it may hold $\boldsymbol{\nu}_j = \mathbf{0}$). Thus,

$$\begin{aligned}\boldsymbol{x} &= \boldsymbol{\mu}_1 K_1 + \boldsymbol{\mu}_2 K_2 + \lambda(\boldsymbol{\nu}_1 L_1 + \boldsymbol{\eta}_1 M_1) + (1 - \lambda)(\boldsymbol{\nu}_2 L_2 + \boldsymbol{\eta}_2 M_2) \\ &= \boldsymbol{\mu}_1 K_1 + \boldsymbol{\mu}_2 K_2 + \lambda \boldsymbol{\nu}_1 L_1 + (1 - \lambda)\boldsymbol{\nu}_2 L_2 + \lambda \boldsymbol{\eta}_1 M_1 + (1 - \lambda)\boldsymbol{\eta}_2 M_2\end{aligned}$$

Note that

$$\lambda \textstyle\sum \boldsymbol{\nu}_1 + (1 - \lambda)\sum \boldsymbol{\nu}_2 + \lambda \sum \boldsymbol{\eta}_1 + (1 - \lambda)\sum \boldsymbol{\eta}_2 = 1.$$

Moreover, as either $\boldsymbol{x}_1 \in \mathcal{P}_1$ and $\lambda > 0$ or $\boldsymbol{x}_2 \in \mathcal{P}_2$ and $\lambda < 1$, we obtain $\lambda \boldsymbol{\nu}_1 \neq \mathbf{0}$ or $(1 - \lambda)\boldsymbol{\nu}_2 \neq \mathbf{0}$. Thus, there exist $K \in \text{matrix}(R_1 \cup R_2)$, $L \in \text{matrix}(P_1 \cup P_2)$, $M \in \text{matrix}(C_1 \cup C_2)$, $\boldsymbol{\mu} \in \mathbb{R}_+^k$, $\boldsymbol{\nu} \in \mathbb{R}_+^\ell$ and $\boldsymbol{\eta} \in \mathbb{R}_+^m$ such that $\sum \boldsymbol{\nu} + \sum \boldsymbol{\eta} = 1$, $\boldsymbol{\nu} \neq \mathbf{0}$, and $\boldsymbol{x} = \boldsymbol{\mu} K + \boldsymbol{\nu} L + \boldsymbol{\eta} M$. Then, by Definition 8, $\boldsymbol{x} \in \mathcal{P}_1 \uplus \mathcal{P}_2$, completing the proof. □

Proof (Proof of Proposition 3).

Item (1) follows directly from condition (3) of Definition 5.

To prove item (2), we show that if $\mathcal{R}_1 \Rrightarrow_\epsilon \mathcal{P}_1$ and $\mathcal{R}_2 \Rrightarrow_\epsilon \mathcal{P}_2$, then $\mathcal{R}_1 \cap \mathcal{R}_2 \Rrightarrow_\epsilon \mathcal{P}_1 \cap \mathcal{P}_2$. We have to show that conditions (1), (2) and (3) of Definition 5 hold for $\mathcal{R}_1 \cap \mathcal{R}_2$ and $\mathcal{P}_1 \cap \mathcal{P}_2$.

We first prove condition (1). As $\mathcal{R}_1$ and $\mathcal{R}_2$ are ϵ-representations there exists $\delta_1, \delta_2 > 0$ such that $\mathcal{R}_1 \subseteq \mathrm{con}\bigl(\{0 \leq \epsilon \leq \delta_1\}\bigr)$ and $\mathcal{R}_2 \subseteq \mathrm{con}\bigl(\{0 \leq \epsilon \leq \delta_2\}\bigr)$. Letting δ be the minimum of $\{\delta_1, \delta_2\}$ we have $\mathcal{R}_1 \cap \mathcal{R}_2 \subseteq \mathrm{con}\bigl(\{0 \leq \epsilon \leq \delta\}\bigr)$.

To prove condition (2), let $(\boldsymbol{p}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}_1 \cap \mathcal{R}_2$. Then, as $\mathcal{R}_1$ and $\mathcal{R}_2$ are ϵ-representations, $(\boldsymbol{p}^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}_1$ and $(\boldsymbol{p}^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}_2$. Hence $(\boldsymbol{p}^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}_1 \cap \mathcal{R}_2$.

To prove condition (3), we have to show $\boldsymbol{p} \in \mathcal{P}_1 \cap \mathcal{P}_2$ if and only if there exists $e > 0$ such that $(\boldsymbol{p}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}_1 \cap \mathcal{R}_2$. First, let $\boldsymbol{p} \in \mathcal{P}_1 \cap \mathcal{P}_2$. Then, by condition (3), there exists $e_1, e_2 > 0$ such that $(\boldsymbol{p}^{\mathrm{T}}, e_1)^{\mathrm{T}} \in \mathcal{R}_1$ and $(\boldsymbol{p}^{\mathrm{T}}, e_2)^{\mathrm{T}} \in \mathcal{R}_2$. Suppose, without loss of generality, that $e_1 \leq e_2$. By condition (2), $(\boldsymbol{p}^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}_2$. Thus, since $\mathcal{R}_2$ is a convex set, $(\boldsymbol{p}^{\mathrm{T}}, e_1)^{\mathrm{T}} \in \mathcal{R}_2$. Hence, $(\boldsymbol{p}^{\mathrm{T}}, e_1)^{\mathrm{T}} \in \mathcal{R}_1 \cap \mathcal{R}_2$. Secondly, suppose that there exists $e > 0$ such that $(\boldsymbol{p}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}_1 \cap \mathcal{R}_2$. Then $(\boldsymbol{p}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}_1$ and $(\boldsymbol{p}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}_2$. By condition (3), $\boldsymbol{p} \in \mathcal{P}_1$ and $\boldsymbol{p} \in \mathcal{P}_2$, so that $\boldsymbol{p} \in \mathcal{P}_1 \cap \mathcal{P}_2$.

To prove item (3), we show that if $\mathcal{P}_1$ and $\mathcal{P}_2$ are not empty, $\mathcal{R}_1 \Rrightarrow_\epsilon \mathcal{P}_2$ and $\mathcal{R}_2 \Rrightarrow_\epsilon \mathcal{P}_2$, then $\mathcal{R}_1 \uplus \mathcal{R}_2 \Rrightarrow_\epsilon \mathcal{P}_1 \uplus \mathcal{P}_2$. We have to show that conditions (1), (2) and (3) of Definition 5 hold for $\mathcal{R}_1 \uplus \mathcal{R}_2$. For $j \in \{1, 2\}$, let $\mathcal{P}_j = \mathrm{gen}\bigl((R_j, P_j, C_j)\bigr)$, where R_j has cardinality k_j. By Lemma 6, for $j \in \{1, 2\}$, there exist finite sets $R'_j, Q_j \subseteq \mathbb{R}^{n+1}_+$ such that $\mathcal{R}_j = \mathrm{gen}\bigl((R'_j, Q_j)\bigr)$, where $R'_j = \bigl\{ (\boldsymbol{r}^{\mathrm{T}}, 0)^{\mathrm{T}} \bigm| \boldsymbol{r} \in R_j \bigr\}$. By Lemma 7, if $(\boldsymbol{p}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}_1 \uplus \mathcal{R}_2$, then for some $0 \leq \lambda \leq 1$, $(\boldsymbol{p_1}^{\mathrm{T}}, e_1)^{\mathrm{T}} \in \mathcal{R}_1$, $(\boldsymbol{p_2}^{\mathrm{T}}, e_2)^{\mathrm{T}} \in \mathcal{R}_2$, $\boldsymbol{r} = \boldsymbol{\mu}_1 K_1 + \boldsymbol{\mu}_2 K_2$ where $K_1 \in \mathrm{matrix}(R_1)$ and $K_2 \in \mathrm{matrix}(R_2)$, $\boldsymbol{\mu}_1 \in \mathbb{R}^{k_1}_+$ and $\boldsymbol{\mu}_2 \in \mathbb{R}^{k_2}_+$, we have

$$\begin{pmatrix}\boldsymbol{p}\\ e\end{pmatrix} = \lambda \begin{pmatrix}\boldsymbol{p}_1\\ e_1\end{pmatrix} + (1-\lambda)\begin{pmatrix}\boldsymbol{p}_2\\ e_2\end{pmatrix} + \begin{pmatrix}\boldsymbol{r}\\ 0\end{pmatrix} = \begin{pmatrix}\lambda\boldsymbol{p}_1 + (1-\lambda)\boldsymbol{p}_2\\ \lambda e_1 + (1-\lambda)e_2\end{pmatrix} + \begin{pmatrix}\boldsymbol{r}\\ 0\end{pmatrix}. \qquad (5)$$

We first prove condition (1). Suppose that $(\boldsymbol{p}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}_1 \uplus \mathcal{R}_2$, so that we can rewrite it according to (5). As $\mathcal{R}_1$ and $\mathcal{R}_2$ are both ϵ-representations, there exist $\delta_1, \delta_2 > 0$ such that $\mathcal{R}_1 \subseteq \mathrm{con}\bigl(\{0 \leq \epsilon \leq \delta_1\}\bigr)$ and $\mathcal{R}_2 \subseteq \mathrm{con}\bigl(\{0 \leq \epsilon \leq \delta_2\}\bigr)$. Letting δ be the maximum of $\{\delta_1, \delta_2\}$, we obtain $0 \leq e_1 \leq \delta$ and $0 \leq e_2 \leq \delta$, so that $e = \lambda e_1 + (1-\lambda)e_2$ satisfies $0 \leq e \leq \delta$. Thus, $(\boldsymbol{p}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathrm{con}\bigl(\{0 \leq \epsilon \leq \delta\}\bigr)$, as required.

To prove condition (2), suppose that $(\boldsymbol{p}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}_1 \uplus \mathcal{R}_2$, so that we can rewrite it according to (5). As $\mathcal{R}_1$ and $\mathcal{R}_2$ are ϵ-representations, $(\boldsymbol{p}_1^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}_1$ and $(\boldsymbol{p}_2^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}_2$ so that

$$\begin{pmatrix}\boldsymbol{p}\\ 0\end{pmatrix} = \lambda \begin{pmatrix}\boldsymbol{p}_1\\ 0\end{pmatrix} + (1-\lambda)\begin{pmatrix}\boldsymbol{p}_2\\ 0\end{pmatrix} + \begin{pmatrix}\boldsymbol{r}\\ 0\end{pmatrix} \in \mathcal{R}_1 \uplus \mathcal{R}_2.$$

To prove condition (3), we have to show $\boldsymbol{p} \in \mathcal{P}_1 \uplus \mathcal{P}_2$ if and only if there exists $e > 0$ such that $(\boldsymbol{p}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}_1 \uplus \mathcal{R}_2$. First suppose that $\boldsymbol{p} \in \mathcal{P}_1 \uplus \mathcal{P}_2$. Then, by Lemma 7, $\boldsymbol{p} = \lambda\boldsymbol{p}_1 + (1-\lambda)\boldsymbol{p}_2 + \boldsymbol{r}$, for some $0 \leq \lambda \leq 1$, $\boldsymbol{p}_1 \in \mathbb{C}(\mathcal{P}_1)$ and $\boldsymbol{p}_2 \in \mathbb{C}(\mathcal{P}_2)$, where $\boldsymbol{p}_1 \in \mathcal{P}_1$ and $\lambda > 0$ or $\boldsymbol{p}_2 \in \mathcal{P}_2$ and $\lambda < 1$, and $\boldsymbol{r} = \boldsymbol{\mu}_1 K_1 + \boldsymbol{\mu}_2 K_2$ where $K_1 \in \mathrm{matrix}(R_1)$ and $K_2 \in \mathrm{matrix}(R_2)$, $\boldsymbol{\mu}_1 \in \mathbb{R}^{k_1}_+$ and $\boldsymbol{\mu}_2 \in \mathbb{R}^{k_2}_+$. Suppose, without loss of generality, that $\boldsymbol{p}_1 \in \mathcal{P}_1$ and $\lambda > 0$. As $\mathcal{R}_1 \Rrightarrow_\epsilon \mathcal{P}_1$, by condition (3), there exists $e_1 > 0$ such that $(\boldsymbol{p}_1^{\mathrm{T}}, e_1)^{\mathrm{T}} \in \mathcal{R}_1$. As $\mathcal{R}_2$ is an ϵ-representation, by Lemma 5, from $\boldsymbol{p}_2 \in \mathbb{C}(\mathcal{R}_2)$ we obtain $(\boldsymbol{p}_2^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}_2$.

Thus, by letting

$$\begin{pmatrix} \boldsymbol{p} \\ e_1 \end{pmatrix} \stackrel{\text{def}}{=} \lambda \begin{pmatrix} \boldsymbol{p}_1 \\ e_1 \end{pmatrix} + (1-\lambda) \begin{pmatrix} \boldsymbol{p}_2 \\ 0 \end{pmatrix} + \begin{pmatrix} \boldsymbol{r} \\ 0 \end{pmatrix},$$

we obtain $(\boldsymbol{p}^{\mathrm{T}}, e_1)^{\mathrm{T}} \in \mathcal{R}_1 \uplus \mathcal{R}_2$, where $e_1 > 0$ as required. Secondly, suppose that there exists $e > 0$ such that $(\boldsymbol{p}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}_1 \uplus \mathcal{R}_2$, so that we can rewrite it according to (5). As $\mathcal{R}_1$ and $\mathcal{R}_2$ are ϵ-representations, by condition (1), $e_1, e_2 \geq 0$. As $e > 0$ and $\lambda \geq 0$, either $e_1 > 0$ and $\lambda > 0$ or $e_2 > 0$ and $\lambda < 1$. Without loss of generality, we assume that $e_1 > 0$ and $\lambda > 0$. As $\mathcal{R}_1 \Rrightarrow_\epsilon \mathcal{P}_1$, by condition (3), $\boldsymbol{p}_1 \in \mathcal{P}_1$. As $\mathcal{R}_2 \Rrightarrow_\epsilon \mathcal{P}_2$, by Lemma 5, $\boldsymbol{p}_2 \in \mathbb{C}(\mathcal{P}_2)$. Thus, by Lemma 7,

$$\boldsymbol{p} = \lambda \boldsymbol{p}_1 + (1-\lambda)\boldsymbol{p}_2 + \boldsymbol{r} \in \mathcal{P}_1 \uplus \mathcal{P}_2.$$

To prove item (4), we show that $g(\mathcal{R}) \Rrightarrow_\epsilon f(\mathcal{P})$. To see this observe that, if $(\boldsymbol{x}^{\mathrm{T}}, \epsilon)^{\mathrm{T}} \in \mathcal{R}$ for any $\boldsymbol{x} \in \mathcal{P}$ and $\epsilon \in \mathbb{R}$, then, by definition of g,

$$g\big((\boldsymbol{x}^{\mathrm{T}}, \epsilon)^{\mathrm{T}}\big) = \big(f(\boldsymbol{x})^{\mathrm{T}}, \epsilon\big)^{\mathrm{T}}.$$

Thus, the ϵ dimension is not affected at all by the affine transformation, so that conditions (1), (2) and (3) of Definition 5 follow trivially from the hypothesis $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P}$. □

The proof of Proposition 4 requires a few additional lemmas.

Lemma 8. *Let* $\operatorname{con}(\mathcal{C}) \Rrightarrow_\epsilon \mathcal{P}$. *Suppose there exists* $c = \big(\langle \boldsymbol{a}, \boldsymbol{x} \rangle + s \cdot \epsilon \geq b\big) \in \mathcal{C}$ *that is saturated by* $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in \operatorname{con}(\mathcal{C})$, *where* $e \neq 0$. *Then* $s \leq 0$.

Proof. As $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in \operatorname{con}(\mathcal{C})$ saturates c, $\langle \boldsymbol{a}, \boldsymbol{v} \rangle + s \cdot e = b$. By condition (2) of Definition 5, $(\boldsymbol{v}^{\mathrm{T}}, 0)^{\mathrm{T}} \in \operatorname{con}(\mathcal{C})$ so that $\langle \boldsymbol{a}, \boldsymbol{v} \rangle \geq b$. By condition (1) of Definition 5, $e \geq 0$. By hypothesis, $e \neq 0$, so that $s \leq 0$. □

Lemma 9. *Let* $(\mathcal{C}, \mathcal{G}) \equiv \mathcal{R} \Rrightarrow_\epsilon \mathcal{P}$, *where* $\mathcal{G} = (R, P)$ *and* $\mathcal{P} \neq \varnothing$. *Then, there exists a constraint* $c^* \in \mathcal{C} \setminus (\mathcal{C}_> \cup \mathcal{C}_\geq \cup \mathcal{C}_\epsilon)$ *such that* $\operatorname{sat\_gen}(c^*, \mathcal{G}) = R \cup \mathcal{G}_C$. *Moreover, if* $(\mathcal{C}, \mathcal{G})$ *is a DD-pair in minimal form, the following hold:*

1. $\{c^*\} = \mathcal{C} \setminus (\mathcal{C}_> \cup \mathcal{C}_\geq \cup \mathcal{C}_\epsilon)$;
2. *if* $\boldsymbol{p} \stackrel{\text{def}}{=} (\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}$ *and* $\boldsymbol{p}_0 \stackrel{\text{def}}{=} (\boldsymbol{v}^{\mathrm{T}}, 0)^{\mathrm{T}}$, *then*

$$\operatorname{sat\_con}(\boldsymbol{p}_0, \mathcal{C}) \setminus \{c^*\} = \operatorname{sat\_con}(\boldsymbol{p}, \mathcal{C}) \cap \mathcal{C}_\geq;$$

3. *for any* $c \in \mathcal{C} \setminus \{c^*\}$, $\operatorname{sat\_gen}(c, \mathcal{G}) \cap \mathcal{G}_P \neq \varnothing$.

Proof. As $\mathcal{P} \neq \varnothing$, by conditions (2) and (3) of Definition 5, there exists a point $(\boldsymbol{v}^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}$. By condition (1), there exists a constraint $c^* = \big(\langle \boldsymbol{a}, \boldsymbol{x} \rangle + s \cdot \epsilon \geq b\big)$ such that, for any $e < 0$, $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}}$ does not satisfy c^*. Thus, for all $e < 0$ we obtain $\langle \boldsymbol{a}, \boldsymbol{v} \rangle + s \cdot e < b$. As $(\boldsymbol{v}^{\mathrm{T}}, 0)^{\mathrm{T}}$ satisfies c^*, $\langle \boldsymbol{a}, \boldsymbol{v} \rangle \geq b$. It follows that $\langle \boldsymbol{a}, \boldsymbol{v} \rangle = b$ and $s > 0$. Hence we have $c^* \in \mathcal{C} \setminus (\mathcal{C}_> \cup \mathcal{C}_\geq \cup \mathcal{C}_\epsilon)$. Also, as $s > 0$, any point $(\boldsymbol{w}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}$ saturates c^* if and only if $e = 0$. Therefore, by Lemma 3, $R \subseteq \mathrm{sat\_gen}(c^*, \mathcal{G})$, so that $\mathrm{sat\_gen}(c^*, \mathcal{G}) = R \cup \mathcal{G}_C$.

We now suppose that $(\mathcal{C}, \mathcal{G})$ is a DD-pair in minimal form and prove all the remaining statements.

As $\mathcal{P} \neq \varnothing$, we have $\mathcal{G}_P \neq \varnothing$, so that $(R, \mathcal{G}_C) \neq \mathcal{G}$. Since the condition $\mathrm{sat\_gen}(c^*, \mathcal{G}) = R \cup \mathcal{G}_C$ holds for any constraint $c^* \in \mathcal{C} \setminus (\mathcal{C}_> \cup \mathcal{C}_\geq \cup \mathcal{C}_\epsilon)$ and $\mathcal{C}$ is in minimal form, there can be only one such constraint, so that item 1 holds.

We now prove item 2. Consider $c \in \mathcal{C}_\geq$, so that $c = \bigl(\langle \boldsymbol{a}, \boldsymbol{x} \rangle + 0 \cdot \epsilon \geq b\bigr)$; then $c \in \mathrm{sat\_con}(\boldsymbol{p}, \mathcal{C})$ if and only if $c \in \mathrm{sat\_con}(\boldsymbol{p}_0, \mathcal{C})$, so that $\mathrm{sat\_con}(\boldsymbol{p}_0, \mathcal{C}_\geq) = \mathrm{sat\_con}(\boldsymbol{p}, \mathcal{C}_\geq)$. Consider now $c \in \mathcal{C}_>$, so that $c = \bigl(\langle \boldsymbol{a}, \boldsymbol{x} \rangle + s \cdot \epsilon \geq b\bigr)$ where $s < 0$; since $e > 0$, we obtain $\langle \boldsymbol{a}, \boldsymbol{v} \rangle > b$, so that $\boldsymbol{p}_0$ satisfies but does not saturate c; thus $\mathrm{sat\_con}(\boldsymbol{p}_0, \mathcal{C}_>) = \varnothing$. Consider now $c \in \mathcal{C}_\epsilon$, so that $c = (\epsilon \leq \delta)$ for some $\delta > 0$; then it follows that $\mathrm{sat\_con}(\boldsymbol{p}_0, \mathcal{C}_\epsilon) = \varnothing$. By all the above relations, together with item 1 proved above, we obtain

$$\begin{aligned}
\mathrm{sat_con}(\boldsymbol{p}_0, \mathcal{C}) \setminus \{c^*\} &= \mathrm{sat_con}(\boldsymbol{p}_0, \mathcal{C}) \cap (\mathcal{C}_> \cup \mathcal{C}_\geq \cup \mathcal{C}_\epsilon) \\
&= \mathrm{sat_con}(\boldsymbol{p}_0, \mathcal{C}_>) \cup \mathrm{sat_con}(\boldsymbol{p}_0, \mathcal{C}_\geq) \cup \mathrm{sat_con}(\boldsymbol{p}_0, \mathcal{C}_\epsilon) \\
&= \mathrm{sat_con}(\boldsymbol{p}, \mathcal{C}) \cap \mathcal{C}_\geq.
\end{aligned}$$

Finally, to prove item 3, suppose $c \in \mathcal{C} \setminus \{c^*\}$. Then, as $(\mathcal{C}, \mathcal{G})$ is in minimal form, $\mathrm{sat\_gen}(c, \mathcal{G}) \setminus (R \cup \mathcal{G}_C) \neq \varnothing$. Thus, there exists a point encoding $\boldsymbol{p} \in \mathcal{G}_P$ such that $\boldsymbol{p} \in \mathrm{sat\_gen}(c, \mathcal{G})$. □

Lemma 10. *Let $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P} \neq \varnothing$, where $(\mathcal{C}, \mathcal{G}) \equiv \mathcal{R}$ is a DD pair in minimal form. If c is ϵ-redundant in $\mathcal{C}$, then $\mathrm{con}(\mathcal{C}') \Rrightarrow_\epsilon \mathcal{P}$, where $\mathcal{C}' \stackrel{\mathrm{def}}{=} \mathcal{C} \setminus \{c\} \cup \{\epsilon \leq 1\}$.*

Proof. By Definition 7, we have $c \in \mathcal{C}_>$; thus, $c = \bigl(\langle \boldsymbol{a}, \boldsymbol{x} \rangle + s \cdot \epsilon \geq b\bigr)$, where $\boldsymbol{a} \in \mathbb{R}^n$ and $s, b \in \mathbb{R}$ are such that $\boldsymbol{a} \neq \boldsymbol{0}$ and $s < 0$. Let $\mathcal{R}' = \mathrm{con}(\mathcal{C}')$. In order to show that $\mathcal{R}' \Rrightarrow_\epsilon \mathcal{P}$, we first prove that $\mathcal{R}'$ is an ϵ-representation.

Consider condition (1) of Definition 5. The inclusion $\mathcal{R}' \subseteq \mathrm{con}\bigl(\{\epsilon \leq \delta\}\bigr)$ holds trivially by taking $\delta = 1$, because the constraint $\epsilon \leq 1$ has been explicitly added in $\mathcal{C}'$. To prove the other inclusion $\mathcal{R}' \subseteq \mathrm{con}\bigl(\{\epsilon \geq 0\}\bigr)$, we apply Lemma 9 to polyhedron $\mathcal{R}$. Namely, there exists a constraint $c^* \in \mathcal{C} \setminus (\mathcal{C}_> \cup \mathcal{C}_\geq \cup \mathcal{C}_\epsilon)$ such that, for all points $\boldsymbol{v}_0 \stackrel{\mathrm{def}}{=} (\boldsymbol{v}^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}$, c^* is saturated by $\boldsymbol{v}_0$. Since $c \in \mathcal{C}_>$, then $c \neq c^*$ and $c^* \in \mathcal{C}'$. Thus, condition (1) holds for $\mathcal{R}'$.

Consider now condition (2). By contraposition, suppose there exists a vector $\boldsymbol{v} \in \mathbb{R}^n$ and a scalar $e > 0$ such that $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}'$, but $(\boldsymbol{v}^{\mathrm{T}}, 0)^{\mathrm{T}} \notin \mathcal{R}'$. As a consequence, there must exist a constraint $c' = \bigl(\langle \boldsymbol{a}, \boldsymbol{x} \rangle + s \cdot \epsilon \geq b\bigr) \in \mathcal{C}'$ and a point $(\boldsymbol{v}^{\mathrm{T}}, e')^{\mathrm{T}}$, where $0 < e' \leq e$, such that $\langle \boldsymbol{a}, \boldsymbol{v} \rangle + s \cdot e' = b$ and $\langle \boldsymbol{a}, \boldsymbol{v} \rangle < b$. Since $e' > 0$, the above conditions imply $s > 0$; as a consequence, $c' \neq (\epsilon \leq 1)$, so that $c' \in \mathcal{C}$. However, this contradicts Lemma 8. Therefore $(\boldsymbol{v}^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}'$.

Thus, $\mathcal{R}'$ is an ϵ-representation. To show that it is indeed an ϵ-representation for $\mathcal{P}$, we have to prove that $[\![\mathcal{R}]\!] = [\![\mathcal{R}']\!]$.

To prove the inclusion $[\![\mathcal{R}]\!] \subseteq [\![\mathcal{R}']\!]$, let $\boldsymbol{v} \in [\![\mathcal{R}]\!]$. Thus, there exists $e > 0$ such that $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}$. By condition (2) of Definition 5, we also have $(\boldsymbol{v}^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}$ so that, as $\mathcal{R}$ is a convex set, there exists $0 < e' < 1$ such that $\boldsymbol{p} = (\boldsymbol{v}^{\mathrm{T}}, e')^{\mathrm{T}} \in \mathcal{R}$.

Note that $\boldsymbol{p}$ satisfies all the constraints in $\mathcal{C}$ and it also satisfies the constraint $\epsilon \leq 1$; as a consequence, $\boldsymbol{p} \in \mathcal{R}'$ and $\boldsymbol{v} \in [\![\mathcal{R}']\!]$, as required.

To show the other inclusion $[\![\mathcal{R}']\!] \subseteq [\![\mathcal{R}]\!]$, let $\boldsymbol{v} \in [\![\mathcal{R}']\!]$. Thus, there exists $e' > 0$ such that $(\boldsymbol{v}^{\mathrm{T}}, e')^{\mathrm{T}} \in \mathcal{R}'$ and, since $\mathcal{R}'$ is an ϵ-representation, by condition (2) of Definition 5, $\boldsymbol{v}_0 \stackrel{\text{def}}{=} (\boldsymbol{v}^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}'$. Let $\mathcal{G}' = (R', P')$ be a generator system such that $(\mathcal{C}', \mathcal{G}') \equiv \mathcal{R}'$. By Lemma 3 and the definition of closure point encodings, $\boldsymbol{v}_0 \in \operatorname{gen}\bigl((R', \mathcal{G}'_C)\bigr)$. Also, for any $c' \in \mathcal{C}'$, $\boldsymbol{v}_0$ satisfies c'.

First we show that $\boldsymbol{v}_0 \in \mathcal{R}$. By contraposition, suppose that $\boldsymbol{v}_0 \notin \mathcal{R}$. Then $\boldsymbol{v}_0$ does not satisfy c and hence $\langle \boldsymbol{a}, \boldsymbol{v} \rangle < b$. By item 3 of Lemma 9, there must exists a point $(\boldsymbol{p}^{\mathrm{T}}, e_p)^{\mathrm{T}} \in \mathcal{G}_P$ that saturates c. Since $e_p > 0$ and $s < 0$, we obtain $\langle \boldsymbol{a}, \boldsymbol{p} \rangle > b$. Thus, there exists $0 < \lambda < 1$ such that, letting $\boldsymbol{q} \stackrel{\text{def}}{=} \lambda \boldsymbol{v} + (1 - \lambda)\boldsymbol{p}$, we have $\langle \boldsymbol{a}, \boldsymbol{q} \rangle = b$, so that the point $\boldsymbol{q}_0 \stackrel{\text{def}}{=} (\boldsymbol{q}^{\mathrm{T}}, 0)^{\mathrm{T}}$ saturates c. Note that we have $\boldsymbol{p} \in \mathcal{P}$, so that $\boldsymbol{p} \in \mathbb{C}(\mathcal{P})$. Since $\mathcal{P} = [\![\mathcal{R}]\!] \subseteq [\![\mathcal{R}']\!]$, we obtain $\boldsymbol{p} \in \mathbb{C}\bigl([\![\mathcal{R}']\!]\bigr)$; thus, by Lemma 5, we obtain $\boldsymbol{p}_0 \stackrel{\text{def}}{=} (\boldsymbol{p}^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}'$. Therefore, being defined as a convex combination of two points of $\mathcal{R}'$, we have $\boldsymbol{q}_0 \in \mathcal{R}'$; since this point also saturates c, we obtain $\boldsymbol{q}_0 \in \mathcal{R} \cap \mathcal{R}'$. Note that this implies $\operatorname{sat\_gen}(c, \mathcal{G}) \cap \mathcal{G}_C \neq \varnothing$ so that, as by hypothesis c is ϵ-redundant in $\mathcal{C}$, there exists a constraint $c' \in \mathcal{C}_> \setminus \{c\}$ such that $\operatorname{sat\_gen}(c, \mathcal{G}) \setminus \mathcal{G}_P \subseteq \operatorname{sat\_gen}(c', \mathcal{G})$. Let $c' = \bigl(\langle \boldsymbol{a}', \boldsymbol{x} \rangle + s' \cdot \epsilon \geq b'\bigr)$, where $\boldsymbol{a}' \neq \boldsymbol{0}$ and $s' < 0$. Since $c' \in \mathcal{C}'$, the point $(\boldsymbol{v}^{\mathrm{T}}, e')^{\mathrm{T}} \in \mathcal{R}'$ satisfies c'. Since $s' < 0$ and $e' > 0$, we obtain $\langle \boldsymbol{a}', \boldsymbol{v} \rangle > b'$. Similarly, point $\boldsymbol{p}_0 \in \mathcal{R}'$ has to satisfy c', so that $\langle \boldsymbol{a}', \boldsymbol{p} \rangle \geq b'$. Since $\boldsymbol{q} \neq \boldsymbol{p}$, the above conditions imply that $\langle \boldsymbol{a}', \boldsymbol{q} \rangle > b'$, so that $\boldsymbol{q}_0$ saturates c but not c'. However, $\boldsymbol{q}_0 \in \mathcal{R}$ is a positive combination of rays and points in $\mathcal{G}_C$ all saturating c. As $\operatorname{sat\_gen}(c, \mathcal{G}) \setminus \mathcal{G}_P \subseteq \operatorname{sat\_gen}(c', \mathcal{G})$, $\boldsymbol{q}_0$ is also a positive combination of rays and points in $\mathcal{G}_C$ that saturate c'. Thus $\boldsymbol{q}_0$ has to saturate c', which is a contradiction. Therefore, we have $\boldsymbol{v}_0 \in \mathcal{R}$, as claimed above.

Secondly we show that $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}$ for some $e > 0$. By contraposition, suppose that for all $e > 0$ it holds $\langle \boldsymbol{a}, \boldsymbol{v} \rangle + s \cdot e < b$. Then, since $\boldsymbol{v}_0$ satisfies the constraint c, it must hold $\langle \boldsymbol{a}, \boldsymbol{v} \rangle = b$, so that $\boldsymbol{v}_0$ actually saturates c. This implies $\boldsymbol{v}_0 \in \operatorname{gen}\bigl((R_c, P_c)\bigr)$, where $R_c = R \cap \operatorname{sat\_gen}(c, \mathcal{G})$ and $P_c = \mathcal{G}_C \cap \operatorname{sat\_gen}(c, \mathcal{G})$. Therefore, it cannot be the case that $\operatorname{sat\_gen}(c, \mathcal{G}) \cap \mathcal{G}_C = \varnothing$ and, since c is ϵ-redundant, there exists a constraint $c' \in \mathcal{C}_>$ such that $c' \neq c$ and

$$\operatorname{sat\_gen}(c, \mathcal{G}) \setminus \mathcal{G}_P \subseteq \operatorname{sat\_gen}(c', \mathcal{G}).$$

Note that $R_c \cup P_c \subseteq \operatorname{sat\_gen}(c, \mathcal{G}) \setminus \mathcal{G}_P$, so that point $\boldsymbol{v}_0$ also saturates c'. Since $c' \in \mathcal{C}'_>$, this implies that for all $e > 0$ the point $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \notin \mathcal{R}'$, therefore contradicting the initial assumption that $\boldsymbol{v} \in [\![\mathcal{R}']\!]$. As a consequence, there exists $e > 0$ such that $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}$ and, by definition, $\boldsymbol{v} \in [\![\mathcal{R}]\!]$. □

Lemma 11. *Let $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P} \neq \varnothing$, where $(\mathcal{C}, \mathcal{G}) \equiv \mathcal{R}$ is a DD pair in minimal form. If $\boldsymbol{p}$ is ϵ-redundant in $\mathcal{G} = (R, P)$, then $\operatorname{gen}(\mathcal{G}') \Rrightarrow_\epsilon \mathcal{P}$, where $\mathcal{G}' = \bigl(R, P \setminus \{\boldsymbol{p}\}\bigr)$.*

Proof. By Definition 7, we have $\boldsymbol{p} \in \mathcal{G}_U$, so that there exists $e > 0$ such that $\boldsymbol{p} = (\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in P$ and $\boldsymbol{p}_0 = (\boldsymbol{v}^{\mathrm{T}}, 0)^{\mathrm{T}} \notin P$. Let $\mathcal{R}' = \operatorname{gen}(\mathcal{G}')$. Note that, since the function 'gen' is monotonic on both components of its argument, we have

$\mathcal{R}' \subseteq \mathcal{R}$. Moreover, for each $\boldsymbol{q}_0 \stackrel{\text{def}}{=} (\boldsymbol{w}^{\mathrm{T}}, 0)^{\mathrm{T}}$, $\boldsymbol{q}_0 \in \mathcal{R}$ if and only if $\boldsymbol{q}_0 \in \mathcal{R}'$. To see this, it is sufficient to observe that, since $\mathcal{G}_C = \mathcal{G}'_C$ (because $\boldsymbol{p} \notin \mathcal{G}_C$), $\boldsymbol{q}_0 \in \mathcal{R}$ if and only if $\boldsymbol{q}_0 \in \operatorname{gen}\bigl((R, \mathcal{G}_C)\bigr)$ if and only if $\boldsymbol{q}_0 \in \mathcal{R}'$.

In order to show that $\mathcal{R}' \Rrightarrow_\epsilon \mathcal{P}$, we first prove that $\mathcal{R}'$ is an ϵ-representation. Consider condition (1) of Definition 5. As $\mathcal{R}' \subseteq \mathcal{R}$, $\mathcal{R}'$ satisfies condition (1) by taking the same value δ used for $\mathcal{R}$.

Consider now condition (2) of Definition 5. Let $\boldsymbol{q} = (\boldsymbol{w}^{\mathrm{T}}, e_w)^{\mathrm{T}} \in \mathcal{R}'$. Since $\mathcal{R}' \subseteq \mathcal{R}$, we have $\boldsymbol{q} \in \mathcal{R}$; since $\mathcal{R}$ is an ϵ-representation, $\boldsymbol{q}_0 \stackrel{\text{def}}{=} (\boldsymbol{w}^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}$. Thus, by the observation made above, $\boldsymbol{q}_0 \in \mathcal{R}'$, as required.

Thus, $\mathcal{R}'$ is an ϵ-representation. To show that it is indeed an ϵ-representation for $\mathcal{P}$, we have to prove that $[\![\mathcal{R}]\!] = [\![\mathcal{R}']\!]$. The inclusion $[\![\mathcal{R}']\!] \subseteq [\![\mathcal{R}]\!]$ holds by monotonicity of function $[\![\cdot]\!]$, since $\mathcal{R}' \subseteq \mathcal{R}$.

To prove the other inclusion, $[\![\mathcal{R}]\!] \subseteq [\![\mathcal{R}']\!]$, suppose that $\boldsymbol{w} \in [\![\mathcal{R}]\!]$. Then there exists $e > 0$ such that $\boldsymbol{q} \stackrel{\text{def}}{=} (\boldsymbol{w}^{\mathrm{T}}, e)^{\mathrm{T}} \in \mathcal{R}$. By condition (2) of Definition 5, we obtain $\boldsymbol{q}_0 \stackrel{\text{def}}{=} (\boldsymbol{w}^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}$. Thus $\boldsymbol{q}_0 \in \operatorname{gen}\bigl((R, \mathcal{G}_C)\bigr)$; since $\boldsymbol{p} \notin \mathcal{G}_C$, we have $\mathcal{G}_C = \mathcal{G}'_C$ so that $\boldsymbol{q}_0 \in \mathcal{R}'$. As $\boldsymbol{q} \in \mathcal{R}$, by definition of 'gen' there exist $0 \leq \lambda_{\boldsymbol{p}} \leq 1$ and $\boldsymbol{p}_1 \in \operatorname{gen}(\mathcal{G}')$ such that $\boldsymbol{q} = \lambda_{\boldsymbol{p}} \boldsymbol{p} + (1 - \lambda_{\boldsymbol{p}}) \boldsymbol{p}_1$. If $\lambda_{\boldsymbol{p}} = 0$, then $\boldsymbol{q} \in \mathcal{R}'$, so that $\boldsymbol{w} \in [\![\mathcal{R}']\!]$ as required.

Suppose now that $\lambda_{\boldsymbol{p}} > 0$; then $\operatorname{sat\_con}(\boldsymbol{q}, \mathcal{C}) \subseteq \operatorname{sat\_con}(\boldsymbol{p}, \mathcal{C})$. Since $\boldsymbol{p}$ is ϵ-redundant in $\mathcal{G}$, by Definition 7, there exists $\boldsymbol{p}' \stackrel{\text{def}}{=} (\boldsymbol{y}^{\mathrm{T}}, e')^{\mathrm{T}} \in P \setminus \{\boldsymbol{p}\}$ such that $\operatorname{sat\_con}(\boldsymbol{p}, \mathcal{C}) \cap \mathcal{C}_{\geq} \subseteq \operatorname{sat\_con}(\boldsymbol{p}', \mathcal{C})$ (also note that $\boldsymbol{p}' \in \mathcal{R}'$). Thus, we obtain

$$\operatorname{sat\_con}(\boldsymbol{q}, \mathcal{C}) \cap \mathcal{C}_{\geq} \subseteq \operatorname{sat\_con}(\boldsymbol{p}', \mathcal{C}).$$

By letting $\boldsymbol{p}'_0 \stackrel{\text{def}}{=} (\boldsymbol{y}^{\mathrm{T}}, 0)^{\mathrm{T}}$ and applying item 2 of Lemma 9 twice, we obtain

$$\begin{aligned}
&\operatorname{sat_con}\bigl(\boldsymbol{q}_0, (\mathcal{C}_{>} \cup \mathcal{C}_{\geq} \cup \mathcal{C}_{\epsilon})\bigr) \\
&\qquad = \operatorname{sat_con}(\boldsymbol{q}, \mathcal{C}_{\geq}) \\
&\qquad \subseteq \operatorname{sat_con}(\boldsymbol{p}', \mathcal{C}_{\geq}) \\
&\qquad = \operatorname{sat_con}\bigl(\boldsymbol{p}'_0, (\mathcal{C}_{>} \cup \mathcal{C}_{\geq} \cup \mathcal{C}_{\epsilon})\bigr) \\
&\qquad \subseteq \operatorname{sat_con}(\boldsymbol{p}'_0, \mathcal{C}).
\end{aligned}$$

Now let $c \stackrel{\text{def}}{=} \bigl(\langle \boldsymbol{a}, \boldsymbol{x} \rangle + s \cdot \epsilon \geq b\bigr)$ be any constraint in $(\mathcal{C}_{>} \cup \mathcal{C}_{\geq} \cup \mathcal{C}_{\epsilon})$ so that, as $\boldsymbol{p}' \in \mathcal{R}$, $\langle \boldsymbol{a}, \boldsymbol{y} \rangle \geq b$. If $c \in \operatorname{sat\_con}(\boldsymbol{q}_0, \mathcal{C})$, then $c \in \operatorname{sat\_con}(\boldsymbol{p}'_0, \mathcal{C})$, so that all the points lying on the line passing through $\boldsymbol{q}_0$ and $\boldsymbol{p}'_0$ saturate c. On the other hand, if $c \notin \operatorname{sat\_con}(\boldsymbol{q}_0, \mathcal{C})$, then $\langle \boldsymbol{a}, \boldsymbol{w} \rangle > b$. Let

$$\begin{aligned}
\mu_c &\stackrel{\text{def}}{=} \begin{cases} \frac{\langle \boldsymbol{a}, \boldsymbol{y} \rangle - \langle \boldsymbol{a}, \boldsymbol{w} \rangle}{\langle \boldsymbol{a}, \boldsymbol{y} \rangle - b} & \text{if } \langle \boldsymbol{a}, \boldsymbol{y} \rangle > \langle \boldsymbol{a}, \boldsymbol{w} \rangle \text{ and } \langle \boldsymbol{a}, \boldsymbol{y} \rangle > b; \\ 1 & \text{otherwise;} \end{cases} \\
\boldsymbol{w}_c &\stackrel{\text{def}}{=} (1 + \mu_c) \boldsymbol{w} - \mu_c \boldsymbol{y}'.
\end{aligned}$$

Thus $(\boldsymbol{w}_c^{\mathrm{T}}, 0)^{\mathrm{T}}$ lies on the line passing through $\boldsymbol{q}_0$ and $\boldsymbol{p}_0'$ and satisfies c. Let

$$\mu \stackrel{\text{def}}{=} \min\bigl\{ \mu_c \in \mathbb{R} \bigm| c \in \mathcal{C} \setminus \operatorname{sat\_con}(\boldsymbol{q}_0, \mathcal{C}) \bigr\};$$
$$\boldsymbol{w}_\mu \stackrel{\text{def}}{=} (1+\mu)\boldsymbol{w} - \mu \boldsymbol{y'}.$$

Then $(\boldsymbol{w}_\mu^{\mathrm{T}}, 0)^{\mathrm{T}}$ satisfies all the constraints in $(\mathcal{C}_> \cup \mathcal{C}_\geq \cup \mathcal{C}_\epsilon)$. By item 1 of Lemma 9, $\{c^*\} = \mathcal{C} \setminus (\mathcal{C}_> \cup \mathcal{C}_\geq \cup \mathcal{C}_\epsilon)$ and $\operatorname{sat\_gen}(c^*, \mathcal{G}) = R \cup \mathcal{G}_C$. Thus $(\boldsymbol{w}_\mu^{\mathrm{T}}, 0)^{\mathrm{T}}$ satisfies all the constraints in $\mathcal{C}$ so that $(\boldsymbol{w}_\mu^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}$. Since $\mathcal{G}_C = \mathcal{G}_C'$, we also have $(\boldsymbol{w}_\mu^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}'$. Letting $\lambda \stackrel{\text{def}}{=} \frac{1}{1+\mu}$ we have $0 < \lambda < 1$ and $\boldsymbol{w} = \lambda \boldsymbol{w}_\mu + (1-\lambda)\boldsymbol{y'}$. Thus, as $\boldsymbol{p'} \in \mathcal{R}'$, we have

$$\bigl(\boldsymbol{w}^{\mathrm{T}}, (1-\lambda)e'\bigr)^{\mathrm{T}} = \lambda(\boldsymbol{w}_\mu^{\mathrm{T}}, 0)^{\mathrm{T}} + (1-\lambda)\boldsymbol{p'} \in \mathcal{R}'.$$

As $e' > 0$ and $\lambda < 1$, $(1-\lambda)e' > 0$ and hence $\boldsymbol{w} \in [\![\mathcal{R}']\!]$ as required. □

Lemma 12. *Let $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P} \neq \varnothing$, where $\mathcal{R} = \operatorname{con}(\mathcal{C})$ and $\mathcal{C}$ is in minimal form. If $\mathcal{C}$ contains no ϵ-redundant constraint then it is in* smf.

Proof. To prove the thesis, we assume that $\mathcal{C}$ is not in smf and show that $\mathcal{C}$ must have an ϵ-redundant constraint. As $\mathcal{C}$ is not in smf, there is a constraint system $\mathcal{C}'$ in weak minimal form such that $(\mathcal{C}_>' \cup \mathcal{C}_\geq') \subset (\mathcal{C}_> \cup \mathcal{C}_\geq)$ and $\mathcal{R}' \stackrel{\text{def}}{=} \operatorname{con}(\mathcal{C}') \Rrightarrow_\epsilon \mathcal{P}$. Let $c \in (\mathcal{C}_> \cup \mathcal{C}_\geq) \setminus \mathcal{C}'$. Then there exists $\boldsymbol{a} \neq \boldsymbol{0} \in \mathbb{R}^n$, $s \leq 0$, and $b \in \mathbb{R}$ such that

$$c = \bigl(\langle \boldsymbol{a}, \boldsymbol{x} \rangle + s \cdot \epsilon \geq b\bigr).$$

First we show that $c \in \mathcal{C}_>$. By contraposition, suppose that $c \in \mathcal{C}_\geq$, so that $s = 0$. As $\mathcal{C}$ is in minimal form, there exists a point $\boldsymbol{p} \stackrel{\text{def}}{=} (\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in \operatorname{con}\bigl(\mathcal{C} \setminus \{c\}\bigr)$ that does not satisfy c. Thus $\boldsymbol{p}_0 \stackrel{\text{def}}{=} (\boldsymbol{v}^{\mathrm{T}}, 0)^{\mathrm{T}}$ does not satisfy c. As $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P}$, by Lemma 5, we have $\boldsymbol{v} \notin \mathbb{C}(\mathcal{P})$. By hypothesis, $\mathcal{R}' \Rrightarrow_\epsilon \mathcal{P}$ so that, again by Lemma 5, we have $\boldsymbol{p}_0 \notin \mathcal{R}'$. By condition (2) of Definition 5, it follows that $(\boldsymbol{v}^{\mathrm{T}}, e')^{\mathrm{T}} \notin \mathcal{R}'$, for all $e' \geq 0$. However, for some $e' > 0$, $(\boldsymbol{v}^{\mathrm{T}}, e')^{\mathrm{T}}$ satisfies every constraint in $\mathcal{C}_\epsilon'$ and by hypothesis, satisfies every constraint in $\mathcal{C}_>' \cup \mathcal{C}_\geq'$. As $\mathcal{C}'$ is in weak minimal form, by item 1 in Lemma 9, $\mathcal{C}' \setminus (\mathcal{C}_>' \cup \mathcal{C}_\geq' \cup \mathcal{C}_\epsilon') = \{c^*\}$, where $c^* = \bigl(\langle \boldsymbol{a}^*, \boldsymbol{x} \rangle + s^* \cdot \epsilon \geq b^*\bigr)$ and $s^* > 0$. Therefore $(\boldsymbol{v}^{\mathrm{T}}, e')^{\mathrm{T}}$ does not satisfy c^* and there exists $e^* > e'$ such that $\boldsymbol{p}^* = (\boldsymbol{v}^{\mathrm{T}}, e^*)^{\mathrm{T}}$ saturates c^* and satisfies every constraint in $\mathcal{C}' \setminus \mathcal{C}_\epsilon'$. As $\mathcal{R}' \neq \varnothing$ and $\mathcal{C}'$ is in minimal form, there exists a point $\boldsymbol{q}_0 = (\boldsymbol{w}^{\mathrm{T}}, e_w)^{\mathrm{T}} \in \mathcal{R}'$ that saturates c^*. As $s^* > 0$, we can apply Lemma 8 to obtain $e_w = 0$. Thus, every convex combination of $\boldsymbol{p}^*$ and $\boldsymbol{q}_0$ will also saturate c^* and satisfy all constraints in $\mathcal{C}' \setminus \mathcal{C}_\epsilon'$. Then, for some $0 < \lambda < 1$, the convex combination

$$\lambda \boldsymbol{p}^* + (1-\lambda)\boldsymbol{q}_0 = (\boldsymbol{y}^{\mathrm{T}}, \lambda e^*)^{\mathrm{T}}$$

satisfies all the constraints in $\mathcal{C}'$ and saturates c^*. As $\lambda e^* > 0$ and $s^* > 0$, $\boldsymbol{q}_0' \stackrel{\text{def}}{=} \lambda \boldsymbol{p}_0 + (1-\lambda)\boldsymbol{q}_0$ does not satisfy c^*. Thus condition (2) of Definition 5 does not hold for $\mathcal{R}'$ contradicting the hypothesis that $\mathcal{R}'$ is an ϵ-representation.

We can therefore assume that $c \in \mathcal{C}_>$ and hence, $s < 0$. Suppose first that no points in $\mathrm{gen}\big((R, \mathcal{G}_C)\big)$ saturate c; this implies $\mathrm{sat\_gen}(c, \mathcal{G}) \cap \mathcal{G}_C = \varnothing$ and it follows from Definition 7 that c is ϵ-redundant in $\mathcal{C}$. Otherwise, suppose that c is saturated by at least one point in $\mathrm{gen}\big((R, \mathcal{G}_C)\big)$.

We next show that, for each point

$$\boldsymbol{p}_0 \stackrel{\text{def}}{=} (\boldsymbol{v}^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathrm{gen}\big((R, \mathcal{G}_C)\big),$$

there exist $c' \in \mathcal{C}'_>$ that is saturated by $\boldsymbol{p}_0$ but not satisfied by $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}}$, for any $e > 0$. Since $s < 0$, we have $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \notin \mathcal{R}$, for all $e > 0$. As $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P}$, by condition (3) of Definition 5, $\boldsymbol{v} \notin \mathcal{P}$ and, by Lemma 5, we have $\boldsymbol{v} \in \mathbb{C}(\mathcal{P})$. Since by hypothesis $R' \Rrightarrow_\epsilon \mathcal{P}$, by applying again condition (3) of Definition 5 and Lemma 5, we obtain $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \notin \mathcal{R}'$, for all $e > 0$, and $\boldsymbol{p}_0 \in \mathcal{R}'$. As a consequence, there must exist $c' = \big(\langle \boldsymbol{a}', \boldsymbol{x} \rangle + s' \cdot \epsilon \geq b'\big) \in \mathcal{C}'$ such that c' is saturated by $\boldsymbol{p}_0$ but not satisfied by $(\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}}$, for any $e > 0$. Thus we have $s' < 0$, so that $c' \in \mathcal{C}'_>$.

Let $\mathcal{Q} \stackrel{\text{def}}{=} \mathrm{sat\_gen}(c, \mathcal{G}) \setminus \mathcal{G}_P$ and $\mathcal{G}_\mathcal{Q} \stackrel{\text{def}}{=} (R \cap \mathcal{Q}, P \cap \mathcal{Q})$. Suppose that, for each constraint $c_i \in \mathcal{C}'_> = \{c_1, \ldots, c_k\}$, there exists a point $\boldsymbol{q}_i \in \mathrm{gen}(\mathcal{G}_\mathcal{Q})$ that does not saturate c_i. Then the convex combination

$$\frac{1}{k} \sum_{i=1}^{k} \boldsymbol{q}_i \in \mathrm{gen}(\mathcal{G}_\mathcal{Q})$$

saturates c, but does not saturate any constraint in $\mathcal{C}'_>$, contradicting the previous paragraph. As a consequence, there exists a constraint $c' \in \mathcal{C}'_>$ that is saturated by all the points in $\mathrm{gen}(\mathcal{G}_\mathcal{Q})$. As there exists a point in $\mathcal{R}$ that saturates c, $(P \cap \mathcal{Q}) \neq \varnothing$. Thus, as all points in $\mathrm{gen}(\mathcal{G}_\mathcal{Q})$ are obtained by summing a positive combination of the rays in $R \cap \mathcal{Q}$ with a convex combination of the points in $P \cap \mathcal{Q}$, we obtain $\mathcal{Q} \subseteq \mathrm{sat\_gen}(c', \mathcal{G})$. It then follows from Definition 7 that c is ϵ-redundant in $\mathcal{C}$. □

Lemma 13. *Let $\mathcal{R} \Rrightarrow_\epsilon \mathcal{P} \neq \varnothing$, where $\mathcal{R} = \mathrm{gen}(\mathcal{G})$ and $\mathcal{G}$ is in minimal form. If $\mathcal{G}$ contains no ϵ-redundant generator then it is in* smf.

Proof. To prove the thesis, we assume that $\mathcal{G}$ is not in smf and show that $\mathcal{G}$ must have an ϵ-redundant generator. As $\mathcal{G} = (R, P)$ is not in smf, by Definition 6, there exists a generator system $\mathcal{G}' = (R', P') \neq \mathcal{G}$ such that $R' \subseteq R$, $P' \subseteq P$ and $\mathrm{gen}(\mathcal{G}') \Rrightarrow_\epsilon \mathcal{P}$.

We first show that $P' \subset P$. By contraposition, suppose that $P' = P$, so that $R' \subset R$. Let $\boldsymbol{r} \in R \setminus R'$ and define $R'' = R \setminus \{\boldsymbol{r}\}$, so that $R' \subseteq R'' \subset R$. By monotonicity of function 'gen', we obtain

$$\mathrm{gen}(\mathcal{G}') \subseteq \mathrm{gen}\big((R'', P)\big) \subset \mathrm{gen}(\mathcal{G}),$$

where the strict inclusion holds by Definition 2, since by hypothesis $\mathcal{G}$ is in (weak) minimal form. Note that, since $\mathrm{gen}(\mathcal{G}') \Rrightarrow_\epsilon \mathcal{P}$ and $\mathrm{gen}(\mathcal{G}) \Rrightarrow_\epsilon \mathcal{P}$, by exploiting the monotonicity of $[\![\cdot]\!]$ we also obtain $\mathrm{gen}\big((R'', P)\big) \stackrel{\text{def}}{=} \mathcal{R}'' \Rrightarrow_\epsilon \mathcal{P}$. We will now show that $[\![\mathcal{R}'']\!] \subset \mathcal{P}$, therefore deriving the required contradiction.

Since $\mathcal{P} \neq \varnothing$, there exists $(\boldsymbol{w}^{\mathrm{T}}, e_w)^{\mathrm{T}} \in \mathcal{R}$ such that $e_w > 0$. Then, by condition 2 of Definition 5, $(\boldsymbol{w}^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}$. By Lemma 3, we can assume that $\boldsymbol{r} = (\boldsymbol{y}^{\mathrm{T}}, 0)^{\mathrm{T}}$. As $\boldsymbol{r} \notin R''$ and $\mathcal{G}$ is in minimal form, there exists $\mu \geq 0$ such that, if $\boldsymbol{w}_\mu \stackrel{\text{def}}{=} \boldsymbol{w} + \mu \boldsymbol{y}$, then $(\boldsymbol{w}_\mu, 0) \notin \mathcal{R}''$. Since $\mathcal{R}'' \Rrightarrow_\epsilon \mathcal{P}$, by Lemma 5, $\boldsymbol{w}_\mu \notin \mathbb{C}(\mathcal{P})$. Thus, again by Lemma 5, $(\boldsymbol{w}_\mu, 0) \notin \mathcal{R}$, contradicting the assumption that $\boldsymbol{r} \in R$.

Thus, we have $P' \subset P$. Let $\boldsymbol{p} = (\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \in P \setminus P'$ and define $P'' = P \setminus \{\boldsymbol{p}\}$, so that $P' \subseteq P'' \subset P$. By monotonicity of function 'gen' and because $\mathcal{G}$ is in (weak) minimal form, we obtain

$$\operatorname{gen}(\mathcal{G}') \subseteq \operatorname{gen}\big((R, P'')\big) \subset \operatorname{gen}(\mathcal{G}),$$

and $\boldsymbol{p} \notin \mathcal{R}''$. As already observed, since $\operatorname{gen}(\mathcal{G}') \Rrightarrow_\epsilon \mathcal{P}$ and $\operatorname{gen}(\mathcal{G}) \Rrightarrow_\epsilon \mathcal{P}$, by monotonicity of $[\![\cdot]\!]$ we also obtain $\operatorname{gen}\big((R, P'')\big) \stackrel{\text{def}}{=} \mathcal{R}'' \Rrightarrow_\epsilon \mathcal{P}$. We show that $\boldsymbol{p}$ is ϵ-redundant in $\mathcal{G}$.

We start by showing that $\boldsymbol{p} = (\boldsymbol{v}^{\mathrm{T}}, e)^{\mathrm{T}} \notin \mathcal{G}_C$, i.e., that $e > 0$. Suppose, by contraposition that $e = 0$. Then, by Lemma 5, $\boldsymbol{v} \in \mathbb{C}(\mathcal{P})$. Thus, as $\mathcal{R}'' \Rrightarrow_\epsilon \mathcal{P}$, again by Lemma 5, $\boldsymbol{p} = (\boldsymbol{v}^{\mathrm{T}}, 0)^{\mathrm{T}} \in \mathcal{R}''$ which is a contradiction. Therefore $\boldsymbol{p} \in \mathcal{G}_P$.

Letting $\mathcal{C}$ be a constraint system in weak minimal form such that $\operatorname{con}(\mathcal{C}) = \mathcal{R}$, we next show that there must exist $\boldsymbol{p}' \in \mathcal{G}_P \setminus \{\boldsymbol{p}\}$ such that

$$\operatorname{sat\_con}(\boldsymbol{p}, \mathcal{C}) \cap \mathcal{C}_{\geq} \subseteq \operatorname{sat\_con}(\boldsymbol{p}', \mathcal{C}). \tag{6}$$

By condition 3 of Definition 5, since $\boldsymbol{p} \in \mathcal{R}$ and $e > 0$, we have $\boldsymbol{v} \in \mathcal{P}$; and, since also $[\![\mathcal{R}'']\!] = \mathcal{P}$, there must exists $e_q > 0$ such that $\boldsymbol{q} \stackrel{\text{def}}{=} (\boldsymbol{v}^{\mathrm{T}}, e_q)^{\mathrm{T}} \in \mathcal{R}''$. Hence, $\boldsymbol{q} \in \operatorname{gen}\big((R, P'')\big)$ and, since $e_q > 0$, there must exist a point encoding $\boldsymbol{p}' \in \mathcal{G}_P \setminus \{\boldsymbol{p}\}$ that plays an active role (i.e., has a positive coefficient) in the combination generating $\boldsymbol{q}$. Consider a non-strict inequality encoding $c' \in \operatorname{sat\_con}(\boldsymbol{p}, \mathcal{C}) \cap \mathcal{C}_{\geq}$. Since $c' \in \mathcal{C}_{\geq}$, we have $c' = \big(\langle \boldsymbol{a}, \boldsymbol{x} \rangle + 0 \cdot \epsilon \geq b\big)$ so that c' is also saturated by $\boldsymbol{q}$. This in turn implies that c' is saturated by $\boldsymbol{p}'$. Since the choice of $c' \in \mathcal{C}_{\geq}$ was arbitrary, we obtain the condition (6).

To show that $\boldsymbol{p}$ is ϵ-redundant, we have to show that $\boldsymbol{p} \in \mathcal{G}_U$. By contraposition, suppose that $\boldsymbol{p}_0 \stackrel{\text{def}}{=} (\boldsymbol{v}^{\mathrm{T}}, 0)^{\mathrm{T}} \in P$. Suppose $\boldsymbol{p}' = (\boldsymbol{y}^{\mathrm{T}}, e')^{\mathrm{T}}$ and $\boldsymbol{p}'_0 \stackrel{\text{def}}{=} (\boldsymbol{y}^{\mathrm{T}}, 0)^{\mathrm{T}}$; then $\boldsymbol{p}'_0 \in \mathcal{R}$ by condition (2) of Definition 5. By item 2 of Lemma 9, we have

$$\begin{aligned}
\operatorname{sat_con}(\boldsymbol{p}_0, \mathcal{C}) \cap \mathcal{C}_{\geq} &= \operatorname{sat_con}(\boldsymbol{p}, \mathcal{C}) \cap \mathcal{C}_{\geq}, \\
\operatorname{sat_con}(\boldsymbol{p}'_0, \mathcal{C}) \cap \mathcal{C}_{\geq} &= \operatorname{sat_con}(\boldsymbol{p}', \mathcal{C}) \cap \mathcal{C}_{\geq},
\end{aligned}$$

so that, by the condition (6) proved above, we obtain

$$\operatorname{sat\_con}(\boldsymbol{p}_0, \mathcal{C}) \cap \mathcal{C}_{\geq} \subseteq \operatorname{sat\_con}(\boldsymbol{p}'_0, \mathcal{C}) \cap \mathcal{C}_{\geq}. \tag{7}$$

Suppose first that $e < e'$. Let $\boldsymbol{p}'_e \stackrel{\text{def}}{=} (\boldsymbol{y}^{\mathrm{T}}, e)^{\mathrm{T}}$; then, letting $\lambda = \frac{e}{e'}$ (note that $0 \leq \lambda \leq 1$), the point $\boldsymbol{q}_e = (\boldsymbol{w}^{\mathrm{T}}, e)^{\mathrm{T}} \stackrel{\text{def}}{=} \lambda \boldsymbol{p}' + (1 - \lambda) \boldsymbol{p}_0$ is a convex combination of $\boldsymbol{p}'$ and $\boldsymbol{p}_0$ and hence in $\mathcal{R}$. Let also $\boldsymbol{r} \stackrel{\text{def}}{=} \boldsymbol{p} - \boldsymbol{q}_e$; then $\boldsymbol{r}$ cannot be a ray of $\mathcal{R}$, since otherwise we would have $\boldsymbol{p} \in \text{gen}\big(R, \{\boldsymbol{p}', \boldsymbol{p}_0\}\big)$, contradicting the hypothesis that $\mathcal{G}$ is in minimal form. For each $\mu \in \mathbb{R}_+$, let $\boldsymbol{p}_\mu \stackrel{\text{def}}{=} \boldsymbol{p}_0 + \mu\boldsymbol{r}$. Since $\boldsymbol{r}$ is not a ray of $\mathcal{R}$, there must exist $\mu' \in \mathbb{R}_+$ such that, for all $\mu > \mu'$, we have $\boldsymbol{p}_\mu \notin \mathcal{R}$. If $\mu' > 0$, then $\boldsymbol{p}_{\mu'} \neq \boldsymbol{p}_0$; thus, $\boldsymbol{p}_0$ can be expressed as a convex combination of $\boldsymbol{p}_{\mu'}$ and $\boldsymbol{p}'_0$, contradicting the hypothesis that $\mathcal{G}$ is in minimal form. Therefore, it must hold $\mu' = 0$ (i.e., $\boldsymbol{p}_{\mu'} = \boldsymbol{p}_0$). As a consequence, there must exist a constraint $c \in \mathcal{C}$ such that $\boldsymbol{p}_0$ saturates c but $\boldsymbol{p}'_0$ does not saturate c. As $\mathcal{C}$ is in minimal form, by item 2 of Lemma 9, $\mathcal{C} \setminus \mathcal{C}_> \cup \mathcal{C}_\geq \cup \mathcal{C}_\epsilon = \{c^*\}$ and $\text{sat\_gen}(c^*, \mathcal{G}) = R \cup \mathcal{G}_C$ so that p_0 and p'_0 saturate c^*. Thus, by item 1 of Lemma 9, since $\boldsymbol{p}'_0$ does not saturate c we obtain $c \in \mathcal{C}_> \cup \mathcal{C}_\geq \cup \mathcal{C}_\epsilon$; from this and the fact that $\boldsymbol{p}_0$ saturates c, by item 2 of Lemma 9, we obtain $c \in \mathcal{C}_\geq$; However, this contradicts the condition (7) established above, so that we cannot have $e < e'$.

Secondly suppose that $e \geq e'$. Let $\boldsymbol{p}_{e'} \stackrel{\text{def}}{=} (\boldsymbol{v}^{\mathrm{T}}, e')^{\mathrm{T}}$; then, $\boldsymbol{p}_{e'} \in \mathcal{R}$ can be obtained as a convex combination of $\boldsymbol{p}$ and $\boldsymbol{p}_0$. Let also $\boldsymbol{r}' \stackrel{\text{def}}{=} \boldsymbol{p}' - \boldsymbol{p}_{e'}$; then r cannot be a ray of $\mathcal{R}$, since otherwise we would have $\boldsymbol{p}' \in \text{gen}\big(R, \{\boldsymbol{p}, \boldsymbol{p}_0\}\big)$, contradicting the hypothesis that $\mathcal{G}$ is in minimal form. By letting $\boldsymbol{p}_\mu \stackrel{\text{def}}{=} \boldsymbol{p}_0 - \mu\boldsymbol{r}'$, we obtain $\boldsymbol{p}_\mu \notin \mathcal{R}$ for all $\mu > 0$. Thus, we are in the same situation identified before and a contradiction can be derived by the same argument. As a consequence, $\boldsymbol{p}_0 \in P$ cannot hold so that $\boldsymbol{p} \in \mathcal{G}_U$, completing the proof. □

Proof (Proof of Proposition 4). Items 1, 2, 3, and 4 have been proved as Lemmas 10, 11, 12, and 13 respectively. □